Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

ANALISIS STATISTIK DENGAN R

YAYASAN PRIMA AGUS TEKNIK

ANALISIS STATISTIK DENGAN R

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

BIO DATA PENULIS

Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK

JL. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-5734-36-1

9 786235 734361

ANALISIS STATISTIK DENGAN R

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK

JL. Majapahit No. 605 Semarang

Telp. (024) 6723456. Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Analisis Statistik Dengan R

Penulis :
Dr. Ir. Agus Wibowo, M.Kom., M.Si., MM.

ISBN : 9 786235 734361

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom.

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji Syukur penulis panjatkan karena buku yang berjudul "*Analisis Statistik Dengan R*" dapat terselesaikan. Banyak buku statistik mengajarkan pembaca konsepnya tetapi tidak mengajarkan pembaca cara mudah untuk menerapkannya. Hal ini sering menyebabkan kurangnya pemahaman. R adalah alat untuk menerapkan dan mempelajari konsep statistik. Sebelum penulis menjelaskan tentang salah satu fitur R, penulis menjelaskan dasar statistik yang mendasarinya. Dengan begitu, pembaca akan lebih memahami fitur tersebut saat menggunakannya dan menggunakannya dengan lebih efektif.

Buku ini mencakup detail R dan memperkenalkan beberapa teknik pengkodean yang cerdas. Dalam buku apa pun yang menunjukkan kepada Anda cara menggunakan alat perangkat lunak seperti R. Buku ini bukan hanya tentang statistik atau hanya tentang R — buku ini dengan tegas berada di persimpangan keduanya. Dalam konteks yang tepat, R dapat menjadi alat yang hebat untuk statistik pengajaran dan pembelajaran.

Meskipun bidang statistik berjalan dengan cara yang logis, penulis telah mengatur buku ini sehingga pembaca dapat membukanya di setiap bab dan mulai membaca. Idenya adalah agar penulis menemukan informasi dicari dengan tergesa-gesa dan menggunakannya.

Buku ini yang menjelaskan konsep statistik sangat mirip dengan bagian terkait di dalamnya. Penulis memberikan contoh yang sama dan, dalam banyak kasus, kata-kata yang sama. Buku ini disusun menjadi 5 bagian lima bagian.

Di Bagian 1, saya memberikan pengantar umum statistik dan membahas konsep statistik serta menjelaskan teknik R yang digunakan. Serta dasar-dasar tentang statistika. Bagi pembaca yang belum pernah belajar tentang Statistik, maka lebih baik untuk menguasai Bagian 1 dari buku ini. Bagian 2 menjelaskan tentang statistik tambahan dan mengajarkan tentang bagaimana menggunakan R untuk bekerja dengan statistik tersebut. Penulis juga memperkenalkan grafik R di bagian ini.

Bagian 3 yaitu tentang Menarik Kesimpulan dari Data**.** Bagian ini membahas tujuan mendasar dari analisis statistic, untuk melampaui data dan membuat keputusan. Biasanya, data adalah ukuran sampel yang diambil dari populasi yang besar. Tujuannya adalah menggunakan data ini untuk mencari tahu apa yang terjadi dalam populasi. Bagian 4 yaitu tentang Probabilitas, probabilitas adalah dasar untuk analisis statistik dan pengambilan keputusan. Di Bagian 4, penulis menunjukkan bagaimana menerapkan probabilitas, khususnya di bidang pemodelan. R menyediakan satu set kaya kemampuan yang berhubungan dengan probabilitas.

Bagian terakhir dalam buku ini adalah bagian 5, yang terdiri dari dua bab. Pada bab pertama, penulis memberikan sepuluh tips kepada pengguna Excel untuk pindah ke R. Pada bab selanjutnya, membahas sepuluh topik terkait statistik dan R yang tidak sesuai dengan bab lain mana pun. Akhir kata semoga buku ini berguna bagi para pembaca.

Semarang, Februari 2022
Penulis

Dr. Ir. Agus Wibowo, M.Kom., M.Si., M.M.

DAFTAR ISI

BAGIAN 1
MEMULAI ANALISIS STATISTIK DENGAN R
BAB 1
DATA, STATISTIK, DAN KEPUTUSAN

Statistik? Itu semua tentang mengolah angka menjadi formula yang tampak misterius, bukan? Tidak juga. Statistik, pertama dan terpenting, adalah tentang pengambilan keputusan. Beberapa penghitung angka terlibat, tentu saja, tetapi tujuan utamanya adalah menggunakan angka untuk membuat keputusan. Ahli statistik melihat data dan bertanya-tanya apa yang dikatakan angka-angka itu. Jenis tren apa yang ada dalam data? Jenis prediksi apa yang mungkin? Kesimpulan apa yang bisa kita buat?

Untuk memahami data dan menjawab pertanyaan-pertanyaan ini, ahli statistik telah mengembangkan berbagai macam alat analisis. Tentang bagian menghitung angka: Jika Anda harus melakukannya melalui pensil dan kertas (atau dengan bantuan kalkulator saku), Anda akan segera berkecil hati dengan jumlah perhitungan yang terlibat dan kesalahan yang mungkin muncul. Perangkat lunak seperti R membantu Anda mengolah data dan menghitung angka. Sebagai bonus, R juga dapat membantu Anda memahami konsep statistik.

Dikembangkan secara khusus untuk analisis statistik, R adalah bahasa komputer yang mengimplementasikan banyak alat analisis yang telah dikembangkan oleh para ahli statistik untuk pengambilan keputusan. Saya menulis buku ini untuk menunjukkan bagaimana menggunakan alat-alat ini dalam pekerjaan Anda.

1.1 GAGASAN STATISTIK

Alat analisis yang disediakan R didasarkan pada konsep statistik yang saya bantu Anda jelajahi di sisa bab ini. Seperti yang akan Anda lihat, konsep-konsep ini didasarkan pada akal sehat.

Sampel dan populasi

Jika Anda menonton TV pada malam pemilihan, Anda tahu bahwa salah satu peristiwa utama adalah prediksi hasil segera setelah pemungutan suara ditutup (dan sebelum semua suara dihitung). Bagaimana para pakar hampir selalu melakukannya dengan benar?

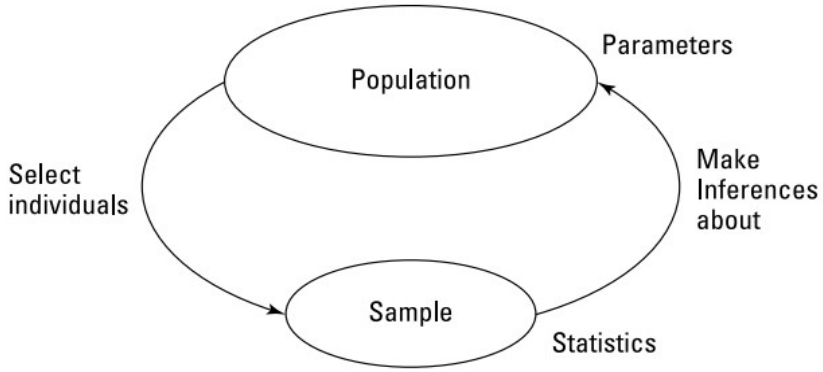
Idenya adalah untuk berbicara dengan sampel pemilih tepat setelah mereka memilih. Jika mereka jujur tentang bagaimana mereka menandai surat suara mereka, dan jika sampel mewakili populasi pemilih, analis dapat menggunakan data sampel untuk menarik kesimpulan tentang populasi. Singkatnya, itulah yang dimaksud dengan statistik — menggunakan data dari sampel untuk menarik kesimpulan tentang populasi.

Berikut contoh lain. Bayangkan bahwa tugas Anda adalah mencari rata-rata tinggi badan anak-anak berusia 10 tahun di Amerika Serikat. Karena Anda mungkin tidak memiliki waktu atau sumber daya untuk mengukur setiap anak, Anda akan mengukur tinggi badan sampel yang mewakili. Kemudian Anda akan menghitung rata-rata ketinggian tersebut dan menggunakan

rata-rata tersebut sebagai perkiraan rata-rata populasi. Memperkirakan rata-rata populasi adalah salah satu jenis inferensi yang dibuat oleh ahli statistik dari data sampel. Saya membahas inferensi secara lebieh rinci di bagian mendatang "Statistik Inferensial: Menguji Hipotesis."

Berikut ini beberapa terminologi penting: Sifat populasi (seperti rata-rata populasi) disebut parameter, dan sifat sampel (seperti rata-rata sampel) disebut statistik. Jika satu-satunya perhatian Anda adalah properti sampel (seperti tinggi badan anak-anak dalam sampel Anda), statistik yang Anda hitung bersifat deskriptif. Jika Anda khawatir tentang memperkirakan properti populasi, statistik Anda bersifat inferensial.

Sekarang untuk konvensi penting tentang notasi: Ahli statistik menggunakan huruf Yunani (μ, σ, ρ) untuk menyatakan parameter, dan huruf Inggris ($\bar{X}$, s, r) untuk menyatakan statistik. Gambar 1.1 merangkum hubungan antara populasi dan sampel, dan antara parameter dan statistik.



Gambar 1.1 Hubungan antara populasi, sampel, parameter, dan statistic

Variabel: Dependen dan independen

Variabel adalah sesuatu yang dapat mengambil lebih dari satu nilai — seperti usia Anda, nilai dolar terhadap mata uang lain, atau jumlah permainan yang dimenangkan tim olahraga favorit Anda. Sesuatu yang hanya memiliki satu nilai adalah konstanta. Para ilmuwan memberi tahu kami bahwa kecepatan cahaya adalah konstan, dan kami menggunakan konstanta untuk menghitung luas lingkaran.

Ahli statistik bekerja dengan variabel independen dan variabel dependen. Dalam penelitian atau eksperimen apa pun, Anda akan menemukan kedua jenis itu. Ahli statistik menilai hubungan di antara mereka. Misalnya, bayangkan metode pelatihan terkomputerisasi yang dirancang untuk meningkatkan IQ seseorang. Bagaimana seorang peneliti mengetahui apakah metode ini melakukan apa yang seharusnya dilakukan? Pertama, dia akan secara acak menetapkan sampel orang ke salah satu dari dua kelompok. Satu kelompok akan menerima metode pelatihan, dan yang lain akan menyelesaikan jenis aktivitas berbasis komputer lainnya — seperti membaca teks di situs web. Sebelum dan sesudah setiap kelompok menyelesaikan kegiatannya, peneliti mengukur IQ masing-masing orang. Apa yang terjadi selanjutnya? Saya membahas topik itu di bagian mendatang "Statistik Inferensial: Menguji Hipotesis."

Untuk saat ini, pahami bahwa variabel independen di sini adalah Jenis Aktivitas. Dua kemungkinan nilai dari variabel ini adalah IQ Training dan Reading Text. Variabel dependennya adalah perubahan IQ dari *Before ke After*. Variabel terikat adalah apa yang diukur oleh peneliti. Dalam sebuah eksperimen, variabel independen adalah apa yang dimanipulasi oleh peneliti. Dalam konteks lain, seorang peneliti tidak dapat memanipulasi variabel independen. Sebaliknya, ia mencatat nilai-nilai yang terjadi secara alami dari variabel independen dan bagaimana mereka mempengaruhi variabel dependen.

Secara umum, tujuannya adalah untuk mengetahui apakah perubahan pada variabel bebas berhubungan dengan perubahan variabel terikat. Dalam contoh-contoh yang muncul di seluruh buku ini, saya menunjukkan kepada Anda bagaimana menggunakan R untuk menghitung karakteristik kelompok skor, atau untuk membandingkan kelompok skor. Setiap kali saya menunjukkan sekelompok skor, saya berbicara tentang nilai-nilai variabel dependen.

Jenis data

Saat Anda melakukan pekerjaan statistik, Anda dapat menemukan empat jenis data. Dan ketika Anda bekerja dengan sebuah variabel, cara Anda bekerja dengannya bergantung pada jenis datanya. Jenis pertama adalah data nominal. Jika sekumpulan angka merupakan data nominal, angka tersebut adalah label – nilainya tidak berarti apa-apa. Pada tim olahraga, nomor jersey adalah nominal. Mereka hanya mengidentifikasi para pemain.

Jenis selanjutnya adalah data ordinal. Dalam tipe data ini, angka lebih dari sekedar label. Seperti yang mungkin dikatakan oleh nama “ordinal”, urutan nomornya penting. Jika saya meminta Anda untuk memberi peringkat sepuluh makanan dari yang paling Anda sukai (satu), hingga yang paling tidak Anda sukai (sepuluh), kami akan memiliki satu set data ordinal. Tetapi perbedaan antara makanan favorit ketiga Anda dan makanan favorit keempat Anda mungkin tidak sama dengan perbedaan antara favorit kesembilan dan favorit kesepuluh Anda. Jadi jenis data ini tidak memiliki interval yang sama dan perbedaan yang sama.

Data interval memberi kita perbedaan yang sama. Skala suhu Fahrenheit adalah contoh yang baik. Perbedaan antara 30^o dan 40^o sama dengan perbedaan antara 90^o dan 100^o. Jadi setiap derajat adalah interval. Orang terkadang terkejut saat mengetahui bahwa pada skala Fahrenheit, suhu 80^o tidak dua kali lebih panas dari 40^o. Agar pernyataan rasio ("dua kali lipat", "setengah dari") masuk akal, "nol" harus berarti tidak adanya sama sekali dari hal yang Anda ukur. Suhu 0^o F tidak berarti sama sekali tidak ada panas – itu hanya titik sembarang pada skala Fahrenheit. (Hal yang sama berlaku untuk Celcius.)

Jenis data keempat, rasio, memberikan titik nol yang berarti. Pada Skala Kelvin suhu, nol berarti "nol mutlak", di mana semua gerakan molekul (dasar panas) berhenti. Jadi 200^o Kelvin dua kali lebih panas dari 100^o Kelvin. Contoh lainnya adalah panjang. Delapan inci dua kali lebih panjang dari empat inci. "Nol inci" berarti "tidak ada panjang sama sekali." Variabel bebas atau variabel terikat dapat berupa nominal, ordinal, interval, atau rasio. Alat analisis yang Anda gunakan bergantung pada jenis data yang Anda gunakan.

Kemungkinan kecil

Ketika ahli statistik membuat keputusan, mereka menggunakan probabilitas untuk mengekspresikan keyakinan mereka tentang keputusan tersebut. Mereka tidak pernah bisa benar-benar yakin tentang apa yang mereka putuskan. Mereka hanya bisa memberi tahu Anda seberapa besar kemungkinan kesimpulan mereka. Apa yang kita maksud dengan probabilitas? Matematikawan dan filsuf mungkin memberi Anda definisi yang kompleks. Dalam pengalaman saya, bagaimanapun, cara terbaik untuk memahami probabilitas adalah dalam hal contoh.

Berikut ini contoh sederhananya: Jika Anda melempar koin, berapa peluang koin itu muncul? Jika koinnya adil, Anda mungkin berpikir bahwa Anda memiliki peluang kepala 50-50 dan peluang ekor 50-50. Dan Anda benar. Dalam hal jenis angka yang terkait dengan probabilitas, itu adalah $^1/_2$.

Pikirkan tentang melempar dadu yang adil (satu anggota dari sepasang dadu). Berapa probabilitas Anda mendapatkan angka 4? Nah, sebuah dadu memiliki enam wajah dan salah satunya adalah 4, jadi itu $^1/_6$. Contoh lain: Pilih satu kartu secara acak dari setumpuk standar 52 kartu. Berapa probabilitas bahwa itu adalah berlian? Setumpuk kartu memiliki empat setelan, jadi itu $^1/_4$.

Contoh-contoh ini memberi tahu Anda bahwa jika Anda ingin mengetahui probabilitas suatu peristiwa terjadi, hitung berapa banyak cara peristiwa itu dapat terjadi dan bagi dengan jumlah total peristiwa yang dapat terjadi. Dalam dua contoh pertama (kepala, 4), acara yang Anda minati hanya terjadi satu arah. Untuk koinnya, kita bagi satu per dua. Untuk dadu, kita bagi satu per enam. Pada contoh ketiga (diamond), event dapat terjadi 13 cara (Ace melalui King), jadi kita membagi 13 dengan 52 (mendapatkan $^1/_4$).

Sekarang untuk contoh yang sedikit lebih rumit. Melempar koin dan melempar dadu secara bersamaan. Berapa probabilitas ekor dan 4? Pikirkan tentang semua kemungkinan kejadian yang dapat terjadi ketika Anda melempar koin dan melempar dadu pada saat yang bersamaan. Anda bisa memiliki ekor dan 1 sampai 6, atau kepala dan 1 sampai 6. Itu menambahkan hingga 12 kemungkinan. Kombinasi ekor-dan-4 hanya dapat terjadi dengan satu cara. Jadi peluangnya adalah $^1/_2$.

Secara umum, rumus peluang terjadinya suatu peristiwa tertentu adalah

$$Mencegah = \frac{Banyaknya\ cara\ peristiwa\ itu\ terjadi}{Jumlah\ total\ kemungkinan\ terjadi}$$

Di awal bagian ini, saya mengatakan bahwa ahli statistik mengungkapkan keyakinan mereka tentang kesimpulan mereka dalam hal probabilitas, itulah sebabnya saya mengangkat semua ini di tempat pertama. Garis pemikiran ini mengarah pada probabilitas bersyarat — probabilitas bahwa suatu peristiwa terjadi mengingat beberapa peristiwa lain terjadi. Misalkan saya melempar dadu, lihatlah (agar Anda tidak melihatnya), dan beri tahu Anda bahwa saya melempar angka ganjil. Berapa peluang saya mendapatkan angka 5? Biasanya, probabilitas dari 5

adalah$^1/_6$, tapi "Saya menggulung angka ganjil" mempersempitnya. Informasi tersebut menghilangkan tiga angka genap (2, 4, 6) sebagai kemungkinan. Hanya tiga angka ganjil (1,3, 5) yang mungkin, jadi peluangnya adalah $^1/_3$.

Apa masalah besar tentang probabilitas bersyarat? Peran apa yang dimainkannya dalam analisis statistik? Baca terus.

1.2 STATISTIK INFERENSIAL: MENGUJI HIPOTESIS

Sebelum seorang ahli statistik melakukan penelitian, ia menyusun penjelasan tentatif — sebuah hipotesis yang memberi tahu mengapa data bisa keluar dengan cara tertentu. Setelah mengumpulkan semua data, ahli statistik harus memutuskan apakah akan menolak hipotesis atau tidak.

Keputusan itu adalah jawaban atas pertanyaan probabilitas bersyarat — berapa probabilitas memperoleh data, mengingat hipotesis ini benar? Ahli statistik memiliki alat yang menghitung probabilitas. Jika probabilitas ternyata rendah, ahli statistik menolak hipotesis. Kembali ke lempar koin sebagai contoh: Bayangkan Anda tertarik pada apakah koin tertentu adil — apakah koin itu memiliki peluang kepala atau ekor yang sama pada lemparan apa pun. Mari kita mulai dengan "Koin itu adil" sebagai hipotesis. Untuk menguji hipotesis, Anda akan melempar koin beberapa kali — katakanlah, seratus. 100 lemparan ini adalah data sampel. Jika koinnya adil (sesuai hipotesis), Anda akan mengharapkan 50 kepala dan 50 ekor.

Jika itu 99 kepala dan 1 ekor, Anda pasti akan menolak hipotesis koin yang adil: Probabilitas bersyarat 99 kepala dan 1 ekor diberikan koin yang adil sangat rendah. Tentu saja, koinnya masih bisa adil dan Anda bisa, secara kebetulan, mendapatkan pembagian 99-1, bukan? Tentu. Anda tidak pernah benar-benar tahu. Anda harus mengumpulkan data sampel (100 hasil lemparan) dan kemudian memutuskan. Keputusan Anda mungkin benar, atau mungkin tidak. Juri membuat keputusan seperti ini. Di Amerika Serikat, hipotesis awal adalah bahwa terdakwa tidak bersalah (“tidak bersalah sampai terbukti bersalah”). Pikirkan bukti sebagai "data." Anggota juri mempertimbangkan bukti dan menjawab pertanyaan probabilitas bersyarat: Berapa probabilitas bukti, mengingat terdakwa tidak bersalah? Jawaban mereka menentukan putusan.

Hipotesis nol dan alternatif

Pikirkan lagi tentang pelajaran melempar koin yang baru saja saya sebutkan. Data sampel adalah hasil dari 100 kali lemparan. Saya mengatakan bahwa kita dapat mulai dengan hipotesis bahwa koin itu adil. Titik awal ini disebut hipotesis nol. Notasi statistik untuk hipotesis nol adalah H0. Menurut hipotesis ini, setiap potongan kepala-ekor dalam data konsisten dengan koin yang adil. Anggap saja sebagai gagasan bahwa tidak ada data sampel yang luar biasa.

Hipotesis alternatif dimungkinkan — bahwa koin itu tidak adil dan dimuat untuk menghasilkan jumlah kepala dan ekor yang tidak sama. Hipotesis ini mengatakan bahwa setiap perpecahan kepala-ekor konsisten dengan koin yang tidak adil. Hipotesis alternatif ini disebut, percaya atau tidak, hipotesis alternatif. Notasi statistik untuk hipotesis alternatif adalah H1. Sekarang lempar koin 100 kali dan perhatikan jumlah kepala dan ekornya. Jika hasilnya kira-kira

90 kepala dan 10 ekor, ada baiknya menolak H_0. Jika hasilnya sekitar 50 ekor dan 50 ekor, jangan tolak H_0.

Ide serupa berlaku untuk contoh IQ yang saya berikan sebelumnya. Satu sampel menerima metode pelatihan IQ berbasis komputer, dan sampel lainnya berpartisipasi dalam aktivitas berbasis komputer yang berbeda — seperti membaca teks di situs web. Sebelum dan sesudah setiap kelompok menyelesaikan kegiatannya, peneliti mengukur IQ setiap orang. Hipotesis nol, H_0, adalah bahwa peningkatan satu kelompok tidak berbeda dari yang lain. Jika peningkatan lebih besar dengan pelatihan IQ dibandingkan dengan aktivitas lain — jauh lebih besar sehingga tidak mungkin keduanya tidak berbeda satu sama lain — tolak H_0. Jika tidak, jangan tolak H_0.

Perhatikan bahwa saya tidak mengatakan "terima H_0." Cara logika bekerja, Anda tidak pernah menerima hipotesis. Anda menolak H_0 atau tidak menolak H_0. Dalam persidangan juri, putusannya adalah "bersalah" (tolak hipotesis nol "tidak bersalah") atau "tidak bersalah" (jangan tolak H_0). "Tidak bersalah" (penerimaan hipotesis nol) bukanlah keputusan yang memungkinkan. Perhatikan juga bahwa dalam contoh pelemparan koin saya mengatakan “sekitar 50 kepala dan 50 ekor.” Apa artinya sekitar? Juga, saya mengatakan bahwa jika 90-10, tolak H0. Bagaimana dengan 85-15? 80-20? 70-30? Seberapa jauh perbedaan dari 50-50 split yang harus Anda lakukan untuk menolak H_0? Dalam contoh pelatihan IQ, seberapa besar peningkatan IQ yang harus dilakukan untuk menolak H_0?

Saya tidak akan menjawab pertanyaan-pertanyaan ini sekarang. Ahli statistik telah merumuskan aturan keputusan untuk situasi seperti ini, dan kami akan menjelajahi aturan tersebut di seluruh buku ini.

Dua jenis kesalahan

Kapanpun Anda mengevaluasi data dan memutuskan untuk menolak H_0 atau tidak menolak H_0, Anda tidak pernah bisa benar-benar yakin. Anda tidak pernah benar-benar tahu keadaan dunia yang "sebenarnya". Dalam contoh pelemparan koin, itu berarti Anda tidak dapat memastikan apakah koin itu adil atau tidak. Yang bisa Anda lakukan hanyalah membuat keputusan berdasarkan data sampel. Jika Anda ingin tahu pasti tentang koin, Anda harus memiliki data untuk seluruh populasi lemparan — yang berarti Anda harus terus melempar koin sampai akhir zaman.

Karena Anda tidak pernah yakin dengan keputusan Anda, Anda dapat membuat kesalahan dengan cara apa pun yang Anda putuskan. Seperti yang saya sebutkan sebelumnya, koinnya bisa adil dan Anda kebetulan mendapatkan 99 kepala dalam 100 lemparan. Itu tidak mungkin, dan itulah mengapa Anda menolak H_0 jika itu terjadi. Mungkin juga koinnya bias, namun Anda hanya melemparkan 50 kepala dalam 100 lemparan. Sekali lagi, itu tidak mungkin dan Anda tidak menolak H_0 dalam kasus itu. Meskipun kesalahan itu tidak mungkin, itu mungkin terjadi. Mereka mengintai di setiap studi yang melibatkan statistik inferensial. Ahli statistik menamakannya kesalahan Tipe I dan kesalahan Tipe II.

Jika Anda menolak H_0 dan tidak seharusnya, itu adalah kesalahan Tipe I. Dalam contoh koin, itu menolak hipotesis bahwa koin itu adil, padahal kenyataannya itu adalah koin yang adil.

Jika Anda tidak menolak H_0 dan seharusnya Anda menerimanya, itu adalah kesalahan Tipe II. Itu terjadi jika Anda tidak menolak hipotesis bahwa koin itu adil, dan pada kenyataannya itu bias. Bagaimana Anda tahu jika Anda telah membuat salah satu jenis kesalahan? Anda tidak — setidaknya tidak tepat setelah Anda membuat keputusan untuk menolak atau tidak menolak H_0. (Jika mungkin untuk mengetahuinya, Anda tidak akan membuat kesalahan sejak awal!) Yang dapat Anda lakukan hanyalah mengumpulkan lebih banyak data dan melihat apakah data tambahan tersebut konsisten dengan keputusan Anda.

Jika Anda menganggap H_0 sebagai kecenderungan untuk mempertahankan status quo dan tidak menafsirkan sesuatu sebagai sesuatu yang tidak biasa (tidak peduli bagaimana kelihatannya), kesalahan Tipe II berarti Anda melewatkan sesuatu yang besar. Faktanya, beberapa kesalahan ikonik adalah kesalahan Tipe II. Inilah yang saya maksud. Pada hari Tahun Baru tahun 1962, sebuah grup rock yang terdiri dari tiga gitaris dan seorang drummer mengikuti audisi di studio London dari sebuah perusahaan rekaman besar. Legenda mengatakan bahwa eksekutif rekaman tidak menyukai apa yang mereka dengar, tidak menyukai apa yang mereka lihat, dan percaya bahwa grup gitar akan keluar. Meskipun para musisi bermain sepenuh hati, grup tersebut gagal dalam audisi. Siapa kelompok itu? The Beatles! Dan itu adalah kesalahan Tipe II.

BAB 2
R: APA YANG DILAKUKAN DAN BAGAIMANA CARA KERJANYA

R adalah bahasa komputer. Ini adalah alat untuk melakukan perhitungan dan pemecahan angka yang mengatur panggung untuk analisis statistik dan pengambilan keputusan. Aspek penting dari analisis statistik adalah menyajikan hasil dengan cara yang dapat dipahami. Untuk alasan ini, grafik merupakan komponen utama dari R.

Ross Ihaka dan Robert Gentleman mengembangkan R pada 1990-an di University of Auckland, Selandia Baru. Didukung oleh Foundation for Statistical Computing, R semakin populer dari hari ke hari. RStudio adalah lingkungan pengembangan terintegrasi (IDE) open source untuk membuat dan menjalankan kode R. Ini tersedia dalam versi untuk Windows, Mac, dan Linux. Meskipun Anda tidak memerlukan IDE untuk bekerja dengan R, RStudio membuat hidup jauh lebih mudah.

2.1 MENGUNDUH R DAN RSTUDIO

Hal pertama yang pertama. Unduh R dari Jaringan Arsip R Komprehensif (CRAN). Di browser Anda, ketik alamat ini jika Anda bekerja di Windows:

```
cran.r-project.org/bin/windows/base/
```

Ketik yang ini jika Anda bekerja di Mac:

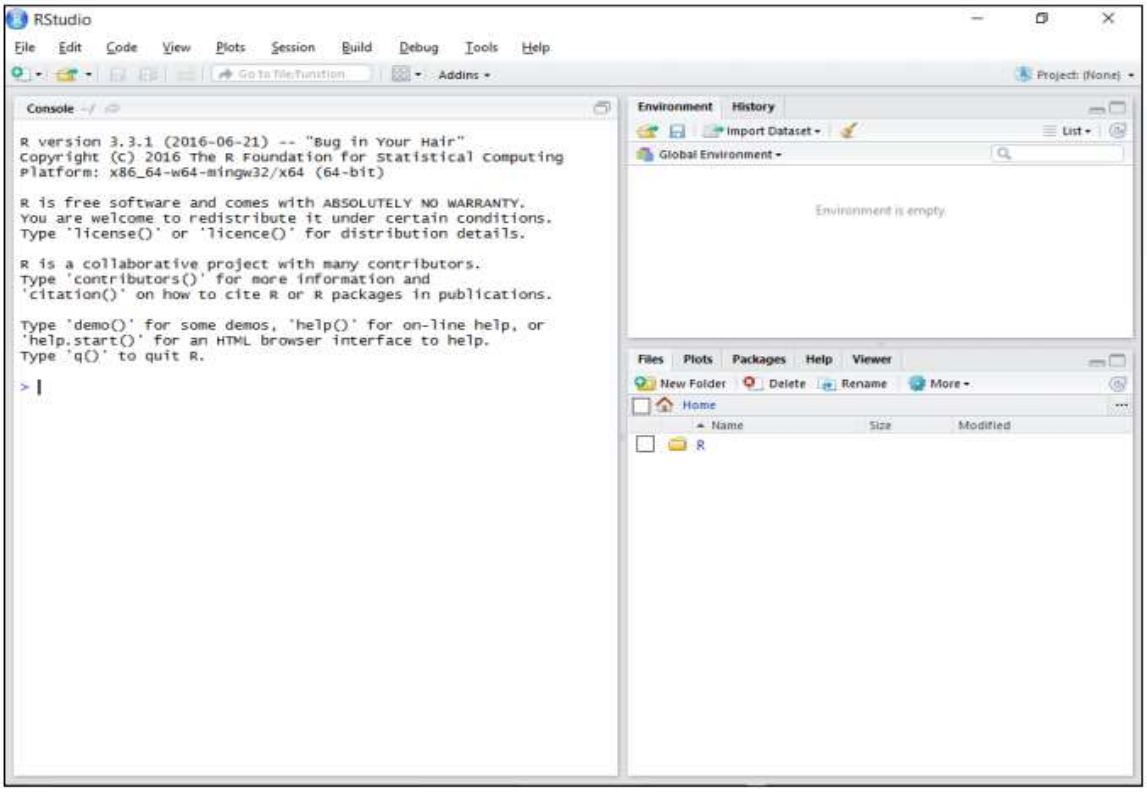
```
cran.r-project.org/bin/macosx/
```

Klik tautan untuk mengunduh R. Ini menempatkan file win.exe di komputer Windows Anda, atau file .pkg di Mac Anda. Dalam kedua kasus, ikuti prosedur instalasi biasa. Saat penginstalan selesai, pengguna Windows melihat ikon R di desktop mereka, pengguna Mac melihatnya di folder Aplikasi mereka. Kedua URL menyediakan tautan bermanfaat ke FAQ. URL terkait Windows juga tertaut ke "Instalasi dan instruksi lainnya". Sekarang untuk RStudio. Berikut URL-nya: www.rstudio.com/products/rstudio/download.

Klik tautan untuk penginstal untuk komputer Anda, dan ikuti kembali prosedur penginstalan yang biasa. Setelah instalasi RStudio selesai, klik ikon RStudio untuk membuka jendela yang ditunjukkan pada Gambar 2.1. Jika Anda sudah memiliki versi RStudio yang lebih lama dan Anda menjalani prosedur penginstalan ini, penginstalan akan memperbarui ke versi terbaru (dan Anda tidak perlu menghapus instalan versi yang lebih lama).

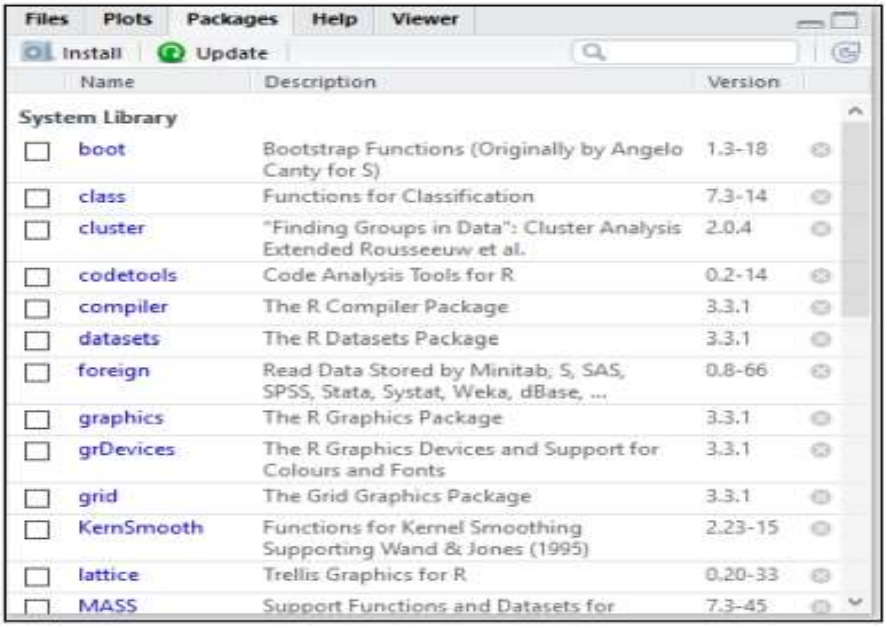
Panel Konsol besar di sebelah kiri menjalankan kode R. Salah satu cara untuk menjalankan kode R adalah dengan mengetikkannya langsung ke panel Konsol. Saya tunjukkan cara lain sebentar lagi. Dua panel lainnya memberikan informasi yang berguna saat Anda bekerja dengan R. Panel Lingkungan dan Sejarah ada di kanan atas. Tab Lingkungan melacak hal-hal yang Anda buat (yang disebut objek oleh R) saat Anda bekerja dengan R. Tab Riwayat melacak kode R yang Anda masukkan. Biasakan dengan kata objek. Segala sesuatu di R adalah objek.

Tab Files, Plots, Packages, dan Help berada di panel di kanan bawah. Tab File menunjukkan file yang Anda buat. Tab Plot menyimpan grafik yang Anda buat dari data Anda. Tab Paket menunjukkan add-on (disebut paket) yang Anda unduh sebagai bagian dari instalasi R. Ingatlah bahwa "diunduh" tidak berarti "siap digunakan". Untuk menggunakan kemampuan paket, satu langkah lagi diperlukan – dan percayalah – Anda pasti ingin menggunakan paket. Gambar 2.2 menunjukkan tab Paket. Paket-paket tersebut ada di perpustakaan pengguna (yang dapat Anda lihat pada gambar) atau perpustakaan sistem (yang harus Anda gulir ke bawah). Saya membahas paket nanti dalam bab ini.

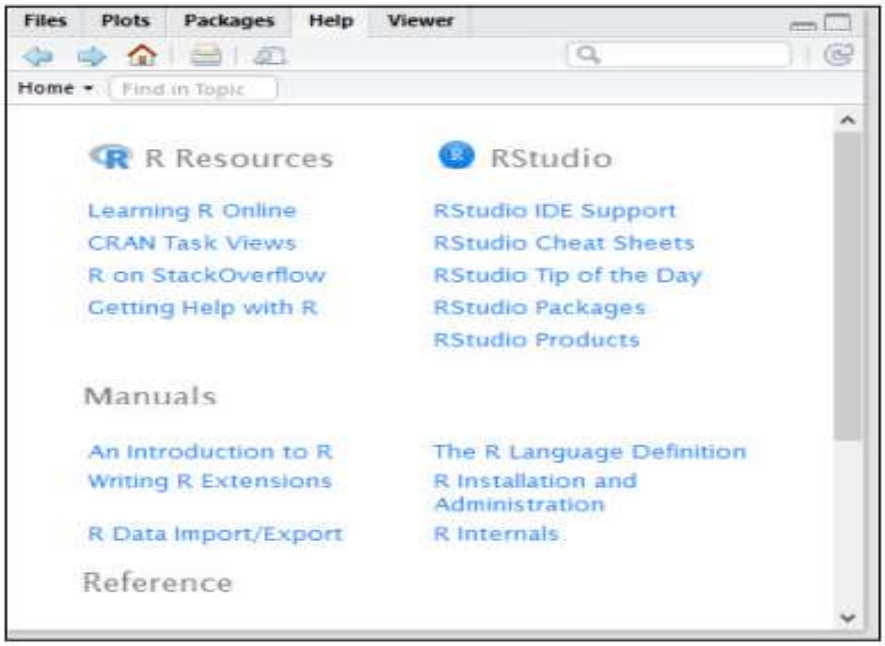


Gambar 2.1 RStudio, segera setelah Anda menginstalnya dan klik ikonnya.

Tab Bantuan, yang ditunjukkan pada Gambar 2.3, menyediakan tautan ke banyak informasi tentang R dan RStudio. Untuk memanfaatkan kekuatan penuh RStudio sebagai IDE, klik ikon yang lebih besar dari dua ikon di sudut kanan atas panel Konsol. Itu mengubah tampilan RStudio sehingga terlihat seperti Gambar 2.4.



Gambar 2.2 Tab Paket RStudio.

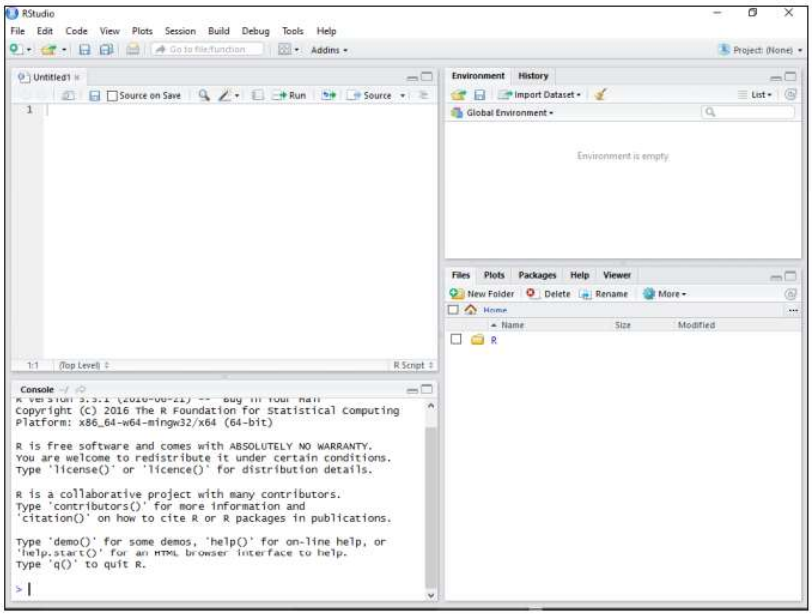


Gambar 2.3 Tab Bantuan RStudio.

Bagian atas panel Konsol dipindahkan ke kiri bawah. Panel baru di kiri atas adalah panel Scripts. Anda mengetik dan mengedit kode di panel Skrip dan tekan Ctrl+R (Command+Enter di Mac), lalu kode dijalankan di panel Konsol.

Ctrl+Enter berfungsi seperti Ctrl+R. Anda juga dapat memilih

Code ➪ Run Selected Line(s)



Gambar 2.4 RStudio, setelah Anda mengklik ikon yang lebih besar di sudut kanan atas panel Konsol.

2.2 SESI DENGAN R

Sebelum Anda mulai bekerja, pilih File ➪ Save As . . . dan kemudian simpan sebagai Sesi R Pertama Saya. Ini memberi label ulang tab di panel Script dengan nama file dan menambahkan ekstensi .R. Ini juga menyebabkan nama file (bersama dengan ekstensi .R) muncul di tab File.

Direktori kerja

Apa sebenarnya yang disimpan R, dan di mana R menyimpannya? Apa yang disimpan R disebut ruang kerja, yang merupakan lingkungan tempat Anda bekerja. R menyimpan ruang kerja di direktori kerja. Di Windows, direktori kerja default adalah

```
C:\Users\<User Name>\Documents
```

Jika Anda pernah lupa path ke direktori kerja Anda, ketik

```
> getwd()
```

di panel Konsol, dan R mengembalikan jalur di layar.

Di panel Konsol, Anda tidak mengetikkan panah penunjuk kanan di awal baris. Itu permintaan. Direktori kerja saya terlihat seperti ini:

```
> getwd()
[1] "C:/Users/Joseph Schmuller/Documents
```

Perhatikan ke arah mana garis miring itu miring. Mereka berlawanan dengan apa yang biasanya Anda lihat di jalur file Windows. Ini karena R menggunakan \ sebagai karakter pelarian, artinya apa pun yang mengikuti \ berarti sesuatu yang berbeda dari biasanya. Misalnya, \t di R berarti tombol Tab. Anda juga dapat menulis jalur file Windows di R sebagai:

```
C:\\Users\\<User Name>\\Documents
```

Jika mau, Anda dapat mengubah direktori kerja:

```
> setwd(<file path>)
```

Cara lain untuk mengubah direktori kerja adalah dengan memilih

```
Session➪ Set Working Directory➪ Choose Directory
```

Jadi mari kita mulai, sudah

Dan sekarang untuk beberapa R! Di jendela Script, ketik

```
x <- c(3,4,5)
```

lalu Ctrl+R.

Itu menempatkan baris berikut ke panel Konsol:

```
> x <- c(3,4,5)
```

Seperti yang saya sebutkan di Tip sebelumnya, panah penunjuk kanan (tanda lebih besar dari) adalah prompt yang disediakan R di panel Konsol. Anda tidak melihatnya di panel Scripts. Apa yang baru saja R lakukan? Tanda panah mengatakan bahwa x diberikan apa pun yang ada di sebelah kanan tanda panah. Jadi tanda panah adalah operator penugasan R. Di sebelah kanan tanda panah, c adalah singkatan dari concatenate, cara yang bagus untuk mengatakan "Ambil item apa pun yang ada di dalam tanda kurung dan satukan." Jadi himpunan angka 3, 4, 5 sekarang ditugaskan ke x.

R mengacu pada satu set angka seperti ini sebagai vektor. (Saya memberi tahu Anda lebih banyak tentang ini di bagian "Struktur R" nanti.) Anda dapat membaca baris kode R itu sebagai "x mendapatkan vektor 3, 4, 5." Ketik x ke dalam panel Scripts dan tekan Ctrl+R, dan inilah yang Anda lihat di panel Console:

```
> x
[1] 3 4 5
```

Tanda 1 dalam tanda kurung siku adalah label untuk nilai pertama di baris keluaran. Di sini Anda hanya memiliki satu nilai, tentu saja. Apa yang terjadi ketika R menghasilkan banyak nilai melalui banyak baris? Setiap baris mendapat label numerik tanda kurung, dan nomor tersebut sesuai dengan nilai pertama di baris. Misalnya, jika output terdiri dari 21 nilai dan nilai ke-18 adalah yang pertama pada baris kedua, baris kedua dimulai dengan [18]. Membuat vektor x menyebabkan tab Environment terlihat seperti Gambar 2.5.

Gambar 2.5 Tab RStudio Environment, setelah membuat vektor x.

Cara lain untuk melihat objek di lingkungan adalah dengan mengetik

```
> ls()
```

Sekarang Anda dapat bekerja dengan x. Pertama, tambahkan semua angka dalam vektor. Mengetik

```
sum(x)
```

di panel Script (ingat untuk mengikuti dengan Ctrl+R) jalankan baris berikut di panel Console:

```
> sum(x)
[1] 12
```

Bagaimana dengan rata-rata bilangan pada vektor x? itu

```
mean(x)
```

di panel Script, yang (ketika diikuti oleh Ctrl+R) dijalankan ke

```
>  mean(x)
[1] 4
```

di panel Konsol.

Saat Anda mengetik di panel Skrip atau di panel Konsol, Anda akan melihat bahwa informasi bermanfaat muncul. Saat Anda mendapatkan pengalaman dengan RStudio, Anda akan belajar cara menggunakan informasi itu.

Seperti yang saya tunjukkan di Bab 5, varians adalah ukuran seberapa banyak satu set angka berbeda dari rata-ratanya. Apa sebenarnya varians itu, dan bagaimana cara menghitungnya? Saya akan meninggalkannya untuk Bab 5. Untuk saat ini, inilah cara Anda menggunakan R untuk menghitung varians:

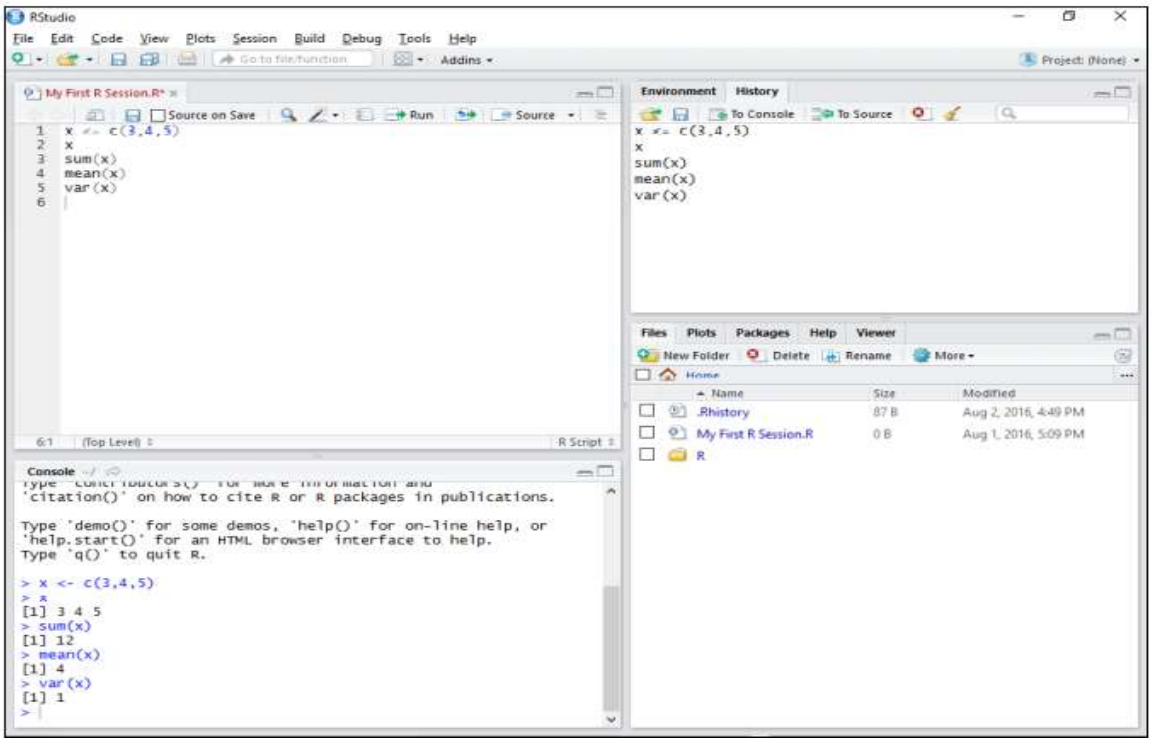
```
> var(x)
[1] 1
```

Dalam setiap kasus, Anda mengetik perintah dan R mengevaluasinya dan menampilkan hasilnya.

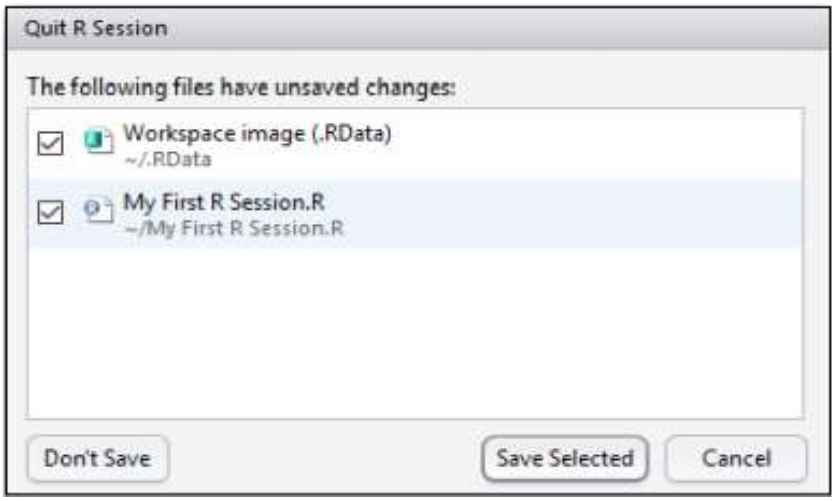
Gambar 2.6 menunjukkan seperti apa RStudio setelah semua perintah ini. Untuk mengakhiri sesi, pilih File Quit Session atau tekan Ctrl+Q. Seperti yang ditunjukkan Gambar 2.7, sebuah kotak dialog terbuka dan menanyakan apa yang ingin Anda simpan dari sesi. Menyimpan pilihan memungkinkan Anda untuk membuka kembali sesi yang Anda tinggalkan saat berikutnya Anda membuka RStudio (walaupun panel Konsol tidak menyimpan pekerjaan Anda). Cukup membantu, RStudio ini.

Ke depan, sebagian besar waktu saya tidak mengatakan "Ketik kode R ini ke dalam panel Script dan tekan Ctrl + Enter" setiap kali saya membawa Anda melalui sebuah contoh. Saya hanya menunjukkan kode dan outputnya, seperti pada contoh var().

Juga, terkadang saya menampilkan kode dengan > prompt, dan terkadang tanpa. Umumnya, saya menunjukkan prompt ketika saya ingin Anda melihat kode R dan hasilnya. Saya tidak menampilkan prompt ketika saya hanya ingin Anda melihat kode R yang saya buat di panel Scripts.



Gambar 2.6 RStudio setelah membuat dan bekerja dengan vektor.



Gambar 2.7 Kotak dialog Quit R Session.

Data hilang

Dalam contoh analisis statistik yang saya berikan, saya biasanya menangani skenario kasus terbaik di mana kumpulan data dalam kondisi yang baik dan memiliki semua data yang seharusnya mereka miliki. Namun, di dunia nyata, segala sesuatunya tidak selalu berjalan mulus. Seringkali, Anda menemukan kumpulan data yang memiliki nilai yang hilang karena satu dan lain alasan. R menunjukkan nilai yang hilang sebagai NA (untuk Tidak Tersedia).

Misalnya, berikut adalah beberapa data (dari kumpulan data yang jauh lebih besar) tentang kapasitas bagasi, dalam kaki kubik, dari sembilan kendaraan:

```
capacity <- c(14,13,14,13,16,NA,NA,20,NA)
```

Tiga dari kendaraan adalah van, dan istilah kapasitas bagasi tidak berlaku untuk mereka — oleh karena itu, tiga contoh NA. Inilah yang terjadi ketika Anda mencoba menemukan rata-rata grup ini:

```
> mean(capacity)
[1] NA
```

Untuk menemukan mean, Anda harus menghapus NA sebelum menghitung:

```
> mean(capacity, na.rm=TRUE)
[1] 15
```

Jadi rm di na.rm berarti "hapus" dan =TRUE berarti "selesaikan."

Untuk berjaga-jaga jika Anda harus memeriksa serangkaian skor untuk data yang hilang, fungsi is,na() melakukannya untuk Anda:

```
> is.na(capacity)
[1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
```

2.3 FUNGSI R

Di bagian sebelumnya, saya menggunakan c(), sum(), mean(), dan var(). Ini adalah contoh fungsi yang dibangun ke dalam R. Masing-masing terdiri dari nama fungsi yang langsung diikuti oleh tanda kurung. Di dalam kurung adalah argumen. Dalam konteks ini, “argumen” tidak berarti “ketidaksepakatan”, “konfrontasi”, atau semacamnya. Itu hanya istilah matematika untuk fungsi apa pun yang beroperasi. Bahkan jika suatu fungsi tidak membutuhkan argumen, Anda tetap menyertakan tanda kurung.

Empat fungsi R yang saya tunjukkan kepada Anda cukup sederhana dalam hal argumen dan outputnya. Namun, saat Anda bekerja dengan R, Anda menemukan fungsi yang membutuhkan lebih dari satu argumen. R menyediakan beberapa cara bagi Anda untuk menangani fungsi multiargumen. Salah satu caranya adalah dengan membuat daftar argumen dalam urutan kemunculannya dalam definisi fungsi. R menyebut pencocokan posisi ini.

Inilah yang saya maksud. Fungsi substr() membutuhkan tiga argumen. Yang pertama adalah string karakter seperti "abcdefg", yang disebut R sebagai vektor karakter. Argumen kedua adalah posisi awal dalam string (1 adalah posisi pertama, 2 adalah posisi kedua, dan seterusnya). Yang ketiga adalah posisi stop di dalam string (angka yang lebih besar dari atau sama dengan posisi awal).

Faktanya, jika Anda mengetik substr ke dalam panel Scripts, Anda akan melihat pesan pop-up yang terlihat seperti ini:

```
substr(x, start, stop)
Extract or replace substrings in a character vector
```

di mana x adalah vektor karakter. Fungsi ini mengembalikan substring, yang terdiri dari karakter antara posisi awal dan berhenti. Berikut ini contohnya:

```
> substr("abcdefg",2,4)
[1] "bcd"
```

Apa yang terjadi jika Anda menukar 2 dan 4?

```
> substr(“abcdefg”,4,2)
[1] “”
```

Hasil ini dapat dimengerti sepenuhnya: Tidak ada substring yang dapat dimulai di posisi keempat dan berhenti di posisi kedua.

Tetapi jika Anda menyebutkan argumennya, tidak masalah bagaimana Anda mengurutkannya:

```
> substr(“abcdefg”,stop=4,start=2)
[1] “bcd”
```

Bahkan ini berfungsi:

```
> substr(“abcdefg”,stop=4,start=2)
[1] “bcd”
```

Jadi, ketika Anda menggunakan suatu fungsi, Anda dapat menempatkan argumennya di luar urutan, jika Anda menamainya. R memanggil pencocokan kata kunci ini, yang berguna saat Anda menggunakan fungsi R yang memiliki banyak argumen. Jika Anda tidak dapat mengingat pesanan mereka, cukup gunakan nama mereka dan fungsinya berfungsi.

Jika Anda membutuhkan bantuan untuk fungsi tertentu — substr(), misalnya — ketik?substr dan lihat informasi bermanfaat yang muncul di tab Bantuan.

2.4 FUNGSI YANG DITENTUKAN PENGGUNA

Sebenarnya, ini bukan buku tentang pemrograman R. Namun, untuk kelengkapan, saya pikir setidaknya saya akan memberi tahu Anda bahwa Anda dapat membuat fungsi Anda sendiri di R, dan menunjukkan dasar-dasar membuatnya.

Bentuk fungsi R adalah

```
myfunction <- function(argument1, argument2, ...){
            statements
            return(object)
            }
```

Berikut adalah fungsi sederhana untuk menghitung jumlah kuadrat dari tiga bilangan:

```
sumofsquares <- function(x,y,z){
  sumsq <- sum(c(x^2,y^2,z^2))
  return(sumsq)
}
```

Ketik cuplikan itu ke dalam panel Scripts dan sorot. Kemudian tekan Ctrl+R. Cuplikan berikut muncul di panel Konsol:

```
> sumofsquares <- function(x,y,z ){
+   sumsq <- sum(c(x^2,y^2,z^2))
+   return(sumsq)
+ }
```

Setiap tanda plus adalah prompt kelanjutan. Itu hanya menunjukkan bahwa garis berlanjut dari baris sebelumnya.

Dan inilah cara menggunakan fungsinya:

```
> sumofsquares(3,4,5)
[1] 50
```

2.5 KOMENTAR

Komentar adalah cara untuk membubuhi keterangan kode. Mulailah komentar dengan simbol #, yang tentu saja adalah octothorpe. (Apa yang Anda katakan? "Hashtag"? Tentunya Anda bercanda.) Simbol ini memberitahu R untuk mengabaikan semua yang ada di sebelah kanannya.

Komentar sangat membantu seseorang yang harus membaca kode yang Anda tulis. Sebagai contoh:

```
sumofsquares <- function(x,y,z){ # list the arguments
  sumsq <- sum(c(x^2,y^2,z^2))  # perform the operations
  return(sumsq)    # return the value
}
```

Perhatian: Saya tidak menambahkan komentar pada baris kode dalam buku ini. Sebagai gantinya, saya memberikan deskripsi terperinci. Dalam buku seperti ini, saya merasa itulah cara paling efektif untuk menyampaikan pesan.

Seperti yang Anda bayangkan, menulis fungsi R dapat mencakup JAUH lebih dari yang saya paparkan di sini. Untuk mempelajari lebih lanjut, lihat R For Dummies, oleh Andrie de Vries dan Joris Meys (John Wiley & Sons).

2.6 STRUKTUR R

Saya menyebutkan di bagian "Fungsi R", di awal bab ini, bahwa fungsi R dapat memiliki banyak argumen. Ini juga kasus bahwa fungsi R dapat memiliki banyak keluaran. Untuk memahami kemungkinan keluaran (dan juga masukan), Anda harus memahami struktur yang bekerja dengan R.

Vektor

Vektor adalah struktur dasar R, dan saya menunjukkannya kepada Anda dalam contoh sebelumnya. Ini adalah array elemen data dengan tipe yang sama. Elemen data dalam sebuah vektor disebut komponen. Untuk membuat vektor, gunakan fungsi c(), seperti yang saya lakukan pada contoh sebelumnya:

```
> x <- c(3,4,5)
```

Di sini, tentu saja, komponennya adalah angka. Dalam vektor karakter, komponennya adalah string teks yang dikutip (“Moe,” “Larry,” “Curly”):

```
> stooges <- c("Moe","Larry", "Curly")
```

Sebenarnya, dalam contoh substr(), "abcdefg" adalah vektor karakter dengan satu elemen. Dimungkinkan juga untuk memiliki vektor logis, yang elemennya BENAR dan SALAH, atau singkatan T dan F:

```
> z <- c(T,F,T,F,T,T)
```

Untuk merujuk ke komponen tertentu dari sebuah vektor, ikuti nama vektor dengan nomor kurung:

```
> stooges[2]
[1] "Larry"
```

Vektor numerik

Selain c(), R menyediakan seq() dan rep() untuk pembuatan vektor numerik pintasan. Misalkan Anda ingin membuat vektor angka dari 10 hingga 30 tetapi Anda tidak ingin mengetik semua angka itu. Berikut cara melakukannya:

```
> y <- seq(10,30)
> y
 [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[18] 27 28 29 30
```

Di layar saya, dan mungkin juga di layar Anda, semua elemen di y muncul dalam satu baris. Namun, halaman yang dicetak tidak selebar panel Konsol. Oleh karena itu, saya memisahkan output menjadi dua baris. Saya melakukan itu di seluruh buku, jika perlu.
R memiliki sintaks khusus untuk vektor numerik yang elemennya bertambah 1:

```
> y <- 10:30
> y
 [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[18] 27 28 29 30
```

Jika Anda ingin elemen meningkat dalam langkah 2, gunakan seq seperti ini:

```
> w <- seq(10,30,2)
> w
 [1] 10 12 14 16 18 20 22 24 26 28 30
```

Anda mungkin ingin membuat vektor nilai berulang. Jika demikian, rep() adalah fungsi yang digunakan:

```
> trifecta <- c(6,8,2)
> repeated_trifecta <- rep(trifecta,4)
> repeated_trifecta
 [1] 6 8 2 6 8 2 6 8 2 6 8 2
```

Cara lain untuk menggunakan rep() adalah dengan menyediakan vektor sebagai argumen kedua. Ingat dari contoh sebelumnya bahwa x adalah vektor (3,4,5) Apa yang terjadi jika Anda memberikan x sebagai argumen kedua untuk rep()?

```
> repeated_trifecta <- rep(trifecta,x)
> repeated_trifecta
 [1] 6 6 6 8 8 8 8 2 2 2 2 2
```

Elemen pertama berulang tiga kali; elemen kedua, empat kali; dan elemen ketiga, lima kali.

Matriks

Matriks adalah array 2 dimensi dari elemen data dengan tipe yang sama. Dalam statistik, matriks berguna sebagai tabel yang menyimpan data. (Statistik tingkat lanjut memiliki aplikasi lain untuk matriks, tetapi itu di luar cakupan buku ini.)

Anda dapat memiliki matriks angka:

5	30	55	80
10	35	60	85
15	40	65	90
20	45	70	95
25	50	75	100

atau matriks string karakter:

"Moe"	"Larry"	"Curly"	"Shemp"
"Groucho"	"Harpo"	"Chico"	"Zeppo"
"Ace"	"King"	"Queen"	"Jack"

Angka-angka tersebut merupakan matriks 5 (baris) X 4 (kolom); matriks karakter string adalah 3 X 4.

Untuk membuat matriks numerik 5 X 4, pertama-tama Anda membuat vektor angka dari 5 hingga 100 dalam langkah 5:

```
> num_matrix <- seq(5,100,5)
```

Kemudian Anda menggunakan fungsi dim() untuk mengubah vektor menjadi matriks 2 dimensi:

```
> dim(num_matrix) <-c(5,4)
> num_matrix
     [,1] [,2] [,3] [,4]
[1,]    5   30   55   80
[2,]   10   35   60   85
[3,]   15   40   65   90
[4,]   20   45   70   95
[5,]   25   50   75  100
```

Perhatikan bagaimana R menampilkan nomor baris dalam kurung di sepanjang sisi, dan nomor kolom dalam kurung di sepanjang bagian atas. Transpose matriks menukar baris dengan kolom. Dalam R, fungsi t() menanganinya:

```
> t(num_matrix)
     [,1] [,2] [,3] [,4] [,5]
```

```
[1,]    5   10   15   20   25
[2,]   30   35   40   45   50
[3,]   55   60   65   70   75
[4,]   80   85   90   95  100
```

Fungsi matrix() menyediakan cara lain untuk membuat matriks:

```
> num_matrix <- matrix(seq(5,100,5),nrow=5)
> num_matrix
     [,1] [,2] [,3] [,4]
[1,]    5   30   55   80
[2,]   10   35   60   85
[3,]   15   40   65   90
[4,]   20   45   70   95
[5,]   25   50   75  100
```

Jika Anda menambahkan argumen byrow=T, R mengisi matriks dengan baris, seperti ini:

```
> num_matrix <- matrix(seq(5,100,5),nrow=5)
> num_matrix
     [,1] [,2] [,3] [,4]
[1,]    5   30   55   80
[2,]   10   35   60   85
[3,]   15   40   65   90
[4,]   20   45   70   95
[5,]   25   50   75  100
```

Bagaimana Anda merujuk ke komponen matriks tertentu? Anda mengetik nama matriks dan kemudian, dalam tanda kurung, nomor baris, koma, dan nomor kolom:

```
> num_matrix[5,4]
[1] 100
```

Faktor

Dalam Bab 1, saya menjelaskan empat jenis data: nominal, ordinal, interval, dan rasio. Dalam data nominal, angka hanyalah label, dan besarnya tidak memiliki signifikansi. Misalkan Anda sedang melakukan survei warna mata orang. Saat Anda merekam warna mata seseorang, Anda mencatat angka: 1 = kuning, 2 = biru, 3 = coklat, 4 = abu-abu, 5 = hijau, dan 6 = cokelat. Salah satu cara untuk memikirkan proses ini adalah bahwa warna mata adalah sebuah faktor, dan setiap warna adalah tingkatan dari faktor tersebut. Jadi dalam hal ini, faktor warna mata memiliki enam tingkatan. Faktor adalah suku R untuk variabel nominal (juga dikenal sebagai variabel kategori).

Sekarang bayangkan Anda telah menggunakan kode numerik untuk mentabulasi warna mata 14 orang dan kemudian mengubah kode tersebut menjadi vektor:

```
> eye_color <- c(2,2,4,1,5,5,5,6,1,3,6,3,1,4)
```

Selanjutnya, Anda menggunakan fungsi factor() untuk mengubah eye_color menjadi faktor:

```
> feye_color <- factor(eye_color)
```

Akhirnya, Anda menetapkan level faktor:

```
> levels(feye_color) <- c("amber","blue", "brown","gray","green",
  "hazel")
```

Sekarang, jika Anda memeriksa data warna mata dalam hal tingkat faktor, terlihat seperti ini:

```
> feye_color
 [1] blue  blue  gray  amber green green green hazel amber
[10] brown hazel brown amber gray
Levels: amber blue brown gray green hazel
```

Daftar

Di R, daftar adalah kumpulan objek yang tidak harus bertipe sama. Misalkan selain warna mata setiap orang dalam contoh di bagian sebelumnya, Anda mengumpulkan "skor empati" berdasarkan tes kepribadian. Skalanya berkisar dari 0 (kurang empati) sampai 100 (paling empati). Berikut adalah vektor untuk data empati orang-orang ini:

```
> empathy_score <- c(15,21,45,32,61,74,53,92,83,22,67,55,42,44)
```

Anda ingin menggabungkan vektor warna mata dalam bentuk kode, vektor warna mata dalam bentuk faktor, dan vektor skor empati menjadi satu koleksi bernama eyes_and_empati. Anda menggunakan fungsi list() untuk tugas ini:

```
> eyes_and_empathy <- list(eyes_code=eye_color, eyes=feye_color,
  empathy=empathy_score)
```

Perhatikan bahwa Anda memberi nama setiap argumen (kode_mata, mata, dan empati). Hal ini menyebabkan R menggunakan nama tersebut sebagai nama komponen daftar.

Berikut tampilan daftarnya:

```
> eyes_and_empathy
$eyes_code
 [1] 2 2 4 1 5 5 5 6 1 3 6 3 1 4

$eyes
 [1] blue  blue  gray  amber green green green hazel amber
[10] brown hazel brown amber gray
Levels: amber blue brown gray green hazel

$empathy
 [1] 15 21 45 32 61 74 53 92 83 22 67 55 42 44
```

Seperti yang Anda lihat, R menggunakan tanda dolar ($) untuk menunjukkan setiap komponen daftar. Jadi, jika Anda ingin merujuk ke komponen daftar, ketikkan nama daftar, tanda dolar, dan

nama komponen:

```
> eyes_and_empathy$empathy
 [1] 15 21 45 32 61 74 53 92 83 22 67 55 42 44
```

Bagaimana dengan memusatkan perhatian pada skor tertentu, seperti yang keempat? Saya pikir Anda dapat melihat ke mana arahnya:

```
> eyes_and_empathy$empathy[4]
[1] 32
```

Daftar dan statistik

Daftar penting karena banyak fungsi statistik mengembalikan daftar objek. Salah satu fungsi statistik adalah t.test(). Dalam Bab 10, saya menjelaskan tes ini dan teori di baliknya. Untuk saat ini, hanya berkonsentrasi pada outputnya.

Saya menggunakan tes ini untuk melihat apakah rata-rata skor empati berbeda dari angka arbitrer — 30, misalnya. Berikut tesnya:

```
> t.result <- t.test(eyes_and_empathy$empathy, mu = 30)
```

Mari kita periksa outputnya:

```
> t.result

        One Sample t-test
```

```
data:  eyes_and_empathy$empathy
t = 3.2549, df = 13, p-value = 0.006269
alternative hypothesis: true mean is not equal to 30
95 percent confidence interval:
 36.86936 63.98778
sample estimates:
mean of x
 50.42857
```

Tanpa masuk ke detailnya, pahami bahwa output ini, t.result, adalah daftar. Untuk menunjukkan ini, Anda menggunakan $ untuk fokus pada beberapa komponen:

```
> t.result$data.name
[1] "eyes_and_empathy$empathy"
> t.result$p.value
[1] 0.006269396
> t.result$statistic
       t
3.254853
```

Bingkai data

Daftar adalah cara yang baik untuk mengumpulkan data. Sebuah bingkai data bahkan lebih baik. Mengapa? Saat memikirkan data untuk sekelompok individu — seperti 14 orang dalam contoh di bagian sebelumnya — Anda biasanya berpikir dalam hal kolom yang mewakili variabel data (seperti kode_mata, mata, dan empati) dan baris yang mewakili indeks. - video. Dan itu adalah bingkai data. Jika istilah kumpulan data atau matriks data muncul di benak Anda, Anda sudah cukup memahaminya.

Fungsi data.frame() bekerja dengan vektor yang ada untuk menyelesaikan pekerjaan:

```
> e <- data.frame(eye_color,feye_color,empathy_score)
> e
   eye_color feye_color empathy_score
1          2       blue            15
2          2       blue            21
3          4       gray            45
4          1      amber            32
5          5      green            61
6          5      green            74
7          5      green            53
8          6      hazel            92
9          1      amber            83
10         3      brown            22
11         6      hazel            67
12         3      brown            55
13         1      amber            42
14         4       gray            44
```

Ingin skor empati untuk orang ketujuh? Itu

```
> e[7,3]
[1] 53
```

Bagaimana dengan semua informasi untuk orang ketujuh:

```
> e[7,]
  eye_color feye_color empathy_score
7         5      green            53
```

Mengedit bingkai data: Terlihat seperti spreadsheet (tetapi tidak)

R menyediakan cara untuk memodifikasi bingkai data dengan cepat. Fungsi edit() membuka jendela Editor Data yang sangat mirip dengan spreadsheet, dan Anda dapat membuat perubahan dalam sel. Gambar 2.8 menunjukkan apa yang terjadi saat Anda mengetik

```
> edit(e)
```

	eye_color	feye_color	empathy_score	var4	var5	var6
1	2	blue	15			
2	2	blue	21			
3	4	gray	45			
4	1	amber	32			
5	5	green	61			
6	5	green	74			
7	5	green	53			
8	6	hazel	92			
9	1	amber	83			
10	3	brown	22			
11	6	hazel	67			
12	3	brown	55			
13						
14						
15						
16						
17						
18						
19						

Gambar 2.8: Fungsi edit() membuka tampilan seperti spreadsheet dari bingkai data.

Anda harus menutup jendela Data Editor untuk melanjutkan. Untuk pengguna Mac: RStudio versi Mac memerlukan sistem X Window untuk beberapa fungsi, seperti edit(), agar berfungsi. Apple dulu menyertakan kemampuan ini dengan Mac, tetapi sekarang tidak lagi. Saat ini, Anda harus mengunduh dan menginstal XQuartz.

Mengekstrak data dari bingkai data

Misalkan Anda ingin melakukan pemeriksaan cepat pada skor empati rata-rata untuk orang dengan mata biru versus orang dengan mata hijau versus orang dengan mata cokelat. Tugas pertama adalah mengekstrak skor empati untuk setiap warna mata dan membuat vektor:

```
> e.blue <- e$empathy_score[e$feye_color=="blue"]
> e.green <- e$empathy_score[e$feye_color=="green"]
> e.hazel <- e$empathy_score[e$feye_color=="hazel"]
```

Perhatikan tanda sama dengan ganda (==) dalam tanda kurung. Ini adalah operator logika. Anggap saja sebagai "jika e$feye_color sama dengan 'biru.'" Tanda sama dengan ganda (a==b) membedakan operator logika (“jika a sama dengan b”) dari operator penugasan (a=b; “tetapkan a sama dengan b”).

Selanjutnya, Anda membuat vektor rata-rata:

```
> e.averages <- c(mean(e.blue),mean(e.green),mean(e.hazel))
```

Kemudian Anda menggunakan length() untuk membuat vektor jumlah skor di setiap kelompok warna mata:

```
> e.amounts <- c(length(e.blue), length(e.green),
  length(e.hazel))
```

Dan kemudian Anda membuat vektor warna:

```
> colors <- c("blue","green","hazel")
```

Sekarang Anda membuat kerangka data 3 kolom dengan warna dalam satu kolom, rata-rata empati yang sesuai di kolom berikutnya, dan jumlah skor di setiap kelompok warna mata di kolom terakhir:

```
> e.averages.frame <- data.frame(color=colors,
  average=e.averages, n=e.amounts)
```

Seperti halnya dengan daftar, penamaan argumen menetapkan nama argumen ke komponen bingkai data (vektor, yang muncul di layar sebagai kolom).
Dan inilah tampilannya:

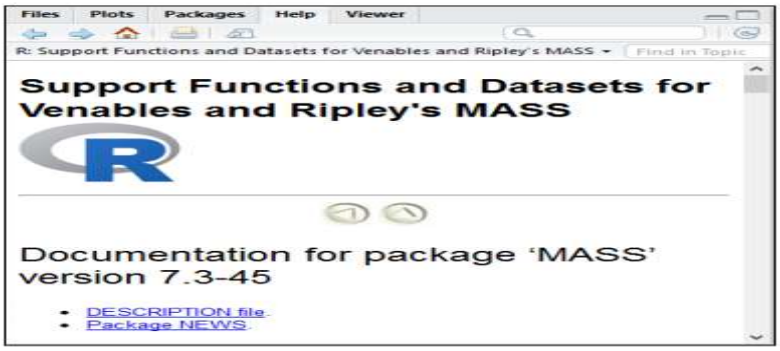
```
> e.averages.frame
  color  average n
1  blue 18.00000 2
2 green 62.66667 3
3 hazel 79.50000 2
```

2.7 PAKET

Paket adalah kumpulan fungsi dan data yang menambah R. Jika Anda seorang ilmuwan data yang bercita-cita tinggi dan Anda sedang mencari data untuk dikerjakan, Anda akan menemukan banyak bingkai data dalam paket R. Jika Anda mencari fungsi statistik khusus yang tidak ada dalam instalasi R dasar, Anda mungkin dapat menemukannya dalam sebuah paket.

R menyimpan paket dalam direktori yang disebut perpustakaan. Bagaimana Anda memasukkan paket ke perpustakaan? Klik tab Paket di File, Plot, Paket, dan panel Bantuan. (Lihat Gambar 2.2.) Dalam contoh yang akan datang, saya menggunakan paket MASS yang terkenal, yang berisi lebih dari 150 bingkai data dari berbagai bidang.

Jika Anda ingin melihat apa yang ada di dalam paket MASS, klik MASS di tab Packages. (Ada di bagian System Library dari tab ini.) Itu akan membuka halaman di tab Help, yang muncul di Gambar 2.9.



Gambar 2.9 Tab Bantuan, menampilkan informasi tentang paket MASS.

Menggulir ke bawah menunjukkan nama frame data dan fungsi. Mengklik nama bingkai data akan membuka halaman informasi tentangnya. Kembali ke tab Packages, Anda klik kotak centang di sebelah MASS untuk menginstal paket. Itu menyebabkan baris ini muncul di jendela Konsol:

```
> library("MASS", lib.loc="C:/Program Files/R/R-3.3.1/library")
```

Dan paket MASS diinstal.

Salah satu frame data di MASS bernama anoreksia. Ini berisi data berat badan untuk 72 pasien anoreksia wanita muda. Setiap pasien menyelesaikan satu dari tiga jenis terapi. Seperti apa bingkai data itu? Anda mengetik baris ini ke panel Konsol:

```
> edit(anorexia)
```

untuk membuka jendela Data Editor, yang ditunjukkan pada Gambar 2.10.

Data Editor

File Edit Help

	Treat	Prewt	Postwt	var4	var5	var6	var7
1	Cont	80.7	80.2				
2	Cont	89.4	80.1				
3	Cont	91.8	86.4				
4	Cont	74	86.3				
5	Cont	78.1	76.1				
6	Cont	88.3	78.1				
7	Cont	87.3	75.1				
8	Cont	75.1	86.7				
9	Cont	80.6	73.5				
10	Cont	78.4	84.6				
11	Cont	77.6	77.4				
12	Cont	88.7	79.5				
13	Cont	81.3	89.6				
14	Cont	78.1	81.4				
15	Cont	70.5	81.8				
16	Cont	77.3	77.3				
17	Cont	85.2	84.2				
18	Cont	86	75.4				
19	Cont	84.1	79.5				

Gambar 2.10 Kerangka data anoreksia dalam paket MASS.

Sepertinya itu hanya menunggu Anda untuk menganalisis, bukan? Saya belum membahas analisis statistik apa pun, tetapi Anda dapat bekerja sedikit pada kerangka data ini dengan apa yang telah saya tunjukkan kepada Anda.

Kerangka data memberikan bobot pra-terapi (Prewt) dan bobot pasca-terapi (Postwt) untuk setiap pasien. Bagaimana dengan perubahan beratnya? Dapatkah R menghitungnya untuk setiap pasien? Tentu saja!

```
> anorexia$Postwt-anorexia$Prewt
 [1]  -0.5  -9.3  -5.4  12.3  -2.0 -10.2 -12.2  11.6  -7.1
[10]   6.2  -0.2  -9.2   8.3   3.3  11.3   0.0  -1.0 -10.6
[19]  -4.6  -6.7   2.8   0.3   1.8   3.7  15.9 -10.2   1.7
[28]   0.7  -0.1  -0.7  -3.5  14.9   3.5  17.1  -7.6   1.6
[37]  11.7   6.1   1.1  -4.0  20.9  -9.1   2.1  -1.4   1.4
[46]  -0.3  -3.7  -0.8   2.4  12.6   1.9   3.9   0.1  15.4
[55]  -0.7  11.4  11.0   5.5   9.4  13.6  -2.9  -0.1   7.4
[64]  21.5  -5.3  -3.8  13.4  13.1   9.0   3.9   5.7  10.7
```

Hmmm. Ingat uji-t yang saya tunjukkan sebelumnya di bab ini? Saya menggunakannya di sini untuk melihat apakah perubahan berat badan sebelum terapi/pasca terapi berbeda dari 0. Anda akan berharap bahwa, rata-rata, perubahannya positif. Berikut uji-t:

```
> t.test(anorexia$Postwt-anorexia$Prewt, mu=0)

  One Sample t-test

data:  anorexia$Postwt - anorexia$Prewt
t = 2.9376, df = 71, p-value = 0.004458
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.8878354 4.6399424
sample estimates:
mean of x
 2.763889
```

Hasil uji-t menunjukkan bahwa rata-rata perubahan berat badan adalah positif (2,763889 lbs). Nilai t yang tinggi (2,9376), bersama dengan nilai p yang rendah (0,004458), menunjukkan bahwa perubahan ini signifikan secara statistik. (Apa artinya itu?) Jika saya memberi tahu Anda lagi, saya akan bertindak lebih dulu. (Lihat Bab 10 untuk detailnya.)

Ini hal lain: Saya mengatakan bahwa setiap pasien menyelesaikan satu dari tiga jenis terapi. Apakah satu terapi lebih efektif daripada yang lain? Atau apakah mereka hampir sama? Sekarang saya benar-benar akan mendahului diri saya sendiri! (Penjelasan itu ada di Bab 12, tetapi lihat bagian "Rumus R," nanti di bab ini.)

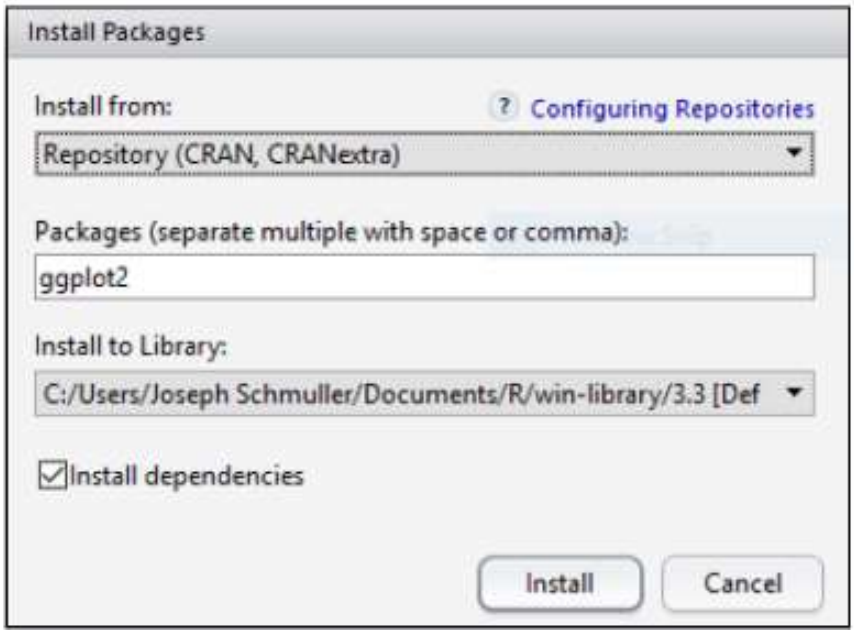
Lebih Banyak Paket

Komunitas R sangat aktif. Anggotanya membuat dan menyumbangkan paket baru yang berguna setiap saat ke CRAN (Comprehensive R Archive Network). Jadi tidak setiap paket R ada di tab Paket RStudio.

Saat Anda mengetahui tentang paket baru yang menurut Anda mungkin berguna, mudah untuk menginstalnya ke perpustakaan Anda. Saya mengilustrasikannya dengan menginstal ggplot2, paket berguna yang memperluas kemampuan grafis R.

Salah satu cara untuk menginstalnya adalah melalui tab Packages. (Lihat Gambar 2.2.) Klik ikon Instal di sudut kiri atas tab. Ini akan membuka kotak dialog Install Packages, yang ditunjukkan pada Gambar 2.11.

Cara lain untuk membuka kotak dialog Instal Paket adalah dengan memilih Instal Paket dari menu Alat di bilah menu di bagian atas RStudio.



Gambar 2.11 Kotak dialog Instal Paket.

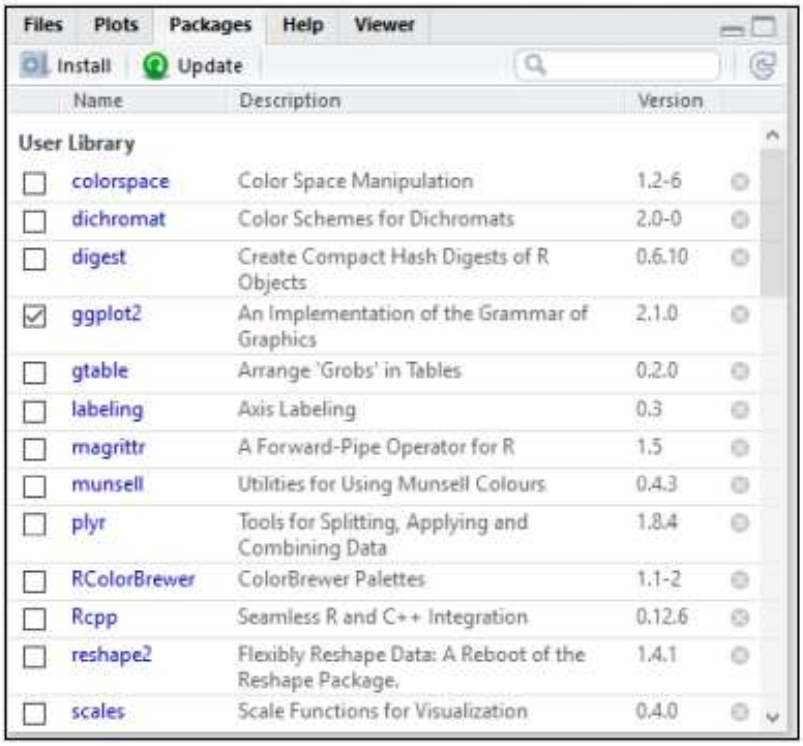
Di bidang Paket, saya mengetik ggplot2. Klik Instal, dan baris berikut muncul di panel Konsol:

```
> install.packages("ggplot2")
```

Akan tetapi, sulit untuk melihat baris ini, karena banyak hal lain terjadi segera di panel Konsol dan di bilah status di layar. Setelah semua selesai, ggplot2 ada di tab Packages. Langkah terakhir adalah mengklik kotak centang di sebelah ggplot2 untuk meletakkannya di perpustakaan. Kemudian Anda dapat menggunakan paket tersebut. Gambar 2-12 menunjukkan tab Packages dengan ggplot2 dan kotak yang dicentang.

Mengklik kotak centang menempatkan baris berikut di panel Konsol:

```
> library("ggplot2", lib.loc="~/R/win-library/3.3")
```



Gambar 2.12 Tab Paket setelah menginstal ggplot2 dan meletakkannya di perpustakaan.

Cara lain untuk memulai proses instalasi adalah dengan mengetik langsung ke panel Konsol.

```
> install.packages("ggplot2")
```

2.8 RUMUS R

Dalam Bab 1, saya membahas variabel bebas dan variabel terikat. Saya menunjukkan bahwa, dalam sebuah eksperimen, variabel independen adalah apa yang dimanipulasi oleh peneliti dan variabel dependen adalah apa yang diukur oleh peneliti. Dalam contoh anoreksia sebelumnya, Perlakukan (jenis terapi) adalah variabel independen, dan Postwt-Prewt (berat badan pasca terapi dikurangi berat badan sebelum terapi) adalah variabel terikat. Dalam istilah praktis, "memanipulasi" berarti bahwa peneliti secara acak menugaskan setiap pasien anoreksia ke salah satu dari tiga terapi.

Dalam jenis penelitian lain, peneliti tidak dapat memanipulasi variabel independen. Sebaliknya, dia mencatat nilai-nilai yang terjadi secara alami dari variabel independen dan menilai efeknya pada variabel dependen. Pada contoh warna mata dan empati sebelumnya, warna mata adalah variabel bebas dan skor empati adalah variabel terikat. Rumus R menggabungkan konsep-konsep ini dan merupakan dasar dari banyak fungsi statistik R dan fungsi grafik. Ini adalah struktur dasar dari rumus R:

```
function(dependent_var ~ independent_var, data=data_frame)
```

Baca operator tilde (~) sebagai “tergantung pada.”

Kerangka data anoreksia memberikan contoh. Untuk menganalisis perbedaan efektivitas ketiga terapi untuk anoreksia, saya akan menggunakan teknik yang disebut analisis varians. (Ini dia, maju sendiri!) Fungsi R untuk ini bernama aov(), dan inilah cara menggunakannya:

```
> aov(Postwt-Prewt ~ Treat, data=anorexia)
```

Tapi ini baru awal dari analisis. Bab 12 memiliki semua detail, serta pemikiran statistik di baliknya.

2.9 *READ AND WRITE*

Sebelum saya menutup bab ini tentang kemampuan R, saya harus memberi tahu Anda cara mengimpor data dari format lain serta cara mengekspor data ke format tersebut. Bentuk umum dari fungsi R untuk membaca file adalah:

```
> read.<format>(“File Name”, arg1, arg2, ...)
```

Bentuk umum dari fungsi R untuk menulis data ke file adalah:

```
> write.<format>(dataframe, “File Name”, arg1, arg2, ...)
```

Di bagian ini, saya membahas spreadsheet, file CSV (nilai yang dipisahkan koma), dan file teks. <format> adalah xlsx, csv, atau tabel. Argumen setelah "Nama File" adalah argumen opsional yang bervariasi untuk format yang berbeda.

Spreadsheet

Informasi di bagian ini akan penting bagi Anda jika Anda telah membaca klasik abadi saya, Analisis Statistik dengan Excel For Dummies (John Wiley & Sons). (Oke, jadi itu adalah plug tak tahu malu untuk klasik abadi saya.) Jika Anda memiliki data pada spreadsheet dan Anda ingin menganalisis dengan R, perhatikan baik-baik.

Urutan pertama bisnis adalah mengunduh paket xlsx dan meletakkannya di perpustakaan. Lihat bagian “Paket Lainnya” di awal bab ini, untuk mengetahui lebih lanjut tentang cara melakukannya. Di drive C saya, saya memiliki spreadsheet bernama Scores di folder bernama Spreadsheets. Itu ada di Sheet1 dari lembar kerja. Ini memegang skor kuis matematika dan skor kuis sains untuk sepuluh siswa. Untuk membaca spreadsheet itu ke dalam R, kodenya adalah:

```
> scores_frame <- read.xlsx("C:/Spreadsheets/Scores.xlsx",
  sheetName="Sheet1")
```

Inilah bingkai data itu:

```
> scores_frame
   Student Math_Score Science_Score
1        1         85            90
2        2         91            87
3        3         78            75
4        4         88            78
5        5         93            99
6        6         82            89
7        7         67            71
8        8         79            84
9        9         89            88
10      10         98            97
```

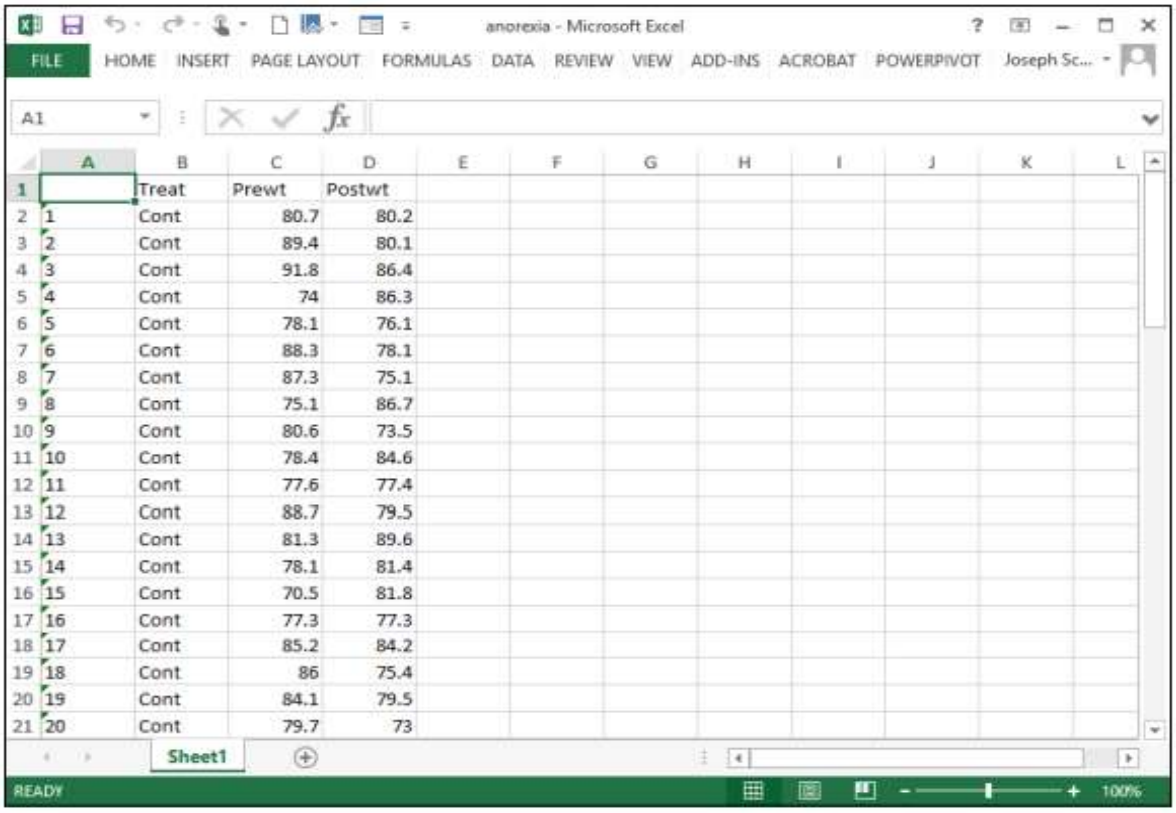
Seperti halnya dengan kerangka data apa pun, jika Anda menginginkan skor matematika untuk siswa keempat, cukup

```
> scores_frame$Math_Score[4]
[1] 88
```

Paket xlsx juga memungkinkan penulisan ke spreadsheet. Jadi, jika Anda ingin teman-teman Excel-sentris Anda melihat kerangka data anoreksia, inilah yang Anda lakukan:

```
> write.xlsx(anorexia,"C:/Spreadsheets/anorexia.xlsx")
```

Baris ini menempatkan bingkai data ke dalam spreadsheet di folder yang ditunjukkan pada drive C. Jika Anda tidak percaya, Gambar 2.13 menunjukkan seperti apa spreadsheet itu.



	A	B	C	D
1		Treat	Prewt	Postwt
2	1	Cont	80.7	80.2
3	2	Cont	89.4	80.1
4	3	Cont	91.8	86.4
5	4	Cont	74	86.3
6	5	Cont	78.1	76.1
7	6	Cont	88.3	78.1
8	7	Cont	87.3	75.1
9	8	Cont	75.1	86.7
10	9	Cont	80.6	73.5
11	10	Cont	78.4	84.6
12	11	Cont	77.6	77.4
13	12	Cont	88.7	79.5
14	13	Cont	81.3	89.6
15	14	Cont	78.1	81.4
16	15	Cont	70.5	81.8
17	16	Cont	77.3	77.3
18	17	Cont	85.2	84.2
19	18	Cont	86	75.4
20	19	Cont	84.1	79.5
21	20	Cont	79.7	73

Gambar 2.13 Bingkai data anoreksia, diekspor ke spreadsheet Excel.

File CSV

Fungsi untuk membaca dan menulis file CSV dan file teks ada di dalam instalasi R, jadi tidak diperlukan paket tambahan. File CSV terlihat seperti spreadsheet saat Anda membukanya di Excel. Sebenarnya, saya membuat file CSV untuk spreadsheet Skor dengan menyimpan spreadsheet sebagai file CSV di folder CSVFiles di drive C. (Untuk melihat semua koma, Anda harus membukanya di editor teks, seperti Notepad++.)
Berikut cara membaca file CSV tersebut ke dalam R:

```
> read.csv("C:/CSVFiles/Scores.csv")
   Student Math_Score Science_Score
1        1         85            90
2        2         91            87
3        3         78            75
4        4         88            78
5        5         93            99
6        6         82            89
7        7         67            71
8        8         79            84
9        9         89            88
10      10         98            97
```

Untuk menulis bingkai data anoreksia ke file CSV,

```
> write.csv(anorexia,"C:/CSVFiles/anorexia.csv")
```

File teks

Jika Anda memiliki beberapa data yang disimpan dalam file teks, R dapat mengimpornya ke dalam bingkai data. Fungsi read.table() menyelesaikannya. Saya menyimpan data Skor sebagai file teks dalam direktori bernama TextFiles. Begini cara R mengubahnya menjadi bingkai data:

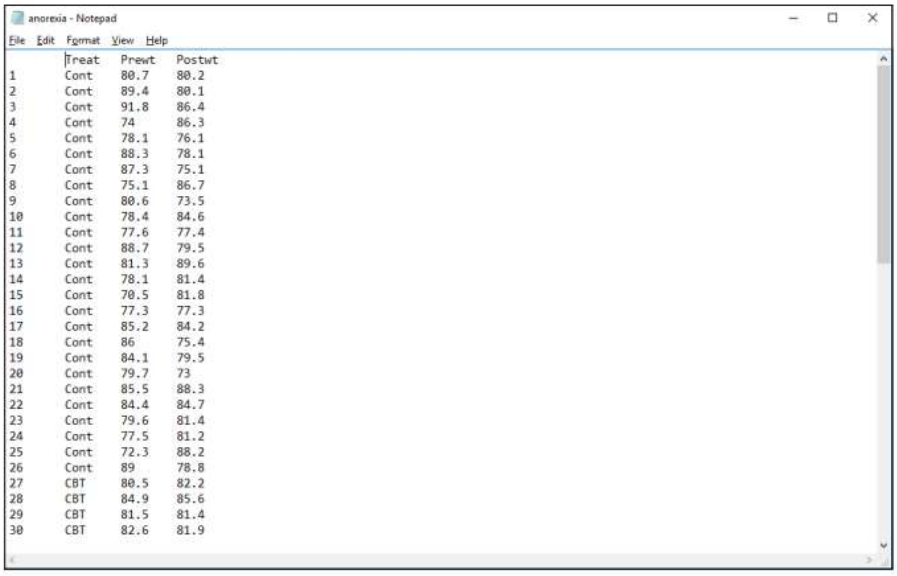
```
> read.table("C:/TextFiles/ScoresText.txt", header=TRUE)
   Student Math_Score Science_Score
1        1         85            90
2        2         91            87
3        3         78            75
4        4         88            78
5        5         93            99
6        6         82            89
7        7         67            71
8        8         79            84
9        9         89            88
10      10         98            97
```

Argumen kedua (header=TRUE) membuat R mengetahui bahwa baris pertama file berisi header kolom. Anda menggunakan write.table() untuk menulis bingkai data anoreksia ke file teks:

```
> write.table(anorexia, "C:/TextFiles/anorexia.txt", quote =
  FALSE, sep = "\t")
```

Ini menempatkan file anorexia.txt di folder TextFiles pada drive C. Argumen kedua (quote = FALSE) memastikan bahwa tidak ada tanda kutip yang muncul, dan argumen ketiga (sep = "\t") membuat file dibatasi tab.

Gambar 2.14 menunjukkan bagaimana file teks terlihat di Notepad. Pengungkapan penuh: Di baris pertama file teks, Anda harus menekan tombol Tab sekali untuk memposisikan header dengan benar.



Gambar 2.14 Bingkai data anoreksia sebagai file teks berbatas tab.

Dalam setiap contoh ini, Anda menggunakan jalur file lengkap untuk setiap file. Itu tidak perlu jika file berada di direktori kerja. Jika, misalnya, Anda meletakkan spreadsheet Skor di direktori kerja, inilah yang harus Anda lakukan untuk membacanya ke R:

```
> read.xlsx("Scores.xlsx","Sheet1")
```

BAGIAN 2
MENJELASKAN DATA
BAB 3 MENDAPATKAN GRAFIK

Visualisasi data merupakan bagian penting dari statistik. Grafik yang baik memungkinkan Anda melihat tren dan hubungan yang mungkin terlewatkan jika Anda hanya melihat angka. Grafik berharga untuk alasan lain: Grafik membantu Anda mempresentasikan ide Anda kepada kelompok. Ini sangat penting dalam bidang ilmu data. Organisasi bergantung pada ilmuwan data untuk memahami sejumlah besar data sehingga pengambil keputusan dapat merumuskan strategi. Grafik memungkinkan ilmuwan data untuk menjelaskan pola dalam data kepada manajer dan personel nonteknis.

3.1 MENEMUKAN POLA

Data sering berada dalam tabel yang panjang dan kompleks. Seringkali, Anda harus memvisualisasikan hanya sebagian dari tabel untuk menemukan pola atau tren. Contoh yang baik adalah kerangka data Cars93, yang berada dalam paket MASS. (Dalam Bab 2, saya menunjukkan cara memasukkan paket ini ke perpustakaan R Anda.) Kerangka data ini menyimpan data pada 27 variabel untuk 93 model mobil yang tersedia pada tahun 1993.

Gambar 3.1 menunjukkan bagian dari bingkai data di jendela Editor Data yang terbuka setelah Anda mengetik

```
> edit(Cars93)
```

Data Editor

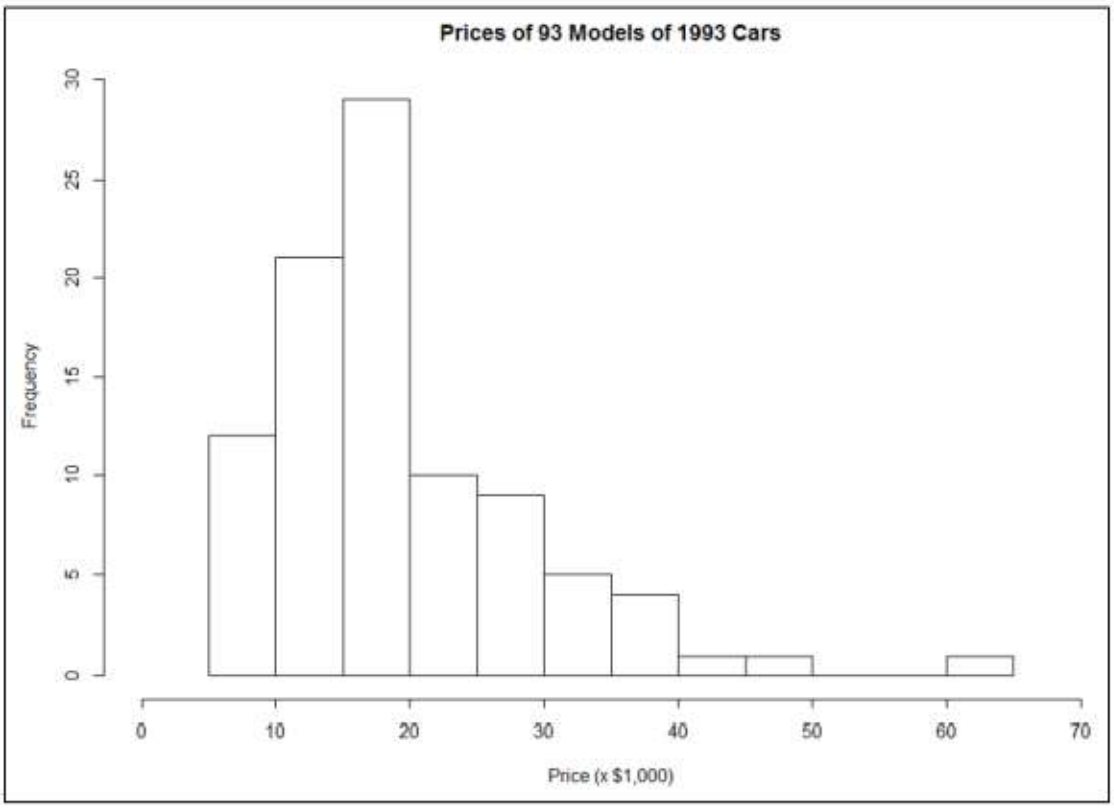
File Edit Help

	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain
1	Acura	Integra	Small	12.9	15.9	18.8	25	31	None	Front
2	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	Front
3	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	Front
4	Audi	100	Midsize	30.8	37.7	44.6	19	26	Driver & Passenger	Front
5	BMW	535i	Midsize	23.7	30	36.2	22	30	Driver only	Rear
6	Buick	Century	Midsize	14.2	15.7	17.3	22	31	Driver only	Front
7	Buick	LeSabre	Large	19.9	20.8	21.7	19	28	Driver only	Front
8	Buick	Roadmaster	Large	22.6	23.7	24.9	16	25	Driver only	Rear
9	Buick	Riviera	Midsize	26.3	26.3	26.3	19	27	Driver only	Front
10	Cadillac	DeVille	Large	33	34.7	36.3	16	25	Driver only	Front
11	Cadillac	Seville	Midsize	37.5	40.1	42.7	16	25	Driver & Passenger	Front
12	Chevrolet	Cavalier	Compact	8.5	13.4	18.3	25	36	None	Front
13	Chevrolet	Corsica	Compact	11.4	11.4	11.4	25	34	Driver only	Front
14	Chevrolet	Camaro	Sporty	13.4	15.1	16.8	19	28	Driver & Passenger	Rear
15	Chevrolet	Lumina	Midsize	13.4	15.9	18.4	21	29	None	Front
16	Chevrolet	Lumina_APV	Van	14.7	16.3	18	18	23	None	Front
17	Chevrolet	Astro	Van	14.7	16.6	18.6	15	20	None	4WD
18	Chevrolet	Caprice	Large	18	18.8	19.6	17	26	Driver only	Rear
19	Chevrolet	Corvette	Sporty	34.6	38	41.5	17	25	Driver only	Rear

Gambar 3.1 Bagian dari kerangka data Cars93.

Menggambarkan distribusi

Salah satu pola yang mungkin menarik adalah distribusi harga semua mobil yang terdaftar dalam kerangka data Cars93. Jika Anda harus memeriksa seluruh kerangka data untuk menentukan ini, itu akan menjadi tugas yang membosankan. Namun, grafik memberikan informasi dengan segera. Gambar 3.2, sebuah histogram, menunjukkan apa yang saya maksud.



Gambar 3.2 Histogram harga mobil dalam kerangka data Cars93.

Histogram cocok bila variabel pada sumbu x merupakan variabel interval atau variabel rasio. (Lihat Bab 1.) Dengan jenis variabel ini, angka memiliki arti.

Dalam Bab 1, saya membedakan antara variabel bebas dan variabel terikat. Di sini, Harga adalah variabel independen, dan Frekuensi adalah variabel dependen. Pada sebagian besar (tetapi tidak semua) grafik, variabel bebas berada pada sumbu x, dan variabel terikat berada pada sumbu y.

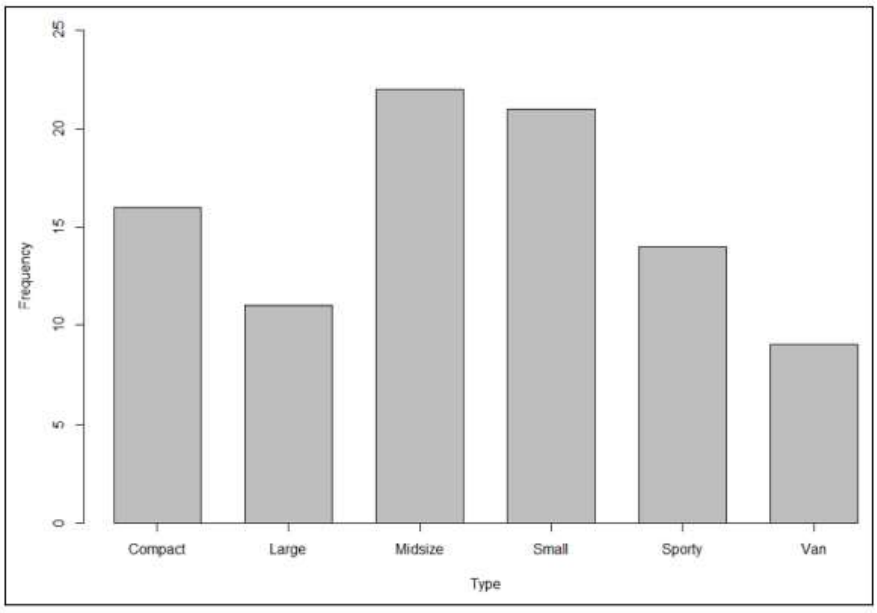
Bar-hopping

Untuk variabel nominal (sekali lagi, lihat Bab 1), angka hanyalah label. Faktanya, level dari variabel nominal (juga disebut faktor — lihat Bab 2) dapat berupa nama. Contoh kasus: Hal menarik lainnya yang mungkin adalah frekuensi dari berbagai jenis mobil (sporty, menengah, van, dan sebagainya) dalam kerangka data. Jadi, "Jenis" adalah variabel nominal. Jika Anda melihat setiap entri dalam bingkai data dan membuat tabel frekuensi ini, itu akan terlihat seperti Tabel 3.1.

Tabel 3.1 Jenis dan Frekuensi Mobil dalam kerangka data Cars93

Jenis	Frekuensi
kompak	16
Besar	11
Ukuran sedang	22
Kecil	21
sporty	14
mobil van	9

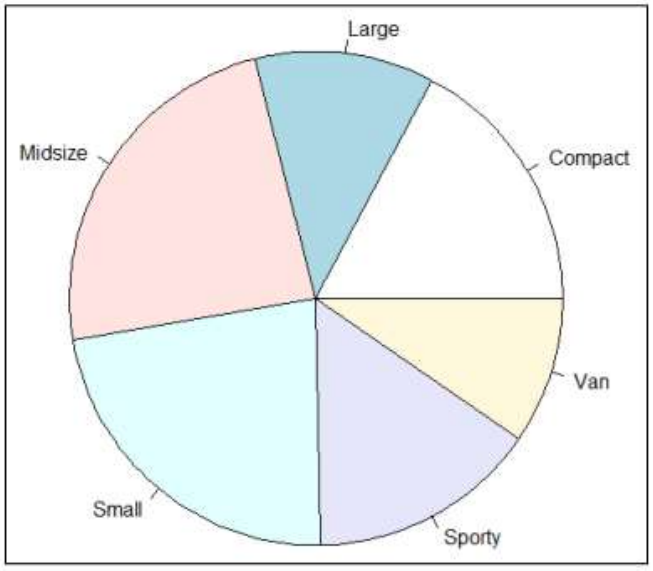
Tabel menunjukkan beberapa tren — lebih banyak model mobil menengah dan kecil daripada mobil dan van besar. Mobil kompak dan mobil sporty ada di tengah. Gambar 3.3 menunjukkan informasi ini dalam bentuk grafik. Jenis grafik ini adalah grafik batang. Spasi antara batang menekankan bahwa Jenis, pada sumbu x, adalah variabel nominal. Meskipun tabelnya cukup sederhana, saya pikir kami akan setuju bahwa penonton lebih suka melihat gambarnya. Seperti yang saya suka katakan, mata yang berkaca-kaca saat melihat angka sering kali bersinar lebih terang saat melihat gambar.



Gambar 3.3 dari Tabel 3.1 diubah menjadi grafik batang.

Mengiris pai

Grafik pai adalah jenis gambar lain yang menunjukkan data yang sama dengan cara yang sedikit berbeda. Setiap frekuensi muncul sebagai sepotong kue. Gambar 3.4 menunjukkan apa yang saya maksud. Dalam grafik pai, luas irisan mewakili frekuensi.



Gambar 3.4 dari Tabel 3.1 sebagai diagram lingkaran.

PEDOMAN GRAFIK PIE
Maafkan saya jika Anda pernah mendengar yang ini sebelumnya. Ini adalah anekdot lucu yang berfungsi sebagai aturan praktis untuk grafik pai.

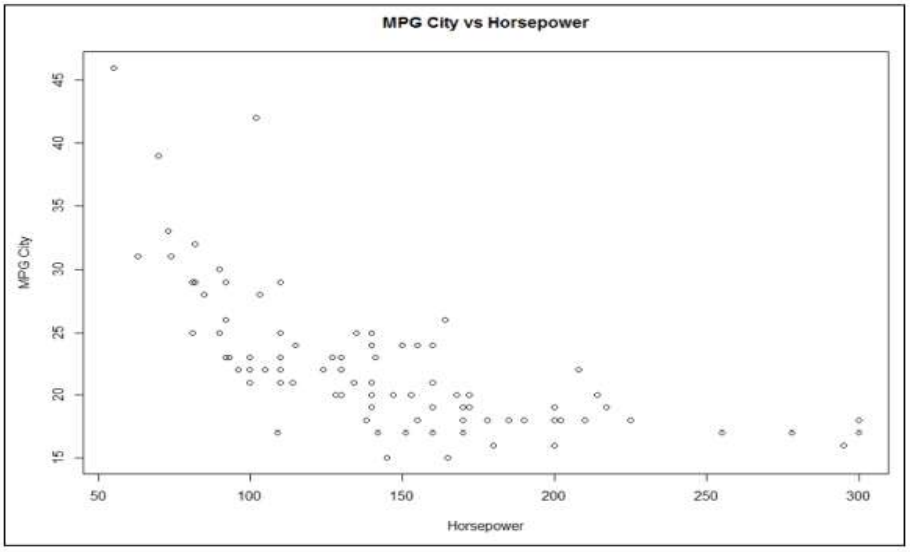
Almarhum, Yogi Berra yang hebat sering membuat pernyataan yang salah yang menjadi bagian dari budaya kita. Dia pernah konon masuk ke restoran pizza dan memesan pizza utuh.

"Haruskah aku memotongnya menjadi empat atau delapan?" tanya pelayan itu.
“Lebih baik jadi empat saja,” kata Yogi. "Aku tidak cukup lapar untuk makan delapan."
Kesimpulan: Jika suatu faktor memiliki banyak level, menghasilkan grafik pai dengan banyak irisan, mungkin informasinya berlebihan. Pesan akan terlihat lebih baik dalam grafik batang.

(Apakah insiden Yogi itu benar-benar terjadi? Tidak jelas. Meringkas ucapan seumur hidup yang dikaitkan dengannya, Tuan Berra berkata: "Setengah kebohongan yang mereka katakan tentang saya tidak benar.")

Plot pencar

Pola minat potensial lainnya adalah hubungan antara mil per galon untuk mengemudi di kota dan tenaga kuda. Jenis grafik ini adalah plot pencar. Gambar 3.5 menunjukkan plot pencar untuk kedua variabel ini.

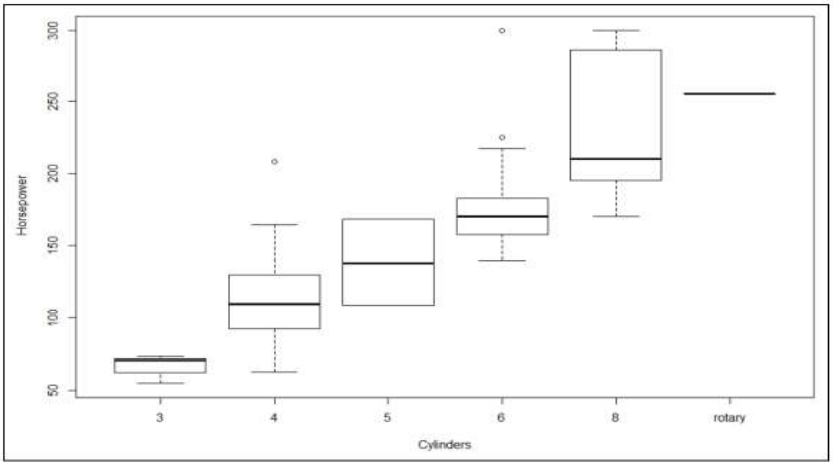


Gambar 3.5 MPG dalam mengemudi kota dan tenaga kuda untuk data di Cars93.

Setiap lingkaran kecil mewakili salah satu dari 93 mobil. Posisi lingkaran di sepanjang sumbu x (koordinat x) adalah tenaga kudanya, dan posisinya di sepanjang sumbu y (koordinat y) adalah MPG untuk mengemudi di kota. Sekilas melihat bentuk plot pencar menunjukkan adanya hubungan: Saat tenaga kuda meningkat, MPG-kota tampaknya menurun. (Para ahli statistik akan mengatakan "MPG-kota berkurang dengan tenaga kuda.") Apakah mungkin menggunakan statistik untuk menganalisis hubungan ini dan mungkin membuat prediksi? (Lihat Bab 14.)

Dari kotak dan kumis

Bagaimana dengan hubungan antara tenaga kuda dan jumlah silinder pada mesin mobil? Anda akan mengharapkan tenaga kuda meningkat dengan silinder, dan Gambar 3-6 menunjukkan bahwa memang demikian. Diciptakan oleh ahli statistik terkenal John Tukey, jenis grafik ini disebut plot kotak, dan ini adalah cara yang bagus dan cepat untuk memvisualisasikan data.



Gambar 3.6 Plot kotak tenaga kuda versus jumlah silinder dalam kerangka data Cars93.

Setiap kotak mewakili sekelompok angka. Kotak paling kiri, misalnya, mewakili tenaga kuda mobil dengan tiga silinder. Garis hitam pekat di dalam kotak adalah median — nilai tenaga kuda yang berada di antara bagian bawah angka dan bagian atas. Tepi bawah dan atas kotak disebut engsel. Engsel bawah adalah kuartil bawah, angka di bawahnya yang jatuh di bawah 25 persen. Engsel atas adalah kuartil atas, angka yang melebihi 75 persen angka. (Saya membahas median di Bab 4 dan persentil di Bab 6.)

Elemen yang mencuat dari engsel disebut kumis (jadi Anda kadang-kadang melihat jenis grafik ini disebut sebagai plot kotak-dan-kumis). Kumis menyertakan nilai data di luar engsel. Batas kumis atas adalah nilai maksimum atau engsel atas ditambah 1,5 kali panjang kotak, mana saja yang lebih kecil. Batas kumis bawah adalah nilai minimum atau engsel bawah dikurangi 1,5 kali panjang kotak, mana yang lebih besar. Titik data di luar kumis adalah outlier. Plot kotak menunjukkan bahwa data untuk empat silinder dan enam silinder memiliki outlier. Perhatikan bahwa grafik hanya menunjukkan garis padat untuk "putar", jenis mesin yang muncul hanya sekali dalam data.

3.2 GRAFIK R DASAR

Kemampuan untuk membuat grafik seperti yang saya tunjukkan di bagian sebelumnya hadir dengan instalasi R Anda, yang membuat grafik ini menjadi bagian dari grafik R dasar. Saya mulai dengan itu. Kemudian di bagian selanjutnya saya menunjukkan paket ggplot2 yang sangat berguna.

Di basis R, format umum untuk membuat grafik adalah:

```
graphics_function(data, arg1, arg2, ...)
```

Setelah Anda membuat grafik di RStudio, klik Zoom pada tab Plot RStudio untuk membuka grafik di jendela yang lebih besar. Grafik lebih jelas di jendela Zoom daripada di tab Plot.

Histogram

Saatnya melihat kembali kerangka data Cars93 yang saya perkenalkan di bagian "Menemukan Pola", di awal bab ini. Untuk membuat histogram distribusi harga dalam kerangka data tersebut, Anda harus memasukkan:

```
> hist(Cars93$Price)
```

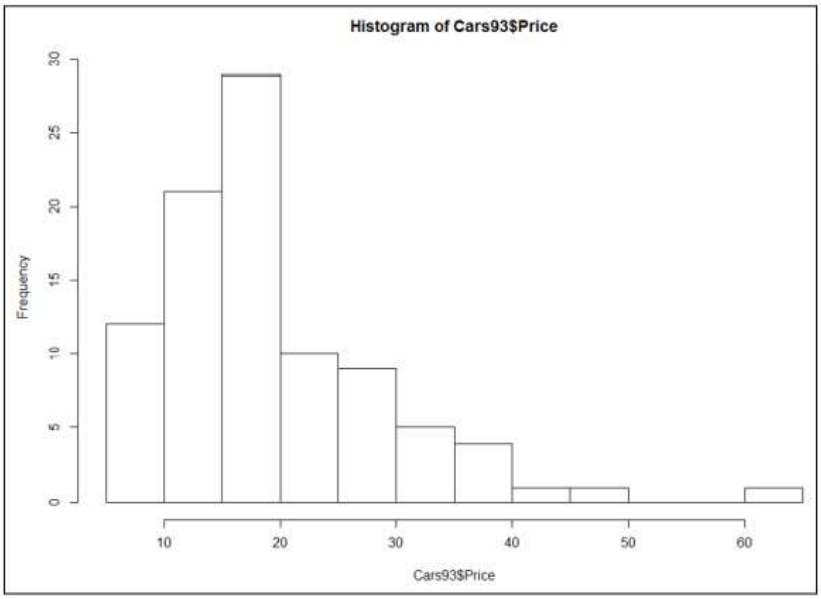
yang menghasilkan Gambar 3.7.

Anda akan melihat bahwa ini tidak terlalu keren seperti Gambar 3.2. Bagaimana Anda merapikannya? Dengan menambahkan argumen.

Salah satu argumen yang sering digunakan dalam grafik R dasar mengubah label sumbu x dari default R menjadi sesuatu yang lebih bermakna. Ini disebut xlab. Untuk sumbu x pada Gambar 3.2, saya menambahkan ke argumen.

```
xlab= “Price (x $1,000)”
```

Anda dapat menggunakan ylab untuk mengubah label sumbu y, tetapi saya membiarkannya di sini.



Gambar 3.7 Histogram awal distribusi harga di Mobil93.

Saya ingin sumbu x memanjang dari batas bawah 0 ke batas atas 70, dan itulah provinsi dari argumen xlim. Karena argumen ini bekerja dengan vector.

```
xlim = c(0,70)
```

Saya juga menginginkan judul yang berbeda, dan untuk itu saya menggunakan main:

```
main = “Prices of 93 Models of 1993 Cars”
```

Untuk menghasilkan histogram pada Gambar 3.2, seluruh megillah adalah:

```
> hist(Cars93$Price, xlab="Price (x $1,000)", xlim = c(0,70),
          main = "Prices of 93 Models of 1993 Cars")
```

Saat membuat histogram, R menghitung jumlah kolom terbaik untuk tampilan yang bagus. Di sini, R memutuskan bahwa 12 adalah angka yang cukup bagus. Anda dapat memvariasikan jumlah kolom dengan menambahkan argumen yang disebut jeda dan menyetel nilainya. R tidak selalu memberi Anda nilai yang Anda tetapkan. Sebaliknya, ia memberikan sesuatu yang mendekati nilai itu dan mencoba mempertahankan penampilan yang tampak bagus. Tambahkan argumen ini, atur nilainya (breaks =4, misalnya), dan Anda akan mengerti maksud saya.

Menambahkan fitur grafik

Aspek penting dari grafik R dasar adalah kemampuan untuk menambahkan fitur ke grafik setelah Anda membuatnya. Untuk menunjukkan kepada Anda apa yang saya maksud, saya harus mulai dengan jenis grafik yang sedikit berbeda.

Cara lain untuk menunjukkan informasi histogram adalah dengan memikirkan data sebagai probabilitas daripada frekuensi. Jadi, alih-alih frekuensi kisaran harga tertentu, Anda membuat grafik probabilitas bahwa sebuah mobil yang dipilih dari data berada dalam kisaran harga tersebut. Untuk melakukan ini, Anda menambahkan ke argumen.

```
probability = True
```

Sekarang kode R terlihat seperti ini:

```
> hist(Cars93$Price, xlab="Price (x $1,000)", xlim = c(0,70),
          main = "Prices of 93 Models of 1993 Cars",probability
          = TRUE)
```

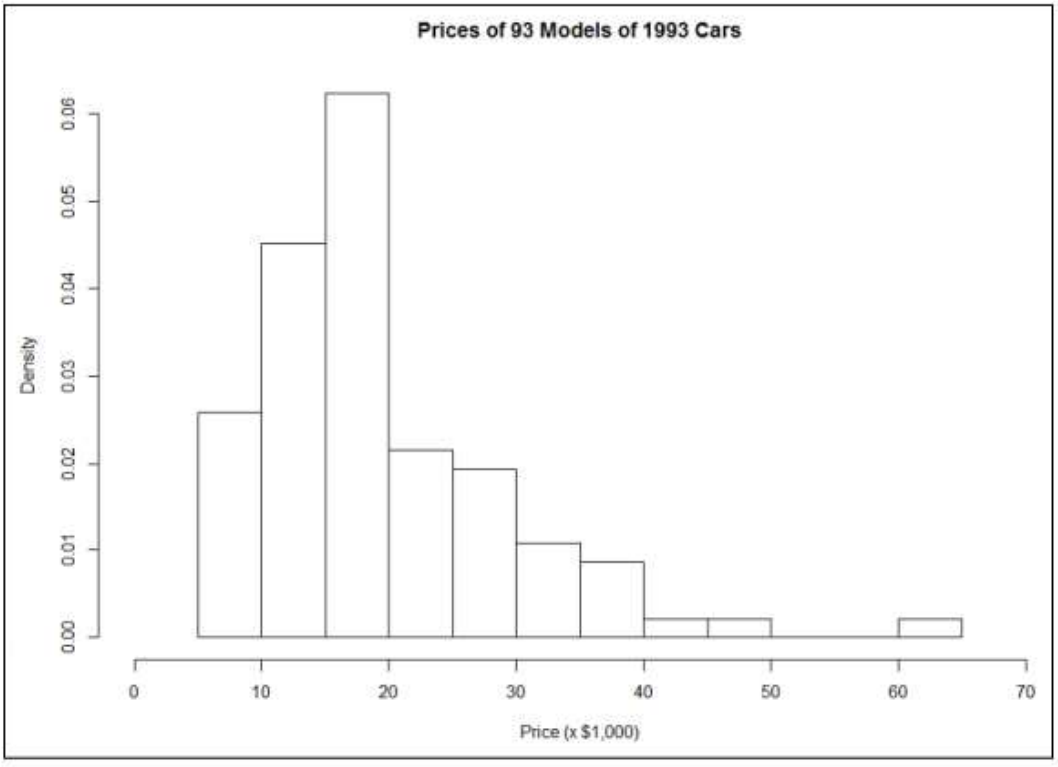
Hasilnya tampak pada Gambar 3.8. Sumbu y mengukur Densitas — sebuah konsep yang berhubungan dengan probabilitas, yang saya bahas di Bab 8. Grafik ini disebut plot kepadatan.

Inti dari semua ini adalah apa yang Anda lakukan selanjutnya. Setelah Anda membuat grafik, Anda dapat menggunakan fungsi tambahan yang disebut lines() untuk menambahkan garis ke plot kepadatan:

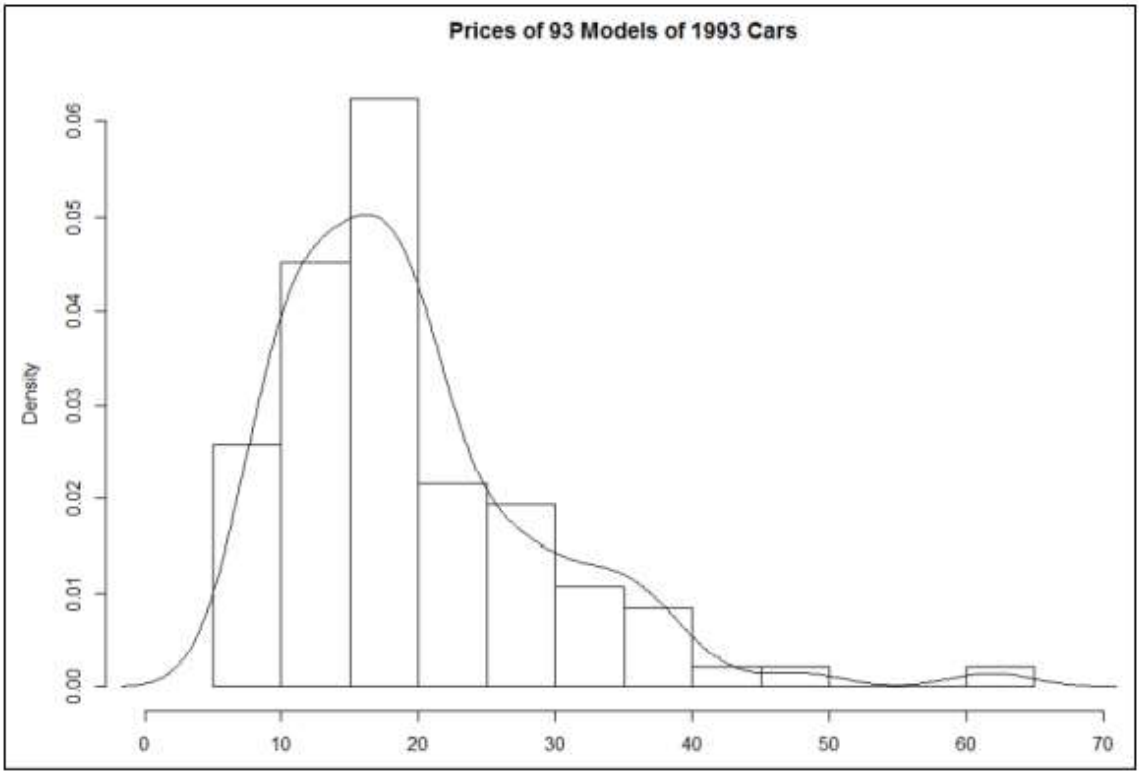
```
> lines(density(Cars93$Price))
```

Grafik sekarang terlihat seperti Gambar 3.9.

Jadi dalam grafik R dasar, Anda dapat membuat grafik dan kemudian mulai menambahkannya setelah Anda melihat seperti apa grafik awal. Ini seperti melukis gambar danau dan kemudian menambahkan gunung dan pohon sesuai keinginan Anda.



Gambar 3.8 Densitas plot distribusi harga di Mobil93.



Gambar 3.9 Plot kepadatan dengan garis tambahan.

Plot bar

Kembali ke bagian "Menemukan Pola", di awal bab ini, saya menunjukkan kepada Anda grafik batang yang menggambarkan jenis dan frekuensi mobil, saya juga menunjukkan Tabel 3.1. Ternyata, Anda harus membuat tabel semacam ini sebelum Anda dapat menggunakan barplot() untuk membuat grafik batang.

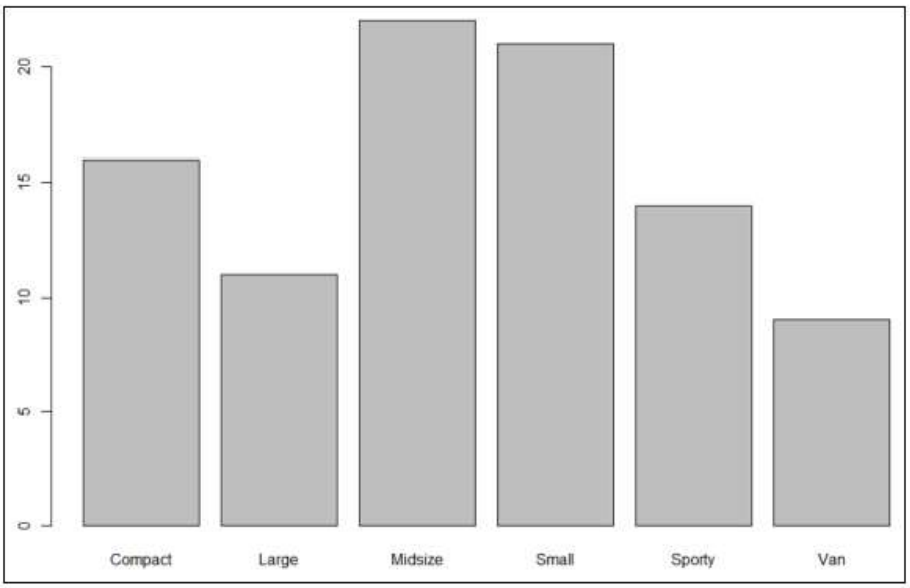
Untuk menempatkan Tabel 3.1 bersama-sama, kode R adalah (cukup tepat)

```
> table(Cars93$Type)

Compact   Large Midsize   Small  Sporty     Van
     16      11      22      21      14       9
```

Untuk grafik batang, maka

```
> barplot(table(Cars93$Type))
```

yang membuat grafik pada Gambar 3.10.



Gambar 3.10 Plot bar awal dari tabel (Cars93 $Type).

Sekali lagi, tidak semanis produk akhir yang ditunjukkan pada Gambar 3.3. Argumen tambahan berhasil. Untuk menempatkan 0 hingga 25 pada sumbu y, Anda menggunakan ylim, yang, seperti xlim, bekerja dengan vektor:

```
ylim = c(0,25)
```

Untuk label sumbu x dan label sumbu y, Anda menggunakan

```
xlab = “Type”
ylab = “Frequency”
```

Untuk menggambar sumbu padat, Anda bekerja dengan axis.lty. Anggap ini sebagai "tipe garis sumbu" yang Anda atur menjadi padat dengan mengetik

```
axis.lty = “solid”
```

Nilai putus-putus dan putus-putus untuk axis.lty menghasilkan tampilan yang berbeda untuk sumbu x. Terakhir, Anda menggunakan spasi untuk menambah jarak antar batang:

```
space = .05
```

Berikut seluruh fungsi untuk menghasilkan grafik pada Gambar 3.3:

```
› barplot(table(Cars93$Type),ylim=c(0,25), xlab="Type",
          ylab="Frequency", axis.lty = "solid", space = .05)
```

Grafik pai

Jenis grafik ini sangat sederhana. Garis

```
› pie(table(Cars93$Type))
```

membawa Anda langsung ke Gambar 3.4.

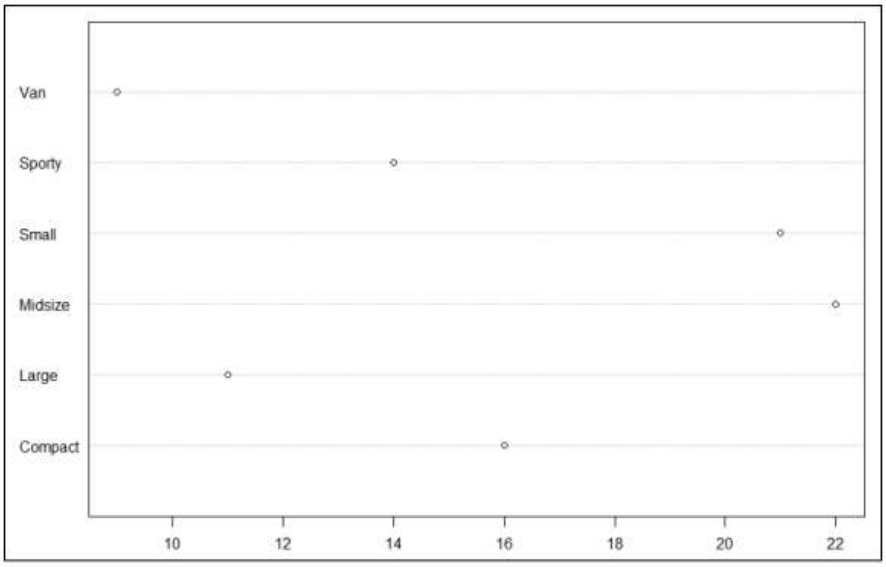
Diagram titik

Tunggu. Apa? Dari mana asalnya? Ini adalah cara lain untuk memvisualisasikan data pada Tabel 3.1. Ahli grafis terkenal William Cleveland percaya bahwa orang mempersepsikan nilai di sepanjang skala umum (seperti dalam plot batang) lebih baik daripada yang mereka rasakan di area (seperti dalam grafik pai). Jadi dia membuat diagram titik, yang saya tunjukkan pada Gambar 3.11. Terlihat sedikit seperti sempoa yang diletakkan miring, bukan? Ini adalah salah satu kasus yang sering terjadi di mana variabel bebas berada pada sumbu y dan variabel terikat berada pada sumbu x.

Format untuk fungsi yang membuat diagram titik adalah:

```
> dotchart(x, labels, arg1, arg2 ...)
```

Dua argumen pertama adalah vektor, dan yang lainnya adalah argumen opsional untuk mengubah tampilan diagram titik. Vektor pertama adalah vektor nilai (frekuensi). Yang kedua cukup jelas — dalam hal ini, ini adalah label untuk jenis kendaraan.



Gambar 3.11 Diagram titik untuk data pada Tabel 3-1.

Untuk membuat vektor yang diperlukan, Anda mengubah tabel menjadi bingkai data:

```
> type.frame <- data.frame(table(Cars93$Type))
> type.frame
     Var1 Freq
1 Compact   16
2   Large   11
3 Midsize   22
4   Small   21
5  Sporty   14
6     Van    9
```

Setelah Anda memiliki bingkai data, baris ini menghasilkan diagram titik:

```
> dotchart(type.frame$Freq,type.frame$Var1)
```

Type.frame$Freq menetapkan bahwa kolom Frekuensi dalam bingkai data adalah sumbu x, dan type.frame$Var1 menetapkan bahwa kolom Var1 (yang menampung jenis mobil) adalah sumbu y.

Baris ini juga berfungsi:

```
> dotchart(type.frame[,2],type.frame[,1])
```

Ingat dari Bab 2 bahwa [,2] berarti "kolom 2" dan [,1] berarti "kolom 1."

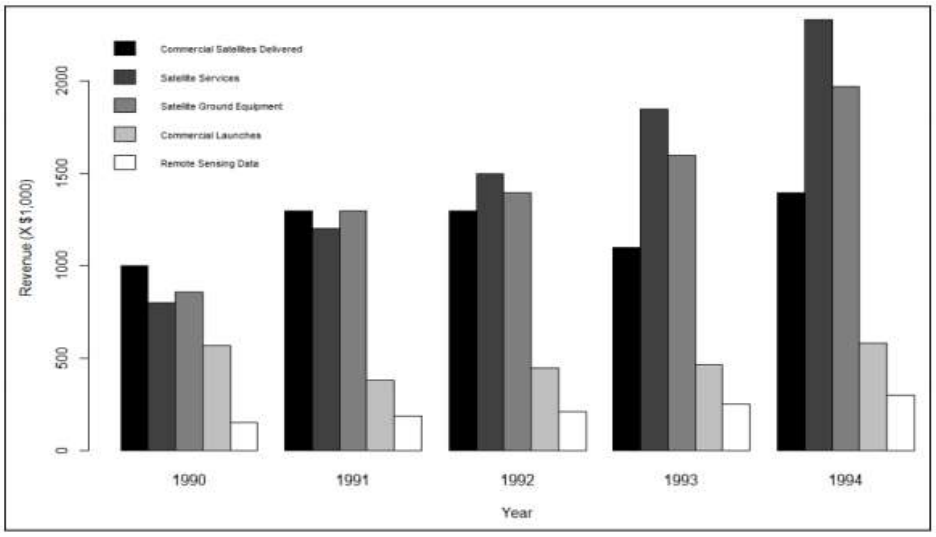
3.3 PENINJAUAN PLOT BAR

Dalam semua grafik sebelumnya, variabel dependen adalah frekuensi. Namun, seringkali, variabel dependen adalah titik data daripada frekuensi. Inilah yang saya maksud. Tabel 3.2 menunjukkan data pendapatan ruang komersial untuk awal 1990-an. (Omong-omong, datanya berasal dari Departemen Perdagangan A.S., melalui Abstrak Statistik A.S.)

Tabel 3.2 Pendapatan Ruang Komersial AS 1990–1994 (Dalam Jutaan Dolar)

Industri	1990	1991	1992	1993	1994
Satelit Komersial Dikirim	1,000	1,300	1,300	1,100	1,400
Layanan Satelit	800	1,200	1,500	1,850	2,330
Peralatan Darat Satelit	860	1,300	1,400	1,600	1,970
Peluncuran Komersial	570	380	450	465	580
Data Penginderaan Jauh	155	190	210	250	300

Data adalah angka dalam sel, yang mewakili pendapatan dalam ribuan dolar. Plot batang R dasar dari data dalam tabel ini muncul pada Gambar 3.12.



Gambar 3.12 Plot batang dari data pada Tabel 3.2.

Jika Anda harus membuat presentasi tentang data ini, saya pikir Anda akan setuju bahwa audiens Anda lebih suka grafik daripada tabel. Meskipun tabelnya informatif, itu tidak menarik perhatian orang. Lebih mudah untuk melihat tren dalam grafik — Layanan Satelit naik paling cepat sementara Peluncuran Komersial tetap stabil, misalnya. Grafik ini disebut plot batang berkelompok. Bagaimana Anda membuat plot seperti ini di basis R?
Hal pertama yang harus dilakukan adalah membuat vektor nilai dalam sel:

```
rev.values <-
          c(1000,1300,1300,1100,1400,800,1200,1500,1850,
          2330,860,1300,1400,1600,1970,570,380,450,465,580,
          155,190,210,250,300)
```

Meskipun koma muncul dalam nilai dalam tabel (untuk nilai yang lebih besar dari seribu), Anda tidak boleh memiliki koma dalam nilai dalam vektor! (Untuk alasan yang jelas: Koma memisahkan nilai berurutan dalam vektor.) Selanjutnya, Anda mengubah vektor ini menjadi matriks. Anda harus memberi tahu R berapa banyak baris (atau kolom) yang akan ada dalam matriks, dan bahwa nilainya dimuat ke dalam matriks baris demi baris:

```
space.rev <- matrix(rev.values,nrow=5,byrow = T)
```

Terakhir, Anda memberikan nama kolom dan nama baris ke matriks:

```
colnames(space.rev) <-
          c("1990","1991","1992","1993","1994")
rownames(space.rev) <- c("Commercial Satellites
          Delivered","Satellite Services","Satellite Ground
          Equipment","Commercial Launches","Remote Sensing Data")
```

Mari kita lihat matriksnya:

```
> space.rev
                              1990 1991 1992 1993 1994
Commercial Satellites Delivered 1000 1300 1300 1100 1400
Satellite Services               800 1200 1500 1850 2330
Satellite Ground Equipment       860 1300 1400 1600 1970
Commercial Launches              570  380  450  465  580
Remote Sensing Data              155  190  210  250  300
```

Sempurna. Ini terlihat seperti Tabel 3.2.
Dengan data di tangan, Anda melanjutkan ke plot batang. Anda membuat vektor warna untuk batang:

```
color.names = c("black","grey25","grey50","grey75","white")
```

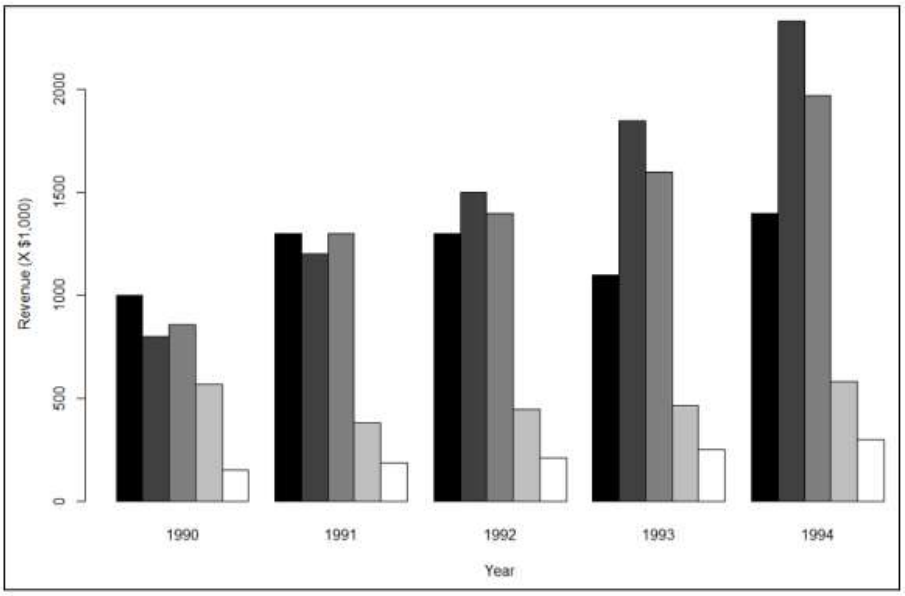
Sepatah kata tentang nama warna tersebut: Anda dapat menggabungkan angka dari 0 hingga 100 dengan "abu-abu" dan mendapatkan warna: "abu-abu0" setara dengan "hitam" dan "abu-abu100" setara dengan "putih". (Jauh lebih dari lima puluh warna, jika Anda tahu apa yang saya maksud ...)

Dan sekarang untuk plotnya:

```
> barplot(space.rev, beside = T, xlab= "Year",ylab= "Revenue
        (X $1,000)", col=color.names)
```

di samping = T berarti jeruji akan, yah, berdampingan satu sama lain. (Anda harus mencoba ini tanpa argumen itu dan lihat apa yang terjadi.) Argumen col = color.names memasok warna yang Anda tentukan dalam vektor.

Plot yang dihasilkan ditunjukkan pada Gambar 3.13.



Gambar 3.13 Plot batang awal dari data pada Tabel 3.2.

Apa yang hilang, tentu saja, adalah legenda. Anda menambahkannya dengan fungsi legend() untuk menghasilkan Gambar 3-12:

```
> legend(1,2300,rownames(space.rev), cex=0.7, fill = color.
  names, bty = "n")
```

Dua nilai pertama adalah koordinat x dan y untuk menemukan legenda. (Itu membutuhkan banyak mengutak-atik!). Argumen berikutnya menunjukkan apa yang masuk ke dalam legenda (nama-nama industri). Argumen cex menentukan ukuran karakter dalam legenda. Nilai, 0,7, menunjukkan bahwa Anda ingin karakter menjadi 70 persen dari ukuran biasanya. Itulah satu-satunya cara untuk menyesuaikan legenda pada grafik. (Pikirkan "cex" sebagai "perluasan karakter," meskipun dalam kasus ini "kontraksi karakter.") fill = color.names menempatkan contoh warna dalam legenda, di sebelah nama baris. Menyetel bty ("tipe batas") ke "n" ("tidak ada") adalah trik kecil lain untuk memasukkan legenda ke dalam grafik.

Plot sebar

Untuk memvisualisasikan hubungan antara tenaga kuda dan MPG untuk mengemudi di kota (seperti yang ditunjukkan pada Gambar 3.5), Anda menggunakan fungsi plot():

```
> plot(Cars93$Horsepower, Cars93$MPG.city,
         xlab="Horsepower",ylab="MPG City", main ="MPG City vs
         Horsepower")
```

Seperti yang Anda lihat, saya menambahkan argumen untuk pelabelan sumbu, dan untuk judul. Cara lain untuk melakukannya adalah dengan menggunakan notasi rumus yang saya tunjukkan di Bab 2. Jadi jika Anda ingin kode R menunjukkan bahwa MPG-city bergantung pada horsepower, Anda mengetik

```
> plot(Cars93$MPG.city ~ Cars93$Horsepower,
         xlab="Horsepower",ylab="MPG City", main ="MPG City vs
         Horsepower")
```

untuk menghasilkan plot pencar yang sama. Operator tilde (~) berarti “tergantung.”

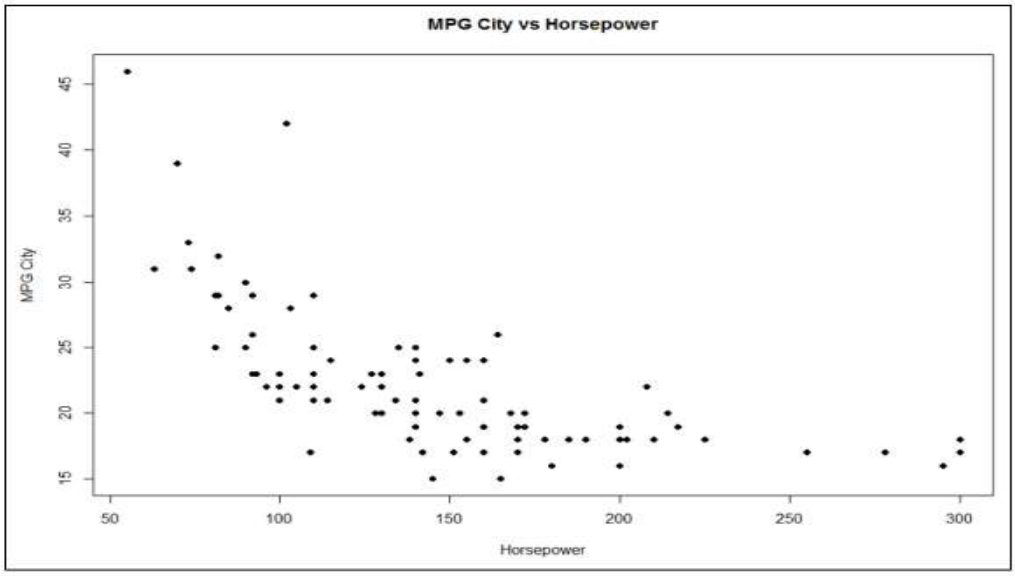
Plot twist

R memungkinkan Anda untuk mengubah simbol yang menggambarkan titik-titik dalam grafik. Gambar 3.5 menunjukkan bahwa simbol default adalah lingkaran kosong. Untuk mengubah simbol, yang disebut karakter plot, atur argumen pch. R memiliki satu set nilai numerik bawaan (0–25) untuk pch yang sesuai dengan satu set simbol. Nilai 0–15 sesuai dengan bentuk yang tidak terisi, dan 16–25 diisi.

Nilai defaultnya adalah 1. Untuk mengubah karakter plot menjadi kotak, setel pch ke 0. Untuk segitiga, nilainya 2, dan untuk lingkaran terisi 16:

```
> plot(Cars93$Horsepower,Cars93$MPG.city, xlab="Horsepower",
         ylab="MPG City", main = "MPG City vs Horsepower",pch=16)
```

Gambar 3.14 menunjukkan plot dengan lingkaran yang terisi. Anda juga dapat mengatur argumen col untuk mengubah warna dari "hitam" menjadi "biru" atau ke berbagai warna lain (yang tidak akan muncul dengan baik di halaman hitam-putih yang Anda lihat).



Gambar 3.14 MPG City vs. Horsepower dengan lingkaran terisi (pch = 16).

Anda tidak terbatas pada nilai numerik bawaan untuk pch. Di sini, misalnya, adalah sentuhan yang menarik: Untuk membantu menemukan pola dalam data, Anda dapat menggambar setiap titik di plot sebagai jumlah silinder di mobil yang sesuai, bukan sebagai simbol.

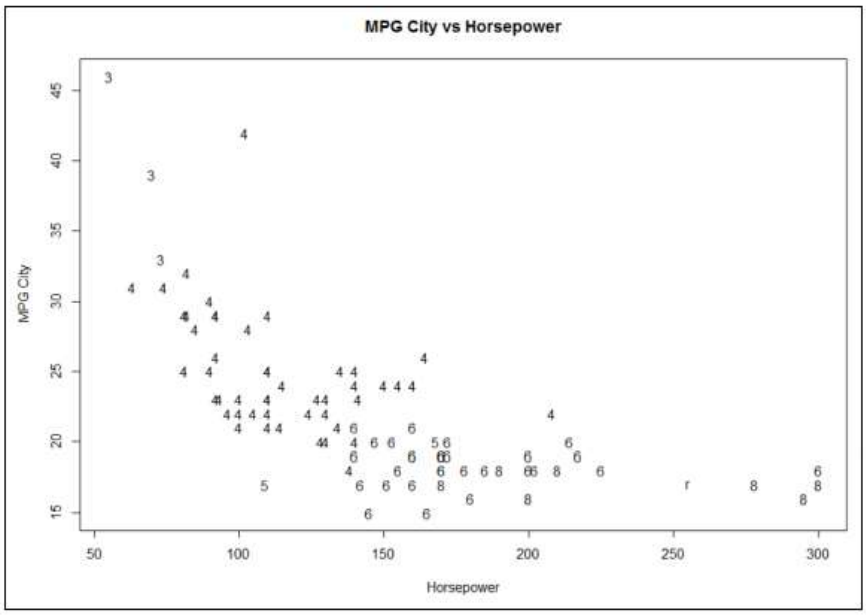
Untuk melakukan itu, Anda harus berhati-hati tentang bagaimana Anda mengatur pch. Anda tidak bisa hanya menetapkan Cars93$. Cylinders sebagai nilainya. Anda harus memastikan bahwa apa yang Anda berikan ke pch adalah karakter (seperti "3", "4" atau "8") daripada angka (seperti 3, 4, atau 8). Komplikasi lain adalah bahwa data berisi "rotary" sebagai salah satu nilai untuk Silinder. Untuk memaksa nilai Silinder menjadi karakter, Anda menerapkan as.character() ke Cars93$Cylinders:

```
pch = as.character(Cars93$Cylinders)
```

dan fungsi plot() adalah

```
> plot(Cars93$Horsepower,Cars93$MPG.city, xlab="Horsepower",
        ylab="MPG City", main = "MPG City vs Horsepower", pch
        = as.character(Cars93$Cylinders))
```

Hasilnya adalah plot pencar pada Gambar 3.15. Menariknya, as.character() melewati "rotary" sebagai "r".



Gambar 3.15 MPG City vs Horsepower dengan poin diplot sebagai jumlah silinder.

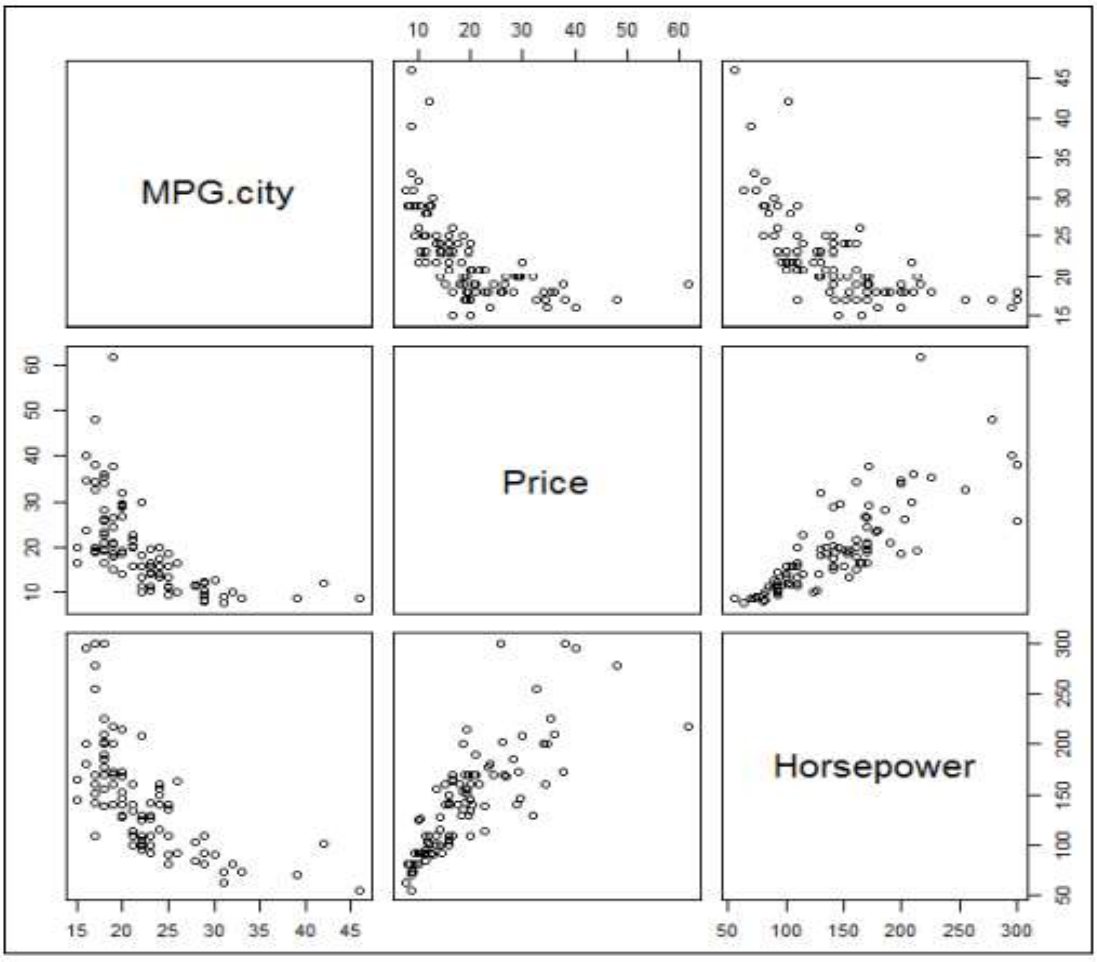
Sejalan dengan intuisi kami tentang mobil, plot ini dengan jelas menunjukkan bahwa jumlah silinder yang lebih rendah dikaitkan dengan tenaga kuda yang lebih rendah dan jarak tempuh yang lebih tinggi, dan jumlah silinder yang lebih tinggi dikaitkan dengan tenaga kuda

yang lebih tinggi dan jarak tempuh yang lebih rendah. Anda juga dapat dengan cepat melihat di mana mesin rotari cocok dengan semua ini (jarak tempuh rendah, tenaga kuda tinggi).

Matriks petak sebar

Basis R menyediakan cara yang bagus untuk memvisualisasikan hubungan di antara lebih dari dua variabel. Jika Anda menambahkan harga ke dalam campuran dan Anda ingin menunjukkan semua hubungan berpasangan antara MPG-kota, harga, dan tenaga kuda, Anda memerlukan beberapa plot pencar. R dapat memplot semuanya bersama-sama dalam sebuah matriks, seperti yang ditunjukkan Gambar 3.16.

Nama-nama variabel ada di sel-sel diagonal utama. Setiap sel off-diagonal menunjukkan plot pencar untuk variabel barisnya (pada sumbu y) dan variabel kolomnya (pada sumbu x). Sebagai contoh, scatter plot pada baris pertama, kolom kedua menunjukkan MPG-city pada sumbu y dan harga pada sumbu x. Pada baris kedua, kolom pertama, sumbunya dibalik: kota MPG pada sumbu x, dan harga pada sumbu y. Fungsi R untuk memplot matriks ini adalah pairs(). Untuk menghitung koordinat untuk semua plot sebar, fungsi ini bekerja dengan kolom numerik dari matriks atau bingkai data.



Gambar 3.16 Beberapa plot pencar untuk hubungan antara MPG-kota, harga, dan tenaga kuda.

Untuk kenyamanan, Anda membuat bingkai data yang merupakan subset dari bingkai data Cars93. Kerangka data baru ini hanya terdiri dari tiga variabel untuk diplot. Fungsi subset() menanganinya dengan baik:

```
> cars.subset <- subset(Cars93, select = c(MPG.
          city,Price,Horsepower))
```

Argumen kedua untuk subset menciptakan vektor dari apa yang harus dipilih dari Cars93. Hanya untuk memastikan bingkai data baru seperti yang Anda inginkan, gunakan fungsi head() untuk melihat enam baris pertama:

```
> head(cars.subset)
  MPG.city Price Horsepower
1       25  15.9        140
2       18  33.9        200
3       20  29.1        172
4       19  37.7        172
5       22  30.0        208
6       22  15.7        110
```

Dan sekarang,

```
> pairs(cars.subset)
```

membuat plot pada Gambar 3.16.

Kemampuan ini tidak terbatas pada tiga variabel, juga tidak pada variabel kontinu. Untuk melihat apa yang terjadi dengan tipe variabel yang berbeda, tambahkan Silinder ke vektor untuk dipilih lalu gunakan fungsi pairs() pada cars.subset.

Plot kotak

Untuk menggambar plot kotak seperti yang ditunjukkan sebelumnya, pada Gambar 3-6, Anda menggunakan rumus untuk menunjukkan bahwa Horsepower adalah variabel dependen dan Silinder adalah variabel independen:

```
> boxplot(Cars93$Horsepower ~ Cars93$Cylinders, xlab="Cylinders",
          ylab="Horsepower")
```

Jika Anda bosan mengetik tanda $, berikut cara lain:

```
> boxplot(Horsepower ~ Cylinders, data = Cars93,
          xlab="Cylinders", ylab="Horsepower")
```

Dengan argumen yang ditata seperti dalam salah satu dari dua contoh kode sebelumnya, plot() bekerja persis seperti boxplot().

3.4 GGPLOT2

Perangkat grafis Base R akan membantu Anda memulai, tetapi jika Anda benar-benar ingin menonjol dalam visualisasi, ada baiknya untuk mempelajari ggplot2. Dibuat oleh R-

megastar Hadley Wickham, "gg" dalam nama paket adalah singkatan dari "tata bahasa grafik" dan itu merupakan indikator yang baik tentang apa yang ada di depan. Itu juga judul buku (oleh Leland Wilkinson) yang menjadi sumber konsep untuk paket ini.

Secara umum, tata bahasa adalah seperangkat aturan untuk menggabungkan hal-hal. Dalam tata bahasa yang paling kita kenal, hal-hal yang terjadi adalah kata, frasa, dan klausa: Tata bahasa bahasa kami memberi tahu Anda cara menggabungkan komponen-komponen ini untuk menghasilkan kalimat yang valid.

Jadi "tata bahasa grafik" adalah seperangkat aturan untuk menggabungkan komponen grafik untuk menghasilkan grafik. Wilkinson mengusulkan bahwa semua grafik memiliki komponen umum yang mendasari — seperti data, sistem koordinat (sumbu x dan y yang Anda kenal dengan baik, misalnya), transformasi statistik (seperti penghitungan frekuensi), dan objek di dalam grafik (misalnya, titik , batang, garis, atau irisan pai), untuk beberapa nama.

Sama seperti menggabungkan kata dan frase menghasilkan kalimat gramatikal, menggabungkan komponen grafis menghasilkan grafik. Dan seperti beberapa kalimat yang gramatikal tetapi tidak masuk akal ("Ide hijau tanpa warna tidur nyenyak."), beberapa kreasi ggplot2 adalah grafik indah yang tidak selalu berguna. Terserah pembicara/penulis untuk masuk akal bagi audiensnya, dan terserah pada pengembang grafis untuk membuat grafik yang berguna bagi orang yang menggunakannya.

Histogram

Di ggplot2, implementasi Wickham dari tata bahasa Wilkinson adalah struktur yang mudah dipelajari untuk kode grafis R. Untuk mempelajari struktur itu, pastikan Anda memiliki ggplot2 di perpustakaan sehingga Anda dapat mengikuti apa yang terjadi selanjutnya. (Temukan ggplot2 pada tab Paket dan klik kotak centangnya.)

Grafik dimulai dengan ggplot(), yang membutuhkan dua argumen. Argumen pertama adalah sumber data. Argumen kedua memetakan komponen data yang diinginkan ke dalam komponen grafik. Fungsi yang melakukan pekerjaan adalah aes().

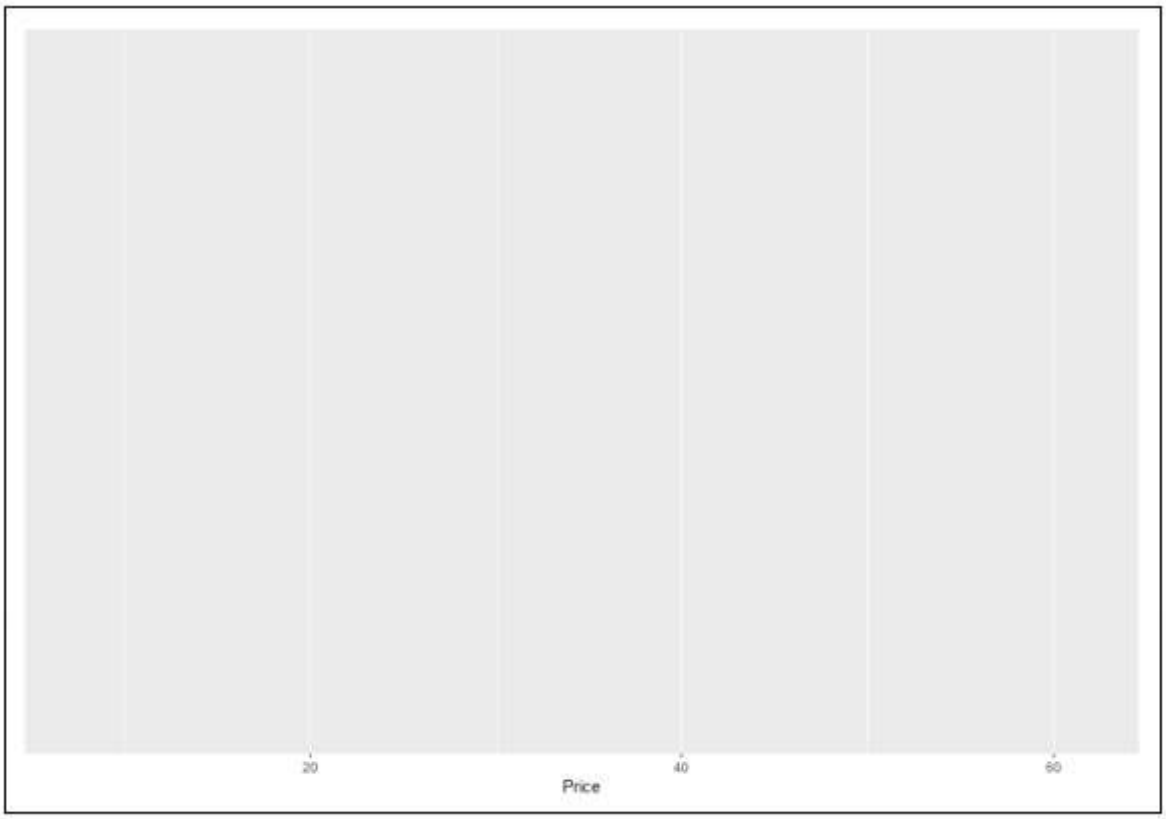
Untuk memulai histogram untuk Harga di Mobil93, fungsinya adalah

```
> ggplot(Cars93, aes(x=Price))
```

Fungsi aes() mengaitkan Harga dengan sumbu x. Di dunia ggplot, ini disebut pemetaan estetika. Faktanya, setiap argumen ke aes() disebut estetika. Baris kode ini menggambar Gambar 3.17, yang hanya berupa kotak dengan latar belakang abu-abu dan Harga pada sumbu x. Nah, bagaimana dengan sumbu y? Apakah sesuatu di peta data ke dalamnya? Tidak. Itu karena ini adalah histogram dan tidak ada data yang secara eksplisit memberikan nilai y untuk setiap x. Jadi Anda tidak bisa mengatakan "y=" di aes(). Sebagai gantinya, Anda membiarkan R melakukan pekerjaan untuk menghitung ketinggian batang di histogram.

Dan bagaimana dengan histogram itu? Bagaimana Anda memasukkannya ke dalam kotak kosong ini? Anda harus menambahkan sesuatu yang menunjukkan bahwa Anda ingin memplot histogram dan membiarkan R mengurus sisanya. Apa yang Anda tambahkan adalah fungsi geom ("geom" adalah kependekan dari "objek geometris"). Fungsi geom ini datang dalam berbagai jenis. ggplot2 menyediakan satu untuk hampir semua kebutuhan grafik, dan memberikan

fleksibilitas untuk bekerja dengan kasus khusus. Untuk menggambar histogram, fungsi geom yang digunakan disebut geom_histogram().



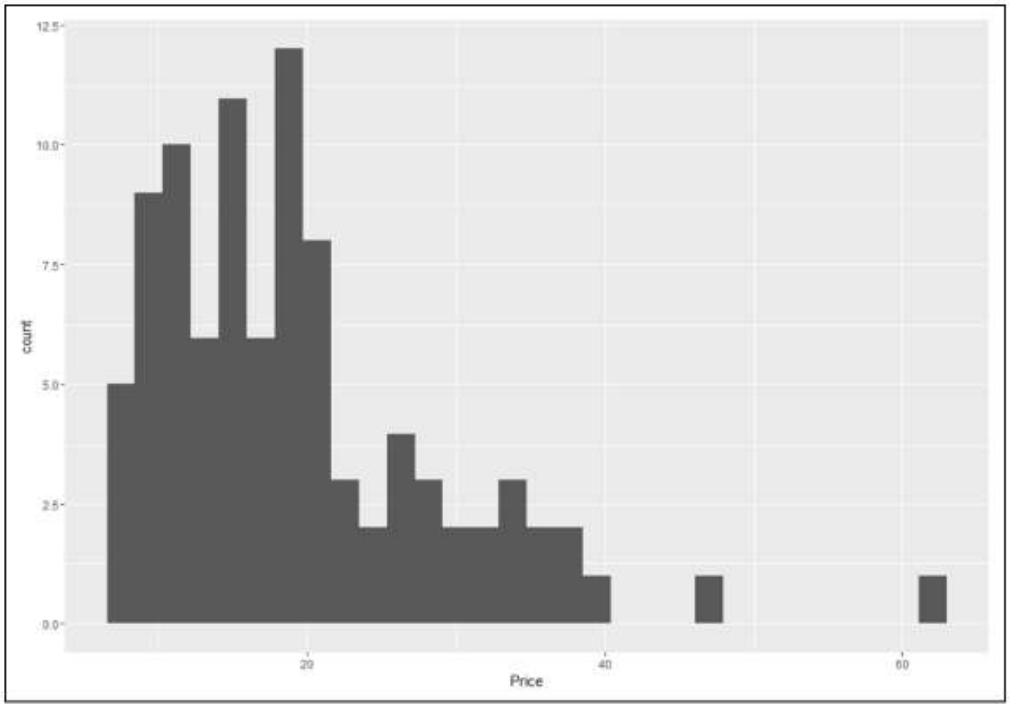
Gambar 3.17 Menerapkan ggplot() dan tidak ada yang lain.

Bagaimana Anda menambahkan geom_histogram() ke ggplot()? Dengan tanda plus:

```
ggplot(Cars93, aes(x=Price)) +
  geom_histogram()
```

Ini menghasilkan Gambar 3.18. Aturan tata bahasa memberi tahu ggplot2 bahwa ketika objek geometris adalah histogram, R melakukan perhitungan yang diperlukan pada data dan menghasilkan plot yang sesuai.

Minimal, kode grafis ggplot2 harus memiliki data, pemetaan estetika, dan objek geometris. Ini seperti menjawab urutan pertanyaan yang logis: Apa sumber datanya? Bagian mana dari data yang Anda minati? Bagian mana dari data yang sesuai dengan bagian grafik yang mana? Bagaimana Anda ingin grafik terlihat?



Gambar 3.18 Histogram awal untuk Harga di Mobil93.

Di luar persyaratan minimum tersebut, Anda dapat memodifikasi grafik. Setiap bilah disebut bin, dan secara default, ggplot() menggunakan 30 di antaranya. Setelah memplot histogram, ggplot() menampilkan pesan di layar yang menyarankan bereksperimen dengan binwidth (yang, tidak mengejutkan, menentukan lebar setiap bin) untuk mengubah tampilan grafik. Dengan demikian, Anda menggunakan binwidth = 5 sebagai argumen di geom_histogram().

Argumen tambahan mengubah tampilan bilah:

```
geom_histogram(binwidth=5, color = “black”, fill = “white”)
```

Dengan fungsi lain, labs(), Anda memodifikasi label untuk sumbu dan memberikan judul untuk grafik:

```
labs(x = "Price (x $1000)", y="Frequency",title="Prices of 93
         Models of 1993 Cars")
```

Sama sekali sekarang:

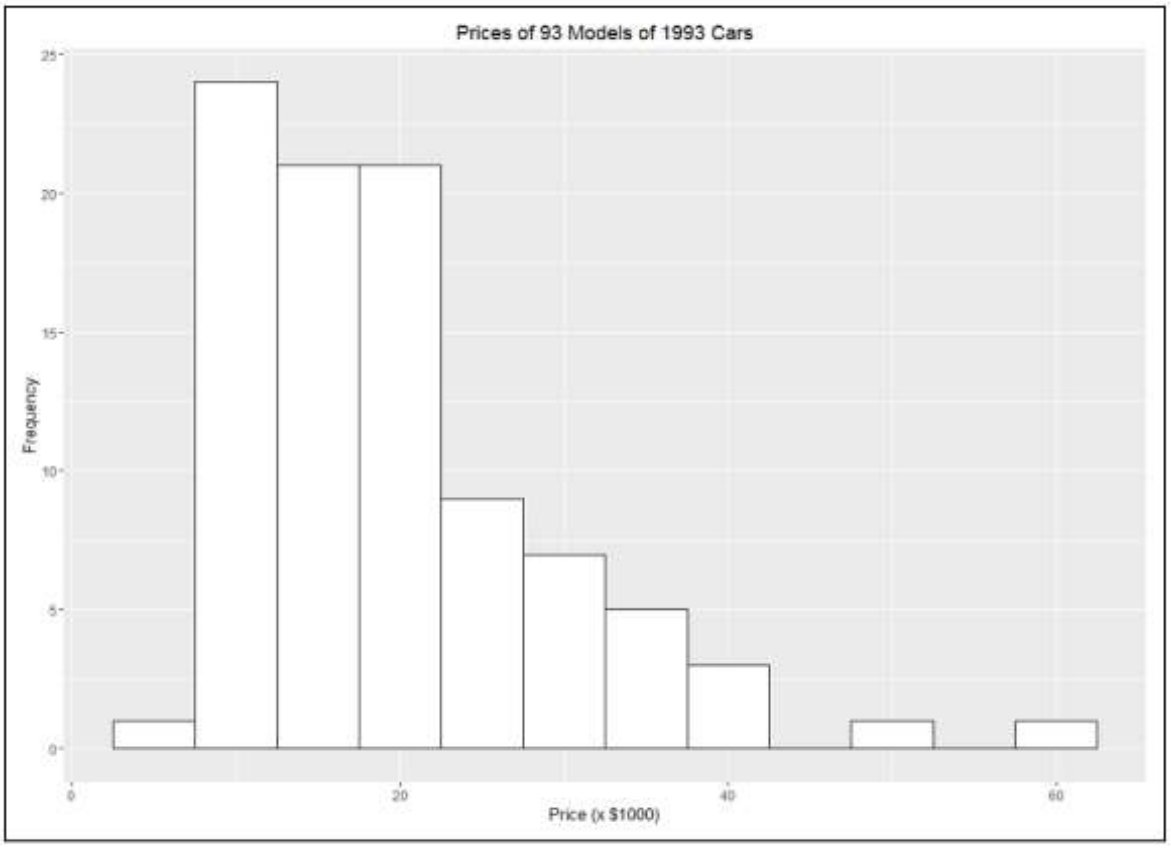
```
ggplot(Cars93, aes(x=Price)) +
  geom_histogram(binwidth=5,color="black",fill="white") +
  labs(x = "Price (x $1000)", y="Frequency", title="Prices of
          93 Models of 1993 Cars")
```

Hasilnya adalah Gambar 3-19. (Perhatikan bahwa ini sedikit berbeda dari Gambar 3-2. Saya harus sedikit mengotak-atik keduanya untuk membuatnya sama.)

Plot bar

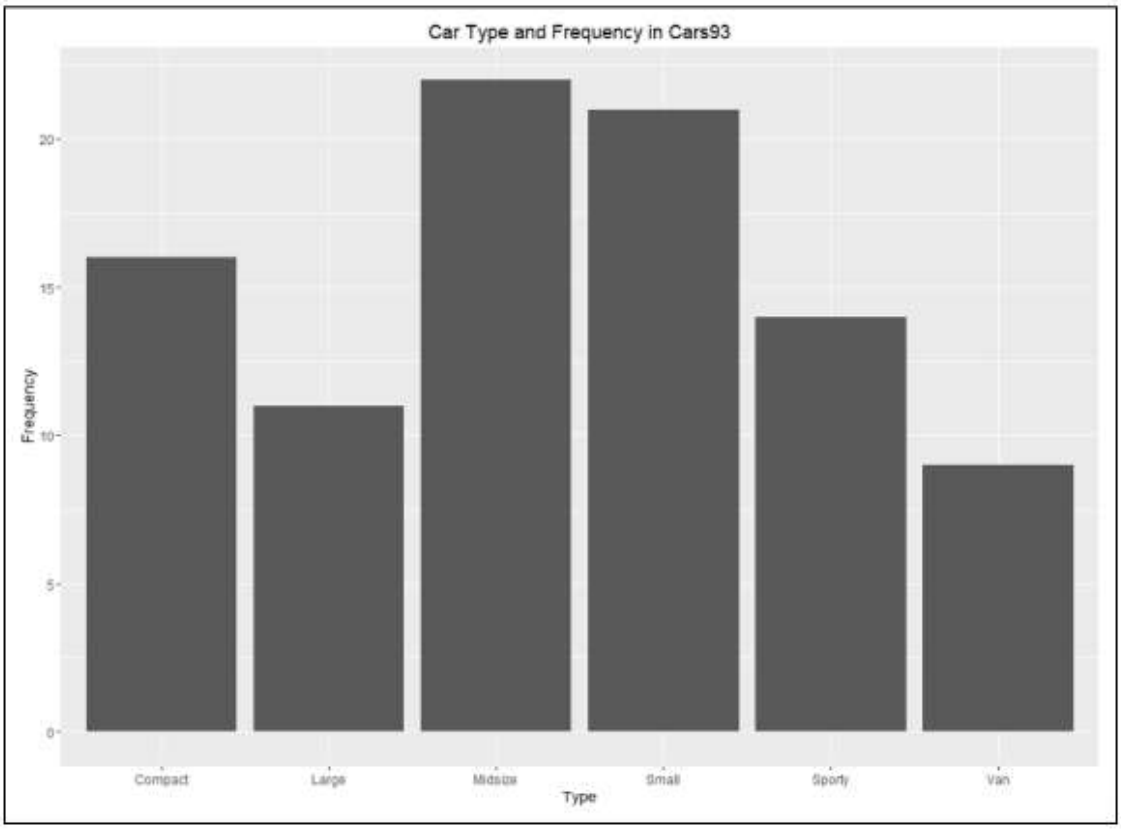
Menggambar plot batang di ggplot2 sedikit lebih mudah daripada menggambar di basis R: Tidak perlu membuat tabel seperti Tabel 3.1 terlebih dahulu untuk menggambar grafik. Seperti pada contoh di bagian sebelumnya, Anda tidak menentukan pemetaan estetika untuk y. Kali ini, fungsi geomnya adalah geom_bar(), dan aturan tata bahasanya memberi tahu ggplot2 untuk melakukan pekerjaan yang diperlukan dengan data dan kemudian menggambar plotnya:

```
ggplot(Cars93, aes(x=Type))+
  geom_bar() +
  labs(y="Frequency", title="Car Type and Frequency in Cars93")
```



Gambar 3.19 Histogram Harga yang sudah jadi.

Gambar 3.20 menunjukkan diagram batang yang dihasilkan.



Gambar 3.20 Petak batang untuk Tipe Mobil.

Diagram titik

Sebelumnya dalam bab ini, saya menunjukkan kepada Anda diagram titik sebagai alternatif dari diagram lingkaran. Di bagian ini, saya memberi tahu Anda cara menggunakan ggplot() untuk menggambarnya.

Mengapa saya tidak memimpin dengan diagram lingkaran dan menunjukkan cara membuatnya dengan paket ggplot2? Ini banyak pekerjaan, dan sedikit untung. Jika Anda ingin membuatnya, fungsi base R pie() jauh lebih mudah digunakan.

Membuat bagan titik dimulai dengan cara yang sama seperti di basis R: Anda membuat tabel untuk Type, dan Anda mengubah tabel menjadi bingkai data.

```
type.frame <- data.frame(table(Cars$93.Type))
```

Untuk memastikan bahwa Anda memiliki nama variabel yang bermakna untuk pemetaan estetika, Anda menerapkan fungsi colnames() untuk memberi nama kolom dalam bingkai data ini. (Itu adalah langkah yang tidak saya lakukan di base R.)

```
colnames(type.frame)<- c("Type","Frequency")
```

Sekarang type.frame terlihat seperti Tabel 3.1:

```
> type.frame
     Type Frequency
1 Compact        16
2   Large        11
3 Midsize        22
4   Small        21
5  Sporty        14
6     Van         9
```

Ke grafik. Untuk mengarahkan diagram titik seperti pada Gambar 3.11, Anda memetakan Frekuensi ke sumbu x dan Ketik ke sumbu y:

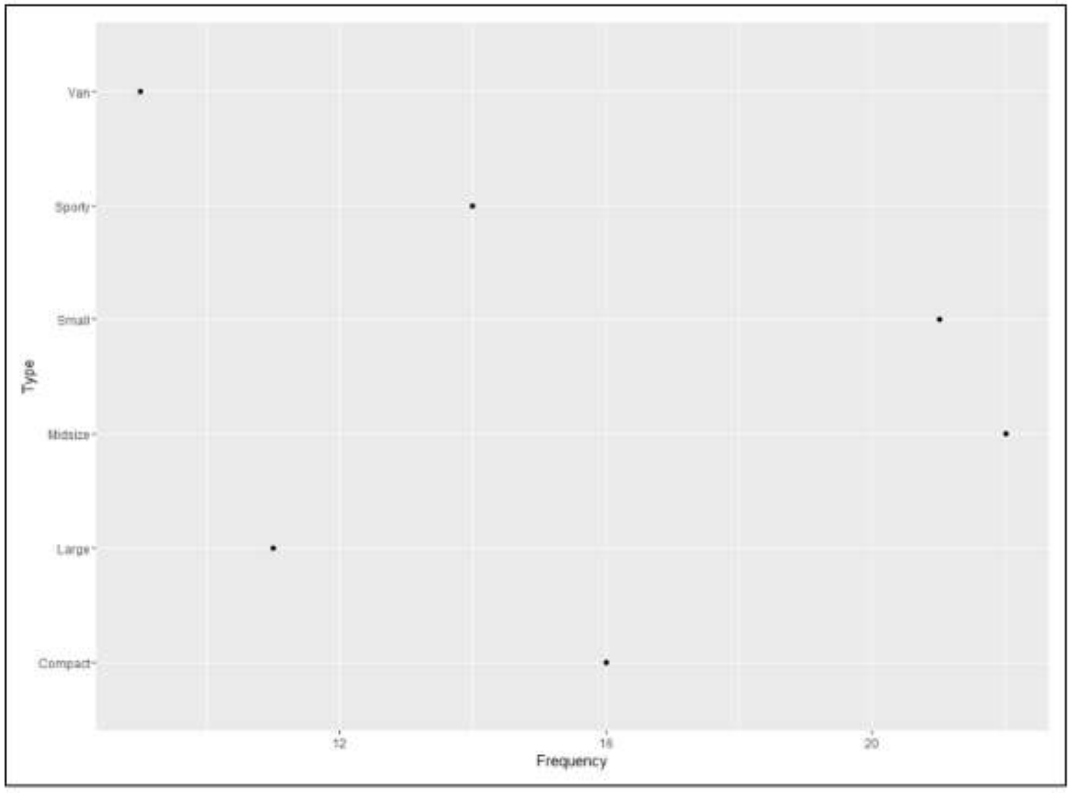
```
ggplot(type.frame, aes(x=Frequency,y= Type))
```

Sekali lagi, biasanya variabel bebas berada pada sumbu x dan variabel terikat berada pada sumbu y, tetapi tidak demikian halnya dalam grafik ini. Selanjutnya, Anda menambahkan fungsi geom. Fungsi geom yang disebut geom_dotplot() tersedia, tetapi yang mengejutkan, itu tidak sesuai di sini. Yang itu menggambar sesuatu yang lain. Di dunia ggplot, plot titik berbeda dari diagram titik. pergilah.

Fungsi geom untuk diagram titik adalah geom_point(). Jadi kode ini

```
ggplot(type.frame, aes(x=Frequency,y=Type)) +
  geom_point()
```

hasil pada Gambar 3.21.



Gambar 3.21 Diagram titik awal untuk Tipe.

Beberapa modifikasi sedang dilakukan. Pertama, dengan grafik seperti ini, adalah sentuhan yang bagus untuk mengatur ulang kategori pada sumbu y sehubungan dengan urutannya pada apa yang Anda ukur pada sumbu x. Itu memerlukan sedikit perubahan dalam pemetaan estetika ke sumbu y:

```
ggplot(type.frame, aes(x=Frequency,y=reorder(Type,Frequency))
```

Titik yang lebih besar akan membuat grafik terlihat sedikit lebih bagus:

```
geom_point(size =4)
```

Fungsi tambahan memodifikasi tampilan grafik secara keseluruhan. Satu keluarga dari fungsi-fungsi ini disebut tema. Salah satu anggota keluarga ini, theme_bw(), menghapus latar belakang abu-abu. Menambahkan theme() dengan argumen yang sesuai a) menghapus garis vertikal di grid dan b) menghitamkan garis horizontal dan membuatnya putus-putus:

```
theme_bw() +
theme(panel.grid.major.x=element_blank(),
      panel.grid.major.y=element_line(color = "black",
          linetype = "dotted"))
```

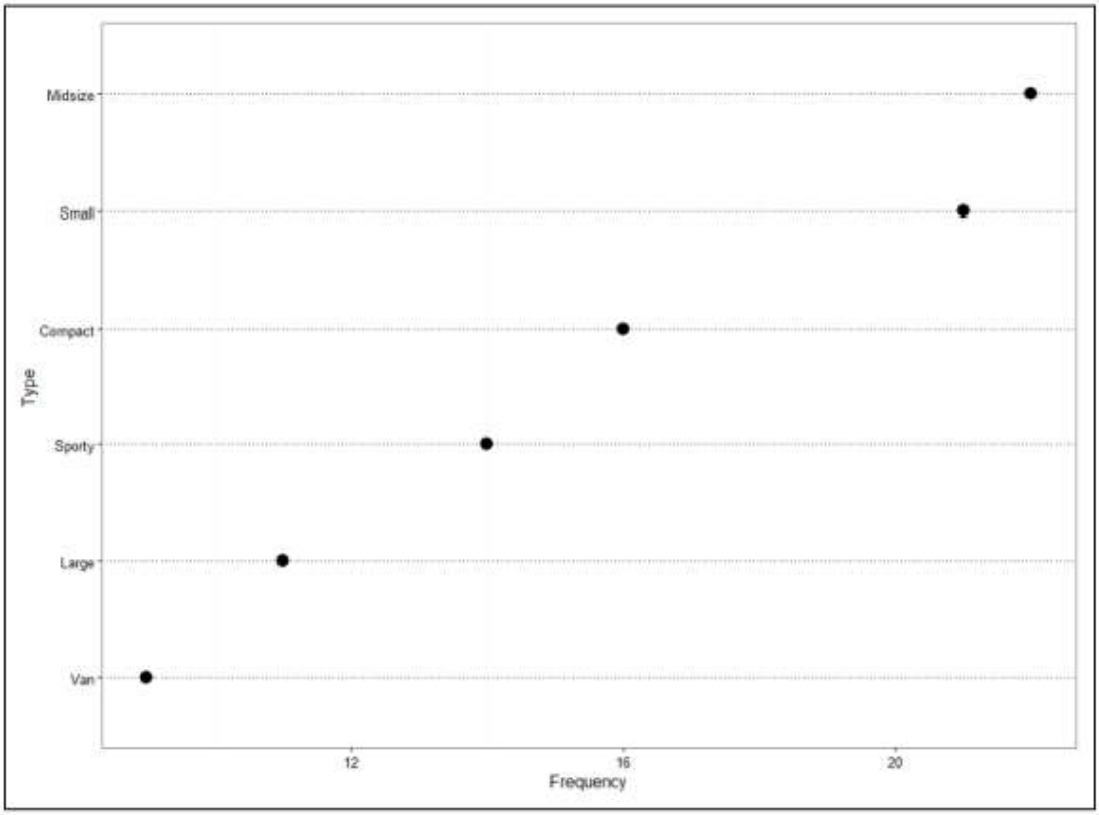
Terakhir, labs() mengubah label sumbu y:

```
labs(y= “Type”)
```

Tanpa perubahan itu, label sumbu y akan menjadi "reorder(Type,Frequency)". Meskipun indah, label itu tidak masuk akal bagi pemirsa rata-rata.
Berikut kode dari awal hingga akhir:

```
ggplot(type.frame, aes(x=Frequency,y=reorder(Type,Frequency))) +
  geom_point(size = 4) +
  theme_bw() +
  theme(panel.grid.major.x=element_blank(),
        panel.grid.major.y=element_line(color = "black",linetype
          = "dotted"))+
  labs(y="Type")
```

Gambar 3.22 menunjukkan diagram titik.



Gambar 3.22 Diagram titik yang dimodifikasi untuk Type.

Plot bar ditinjau kembali

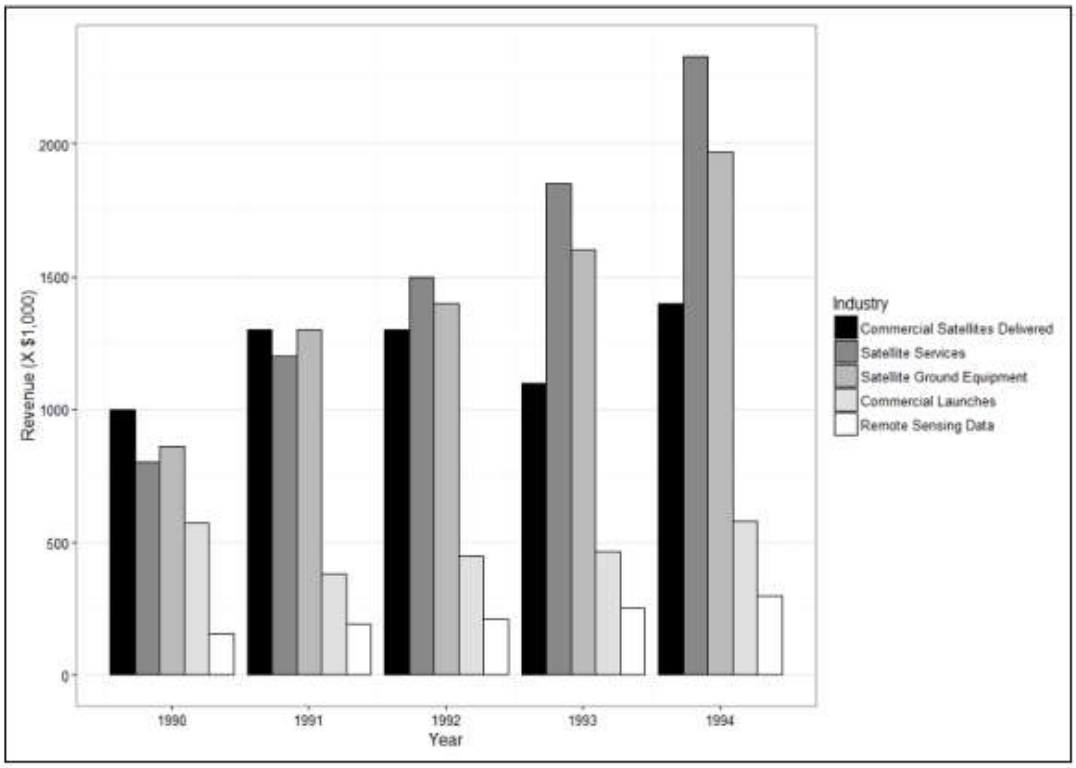
Seperti halnya dengan beberapa grafik pertama di basis R, grafik yang saya tunjukkan sejauh ini di bagian ini memiliki frekuensi (atau "dihitung") sebagai variabel dependen. Dan, tentu saja, seperti yang ditunjukkan Tabel 3.2, tidak selalu demikian.

Di bagian basis R, saya menunjukkan cara membuat plot batang yang dikelompokkan. Di sini, saya menunjukkan cara menggunakan ggplot() untuk membuatnya dari space.rev, kumpulan data yang saya buat dari data di Tabel 3.2. Produk jadi akan terlihat seperti Gambar 3.23.

Urutan pertama bisnis adalah menyiapkan data. Ini bukan dalam format yang digunakan ggplot(). format ini disebut format lebar.

```
> space.rev
                              1990 1991 1992 1993 1994
Commercial Satellites Delivered 1000 1300 1300 1100 1400
Satellite Services               800 1200 1500 1850 2330
Satellite Ground Equipment       860 1300 1400 1600 1970
Commercial Launches              570  380  450  465  580
Remote Sensing Data              155  190  210  250  300
```

ggplot(), bagaimanapun, bekerja dengan format panjang, yang terlihat seperti ini:



Gambar 3.23 Plot batang untuk data pada Tabel 3-2, dibuat dengan ggplot().

```
Industry Year Revenue
1 Commercial Satellites Delivered 1990    1000
2             Satellite Services 1990     800
3     Satellite Ground Equipment 1990     860
4            Commercial Launches 1990     570
5            Remote Sensing Data 1990     155
6 Commercial Satellites Delivered 1991    1300
```

Itu hanya enam baris pertama untuk kumpulan data ini. Jumlah baris adalah 25 (karena 5 baris dan 5 kolom dalam format lebar).

Hadley Wickham (ada nama itu lagi!) membuat paket bernama reshape2 yang menyediakan segalanya untuk transformasi yang mulus. Fungsi melt() mengubah format lebar menjadi panjang. Fungsi lain, cast(), melakukan kebalikannya. Fungsi-fungsi ini sangat membantu karena menghilangkan kebutuhan untuk berpindah-pindah dalam spreadsheet untuk membentuk kembali kumpulan data.

Jadi, dengan reshape2 di perpustakaan (klik kotak centangnya di tab Paket), kodenya adalah:

```
> space.melt <- melt(space.rev)
```

Ya, itu benar-benar semua yang ada untuk itu. Di sini, saya akan membuktikannya kepada Anda:

```
> head(space.melt)
                          Var1 Var2 value
1 Commercial Satellites Delivered 1990  1000
2             Satellite Services 1990   800
3     Satellite Ground Equipment 1990   860
4            Commercial Launches 1990   570
5            Remote Sensing Data 1990   155
6 Commercial Satellites Delivered 1991  1300
```

Selanjutnya, Anda memberi nama yang berarti ke kolom:

```
> colnames(space.melt) <- c("Industry","Year","Revenue")
> head(space.melt)
                        Industry Year Revenue
1 Commercial Satellites Delivered 1990    1000
2             Satellite Services 1990     800
3     Satellite Ground Equipment 1990     860
4            Commercial Launches 1990     570
5            Remote Sensing Data 1990     155
6 Commercial Satellites Delivered 1991    1300
```

Dan sekarang Anda siap untuk berguling. Anda mulai dengan ggplot(). Pemetaan estetika sangat mudah:

```
ggplot(space.melt, aes(x=Year,y=Revenue,fill=Industry))
```

Anda menambahkan fungsi geom untuk bilah, dan Anda menentukan tiga argumen:

```
ggplot(space.melt, aes(x=Year,y=Revenue,fill=Industry))
```

Argumen pertama mutlak diperlukan untuk graf jenis ini. Jika dibiarkan sendiri, default geom_bar ke plot batang yang saya tunjukkan sebelumnya — grafik berdasarkan frekuensi. Karena Anda mendefinisikan pemetaan estetika untuk y, dan jenis grafik tersebut tidak sesuai dengan estetika untuk y, tidak menyetel argumen ini akan menghasilkan pesan kesalahan. Oleh karena itu, Anda memberi tahu ggplot() bahwa ini adalah grafik berdasarkan nilai data eksplisit. Jadi stat="identitas" berarti "menggunakan angka yang diberikan sebagai data."

Nilai untuk argumen berikutnya, posisi, adalah nama lucu yang berarti batang "menghindar" satu sama lain dan berbaris berdampingan. (Abaikan argumen ini dan lihat apa yang terjadi.) Ini analog dengan "di samping =T" di basis R. Argumen ketiga menetapkan warna batas untuk setiap batang. Skema warna isian untuk batang adalah provinsi dari fungsi berikutnya:

```
scale_fill_grey(start = 0,end = 1)
```

Seperti namanya, fungsi ini mengisi bilah dengan nuansa abu-abu (permisi, "abu-abu"). Nilai awal, 0, berwarna hitam, dan nilai akhir, 1, berwarna putih. (Mengingatkan pada "grey0" = "black" dan "grey100" = "white.') Efeknya adalah mengisi lima batang dengan lima warna dari hitam ke putih.

Anda ingin memberi label ulang sumbu y, jadi itu

```
labs(y="Revenue (X $1,000)")
```

dan kemudian hapus latar belakang abu-abu

```
theme_bw()
```

dan, akhirnya, hapus garis vertikal dari kisi

```
theme(panel.grid.major.x = element_blank())
```

Seluruh potongan untuk memproduksi Gambar 3.23 adalah:

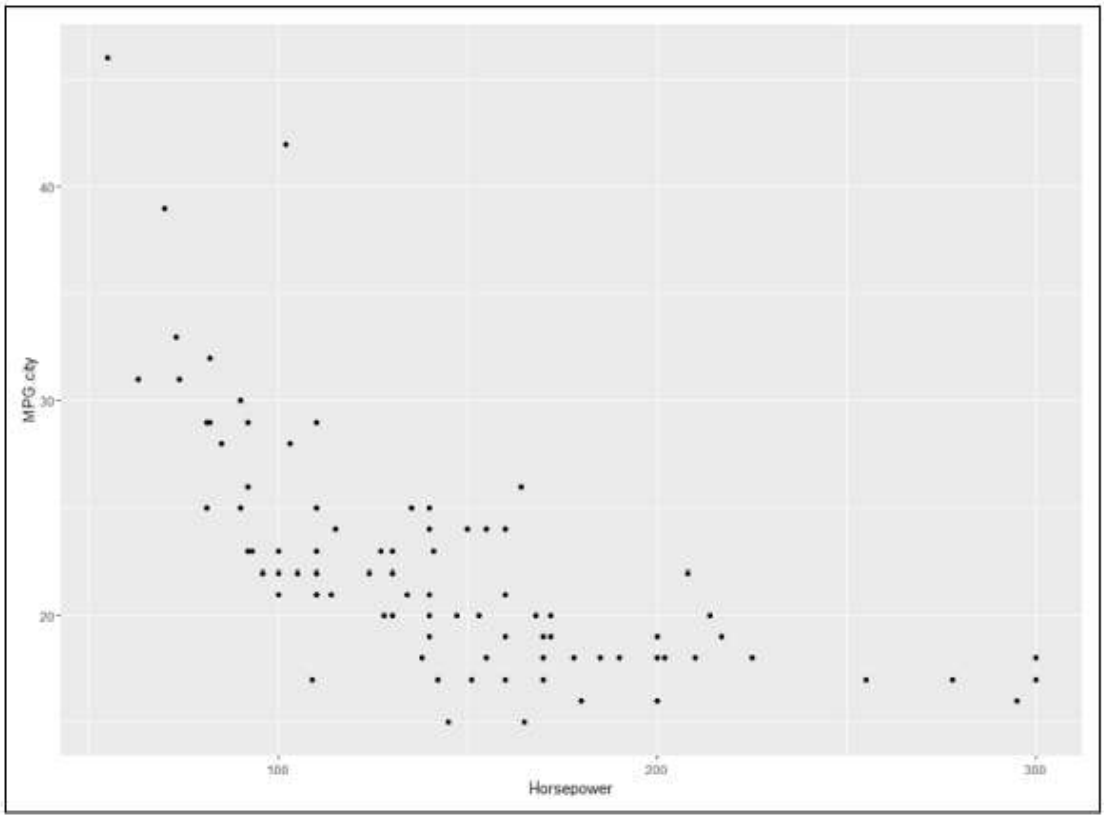
```
ggplot(space.melt, aes(x=Year,y=Revenue,fill=Industry)) +
  geom_bar(stat = "identity", position = "dodge", color="black") +
  scale_fill_grey(start = 0,end = 1)+
  labs(y="Revenue (X $1,000)")+
  theme_bw()+
  theme(panel.grid.major.x = element_blank())
```

Plot sebar

Seperti yang saya jelaskan sebelumnya, plot pencar adalah cara yang bagus untuk menunjukkan hubungan antara dua variabel, seperti tenaga kuda dan mil per galon untuk mengemudi di kota. Dan ggplot() adalah cara yang bagus untuk menggambar plot pencar. Jika Anda telah mengikutinya, tata bahasanya akan mudah bagi Anda:

```
ggplot(Cars93,aes(x=Horsepower,y=MPG.city))+
  geom_point()
```

Gambar 3.24 menunjukkan plot pencar. Saya akan menyerahkan kepada Anda untuk mengubah label sumbu y menjadi "Mil per Galon (Kota)" dan menambahkan judul deskriptif.



Gambar 3.24 MPG.city vs Horsepower di Cars93.

Tentang plot twist itu. . .

Perhatikan lagi Gambar 3.15, hubungan antara MPG.city dan Horsepower. Dalam hal itu, titik-titik dalam plot bukanlah titik. Sebaliknya, setiap titik data adalah jumlah silinder, yang merupakan label yang muncul sebagai karakter teks.

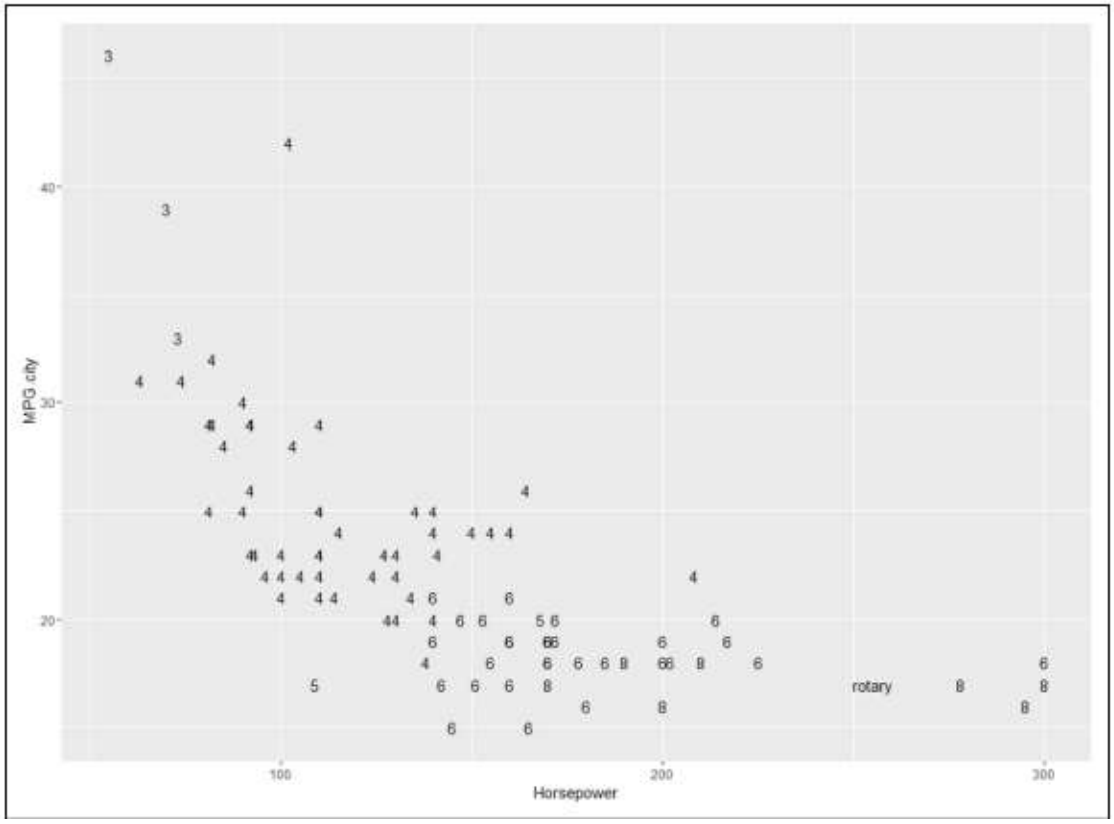
Bagaimana Anda mewujudkannya di dunia ggplot? Pertama, Anda memerlukan pemetaan estetika tambahan di aes(). Pemetaan itu adalah label, dan Anda menyetelnya ke Silinder:

```
ggplot(Cars93, aes(x=Horsepower, y=MPG.city, label = Cylinders))
```

Anda menambahkan objek geometris untuk teks dan voila:

```
ggplot(Cars93, aes(x = Horsepower,y = MPG.city,label =
  Cylinders)) +
  geom_text()
```

Gambar 3.25 menunjukkan grafik yang dihasilkan kode ini. Satu perbedaan dari basis R adalah "rotary" daripada "r" sebagai label titik data.



Gambar 3.25 Plot sebar ggplot2 awal untuk MPG.city vs Horsepower dengan Cylinders sebagai label titik data.

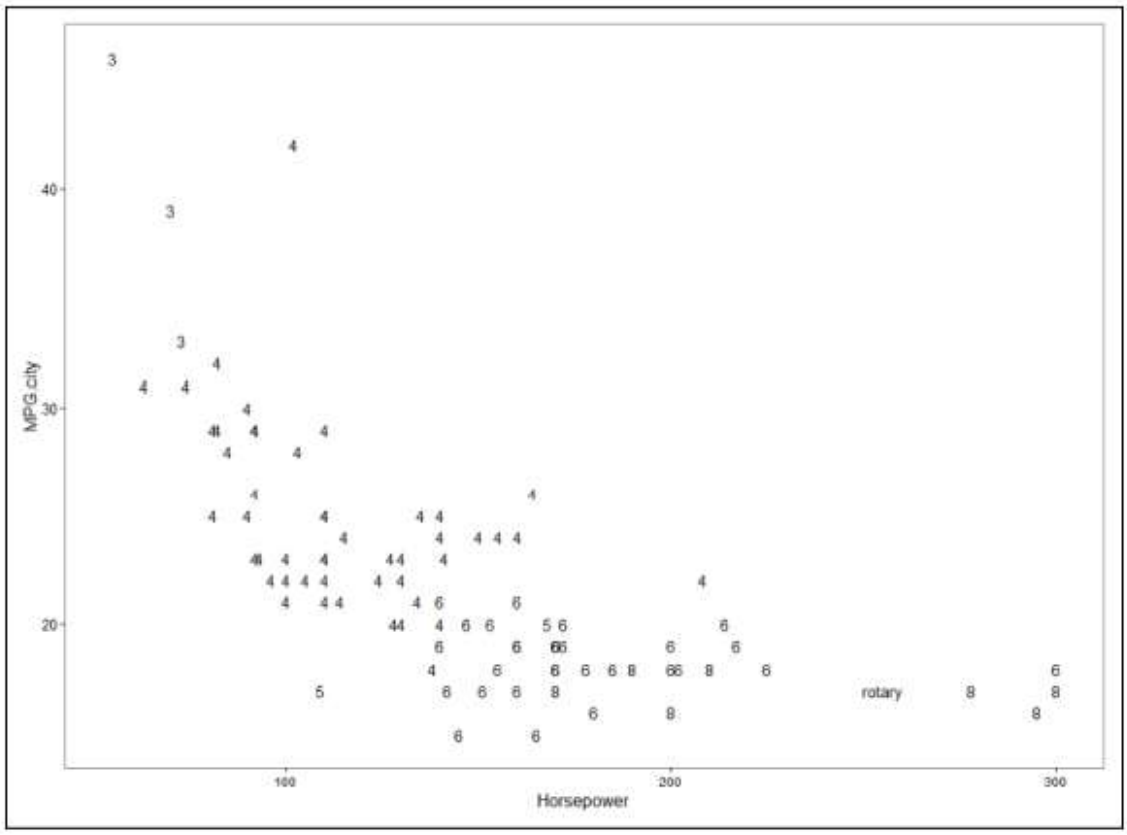
Hanya untuk itu, saya menggunakan fungsi tema (lihat bagian "Diagram titik" sebelumnya) untuk membuat tampilan grafik lebih terlihat seperti yang ditunjukkan pada Gambar 3.15. Seperti pada contoh diagram titik, theme_bw() menghilangkan latar belakang abu-abu. Fungsi theme() (dengan argumen tertentu) menghilangkan grid:

```
theme(panel.grid=element_blank())
```

element_blank() adalah fungsi yang menggambar elemen kosong.
Menyatukan semuanya

```
ggplot(Cars93, aes(x=Horsepower, y=MPG.city, label=Cylinders)) +
  geom_text() +
  theme_bw() +
  theme(panel.grid=element_blank())
```

menghasilkan Gambar 3-26. Sekali lagi, saya serahkan kepada Anda untuk menggunakan labs() untuk mengubah label sumbu y dan menambahkan judul deskriptif.



Gambar 3.26 Plot sebar yang dimodifikasi untuk MPG.city vs Horsepower dengan Cylinders sebagai label titik data.

Matriks petak sebar

Sebuah matriks plot pencar menunjukkan hubungan berpasangan antara lebih dari dua variabel. Gambar 3.16 menunjukkan bagaimana fungsi base R pairs() menggambar matriks semacam ini. Paket ggplot2 memiliki fungsi yang disebut plotpairs() yang melakukan hal serupa, tetapi tidak lagi. GGally, sebuah paket yang dibangun di atas ggplot2, menyediakan ggpairs() untuk menggambar matriks plot sebar, dan ia melakukannya dengan cara yang flamboyan. Paket GGally tidak ada di tab Paket. Anda harus memilih Instal dan ketik GGally di kotak dialog Instal Paket. Ketika muncul di tab Paket, klik kotak centang di sebelahnya.

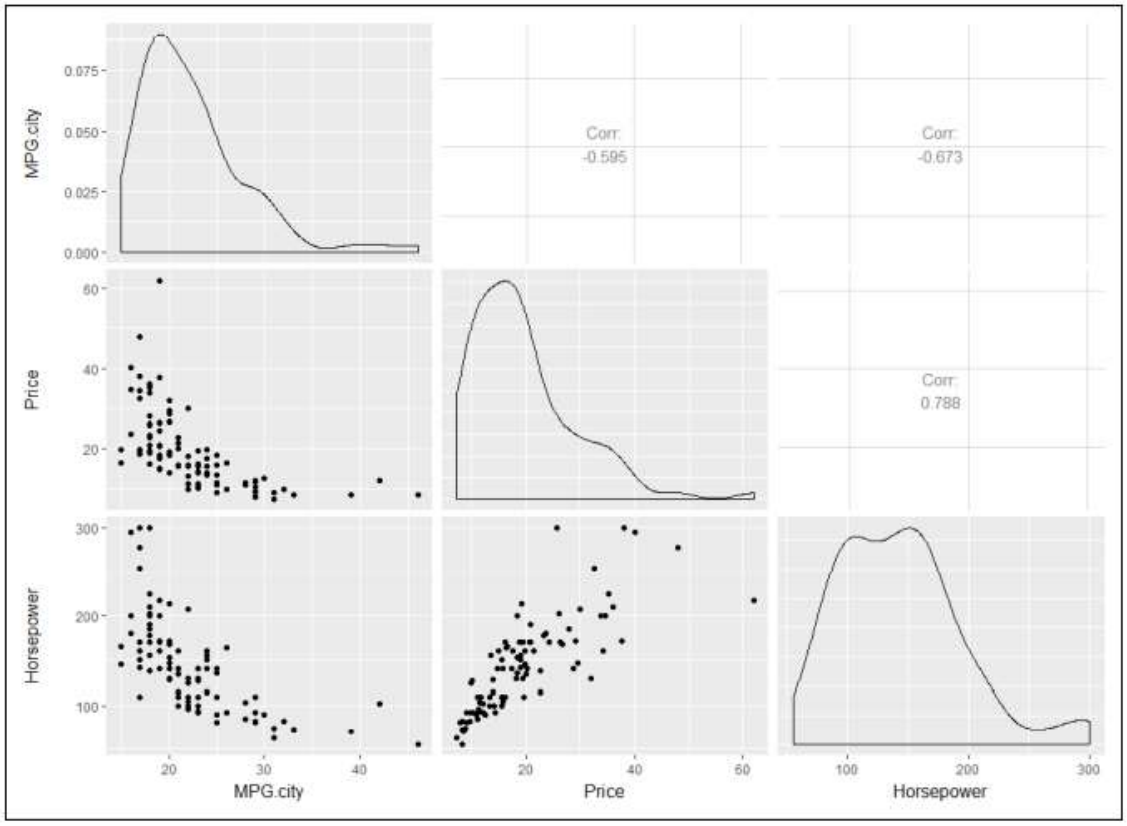
Sebelumnya, saya membuat subset Cars93 yang mencakup MPG.city, Price, dan Horsepower:

```
> cars.subset <- subset(Cars93, select = c(MPG.
  city,Price,Horsepower))
```

Dengan paket GGally di perpustakaan Anda, kode ini membuat matriks plot sebar pada Gambar 3.27:

```
> ggpairs(cars.subset)
```

Seperti yang ditunjukkan Gambar 3.27, yang ini cantik. Sel-sel di sepanjang diagonal utama menyajikan plot kepadatan variabel. (Lihat subbagian sebelumnya “Menambahkan fitur grafik,” dan juga lihat Bab 8.) Salah satu kelemahannya adalah sumbu y terlihat untuk variabel MPG.city hanya di baris pertama dan kolom pertama.



Gambar 3.27 Matriks petak sebar untuk MPG. kota, Harga, dan Horsepower.

Tiga plot pencar berada di sel di bawah diagonal utama. Daripada menunjukkan plot sebar yang sama dengan sumbu terbalik dalam sel di atas diagonal utama (seperti yang dilakukan pasangan()), setiap sel di atas diagonal menunjukkan koefisien korelasi yang merangkum hubungan antara variabel baris sel dan variabel kolomnya . (Koefisien korelasi? Tidak, saya tidak akan menjelaskannya sekarang. Lihat Bab 15.)

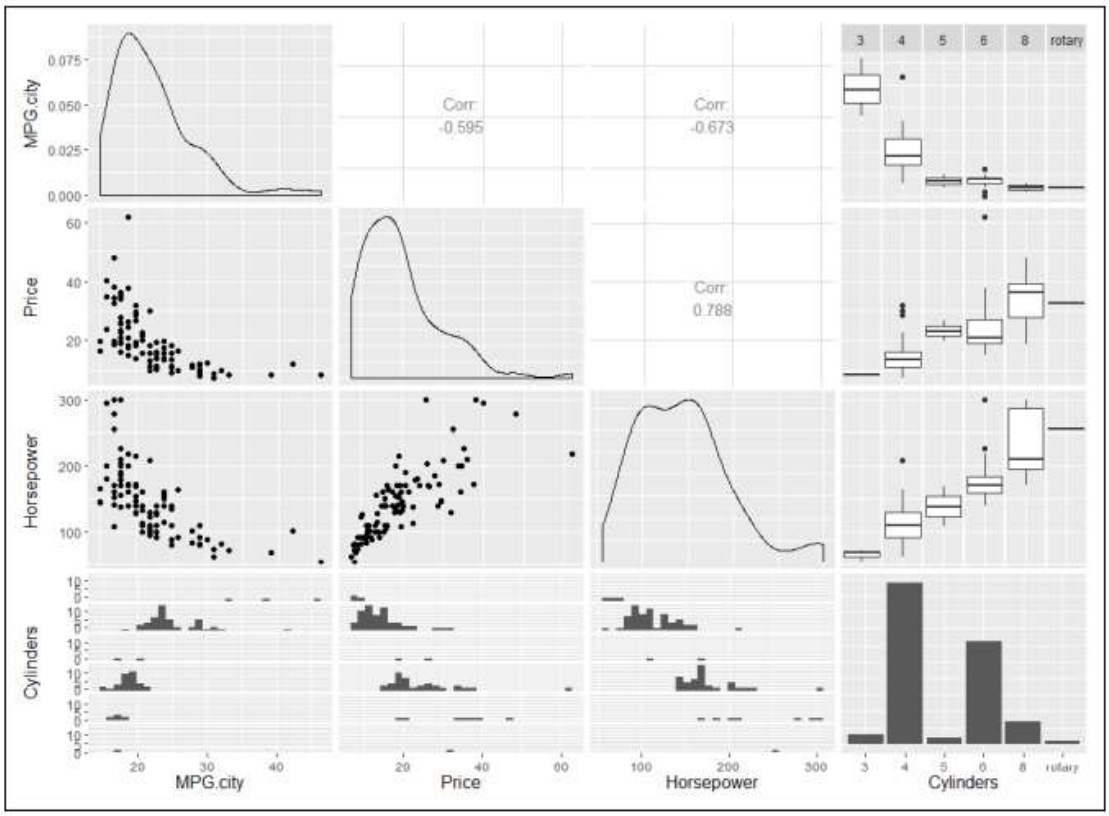
Untuk suguhan visual yang nyata, tambahkan Cylinders ke cars.subset, lalu terapkan ggpairs():

```
> cars.subset <- subset(Cars93, select = c(MPG.city,Price,
          Horsepower,Cylinders))
> ggpairs(cars.subset)
```

Gambar 3.28 menunjukkan matriks plot pencar yang baru, dengan segala kelengkapannya.

Silinder bukanlah variabel yang cocok untuk menyebarkan plot, plot kepadatan, atau koefisien korelasi. (Pertanyaan pikiran: Mengapa tidak?) Jadi, sel di kolom keempat, baris keempat,

memiliki plot batang dan bukan plot kepadatan. Plot batang yang menghubungkan Silinder (pada setiap sumbu y) dengan tiga variabel lainnya (pada sumbu x) berada di tiga sel yang tersisa di baris 4. Plot kotak yang menghubungkan Silinder (pada setiap sumbu x) dengan tiga variabel lainnya (pada sumbu y) berada di tiga sel yang tersisa di kolom 4.



Gambar 3.28 Menambahkan Silinder menghasilkan matriks plot sebar ini.

Plot kotak

Ahli statistik menggunakan plot kotak untuk menunjukkan dengan cepat bagaimana kelompok berbeda satu sama lain. Seperti pada contoh R dasar, saya menunjukkan plot kotak untuk Silinder dan Tenaga Kuda. Ini adalah replikasi dari grafik pada baris 3, kolom 4 dari Gambar 3.28.

Pada titik ini, Anda mungkin dapat mengetahui fungsi ggplot() :

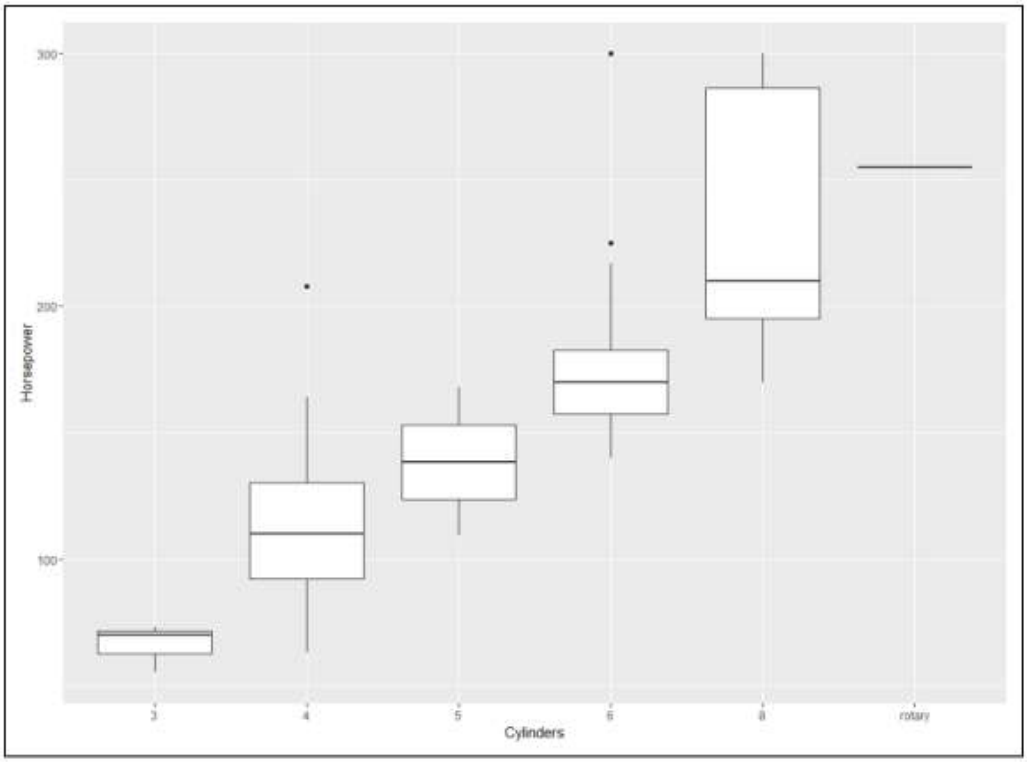
```
ggplot(Cars93, aes(x=Cylinders, y= Horsepower))
```

Apa fungsi geomnya? Jika Anda menebak geom_boxplot(), Anda benar!

Jadi kodenya adalah:

```
ggplot(Cars93, aes(x=Cylinders,y=Horsepower)) +
  geom_boxplot()
```

Dan itu memberi Anda Gambar 3.29.



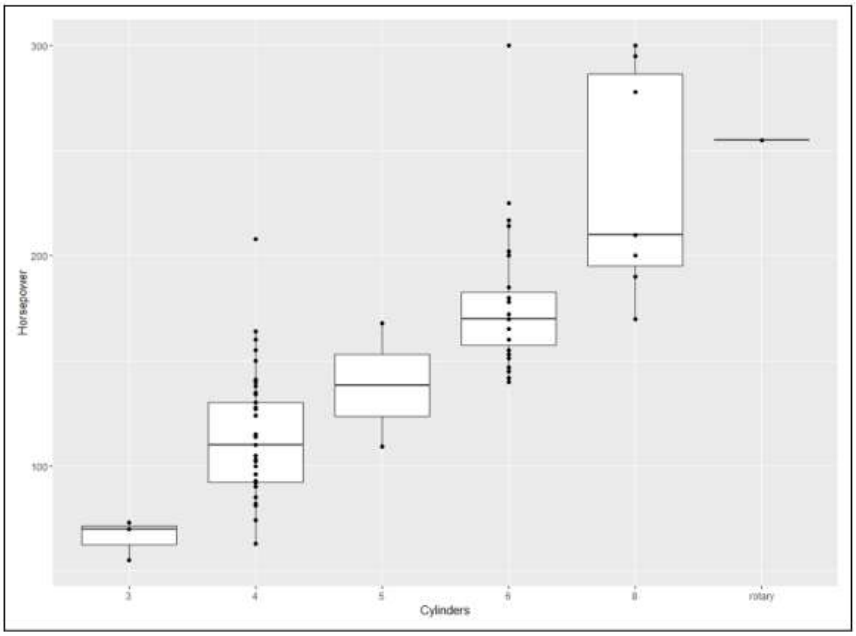
Gambar 3.29 Plot kotak untuk Horsepower vs Cylinders.

Ingin menampilkan semua titik data selain kotak? Tambahkan fungsi geom untuk poin untuk menghasilkan grafik pada Gambar 3.30.

```
ggplot(Cars93, aes(x=Cylinders,y=Horsepower)) +
  geom_boxplot()+
  geom_point()
```

Ingat bahwa ini adalah data untuk 93 mobil. Apakah Anda melihat 93 titik data? Saya juga tidak. Ini, tentu saja, karena banyak poin yang tumpang tindih. Ahli grafis menyebut ini sebagai overplotting.

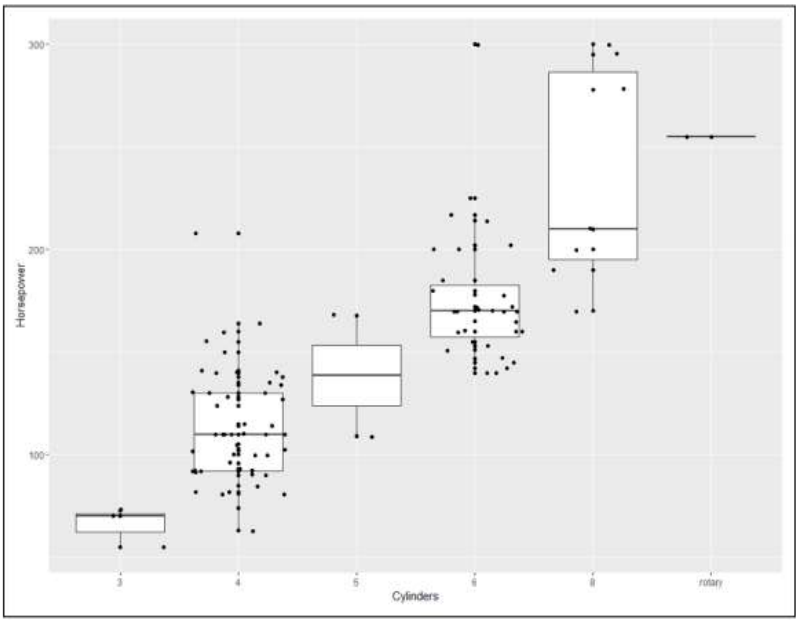
Salah satu cara untuk mengatasi overplotting adalah dengan memposisikan ulang titik-titik secara acak untuk mengungkapkannya tetapi tidak mengubah apa yang diwakilinya. Ini disebut jittering. Dan ggplot2 memiliki fungsi geom untuk itu: geom_jitter(). Menambahkan fungsi ini ke kode



Gambar 3.30 Plot kotak dengan titik data.

```
gplot(Cars93, aes(x=Cylinders,y=Horsepower)) +
  geom_boxplot()+
  geom_point()+
  geom_jitter()
```

menggambar Gambar 3.31.



Gambar 3.31 Plot kotak dengan titik data gelisah.

3.4 KESIMPULAN

Sejauh grafis berjalan, saya baru saja menggores permukaan. R memiliki seperangkat alat dan paket grafis yang kaya — jauh lebih banyak daripada yang bisa saya tunjukkan kepada Anda di bab ini. Dalam bab-bab selanjutnya, setiap kali saya menunjukkan kepada Anda teknik analitik, saya juga menunjukkan kepada Anda bagaimana memvisualisasikan hasilnya. Saya akan menggunakan apa yang telah Anda baca di bab ini, bersama dengan alat dan paket baru jika diperlukan.

BAB 4
MENEMUKAN PUSAT ANDA

Jika Anda pernah bekerja dengan sekumpulan angka dan harus mencari cara untuk meringkasnya dengan satu angka, Anda telah menghadapi situasi yang selalu dihadapi oleh para ahli statistik. Dari mana "angka tunggal" yang ideal ini berasal?

Ide yang baik mungkin untuk memilih nomor dari suatu tempat di tengah set. Angka itu kemudian bisa mewakili seluruh rangkaian angka. Saat Anda melihat sekeliling di tengah set, Anda melihat tendensi sentral. Anda dapat mengatasi tendensi sentral dengan berbagai cara.

4.1 RATA-RATA

Kita semua pernah menggunakan rata-rata. Ahli statistik menyebut rata-rata sebagai mean. Rata-rata adalah cara mudah untuk meringkas pengeluaran Anda, nilai sekolah Anda, kinerja Anda dalam olahraga dari waktu ke waktu.

Dalam perjalanan pekerjaan mereka, para ilmuwan menghitung cara. Ketika seorang peneliti melakukan penelitian, dia menerapkan beberapa jenis perlakuan atau prosedur pada sampel kecil orang atau benda. Kemudian dia mengukur hasil dan memperkirakan efek dari prosedur pada populasi yang menghasilkan sampel. Ahli statistik telah menunjukkan bahwa rata-rata sampel adalah perkiraan rata-rata populasi.

Saya pikir Anda tahu cara menghitung rata-rata, tetapi saya akan tetap melakukannya. Lalu saya tunjukkan rumus statistiknya. Tujuan saya adalah agar Anda memahami rumus statistik secara umum, dan kemudian saya akan menunjukkan kepada Anda bagaimana R menghitung artinya. Rata-rata hanyalah jumlah dari sekumpulan angka dibagi dengan berapa banyak angka yang Anda tambahkan. Misalkan Anda mengukur tinggi (dalam inci) enam anak berusia 5 tahun dan menemukan bahwa tinggi badan mereka adalah:

$$36, 42, 43, 37, 40, 45$$

Rata-rata tinggi badan keenam anak tersebut adalah

$$\frac{36+42+43+37+40+45}{6} = 40.5$$

Jadi, rata-rata sampel ini adalah 40,5 inci.

Upaya pertama pada formula untuk mean mungkin

$$Mean = \frac{jumlah\ angka}{jumlah\ angka\ yang\ anda\ tambahkan}$$

Rumus, meskipun, biasanya melibatkan singkatan. Singkatan umum untuk "Angka" adalah X. Ahli statistik biasanya menyingkat "Jumlah Angka yang Anda Tambahkan" sebagai N.

Jadi rumusnya menjadi

$$\text{Mean} = \frac{\text{Sum of } X}{N}$$

Ahli statistik juga menggunakan singkatan untuk Sum of — huruf besar Yunani untuk S. Diucapkan "sigma," terlihat seperti ini: . Jadi rumus dengan sigma adalah

$$\text{Mean} = \frac{\sum X}{N}$$

Saya belum selesai. Ahli statistik menyingkat "mean," juga. Anda mungkin berpikir bahwa M adalah singkatan, dan beberapa ahli statistik setuju dengan Anda, tetapi kebanyakan lebih memilih simbol yang terkait dengan X. Karena alasan ini, singkatan yang paling populer untuk mean adalah $\bar{X}$, yang diucapkan "X bar." Dan inilah rumusnya:

$$\bar{X} = \frac{\sum X}{N}$$

Saya harus mengikat satu lagi ujung yang longgar. Dalam Bab 1, saya membahas sampel dan populasi. Simbol dalam rumus harus mencerminkan perbedaan antara keduanya. Konvensinya adalah bahwa huruf Inggris, seperti $\bar{X}$, mewakili karakteristik sampel, dan huruf Yunani mewakili karakteristik populasi. Untuk mean populasi, simbolnya adalah padanan Yunani dari M, yaitu . Itu diucapkan seperti "kamu" tetapi dengan "m" di depannya. Maka rumus mean populasi adalah

$$\mu = \frac{\sum X}{N}$$

4.2 RATA-RATA DALAM R: MEAN()

R menyediakan cara yang sangat mudah untuk menghitung mean dari sekumpulan angka: mean(). Saya menerapkannya pada contoh tinggi enam anak.
Pertama, saya membuat vektor ketinggian:

```
> heights <- c(36, 42, 43, 37, 40, 45)
```

Kemudian saya menerapkan fungsi:

```
> mean(heights)
[1] 40.5
```

Dan di sana Anda memilikinya.

Apa kondisi Anda?

Saat Anda bekerja dengan bingkai data, terkadang Anda ingin menghitung rata-rata hanya kasus (baris) yang memenuhi kondisi tertentu, daripada rata-rata semua kasus. Ini mudah dilakukan di R. Untuk pembahasan selanjutnya, saya menggunakan kerangka data Cars93 yang sama dengan yang saya gunakan di Bab 3. Ini adalah salah satu yang memiliki data untuk sampel 93 mobil dari tahun 1993. Ada dalam paket MASS. Jadi pastikan Anda memiliki paket MASS di perpustakaan Anda. (Temukan MASS pada tab Packages dan klik kotak centangnya.)

Misalkan saya tertarik dengan tenaga kuda rata-rata mobil yang dibuat di AS. Pertama saya memilih mobil-mobil itu dan memasukkan tenaga kudanya ke dalam vektor:

```
Horsepower.USA <- Cars93$Horsepower[Cars93$Origin == "USA"]
```

(Jika bagian kanan dari baris itu tampak aneh bagi Anda, baca kembali Bab 2.)
Tenaga kuda rata-rata adalah

```
> mean(Horsepower.USA)
[1] 147.5208
```

Hmm, saya ingin tahu berapa rata-rata untuk mobil yang tidak dibuat di AS:

```
Horsepower.NonUSA <- Cars93$Horsepower[Cars93$Origin ==
          "non-USA"]
> mean(Horsepower.NonUSA)
[1] 139.8889
```

Jadi rata-ratanya sedikit berbeda. (Bisakah kita memeriksa perbedaan itu lebih dekat? Ya kita bisa, itulah yang saya lakukan di Bab 11.)

Hilangkan $-signs sebagainya dengan()

Dalam kode-R sebelumnya, tanda $ menunjukkan variabel dalam kerangka data Cars93. R menyediakan jalan keluar untuk menggunakan nama kerangka data (dan karenanya, tanda $) setiap kali Anda merujuk ke salah satu variabelnya.

Dalam Bab 3, saya menunjukkan bahwa fungsi grafik mengambil, sebagai argumen pertama mereka, sumber data. Kemudian, dalam daftar argumen, tidak perlu mengulang sumber bersama dengan tanda $ untuk menunjukkan variabel yang akan diplot. Fungsi with() melakukan ini untuk fungsi R lainnya. Argumen pertama adalah sumber data, dan argumen kedua adalah fungsi untuk diterapkan ke variabel dalam sumber data tersebut.
Untuk mencari tenaga kuda rata-rata mobil USA di Cars93:

```
> with(Cars93, mean(Horsepower[Origin == "USA"]))
[1] 147.5208
```

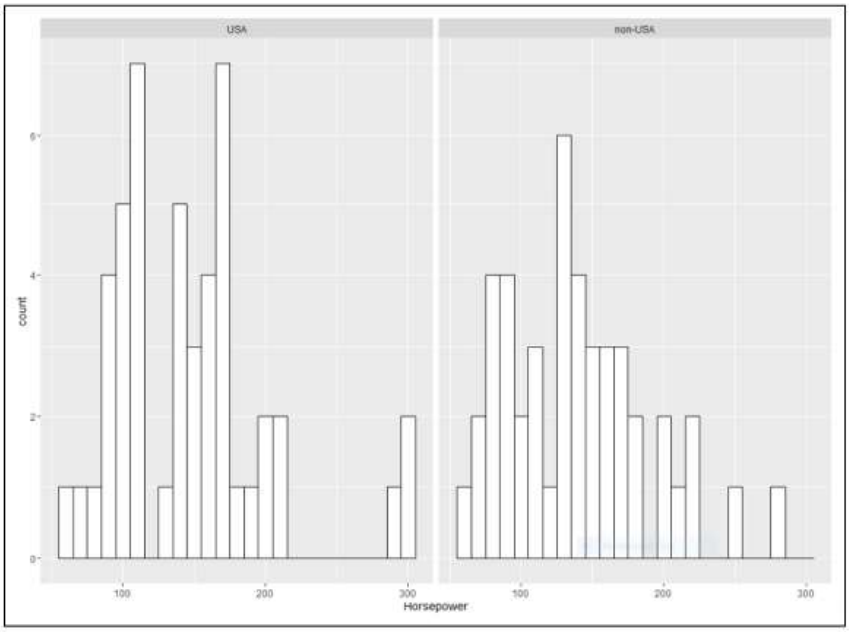
Ini juga melewatkan langkah membuat vektor Horsepower.USA.
Bagaimana dengan berbagai kondisi, seperti tenaga kuda rata-rata mobil 4 silinder AS?

```
> with(Cars93, mean(Horsepower[Origin == "USA" & Cylinders ==4]))
[1] 104.0909
```

R juga menyediakan fungsi attach() sebagai cara untuk menghilangkan tanda $ dan penekanan tombol. Lampirkan bingkai data (attach(Cars93), misalnya) dan Anda tidak perlu merujuknya lagi saat menggunakan variabelnya. Namun, banyak otoritas R merekomendasikan hal ini, karena dapat menyebabkan kesalahan.

Mengeksplorasi data

Sekarang kita telah memeriksa sarana tenaga kuda mobil AS dan non-AS, bagaimana dengan distribusi keseluruhan? Itu membutuhkan sedikit eksplorasi data. Saya menggunakan paket ggplot2 (lihat Bab 3) untuk membuat histogram berdampingan dari kerangka data Cars93 sehingga saya dapat membandingkannya. (Pastikan Anda memiliki ggplot2 di perpustakaan.) Gambar 4.1 menunjukkan apa yang saya maksud.



Gambar 4.1 Histogram tenaga kuda untuk mobil AS dan Non-AS di Cars93.

Untuk membuat histogram pada gambar, saya memulai dengan cara biasa:

```
ggplot(Cars93, aes(x=Horsepower))
```

dan kemudian tambahkan fungsi geom

```
geom_histogram(color="black", fill="white",binwidth = 10)
```

Saya bermain-main sedikit untuk sampai pada nilai binwidth itu.

Kode sejauh ini membuat histogram biasa dengan Horsepower pada sumbu x. Bagaimana cara membuat Gambar 4.1? Untuk melakukan itu, saya menambahkan kemampuan ggplot yang disebut faceting. Sederhananya, faceting membagi data menurut variabel nominal — seperti Origin, yang berupa "USA" atau "non-USA." Beberapa fungsi faceting tersedia. Yang saya gunakan di sini disebut facet_wrap(). Untuk membagi data menurut Asal, itu

```
facet_wrap(~Origin)
```

Sekadar pengingat: Operator tilde (~) berarti "tergantung pada," jadi anggaplah Asal sebagai variabel independen. Kode lengkap untuk Gambar 4.1 adalah:

```
ggplot(Cars93, aes(x=Horsepower)) +
  geom_histogram(color="black", fill="white",binwidth = 10)+
  facet_wrap(~Origin)
```

Seperti yang Anda lihat, distribusi memiliki bentuk keseluruhan yang berbeda. Mobil-mobil AS tampaknya memiliki jarak antara 200-an rendah dan nilai-nilai tertinggi berikutnya, dan mobil-mobil non-AS tidak begitu banyak. Anda juga melihat nilai maksimum yang lebih tinggi untuk

mobil AS. Apa perbedaan lain yang Anda lihat? (Saya membahas perbedaan-perbedaan itu di Bab 7.)

Pencilan: Cacat rata-rata

Pencilan adalah nilai ekstrem dalam kumpulan data. Jika kumpulan data adalah sampel dan Anda mencoba memperkirakan rata-rata populasi, outlier mungkin membiaskan estimasi. Ahli statistik berurusan dengan outlier dengan memangkas mean — menghilangkan nilai ekstrim pada low end dan high end sebelum menghitung mean sampel. Jumlah trim adalah persentase, seperti 5 persen skor atas dan bawah. Misalnya, histogram di sebelah kiri Gambar 4-1 menunjukkan beberapa nilai ekstrim. Untuk memangkas 5 persen atas dan bawah, saya menambahkan argumen trim ke mean():

```
> mean(Horsepower.USA, trim =.05)
[1] 144.1818
```

Hasilnya sedikit lebih rendah dari rata-rata yang belum dipangkas.

Berapa persentase yang tepat untuk trim? Itu terserah Anda. Itu tergantung pada apa yang Anda ukur, seberapa ekstrem skor Anda, dan seberapa baik Anda mengetahui bidang yang Anda pelajari. Saat Anda melaporkan rata-rata yang dipangkas, beri tahu audiens Anda bahwa Anda telah melakukan ini dan beri tahu mereka persentase yang telah Anda pangkas. Di bagian mendatang tentang median, saya menunjukkan cara lain untuk menangani skor ekstrem.

Cara lain untuk mencapai tujuan

Di bagian ini, saya memberi tahu Anda tentang dua rata-rata tambahan yang berbeda dari rata-rata yang biasa Anda gunakan. Rata-rata variasi taman sehari-hari disebut rata-rata aritmatika (diucapkan “arith-MET-ic”). Berapa banyak cara yang berbeda yang mungkin? Matematikawan Yunani kuno menemukan 11.

Rata-rata geometris

Misalkan Anda memiliki investasi 5 tahun yang menghasilkan persentase ini: 10 persen, 15 persen, 10 persen, 20 persen, dan 5 persen. (Ya, ya. Saya tahu. Ini fiksi.) Berapa tingkat pengembalian tahunan rata-rata? Tebakan pertama Anda mungkin rata-rata persentase tersebut. Rata-rata itu adalah 12 persen. Dan itu tidak benar. Mengapa? Itu melewatkan poin penting. Pada akhir tahun pertama, Anda mengalikan investasi Anda dengan 0,10 — Anda tidak menambahkan 1,10 ke dalamnya. Pada akhir tahun kedua, Anda mengalikan hasil tahun pertama dengan 1,15, dan seterusnya.

Rata-rata aritmatika tidak akan memberi Anda tingkat pengembalian rata-rata. Sebagai gantinya, Anda menghitung rata-rata itu dengan cara ini:

Tingkat Pengembalian Rata-rata $\sqrt[5]{1.10 \times 1.15 \times 1.10 \times 1.20 \times 1.05}$ = 1.118847

Tingkat pengembalian rata-rata sedikit kurang dari 12 persen. Rata-rata semacam ini disebut rata-rata geometrik. Dalam contoh ini, mean geometrik adalah akar kelima dari produk lima angka. Apakah selalu akar ke-n dari produk dari n angka? Ya. Basis R tidak menyediakan fungsi untuk menghitung mean geometrik, tetapi cukup mudah untuk menghitungnya.

Saya mulai dengan membuat vektor angka:

```
invest <- c(1.10,1.15,1.10,1.20,1.05)
```

Saya menggunakan fungsi prod() untuk menghitung produk dari angka-angka dalam vektor, dan fungsi length() untuk menghitung berapa banyak angka dalam vektor. Perhitungannya kemudian

```
> gm.invest <- prod(invest)^(1/(length(invest)))
> gm.invest
[1] 1.118847
```

Arti harmonik

Inilah situasi yang terkadang Anda temui dalam kehidupan nyata, tetapi lebih sering terjadi di buku teks aljabar. Misalkan Anda tidak terburu-buru untuk berangkat kerja di pagi hari dan Anda berkendara dari rumah ke tempat kerja Anda dengan kecepatan 30 mil per jam. Di penghujung hari, di sisi lain, Anda ingin cepat pulang. Jadi, dalam perjalanan pulang (dengan jarak yang persis sama), Anda berkendara dari tempat kerja ke rumah dengan kecepatan 50 mil per jam. Berapa tarif rata-rata untuk total waktu Anda di jalan?

Ini bukan 40 mil per jam, karena Anda berada di jalan dengan jumlah waktu yang berbeda untuk setiap bagian perjalanan. Tanpa membahas ini terlalu dalam, rumus untuk mengetahuinya adalah:

$$\frac{1}{\text{Average}} = \frac{1}{2}\left[\frac{1}{30} + \frac{1}{50}\right] = \frac{1}{37.5}$$

Rata-rata adalah 37,5. Jenis rata-rata ini disebut rata-rata harmonik. Contoh ini terdiri dari dua angka, tetapi Anda dapat menghitungnya untuk jumlah angka berapa pun. Masukkan saja setiap angka ke dalam penyebut pecahan dengan 1 sebagai pembilangnya. Matematikawan menyebutnya kebalikan dari suatu bilangan. (Jadi $1/30$ adalah kebalikan dari 30.) Tambahkan semua kebalikannya dan ambil rata-ratanya. Hasilnya adalah kebalikan dari rata-rata harmonik.

Basis R tidak memiliki fungsi untuk rata-rata harmonik, tetapi (sekali lagi) mudah untuk menghitungnya. Anda mulai dengan membuat vektor dari dua kecepatan:

```
speeds <- c(30,50)
```

Mengambil kebalikan dari vektor menghasilkan vektor timbal balik:

```
> 1/speeds
[1] 0.03333333 0.02000000
```

Jadi rata-rata harmoniknya adalah

```
> hm.speeds <- 1/mean(1/speeds)
> hm.speeds
[1] 37.5
```

4.3 MEDIAN: NILAI TENGAH

Mean adalah cara yang berguna untuk meringkas sekelompok angka. Satu kelemahan ("cacat rata-rata") adalah sensitif terhadap nilai ekstrim. Jika satu nomor rusak, berarti juga rusak. Ketika itu terjadi, mean mungkin tidak mewakili kelompok dengan baik.

Di sini, misalnya, adalah kecepatan membaca (dalam kata per menit) untuk sekelompok anak:
56, 78, 45, 49, 55, 62
Meannya adalah:

```
> reading.speeds <- c(56, 78, 45, 49, 55, 62)
> mean(reading.speeds)
[1] 57.5
```

Misalkan anak yang membaca dengan kecepatan 78 kata per menit meninggalkan kelompok dan seorang pembaca yang sangat cepat menggantikannya. Kecepatan membacanya adalah fenomenal 180 kata per menit:

```
> reading.speeds.new <-
          replace(reading.speeds,reading.speeds == 78,180)
> reading.speeds.new
[1]  56 180  45  49  55  62
```

Sekarang artinya adalah:

```
> mean(reading.speeds.new)
[1] 74.5
```

Rata-rata baru menyesatkan. Kecuali anak baru, tidak ada orang lain dalam kelompok yang bisa membaca secepat itu. Dalam kasus seperti ini, ada baiknya menggunakan ukuran tendensi sentral yang berbeda — median.

Median adalah nama yang bagus untuk konsep sederhana: Ini adalah nilai tengah dalam sekelompok angka. Susun angka-angka tersebut secara berurutan, dan median adalah nilai di bawah mana setengah skor jatuh dan di atasnya setengah skor jatuh:

```
> sort(reading.speeds)
[1] 45 49 55 56 62 78
> sort(reading.speeds.new)
[1]  45  49  55  56  62 180
```

Dalam setiap kasus, median berada di tengah antara 55 dan 56, atau 55,5.

4.4 MEDIAN DI R: MEDIAN()

Jadi bukan misteri besar bagaimana menggunakan R untuk menemukan median:

```
> median(reading.speeds)
[1] 55.5
> median(reading.speeds.new)
[1] 55.5
```

Dengan kumpulan data yang lebih besar, Anda mungkin mengalami replikasi skor. Bagaimanapun, median masih merupakan nilai tengah. Sebagai contoh, berikut adalah tenaga kuda untuk mobil 4 silinder di Cars93:

```
> with(Cars93, Horsepower.Four <- Horsepower[Cylinders == 4])
> sort(Horsepower.Four)
 [1]  63  74  81  81  82  82  85  90  90  92  92  92  92  92
[15]  93  96 100 100 100 102 103 105 110 110 110 110 110 110
[29] 110 114 115 124 127 128 130 130 130 134 135 138 140 140
[43] 140 141 150 155 160 164 208
```

Anda melihat sedikit duplikasi dalam angka-angka ini — terutama di sekitar bagian tengah. Hitung melalui nilai yang diurutkan dan Anda akan melihat bahwa 24 skor sama dengan atau kurang dari 110, dan 24 skor lebih besar dari atau sama dengan 110, yang menjadikan median

```
> median(Horsepower.Four)
[1] 110
```

4.5 STATISTIK IA MODE

Satu lagi ukuran tendensi sentral, modus, adalah penting. Ini adalah skor yang paling sering muncul dalam kelompok skor. Terkadang mode adalah ukuran terbaik dari tendensi sentral untuk digunakan. Bayangkan sebuah perusahaan kecil yang terdiri dari 30 konsultan dan dua pejabat tinggi. Setiap konsultan memiliki gaji tahunan sebesar Rp 600.000.000. Setiap petugas memiliki gaji tahunan sebesar Rp 3.750.000.000. Gaji rata-rata di perusahaan ini adalah Rp 796.875.000.

Apakah mean memberi Anda gambaran yang jelas tentang struktur gaji perusahaan? Jika Anda sedang mencari pekerjaan di perusahaan itu, apakah rata-rata akan mempengaruhi harapan Anda? Anda mungkin lebih baik jika Anda mempertimbangkan mode, yang dalam hal ini adalah Rp 600.000.000 (kecuali jika Anda adalah bakat eksekutif dengan harga tinggi!). Tidak ada yang rumit tentang menemukan mode. Lihatlah skor dan temukan salah satu yang paling sering muncul, dan Anda telah menemukan modenya. Apakah dua skor sama untuk kehormatan itu? Dalam hal ini, kumpulan skor Anda memiliki dua mode. (Nama teknisnya adalah bimodal.)

Bisakah Anda memiliki lebih dari dua mode? Sangat. Jika setiap skor terjadi sama seringnya, Anda tidak memiliki mode.

4.6 MODE DI R

Basis R tidak menyediakan fungsi untuk mencari modus. Itu memang memiliki fungsi yang disebut mode(), tetapi itu untuk sesuatu yang jauh berbeda. Sebagai gantinya, Anda memerlukan paket bernama modeest di perpustakaan Anda. (Pada tab Paket, pilih Instal, lalu di kotak dialog Instal, ketik modeest di kotak Paket dan klik Instal. Kemudian centang kotaknya saat muncul di tab Paket).

Satu fungsi dalam paket paling sederhana disebut mfv() ("nilai paling sering"), dan itulah yang Anda butuhkan. Berikut adalah vektor dengan dua mode (2 dan 4):

```
> scores <- c(1,2,2,2,3,4,4,4,5,6)
> mfv(scores)
[1] 2 4
```

BAB 5
MENYIMPANG DARI RATA-RATA

Inilah lelucon ahli statistik terkenal: Tiga ahli statistik pergi berburu rusa dengan busur dan anak panah. Mereka melihat rusa dan membidik. Satu tunas dan panahnya terbang sepuluh kaki ke kiri. Tembakan kedua dan panahnya melesat sepuluh kaki ke kanan. Ahli statistik ketiga dengan gembira berteriak, "Kami mendapatkannya!"

Moral dari cerita: Menghitung mean adalah cara yang bagus untuk meringkas serangkaian angka, tetapi mean mungkin menyesatkan Anda. Bagaimana? Dengan tidak memberi Anda semua informasi yang biasanya Anda butuhkan. Jika Anda hanya mengandalkan mean, Anda mungkin melewatkan sesuatu yang penting tentang kumpulan angka.

Untuk menghindari hilangnya informasi penting, diperlukan jenis statistik lain — statistik yang mengukur variasi. Pikirkan variasi sebagai semacam rata-rata seberapa banyak setiap angka dalam sekelompok angka berbeda dari rata-rata kelompok. Beberapa statistik tersedia untuk mengukur variasi. Semuanya bekerja dengan cara yang sama: Semakin besar nilai statistik, semakin banyak angka yang berbeda dari rata-ratanya. Semakin kecil nilainya, semakin sedikit perbedaannya.

5.1 MENGUKUR VARIASI

Misalkan Anda mengukur tinggi sekelompok anak-anak dan Anda menemukan bahwa tinggi badan mereka (dalam inci) adalah:

48, 48, 48, 48, dan 48.

Kemudian Anda mengukur kelompok lain dan menemukan bahwa tinggi mereka adalah:

50, 47, 52, 46, dan 45.

Jika Anda menghitung rata-rata setiap grup, Anda akan menemukan bahwa mereka sama — 48 inci. Hanya dengan melihat angka-angkanya memberi tahu Anda bahwa kedua kelompok ketinggian itu berbeda: Ketinggian di kelompok pertama semuanya sama, sedangkan ketinggian di kelompok kedua sedikit berbeda.

Rata-rata deviasi kuadrat: Varians dan cara menghitungnya

Salah satu cara untuk menunjukkan ketidakmiripan antara kedua kelompok adalah dengan memeriksa penyimpangan pada masing-masing kelompok. Pikirkan "penyimpangan" sebagai perbedaan antara skor dan rata-rata semua skor dalam suatu kelompok. Inilah yang saya bicarakan. Tabel 5.1 menunjukkan kelompok ketinggian pertama dan penyimpangannya.

Tabel 5.1 Kelompok Ketinggian Pertama dan Penyimpangannya

Tinggi	Tinggi-Rata-rata	Deviasi
48	48-48	0
48	48-48	0

48	48-48	0
48	48-48	0
48	48-48	0

Salah satu cara untuk melanjutkan adalah dengan merata-ratakan deviasinya. Jelas, rata-rata angka di kolom Deviasi adalah nol. Tabel 5.2 menunjukkan kelompok ketinggian kedua dan penyimpangannya.

Tabel 5.2 Kelompok Ketinggian Kedua dan Penyimpangannya

Tinggi	Tinggi-Rata-rata	Deviasi
50	50-48	2
47	47-48	−1
52	52-48	4
46	46-48	−2
45	45-48	−3

Bagaimana dengan rata-rata penyimpangan pada Tabel 5-2? itu. . . nol! Jadi sekarang apa? Rata-rata deviasi tidak membantu Anda melihat perbedaan antara kedua kelompok, karena rata-rata deviasi dari mean dalam setiap kelompok angka selalu nol. Faktanya, ahli statistik veteran akan memberi tahu Anda bahwa itu adalah properti yang menentukan dari mean. Joker di dek di sini adalah angka negatif. Bagaimana ahli statistik menghadapinya? Triknya adalah dengan menggunakan sesuatu yang mungkin Anda ingat dari aljabar: A minus dikalikan minus adalah plus. Terdengar akrab?

Jadi . . . apakah ini berarti Anda mengalikan setiap kali penyimpangan itu sendiri dan kemudian merata-ratakan hasilnya? Sangat. Mengalikan deviasi dengan waktu itu sendiri disebut mengkuadratkan deviasi. Rata-rata deviasi kuadrat sangat penting sehingga memiliki nama khusus: varians. Tabel 5.3 menunjukkan kelompok ketinggian dari Tabel 5.2, bersama dengan deviasi dan deviasi kuadratnya.

Tabel 5.3 Kelompok Ketinggian Kedua dan Penyimpangan Kuadratnya

Tinggi	Tinggi-Rata-rata	Deviasi	Penyimpangan Kuadrat
50	50-48	2	4
47	47-48	−1	1
52	52-48	4	16
46	46-48	−2	4
45	45-48	−3	9

Varians — rata-rata deviasi kuadrat untuk grup ini — adalah (4 + 1 + 16 + 4 + 9) / 5 = 34/5 = 6.8. Ini, tentu saja, sangat berbeda dari kelompok pertama, yang variansnya nol.

Untuk mengembangkan rumus varians untuk Anda dan menunjukkan cara kerjanya, saya menggunakan simbol untuk menunjukkan semua ini. X mewakili judul Tinggi di kolom pertama tabel, dan $\bar{X}$ mewakili rata-rata.

Penyimpangan adalah hasil pengurangan rata-rata dari setiap angka, jadi

$$\left(X-\bar{X}\right)$$

melambangkan penyimpangan. Bagaimana dengan mengalikan deviasi dengan dirinya sendiri? Itu

$$\left(X-\bar{X}\right)^2$$

Untuk menghitung varians, Anda mengkuadratkan setiap deviasi, menjumlahkannya, dan mencari rata-rata deviasi kuadrat. Jika N mewakili jumlah deviasi kuadrat yang Anda miliki (dalam contoh ini, lima), rumus untuk menghitung varians adalah:

$$\frac{\sum\left(X-\bar{X}\right)^2}{N}$$

$\sum$ adalah huruf besar Yunani sigma, dan itu berarti "jumlah dari."

Apa simbol untuk varians? Seperti yang saya sebutkan di Bab 1, huruf Yunani mewakili parameter populasi, dan huruf Inggris mewakili statistik sampel. Bayangkan bahwa kelompok kecil kita yang terdiri dari lima angka adalah seluruh populasi. Apakah alfabet Yunani memiliki huruf yang sesuai dengan V dengan cara yang sama seperti (simbol untuk populasi berarti) sesuai dengan M? Tidak. Sebaliknya, Anda menggunakan sigma huruf kecil! Dan di atas itu, karena Anda berbicara tentang jumlah kuadrat, simbol varians populasi adalah 2.

Intinya: Rumus untuk menghitung varians populasi adalah:

$$\sigma^2=\frac{\sum\left(X-\bar{X}\right)^2}{N}$$

Nilai besar untuk varians memberi tahu Anda bahwa angka-angka dalam suatu kelompok sangat bervariasi dari rata-ratanya. Nilai kecil untuk varians memberi tahu Anda bahwa angkanya sangat mirip dengan rata-ratanya.

Varians sampel

Rumus varians yang baru saja saya tunjukkan cocok jika kelompok lima pengukuran adalah populasi. Apakah ini berarti bahwa varians untuk sampel berbeda? Ya, dan inilah alasannya. Jika kumpulan angka Anda adalah sampel yang diambil dari populasi besar, tujuan Anda kemungkinan besar menggunakan varians sampel untuk memperkirakan varians populasi. Rumus di bagian sebelumnya tidak berfungsi sebagai perkiraan varians populasi. Meskipun mean yang dihitung dengan cara biasa merupakan estimasi akurat dari mean populasi, tidak demikian halnya dengan varians, karena alasan yang jauh di luar cakupan buku ini.

Cukup mudah untuk menghitung perkiraan yang akurat dari varians populasi. Yang harus Anda lakukan adalah menggunakan N-1 dalam penyebut daripada N. (Sekali lagi, untuk alasan yang jauh di luar cakupan buku ini.) Dan karena Anda bekerja dengan karakteristik sampel (bukan

populasi), Anda menggunakan padanan bahasa Inggris dari huruf Yunani — s daripada . Ini berarti bahwa rumus varians sampel (sebagai pendugaan varians populasi) adalah:

$$s^2 = \frac{\sum\left(X - \bar{X}\right)^2}{N-1}$$

Nilai s2, jika diketahui simpangan kuadrat dari himpunan lima bilangan adalah (4 + 1 + 16 + 4 + 9) / 4 = 34/4 = 8.5

Jadi jika angka-angka ini 50, 47, 52, 46, dan 45 adalah seluruh populasi, variansnya adalah 6,8. Jika mereka adalah sampel yang diambil dari populasi yang lebih besar, perkiraan terbaik dari varians populasi tersebut adalah 8,5.

Varians dalam R

Menghitung varians dalam R adalah kesederhanaan itu sendiri. Anda menggunakan fungsi var() . Tetapi varian mana yang diberikannya kepada Anda? Yang penyebutnya N atau yang N-1? Mari kita cari tahu:

```
> heights <- c(50, 47, 52, 46, 45)
```

```
> var(heights)
[1] 8.5
```

Ini menghitung varians yang diperkirakan (dengan N-1 dalam penyebut). Untuk menghitung varians pertama yang saya tunjukkan (dengan N sebagai penyebutnya), saya harus mengalikan angka ini dengan (N–1)/N. Menggunakan panjang() untuk menghitung N, yaitu

```
> var(heights)*(length(heights)-1)/length(heights)
[1] 6.8
```

Jika saya akan sering bekerja dengan varians semacam ini, saya akan mendefinisikan fungsi var.p():

```
var.p = function(x){var(x)*(length(x)-1)/length(x)}
```

Dan berikut cara menggunakannya:

```
> var.p(heights)
[1] 6.8
```

Untuk alasan yang akan menjadi jelas nanti, saya ingin Anda memikirkan penyebut dari estimasi varians (seperti N-1) sebagai derajat kebebasan. Mengapa? Tetap disini. (Bab 12 mengungkapkan semuanya!)

5.2 DEVIASI STANDAR

Setelah Anda menghitung varians dari serangkaian angka, Anda memiliki nilai yang unitnya berbeda dari pengukuran awal Anda. Misalnya, jika pengukuran awal Anda dalam inci, variansnya dalam inci persegi. Ini karena Anda kuadratkan deviasinya sebelum Anda rata-ratakan. Jadi varians dalam populasi lima-skor pada contoh sebelumnya adalah 6,8 inci persegi.

Mungkin sulit untuk memahami apa artinya itu. Seringkali, lebih intuitif jika statistik variasi berada dalam satuan yang sama dengan pengukuran asli. Sangat mudah untuk mengubah varians menjadi statistik semacam itu. Yang harus Anda lakukan adalah mengambil akar kuadrat dari varians. Seperti halnya varians, akar kuadrat ini sangat penting sehingga memiliki nama khusus: simpangan baku.

Simpangan baku populasi

Simpangan baku suatu populasi adalah akar kuadrat dari varians populasi. Simbol simpangan baku populasi adalah (sigma). rumusnya adalah:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum\left(X - \bar{X}\right)^2}{N}}$$

Untuk populasi pengukuran 5-skor ini (dalam inci):

50, 47, 52, 46, dan 45

Varians populasi adalah 6,8 inci persegi, dan standar deviasi populasi adalah 2,61 inci (dibulatkan).

Standar deviasi sampel

Simpangan baku sampel — perkiraan simpangan baku suatu po

$$s = \sqrt{s^2} = \sqrt{\frac{\sum\left(X - \bar{X}\right)^2}{N-1}}$$

Untuk contoh pengukuran ini (dalam inci):

50, 47, 52, 46, dan 45

Varians populasi yang diperkirakan adalah 8,4 inci persegi, dan standar deviasi populasi yang diperkirakan adalah 2,92 inci (dibulatkan).

5.3 DEVIASI STANDAR DALAM R

Seperti halnya varians, menggunakan R untuk menghitung simpangan baku itu mudah: Anda menggunakan fungsi sd(). Dan seperti mitra variansnya, sd() menghitung s, bukan σ :

```
> sd(heights)
[1] 2.915476
```

Untuk σ — memperlakukan lima angka sebagai populasi mandiri, dengan kata lain — Anda harus mengalikan hasil sd() dengan akar kuadrat dari (N-1)/N:

```
> sd(heights)*(sqrt((length(heights)-1)/length(heights)))
[1] 2.607681
```

Sekali lagi, jika Anda akan sering menggunakan yang ini, mendefinisikan fungsi adalah ide yang bagus:

```
sd.p=function(x){sd(x)*sqrt((length(x)-1)/length(x))}
```

Dan inilah cara Anda menggunakan fungsi ini:

```
> sd.p(heights)
[1] 2.607681
```

5.4 KESIMPULAN

Dalam Bab 4, saya menunjukkan bahwa dengan bingkai data yang lebih besar, Anda terkadang ingin menghitung statistik pada kasus (baris) yang memenuhi kondisi tertentu, bukan pada semua kasus.

Seperti pada Bab 3 dan 4, saya menggunakan kerangka data Cars93 untuk pembahasan berikut. Kerangka data tersebut memiliki data untuk sampel 93 mobil dari tahun 1993. Anda akan menemukannya di paket MASS, jadi pastikan Anda memiliki paket MASS di perpustakaan Anda. (Temukan MASS pada tab Packages dan klik kotak centangnya.) Saya menghitung varians tenaga kuda mobil yang berasal dari Amerika Serikat. Menggunakan fungsi with() yang saya tunjukkan di Bab 4, yaitu

```
> with(Cars93, var(Horsepower[Origin == "USA"]))
[1] 2965.319
```

Berapa banyak dari mobil-mobil itu dalam kelompok ini?

```
> with(Cars93, length(Horsepower[Origin == "USA"]))
[1] 48
```

Bagaimana dengan mobil non-AS?

```
> with(Cars93, var(Horsepower[Origin == "non-USA"]))
[1] 2537.283
> with(Cars93, length(Horsepower[Origin == "non-USA"]))
[1] 45
```

Bisakah Anda membandingkan varians itu? Tentu — tetapi tidak sampai Bab 11. Saya akan membiarkannya sebagai latihan bagi Anda untuk menghitung deviasi standar untuk mobil AS dan untuk mobil non-AS.

BAB 6
MEMENUHI STANDAR DAN PERINGKAT

Di tangan kiri saya, saya memegang 100 peso Filipina. Di sebelah kanan saya, saya memegang 1.000 peso Kolombia. Mana yang lebih berharga? Keduanya disebut peso, bukan? Jadi bukankah seharusnya 1.000 lebih besar dari 100? Belum tentu. Peso hanyalah kebetulan nama. Masing-masing berasal dari negara yang berbeda, dan setiap negara memiliki ekonominya sendiri.

Untuk membandingkan dua jumlah uang, Anda harus mengonversi setiap mata uang ke dalam satuan standar. Standar paling intuitif untuk warga AS adalah mata uang kita sendiri. Berapa nilai setiap jumlah dalam dolar dan sen? Saat saya menulis ini, 100 peso Filipina bernilai lebih dari $2. Seribu peso Kolombia bernilai 34 sen.

Jadi ketika Anda membandingkan angka, konteks itu penting. Untuk membuat perbandingan yang valid di seluruh konteks, Anda sering kali harus mengonversi angka menjadi satuan standar. Dalam bab ini, saya menunjukkan kepada Anda bagaimana menggunakan statistik untuk melakukan hal itu. Unit standar menunjukkan kepada Anda di mana skor berdiri dalam kaitannya dengan skor lain dalam suatu grup. Saya juga menunjukkan cara lain untuk menentukan kedudukan skor dalam suatu kelompok.

6.1 MENANGKAP BEBERAPA Z

Angka yang terisolasi tidak memberikan banyak informasi. Untuk memahami sepenuhnya apa arti angka, Anda harus memperhitungkan proses yang menghasilkannya. Untuk membandingkan satu angka dengan angka lainnya, mereka harus berada pada skala yang sama. Saat Anda mengonversi mata uang, mudah untuk mengetahui standarnya. Saat Anda mengubah suhu dari Fahrenheit ke Celsius, atau panjang dari kaki ke meter, sebuah rumus akan memandu Anda.

Jika tidak begitu jelas, Anda dapat menggunakan mean dan standar deviasi untuk menstandarisasi skor yang berasal dari proses yang berbeda. Idenya adalah untuk mengambil satu set skor dan menggunakan meannya sebagai titik nol, dan standar deviasinya sebagai unit ukuran. Kemudian Anda membuat perbandingan: Anda menghitung deviasi setiap skor dari rata-rata, dan kemudian Anda membandingkan deviasi itu dengan standar deviasi. Anda bertanya, "Seberapa besar penyimpangan tertentu relatif terhadap (sesuatu seperti) rata-rata semua penyimpangan?" Untuk membuat perbandingan, Anda membagi simpangan skor dengan simpangan baku. Ini mengubah skor menjadi jenis skor lain. Skor yang diubah disebut skor standar, atau skor-z.

Rumusnya adalah:

$$z = \frac{X - \bar{X}}{s}$$

jika Anda berurusan dengan sampel, dan

$$z = \frac{X - \mu}{\sigma}$$

jika Anda berurusan dengan populasi. Dalam kedua kasus, x mewakili skor yang Anda ubah menjadi skor-z.

Karakteristik skor-z

Sebuah z-score bisa positif, negatif, atau nol. Skor-z negatif mewakili skor yang lebih kecil dari rata-rata, dan skor-z positif mewakili skor yang lebih besar dari rata-rata. Ketika skor sama dengan rata-rata, skor-z-nya adalah nol. Saat Anda menghitung skor-z untuk setiap skor dalam himpunan, rata-rata skor-z adalah 0, dan simpangan baku skor-z adalah 1.

Setelah Anda melakukan ini untuk beberapa set skor, Anda dapat secara sah membandingkan skor dari satu set ke skor dari yang lain. Jika kedua himpunan memiliki mean yang berbeda dan standar deviasi yang berbeda, membandingkan tanpa membakukan seperti membandingkan apel dengan kumquat. Dalam contoh berikut, saya menunjukkan cara menggunakan skor-z untuk membuat perbandingan.

Obligasi versus Bambino

Inilah pertanyaan penting yang sering muncul dalam konteks diskusi metafisik yang serius: Siapa pemukul home run terhebat sepanjang masa: Barry Bonds atau Babe Ruth? Meskipun ini adalah pertanyaan yang sulit untuk dijawab, salah satu cara untuk mengatasinya adalah dengan melihat musim terbaik setiap pemain dan membandingkan keduanya. Obligasi mencapai 73 home run pada tahun 2001, dan Ruth mencapai 60 pada tahun 1927. Di permukaan, Obligasi tampaknya menjadi pemukul yang lebih produktif. Namun, tahun 1927 sangat berbeda dengan tahun 2001. Baseball (dan yang lainnya) mengalami perubahan besar yang telah lama tertunda di tahun-tahun berikutnya, dan statistik pemain mencerminkan perubahan tersebut. Sebuah home run lebih sulit untuk memukul pada tahun 1920-an daripada di tahun 2000-an. Tetap saja, 73 lawan 60? Hmmm.

Skor standar dapat membantu memutuskan musim terbaik siapa yang lebih baik. Untuk membakukan, saya mengambil 50 pemukul home run teratas tahun 1927 dan 50 teratas dari tahun 2001. Saya menghitung rata-rata dan simpangan baku setiap kelompok dan kemudian mengubah 60 Ruth dan 73 Obligasi menjadi skor-z. Rata-rata dari tahun 1927 adalah 12,68 homer dengan standar deviasi 10,49. Rata-rata dari tahun 2001 adalah 37,02 homer dengan standar deviasi 9,64. Meskipun artinya sangat berbeda, standar deviasinya cukup dekat.

Dan skor-z? Ruth adalah:

$$z = \frac{60 - 12.68}{10.49} = 4.51$$

Obligasi adalah:

$$z = \frac{73 - 37.02}{9.64} = 3.73$$

Pemenang yang jelas dalam derby home run musim terbaik z-score adalah Babe Ruth. Periode.

Hanya untuk menunjukkan kepada Anda bagaimana waktu telah berubah, Lou Gehrig mencapai 47 home run pada tahun 1927 (finis kedua setelah Ruth) dengan skor z 3,27. Pada tahun 2001, 47 home run mencapai z-score 1,04.

Nilai ujian

Menjauh dari debat olahraga, salah satu aplikasi praktis skor-z adalah pemberian nilai pada skor ujian. Berdasarkan skor persentase, instruktur secara tradisional mengevaluasi skor 90 poin atau lebih tinggi (dari 100) sebagai A, 80–89 poin sebagai B, 70–79 poin sebagai C, 60–69 poin sebagai D, dan kurang dari 60 poin sebagai F. Kemudian mereka rata-rata mendapat nilai dari beberapa ujian bersama untuk menetapkan nilai kursus.

Apakah itu adil? Sama seperti satu peso dari Filipina bernilai lebih dari satu peso dari Kolombia, dan home run lebih sulit dicapai pada tahun 1927 daripada pada tahun 2001, apakah “poin” pada satu ujian bernilai sama dengan “poin” pada ujian lainnya? Seperti "peso", bukankah "poin" hanya kebetulan? Sangat. Sebuah poin pada ujian yang sulit, menurut definisi, lebih sulit didapat daripada poin pada ujian yang mudah. Karena poin mungkin tidak memiliki arti yang sama dari satu ujian ke ujian lainnya, hal yang paling adil untuk dilakukan adalah mengubah nilai dari setiap ujian menjadi nilai-z sebelum dirata-ratakan. Dengan begitu, Anda merata-ratakan angka di lapangan permainan yang sama.

Saya melakukan itu dalam kursus yang saya ajarkan. Saya sering menemukan bahwa skor numerik yang lebih rendah pada satu ujian menghasilkan skor-z yang lebih tinggi daripada skor numerik yang lebih tinggi dari ujian lain. Misalnya, pada ujian di mana rata-rata adalah 65 dan standar deviasi adalah 12, skor 71 menghasilkan skor-z 0,5. Pada ujian lain, dengan rata-rata 69 dan standar deviasi 14, skor 75 setara dengan skor-z 0,429. (Ya, ini seperti 60 home run Ruth versus 73 dari Bonds.) Moral dari cerita: Angka-angka dalam isolasi memberi tahu Anda sangat sedikit. Anda harus memahami proses yang menghasilkan mereka.

6.2 SKOR STANDAR DI R

Fungsi R untuk menghitung skor standar disebut scale(). Berikan vektor skor, dan scale() mengembalikan vektor skor-z bersama dengan, membantu, rata-rata dan deviasi standar. Untuk menunjukkan scale() beraksi, saya mengisolasi subset dari kerangka data Cars93. (Ada dalam paket MASS. Pada tab Packages, centang kotak di sebelah MASS jika tidak dicentang).

Secara khusus, saya membuat vektor tenaga kuda mobil 8 silinder dari AS:

```
> Horsepower.USA.Eight <- Cars93$Horsepower[Origin ==
        "USA" & Cylinders == 8]
> Horsepower.USA.Eight
[1] 200 295 170 300 190 210
```

Dan sekarang untuk skor-z:

```
> scale(Horsepower.USA.Eight)
           [,1]
[1,] -0.4925263
[2,]  1.2089283
[3,] -1.0298278
[4,]  1.2984785
[5,] -0.6716268
[6,] -0.3134259
attr(,"scaled:center")
[1] 227.5
attr(,"scaled:scale")
[1] 55.83458
```

Nilai terakhir itu adalah s, bukan . Jika Anda harus mendasarkan skor-z Anda pada , bagi setiap elemen dalam vektor dengan akar kuadrat dari (N-1)/N:

```
> N <- length(Horsepower.USA.Eight)
> scale(Horsepower.USA.Eight)/sqrt((N-1)/N)
           [,1]
[1,] -0.5395356
[2,]  1.3243146
[3,] -1.1281198
[4,]  1.4224120
[5,] -0.7357303
[6,] -0.3433408
attr(,"scaled:center")
[1] 227.5
attr(,"scaled:scale")
[1] 55.83458
```

Perhatikan bahwa scale() masih mengembalikan s.

Caching Beberapa Z's

Karena skor-z negatif mungkin memiliki konotasi yang, yah, negatif, pendidik terkadang mengubah skor-z ketika mereka mengevaluasi siswa. Akibatnya, mereka menyembunyikan skor-z, tetapi konsepnya sama — standarisasi dengan standar deviasi sebagai satuan ukuran.

Salah satu transformasi populer disebut T-score. T-score menghilangkan skor negatif karena satu set T-score memiliki rata-rata 50 dan standar deviasi 10. Idenya adalah untuk memberikan ujian, menilai semua tes, dan menghitung mean dan standar deviasi. Selanjutnya, ubah setiap skor menjadi skor-z. Kemudian ikuti rumus ini:

$$T = (z)(10) + 50$$

Orang yang menggunakan T-score biasanya suka membulatkan ke bilangan bulat terdekat. Berikut ini cara mengubah vektor dari contoh menjadi kumpulan nilai-T:

```
T.Hp.USA.Eight <- round((10*scale(Horsepower.USA.Eight)+50),
   digits = 0)
```

Argumen digits=0 dalam fungsi round() membulatkan hasilnya ke bilangan bulat terdekat.

Skor SAT adalah transformasi lain dari skor-z. (Beberapa menyebut SAT sebagai skor-C.) Di bawah sistem penilaian lama, SAT memiliki rata-rata 500 dan standar deviasi 100. Setelah ujian

dinilai, dan mean dan standar deviasinya dihitung, setiap skor ujian menjadi skor-z dengan cara biasa. Rumus ini mengubah skor-z menjadi skor SAT:

$$SAT = (z)(100) + 50$$

Pembulatan ke bilangan bulat terdekat juga merupakan bagian dari prosedur di sini.

Skor IQ masih merupakan transformasi z lainnya. Rata-ratanya adalah 100, dan standar deviasinya adalah 15. Bagaimana prosedur untuk menghitung skor IQ? Anda menebaknya. Dalam kelompok skor IQ, hitung rata-rata dan simpangan baku, lalu hitung skor-z. Maka itu

$$IQ = (z)(15) + 100$$

Seperti dua lainnya, skor IQ dibulatkan ke bilangan bulat terdekat.

6.3 DIMANA PERINGKAT ANDA

Skor standar menunjukkan kepada Anda bagaimana skor berdiri dalam kaitannya dengan skor lain dalam kelompok yang sama. Untuk melakukan ini, mereka menggunakan standar deviasi sebagai satuan ukuran.

Jika Anda tidak ingin menggunakan simpangan baku, Anda dapat menunjukkan kedudukan relatif skor dengan cara yang lebih sederhana. Anda dapat menentukan peringkat skor dalam grup: Dalam urutan menaik, skor terendah memiliki peringkat 1, terendah kedua memiliki peringkat 2, dan seterusnya. Dalam urutan, skor tertinggi adalah peringkat 1, tertinggi kedua 2, dan seterusnya.

Peringkat di R

Tidak mengherankan, fungsi rank() memberi peringkat skor dalam vektor. Urutan default naik:

```
> Horsepower.USA.Eight
[1] 200 295 170 300 190 210
> rank(Horsepower.USA.Eight)
[1] 3 5 1 6 2 4
```

Untuk urutan menurun, beri tanda minus (–) di depan nama vektor:

```
> rank(-Horsepower.USA.Eight)
[1] 4 2 6 1 5 3
```

Skor seri

R menangani skor terikat dengan menyertakan argumen ties.method opsional di rank(). Untuk menunjukkan cara kerjanya, saya membuat vektor baru yang menggantikan nilai keenam (210) di Horsepower.USA.Eight dengan 200:

```
> tied.Horsepower <- replace(Horsepower.USA.Eight,6,200)
> tied.Horsepower
[1] 200 295 170 300 190 200
```

Salah satu cara untuk menangani skor imbang adalah dengan memberi setiap skor imbang rata-rata peringkat yang akan mereka capai. Jadi dua skor dari 200 akan diberi peringkat 3 dan 4, dan rata-rata 3.5 adalah apa yang diberikan metode ini untuk keduanya:

```
> rank(tied.Horsepower, ties.method = "average")
[1] 3.5 5.0 1.0 6.0 2.0 3.5
```

Metode lain menetapkan peringkat minimum:

```
> rank(tied.Horsepower, ties.method = "min")
[1] 3 5 1 6 2 3
```

Dan yang lain lagi menetapkan peringkat maksimum:

```
> rank(tied.Horsepower, ties.method = "max")
[1] 4 5 1 6 2 4
```

Beberapa metode lain tersedia. Ketik ?rank ke jendela konsol untuk detailnya (yang muncul di tab Bantuan).

N terkecil, N terbesar

Anda dapat membalikkan proses peringkat dengan memberikan peringkat (seperti peringkat kedua terendah) dan menanyakan skor mana yang memiliki peringkat tersebut. Prosedur ini dimulai dengan fungsi sort(), yang mengatur skor dalam urutan yang meningkat:

```
> sort(Horsepower.USA.Eight)
[1] 170 190 200 210 295 300
```

Untuk skor terendah kedua, berikan nilai indeks 2:

```
> sort(Horsepower.USA.Eight)[2]
[1] 190
```

Bagaimana dari ujung yang lain? Mulailah dengan menetapkan panjang vektor ke N:

```
> N <- length(Horsepower.USA.Eight)
```

Kemudian, untuk menemukan skor tertinggi kedua, itu

```
> sort(Horsepower.USA.Eight)[N-1]
[1] 295
```

Persentil

Berhubungan erat dengan peringkat adalah persentil, yang mewakili kedudukan skor dalam grup sebagai persentase skor di bawahnya. Jika Anda telah mengikuti tes standar seperti SAT, Anda telah menemukan persentil. Skor SAT dalam persentil ke-80 lebih tinggi dari 80 persen skor SAT lainnya.

Kedengarannya sederhana, bukan? Tidak begitu cepat. "Persentil" dapat memiliki beberapa definisi, dan karenanya, beberapa (atau lebih) cara untuk menghitungnya. Beberapa mendefinisikan persentil sebagai "lebih besar dari" (seperti dalam paragraf sebelumnya), beberapa mendefinisikan persentil sebagai "lebih besar dari atau sama dengan." "Lebih besar dari" sama dengan "eksklusif." “Lebih besar dari atau sama dengan” sama dengan “inklusif.”

Fungsi quantile() menghitung persentil. Jika dibiarkan sendiri, ia menghitung persentil ke-0, ke-25, ke-50, ke-75, dan ke-100. Ini menghitung persentil dengan cara yang konsisten dengan "inklusif" dan (jika perlu) menginterpolasi nilai untuk persentil. Saya mulai dengan mengurutkan vektor Horsepower.USA.Eight sehingga Anda dapat melihat skor secara berurutan dan membandingkannya dengan persentil:

```
> sort(Horsepower.USA.Eight)
[1] 170 190 200 210 295 300
```

Dan sekarang persentilnya:

```
>  quantile(Horsepower.USA.Eight)
    0%    25%    50%    75%   100%
170.00 192.50 205.00 273.75 300.00
```

Perhatikan bahwa persentil ke-25, ke-50, dan ke-75 adalah nilai yang tidak ada dalam vektor. Untuk menghitung persentil yang konsisten dengan "eksklusif", tambahkan argumen tipe dan atur sama dengan 6:

```
> quantile(Horsepower.USA.Eight, type = 6)
    0%    25%    50%    75%   100%
170.00 185.00 205.00 296.25 300.00
```

Tipe default (tipe pertama yang saya tunjukkan) adalah 7, omong-omong. Tujuh jenis lain (cara menghitung persentil) tersedia. Untuk melihatnya, ketik ?quantile ke dalam jendela Konsol (lalu baca dokumentasi pada tab Bantuan.) Selanjutnya, saya menggunakan tipe default untuk persentil. Persentil ke-25, ke-50, ke-75, dan ke-100 sering digunakan untuk meringkas sekelompok skor. Karena mereka membagi sekelompok skor menjadi empat, mereka disebut kuartil.

Namun, Anda tidak terjebak dengan kuartil. Anda bisa mendapatkan quantile() untuk mengembalikan persentil apa pun. Misalkan Anda ingin mencari persentil ke-54, ke-68, dan ke-91. Sertakan vektor dari angka-angka itu (dinyatakan sebagai proporsi) dan Anda berada dalam bisnis:

```
> quantile(Horsepower.USA.Eight, c(.54, .68, .91))
   54%    68%    91%
207.00 244.00 297.75
```

Persen peringkat

Fungsi quantile() memberi Anda skor yang sesuai dengan persentase yang diberikan. Anda juga dapat bekerja dalam arah sebaliknya — temukan peringkat persen yang sesuai dengan skor yang diberikan dalam kumpulan data. Misalnya, di Horsepower.USA.Eight, 170 adalah yang terendah dalam daftar enam, jadi peringkatnya adalah 1 dan peringkat persennya adalah 1/6, atau 16,67 persen.

Base R tidak menyediakan fungsi untuk ini, tetapi cukup mudah untuk membuatnya:

```
percent.ranks <-
          function(x){round((rank(x)/length(x))*100, digits = 2)}
```

Fungsi round() dengan digit = 2 membulatkan hasil ke dua tempat desimal. Menerapkan fungsi ini:

```
> percent.ranks(Horsepower.USA.Eight)
[1]  50.00  83.33  16.67 100.00  33.33  66.67
```

Trik Yang Rapi

Terkadang, Anda mungkin hanya ingin mengetahui peringkat persen dari satu skor dalam kumpulan skor — bahkan jika skor tersebut tidak ada dalam kumpulan data. Misalnya, berapa persen peringkat 273 di Horsepower.USA.Eight?

Untuk menjawab pertanyaan ini, Anda dapat memanfaatkan mean(). Menggunakan fungsi ini bersama dengan operator logika menghasilkan hasil yang menarik. Inilah yang saya maksud:

```
xx <- c(15,20,25,30,35,40,45,50)
```

Inilah hasil yang Anda harapkan:

```
> mean(xx)
[1] 32.5
```

Tapi ini satu yang mungkin tidak Anda:

```
> mean(xx > 15)
[1] 0.875
```

Hasilnya adalah proporsi skor dalam xx yang lebih besar dari 15.

Berikut beberapa lagi:

```
> mean(xx < 25)
[1] 0.25
> mean(xx <= 25)
[1] 0.375
> mean(xx <= 28)
[1] 0.375
```

Operator <= itu, tentu saja, berarti “kurang dari atau sama dengan,” sehingga yang terakhir memberikan proporsi skor dalam xx yang kurang dari atau sama dengan 28. Apakah Anda menangkap maksud saya? Untuk menemukan peringkat persen skor (atau skor potensial) dalam vektor seperti Horsepower.USA.Eight, itu

```
> mean(Horsepower.USA.Eight <= 273)*100
[1] 66.66667
```

6.4 KESIMPULAN

Selain fungsi untuk menghitung persentil dan peringkat, R menyediakan beberapa fungsi yang meringkas data dengan cepat dan melakukan banyak pekerjaan yang saya bahas dalam bab ini.

Satu disebut fivenum(). Fungsi ini, tidak mengejutkan, menghasilkan lima angka. Itu adalah lima angka yang digunakan pembuat plot kotak John Tukey untuk meringkas kumpulan data. Kemudian dia menggunakan angka-angka itu di plot kotaknya. (Lihat Bab 3).

```
> fivenum(Horsepower.USA.Eight)
[1] 170 190 205 295 300
```

Dari kiri ke kanan, itulah minimum, engsel bawah, median, engsel atas, dan maksimum. Ingat fungsi kuantil() dan sembilan cara (tipe) yang tersedia untuk menghitung kuantil? Hasil fungsi ini adalah apa yang dihasilkan oleh tipe = 2 dalam kuantil().
Fungsi lain, ringkasan (), lebih banyak digunakan:

```
> summary(Horsepower.USA.Eight)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  170.0   192.5   205.0   227.5   273.8   300.0
```

Ini memberikan rata-rata bersama dengan kuantil (sebagai tipe default dalam kuantil() menghitungnya). Fungsi ringkasan() serbaguna. Anda dapat menggunakannya untuk meringkas berbagai macam objek, dan hasilnya bisa terlihat sangat berbeda dari satu objek ke objek lainnya. Saya menggunakannya cukup sedikit di bab-bab mendatang.

BAB 7
MERINGKAS SEMUANYA

Ukuran tendensi sentral dan variabilitas yang saya bahas dalam bab-bab sebelumnya bukanlah satu-satunya cara untuk meringkas serangkaian skor. Langkah-langkah ini adalah bagian dari statistik deskriptif. Beberapa statistik deskriptif — seperti maksimum, minimum, dan rentang — mudah dipahami. Beberapa — seperti kemiringan dan kurtosis — tidak. Bab ini mencakup statistik deskriptif dan menunjukkan cara menghitungnya dalam R.

7.1 BERAPA BANYAK?

Mungkin statistik deskriptif mendasar adalah jumlah skor dalam satu set data. Di bab-bab sebelumnya, saya bekerja dengan length(), fungsi R yang menghitung angka ini. Seperti di bab-bab sebelumnya, saya bekerja dengan kerangka data Cars93, yang ada dalam paket MASS. (Jika tidak dipilih, klik kotak centang di sebelah MASS pada tab Paket.) Cars93 menyimpan data pada 27 variabel untuk 93 mobil yang tersedia pada tahun 1993. Apa yang terjadi ketika Anda menerapkan length() ke data frame?

```
> length(Cars93)
[1] 27
```

Jadi panjang() mengembalikan jumlah variabel dalam bingkai data. functionncol() melakukan hal yang sama:

```
> ncol(Cars93)
[1] 27
```

Saya sudah tahu jumlah kasus (baris) dalam bingkai data, tetapi jika saya harus menemukan nomor itu, nrow() akan menyelesaikannya:

```
> nrow(Cars93)
[1] 93
```

Jika Anda ingin mengetahui berapa banyak kasus dalam kerangka data yang memenuhi kondisi tertentu — seperti berapa banyak mobil yang berasal dari AS — Anda harus memperhitungkan cara R memperlakukan kondisi: R menempelkan label “BENAR” pada kotak yang memenuhi a kondisi, dan "FALSE" untuk kasus yang tidak. Juga, R memberikan nilai 1 ke "TRUE" dan 0 ke "FALSE".

Untuk menghitung jumlah mobil yang berasal dari Amerika Serikat, Anda menyatakan kondisinya dan kemudian menjumlahkan semua 1:

```
> sum(Cars93$Origin == "USA")
[1] 48
```

Untuk menghitung jumlah mobil non-USA dalam data frame, Anda dapat mengubah kondisi menjadi “non-USA”, tentu saja, atau Anda dapat menggunakan != — operator “not equal to”:

```
> sum(Cars93$Origin != "USA")
[1] 45
```

Kondisi yang lebih kompleks dimungkinkan. Untuk jumlah mobil USA 4 silinder:

```
> sum(Cars93$Origin == "USA" & Cars93$Cylinders == 4)
[1] 22
```

Atau, jika Anda lebih suka tidak ada $-signs:

```
> with(Cars93, sum(Origin == "USA" & Cylinders == 4))
[1] 22
```

Untuk menghitung jumlah elemen dalam vektor, panjang(), seperti yang mungkin telah Anda baca sebelumnya, adalah fungsi yang digunakan. Berikut adalah vektor tenaga kuda untuk mobil USA 4 silinder:

```
> Horsepower.USA.Four <- Cars93$Horsepower[Origin ==
          "USA" & Cylinders == 4]
```

dan inilah jumlah nilai horsepower dalam vektor itu:

```
> length(Horsepower.USA.Four)
[1] 22
```

7.2 TERTINGGI DAN TERENDAH

Dua statistik deskriptif yang tidak perlu diperkenalkan adalah nilai maksimum dan minimum dalam satu set skor:

```
> max(Horsepower.USA.Four)
[1] 155
> min(Horsepower.USA.Four)
[1] 63
```

Jika Anda membutuhkan kedua nilai secara bersamaan:

```
> range(Horsepower.USA.Four)
[1]  63 155
```

Hidup di dalam momen

Dalam statistika, momen adalah besaran-besaran yang berhubungan dengan bentuk suatu himpunan bilangan. Yang saya maksud dengan “bentuk sekumpulan angka” adalah “seperti apa bentuk histogram berdasarkan angka” — seberapa tersebarnya, seberapa simetrisnya, dan banyak lagi.

Momen mentah orde k adalah rata-rata dari semua angka dalam himpunan, dengan setiap angka dipangkatkan ke k sebelum Anda meratakannya. Jadi momen mentah pertama adalah mean aritmatika. Momen mentah kedua adalah rata-rata skor kuadrat. Momen mentah ketiga adalah rata-rata skor pangkat tiga, dan seterusnya.

Momen sentral didasarkan pada rata-rata penyimpangan angka dari rata-ratanya. (Mulai terdengar samar-samar akrab?) Jika Anda mengkuadratkan deviasi sebelum Anda rata-

rata, Anda memiliki momen sentral kedua. Jika Anda memotong deviasi sebelum Anda rata-rata, itulah momen sentral ketiga. Naikkan masing-masing ke kekuatan keempat sebelum Anda rata-rata, dan Anda memiliki momen sentral keempat. Aku bisa terus dan terus, tapi Anda mendapatkan ide.

Dua pertanyaan singkat:

1. Untuk setiap rangkaian angka, berapa momen sentral pertama?
2. Dengan nama lain apa Anda mengetahui momen sentral kedua?

Dua jawaban cepat: 1. Nol. 2. Varians populasi. Baca ulang Bab 5 jika Anda tidak percaya.

Momen yang bisa diajarkan

Sebelum saya melanjutkan, saya pikir itu ide yang baik untuk menerjemahkan ke dalam R semua yang telah saya katakan sejauh ini dalam bab ini. Dengan begitu, ketika Anda masuk ke paket R berikutnya untuk menginstal (yang menghitung momen), Anda akan tahu apa yang terjadi di balik layar.

Berikut adalah fungsi untuk menghitung momen pusat suatu vektor:

```
cen.mom <-function(x,y){mean((x - mean(x))^y)}
```

Argumen pertama, x, adalah vektor. Argumen kedua, y, adalah urutannya (kedua, ketiga, keempat ...).

Berikut adalah vektor untuk mencobanya:

```
Horsepower.USA <- Cars93$Horsepower[Origin == "USA"]
```

Dan inilah momen sentral kedua, ketiga, dan keempat:

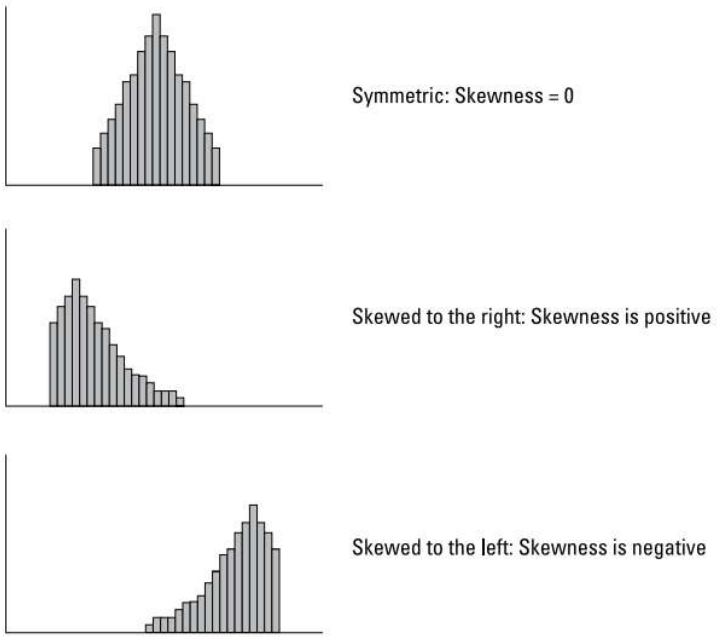
```
> cen.mom(Horsepower.USA,2)
[1] 2903.541
> cen.mom(Horsepower.USA,3)
[1] 177269.5
> cen.mom(Horsepower.USA,4)
[1] 37127741
```

Kembali ke deskriptif

Apa hubungan semua ini tentang momen dengan statistik deskriptif? Seperti yang saya katakan . . . dengan baik . . . beberapa saat yang lalu, pikirkan histogram yang didasarkan pada sekumpulan angka. Momen mentah pertama (rata-rata) menempatkan pusat histogram. Momen sentral kedua menunjukkan penyebaran histogram. Momen sentral ketiga terlibat dalam simetri histogram, yang disebut skewness. Momen sentral keempat menggambarkan seberapa gemuk atau tipisnya ekor (ujung ekstrem) dari histogram. Ini disebut kurtosis. Masuk ke momen-momen dengan tatanan yang lebih tinggi dari itu jauh di luar cakupan buku ini. Tapi mari kita masuk ke simetri dan "tailedness."

Kecondongan

Gambar 7.1 menunjukkan tiga histogram. Yang pertama adalah simetris; dua lainnya tidak. Simetri dan asimetri tercermin dalam statistik skewness.



Gambar 7.1 Tiga histogram, menunjukkan tiga jenis kemiringan.

Untuk histogram simetris, skewness adalah 0. Untuk histogram kedua — histogram yang mengarah ke kanan — nilai statistik skewness adalah positif. Itu juga dikatakan "miring ke kanan." Untuk histogram ketiga (yang mengarah ke kiri), nilai statistik skewness adalah negatif. Itu juga dikatakan "miring ke kiri".

Sekarang untuk formula. Saya akan membiarkan Mk mewakili momen sentral ke-k. Untuk menghitung skew- ness, itu

$$skewness = \frac{\sum\left(X-\bar{X}\right)^3}{(N-1)s^3}$$

Dalam bahasa Inggris, skewness dari sekumpulan angka adalah momen pusat ketiga dibagi dengan momen pusat kedua yang dipangkatkan menjadi tiga bagian. Dengan fungsi R yang saya definisikan sebelumnya, lebih mudah dilakukan daripada mengatakan:

```
> cen.mom(Horsepower.USA,3)/cen.mom(Horsepower.USA,2)^1.5
[1] 1.133031
```

Dengan paket momen, lebih mudah lagi. Pada tab Paket, klik Instal dan ketik momen ke dalam kotak dialog Instal Paket, dan klik Instal. Kemudian pada tab Paket, klik kotak centang di sebelah momen.

Inilah fungsi skewness() dalam aksinya:

```
> skewness(Horsepower.USA)
[1] 1.133031
```

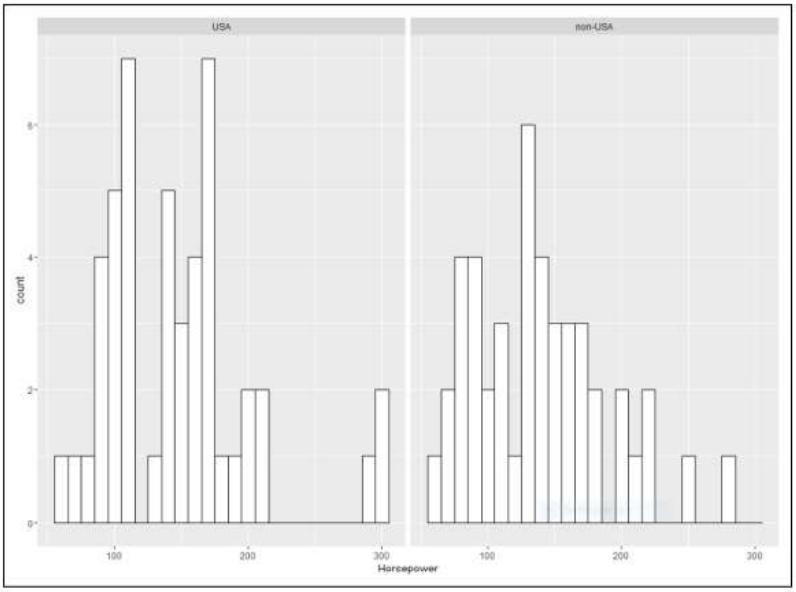
Jadi kemiringannya positif. Bagaimana itu dibandingkan dengan tenaga kuda untuk mobil non-AS?

```
> Horsepower.NonUSA <- Cars93$Horsepower[Origin == "non-USA"]
> skewness(Horsepower.NonUSA)
[1] 0.642995
```

Kemiringannya lebih positif untuk mobil AS daripada mobil non-AS. Seperti apa kedua histogram itu? Saya membuatnya berdampingan pada Gambar 4.1, di Bab 4. Untuk memudahkan, saya menunjukkannya di sini sebagai Gambar 7.2.
Kode yang menghasilkannya adalah:

```
ggplot(Cars93, aes(x=Horsepower)) +
  geom_histogram(color="black", fill="white",binwidth = 10)+
  facet_wrap(~Origin)
```

Konsisten dengan nilai skewness, histogram menunjukkan bahwa di mobil USA, skornya lebih banyak di sebelah kiri daripada di mobil non-USA. Terkadang lebih mudah untuk melihat tren dalam plot kepadatan daripada dalam histogram. Plot kepadatan menunjukkan proporsi skor antara batas bawah yang diberikan dan batas atas yang diberikan (seperti proporsi mobil dengan tenaga kuda antara 100 dan 140). Saya membahas kepadatan secara lebih rinci di Bab 8.

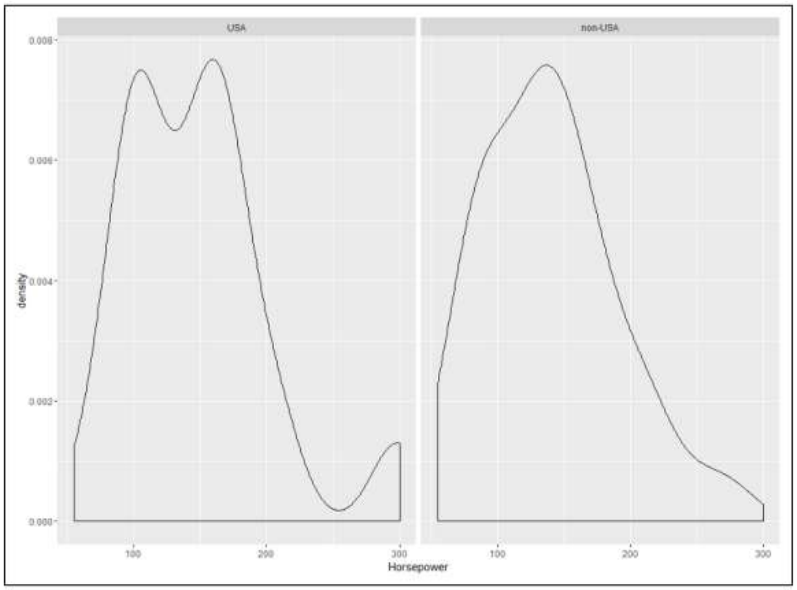


Gambar 7.2 Histogram tenaga kuda untuk mobil AS dan non-AS.

Mengubah satu baris kode menghasilkan plot kepadatan:

```
ggplot(Cars93, aes(x=Horsepower)) +
  geom_density() +
  facet_wrap(~Origin)
```

Gambar 7.3 menunjukkan dua plot kepadatan.

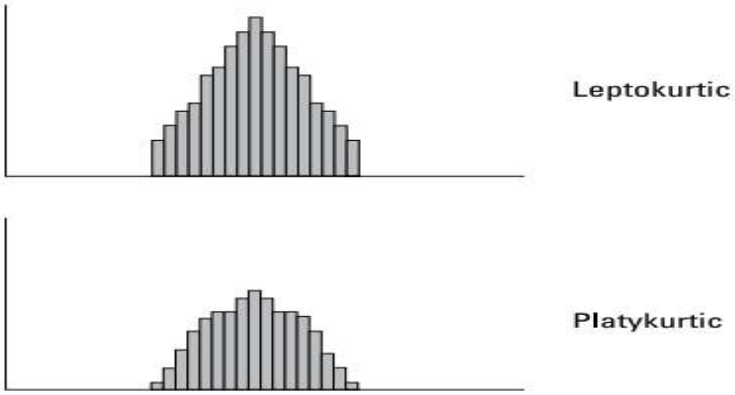


Gambar 7.3 Plot kepadatan tenaga kuda untuk mobil AS dan mobil non-AS.

Dengan plot kepadatan, tampaknya lebih mudah (bagi saya, bagaimanapun) untuk melihat lebih banyak kemiringan ke kiri (dan karenanya, kemiringan yang lebih positif) di plot di sebelah kiri.

Kurtosis

Gambar 7.4 menunjukkan dua histogram. Yang pertama memiliki ekor yang lebih gemuk daripada yang kedua. Yang pertama dikatakan leptokurtik. Yang kedua adalah platikurtik. Kurtosis untuk histogram pertama lebih besar daripada yang kedua.



Gambar 7.4 Dua histogram, menunjukkan dua jenis kurtosis.

Rumus untuk kurtosis adalah:

$$kurtosis = \frac{\sum\left(X - \bar{X}\right)^2}{(N-1)s^4} - 3$$

dimana M4 adalah momen pusat keempat dan M2 adalah momen pusat kedua. Jadi kurtosis adalah momen pusat keempat dibagi dengan kuadrat momen pusat kedua.

Banyak ahli statistik mengurangi 3 dari hasil rumus kurtosis. Mereka menyebut nilai itu sebagai kelebihan kurtosis. Yang dimaksud dengan “berlebihan” adalah kurtosis yang lebih besar (atau mungkin lebih kecil) daripada kurtosis dari sesuatu yang disebut distribusi normal standar, yang saya bahas di Bab 8. Karena pengurangan, kelebihan kurtosis bisa negatif. Mengapa 3 mewakili kurtosis dari distribusi normal standar? Jangan tanya. Menggunakan fungsi yang saya definisikan sebelumnya, kurtosis tenaga kuda untuk mobil AS adalah:

```
> cen.mom(Horsepower.USA,4)/cen.mom(Horsepower.USA,2)^2
[1] 4.403952
```

Tentu saja, fungsi kurtosis() dalam paket momen membuat ini menjadi mudah:

```
> kurtosis(Horsepower.USA)
[1] 4.403952
```

Ekor yang lebih gemuk pada plot kepadatan sisi kiri pada Gambar 7-3 menunjukkan bahwa mobil AS memiliki kurtosis yang lebih tinggi daripada mobil non-AS. Apakah ini benar?

```
> kurtosis(Horsepower.NonUSA)
[1] 3.097339
```

Ya itu!

Selain skewness() dan kurtosis(), paket momen menyediakan fungsi yang disebut moment() yang melakukan semua yang dilakukan cen.mom() dan sedikit lebih banyak lagi. Saya hanya berpikir itu akan menjadi ide yang baik untuk menunjukkan kepada Anda fungsi yang ditentukan pengguna yang menggambarkan apa yang masuk ke dalam menghitung momen sentral. (Apakah saya sedang "menjadi penting" ... atau apakah saya hanya "merebut momen"? Oke. Saya akan berhenti).

7.3 MENGATUR FREKUENSI

Cara yang baik untuk mengeksplorasi data adalah untuk mengetahui frekuensi kemunculan untuk setiap kategori variabel nominal, dan untuk setiap interval variabel numerik.

Variabel nominal: table() et al

Untuk variabel nominal, seperti Type of Automobile di Cars93, cara termudah untuk mendapatkan frekuensi adalah fungsi table() yang saya gunakan sebelumnya:

```
> car.types <-table(Cars93$Type)
> car.types

Compact   Large Midsize   Small  Sporty     Van
     16      11      22      21      14       9
```

Fungsi lain, prop.table(), menyatakan frekuensi ini sebagai proporsi dari jumlah keseluruhan:

```
> prop.table(car.types)

   Compact      Large    Midsize      Small     Sporty
       Van
```

```
0.17204301 0.11827957 0.23655914 0.22580645 0.15053763
         0.09677419
```

Nilai di sini tampak rusak karena halaman tidak selebar jendela Konsol. Jika saya membulatkan proporsi menjadi dua tempat desimal, hasilnya terlihat jauh lebih baik di halaman:

```
> round(prop.table(car.types),2)

Compact   Large Midsize   Small  Sporty     Van
   0.17    0.12    0.24    0.23    0.15    0.10
```

Fungsi lain, margin.table(), menambahkan frekuensi:

```
> margin.table(car.types)
[1] 93
```

Variabel numerik: hist()

Mentabulasi frekuensi untuk interval data numerik adalah bagian tak terpisahkan dari pembuatan histogram. (Lihat Bab 3.) Untuk membuat tabel frekuensi, gunakan fungsi grafik hist(), yang menghasilkan daftar komponen saat argumen plot FALSE:

```
> prices <- hist(Cars93$Price, plot=F, breaks=5)
> prices
$breaks
[1]  0 10 20 30 40 50 60 70

$counts
[1] 12 50 19  9  2  0  1

$density
[1] 0.012903226 0.053763441 0.020430108 0.009677419 0.002150538
         0.000000000
[7] 0.001075269

$mids
[1]  5 15 25 35 45 55 65

$xname
[1] "Cars93$Price"

$equidist
[1] TRUE
```

(Di Cars93, ingat, setiap harga dalam ribuan dolar).

Meskipun saya menentukan lima jeda, hist() menggunakan sejumlah jeda yang membuat semuanya terlihat "lebih cantik." Dari sini, saya dapat menggunakan mids (interval-midpoints) dan menghitung untuk membuat matriks frekuensi, dan kemudian bingkai data:

```
> prices.matrix <- matrix(c(prices$mids,prices$counts), ncol = 2)
> prices.frame <- data.frame(prices.matrix)
> colnames(prices.frame) <- c("Price Midpoint (X
        $1,000)","Frequency")
> prices.frame
  Price Midpoint (X $1,000) Frequency
1                         5        12
2                        15        50
3                        25        19
4                        35         9
5                        45         2
6                        55         0
7                        65         1
```

Frekuensi kumulatif

Cara lain untuk melihat frekuensi adalah dengan memeriksa frekuensi kumulatif: Setiap frekuensi kumulatif interval adalah jumlah dari frekuensinya sendiri dan semua frekuensi dalam interval sebelumnya.

Fungsi cumsum() melakukan aritmatika pada vektor frekuensi:

```
> prices$counts
[1] 12 50 19  9  2  0  1
> cumsum(prices$counts)
[1] 12 62 81 90 92 92 93
```

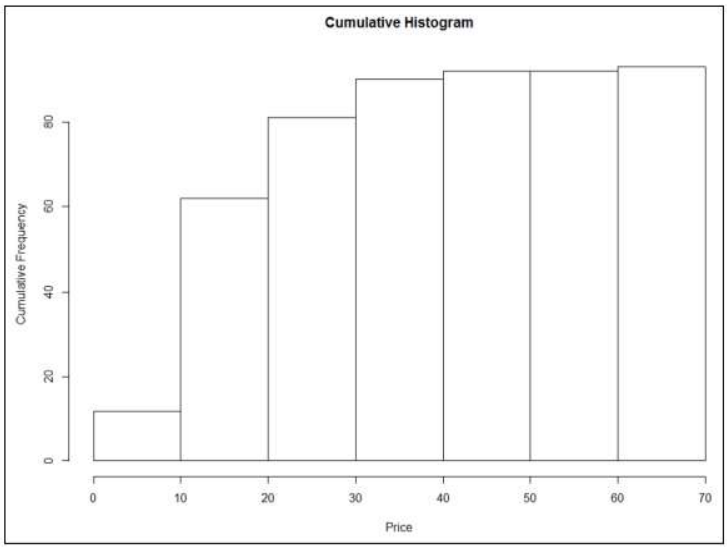
Untuk memplot histogram frekuensi kumulatif, saya mengganti vektor frekuensi kumulatif dengan yang asli:

```
> prices$counts <- cumsum(prices$counts)
```

dan kemudian terapkan plot():

```
> plot(prices, main = "Cumulative Histogram", xlab = "Price",
         ylab = "Cumulative Frequency")
```

Hasilnya adalah Gambar 7.5.



Gambar 7.5 Histogram frekuensi kumulatif dari data harga di Cars93.

Langkah demi langkah: Fungsi distribusi kumulatif empiris

Fungsi distribusi kumulatif empiris (ecdf) berkaitan erat dengan frekuensi kumulatif. Alih-alih menunjukkan frekuensi dalam suatu interval, bagaimanapun, ecdf menunjukkan proporsi skor yang kurang dari atau sama dengan setiap skor. Jika ini terdengar familier, mungkin karena Anda membaca tentang persentil di Bab 6.

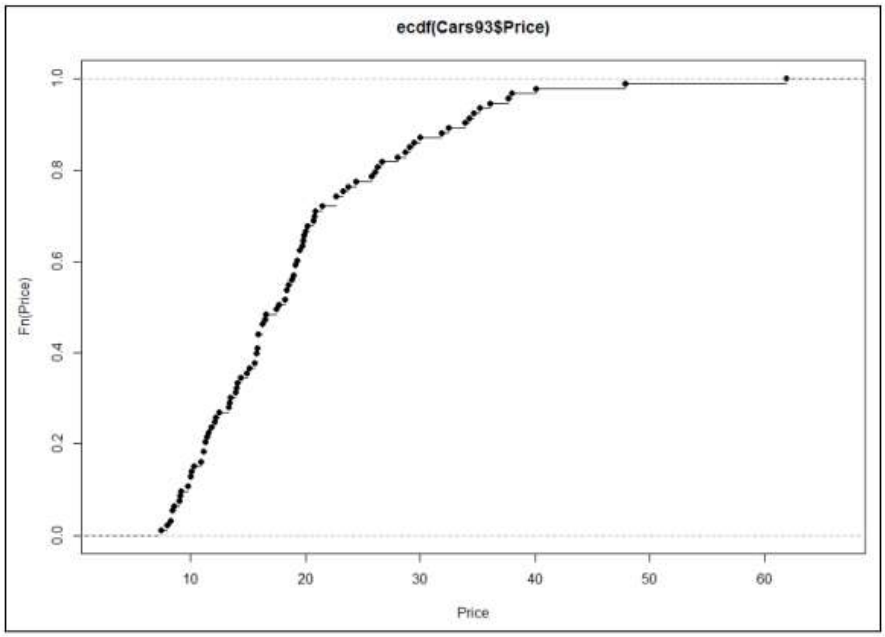
Di basis R, mudah untuk memplot ecdf:

```
> plot(ecdf(Cars93$Price), xlab = "Price", ylab = "Fn(Price)")
```

Ini menghasilkan Gambar 7.6.

Huruf besar F pada sumbu y adalah konvensi notasi untuk distribusi kumulatif. Fn berarti, pada dasarnya, "fungsi kumulatif" sebagai lawan dari f atau fn, yang hanya berarti "fungsi." (Label sumbu y juga bisa berupa Persentil(Harga)).

Perhatikan baik-baik plotnya. Ketika titik-titik berurutan berjauhan (seperti dua di kanan atas), Anda dapat melihat garis horizontal memanjang ke kanan dari suatu titik. (Sebuah garis memanjang dari setiap titik, tetapi garis-garis tersebut tidak terlihat ketika titik-titik tersebut digabungkan.) Pikirkan garis ini sebagai "langkah" dan kemudian titik berikutnya adalah langkah yang lebih tinggi dari titik sebelumnya. Seberapa tinggi? Itu akan menjadi 1/N, di mana N adalah jumlah skor dalam sampel. Untuk Cars93, itu akan menjadi 1/93, yang dibulatkan menjadi 0,011. (Sekarang pertimbangkan kembali judul subbagian ini. Lihat apa yang saya lakukan di sana?)



Gambar 7.6 Fungsi distribusi kumulatif empiris untuk data harga di Mobil93.

Mengapa ini disebut fungsi distribusi kumulatif "empiris"? Sesuatu yang empiris didasarkan pada pengamatan, seperti data sampel. Apakah mungkin untuk memiliki fungsi distribusi kumulatif non-empiris (cdf)? Ya — dan itulah cdf populasi tempat sampel berasal.

(Lihat Bab 1.) Salah satu kegunaan penting dari ecdf adalah sebagai alat untuk memperkirakan populasi cdf.

Jadi ecdf yang diplot adalah perkiraan cdf untuk populasi, dan perkiraan itu didasarkan pada data sampel. Untuk membuat perkiraan, Anda menetapkan probabilitas untuk setiap titik dan kemudian menjumlahkan probabilitas, poin demi poin, dari nilai minimum ke nilai maksimum. Ini menghasilkan probabilitas kumulatif untuk setiap titik.

Probabilitas yang ditetapkan untuk nilai sampel adalah perkiraan proporsi waktu nilai itu muncul dalam populasi. Apa perkiraannya? Itulah 1/N yang disebutkan di atas untuk setiap titik — .011, untuk sampel ini. Untuk nilai apa pun, itu mungkin bukan proporsi yang tepat dalam populasi. Itu hanya perkiraan terbaik dari sampel.

Saya lebih suka menggunakan ggplot() untuk memvisualisasikan ecdf. Karena saya mendasarkan plot pada vektor (Cars93$Price), sumber datanya adalah NULL:

```
ggplot(NULL, aes(x=Cars93$Price))
```

Sesuai dengan sifat langkah demi langkah dari fungsi ini, plot terdiri dari langkah-langkah, dan fungsi geom adalah geom_step. Statistik yang menempatkan setiap langkah pada plot adalah ecdf, jadi

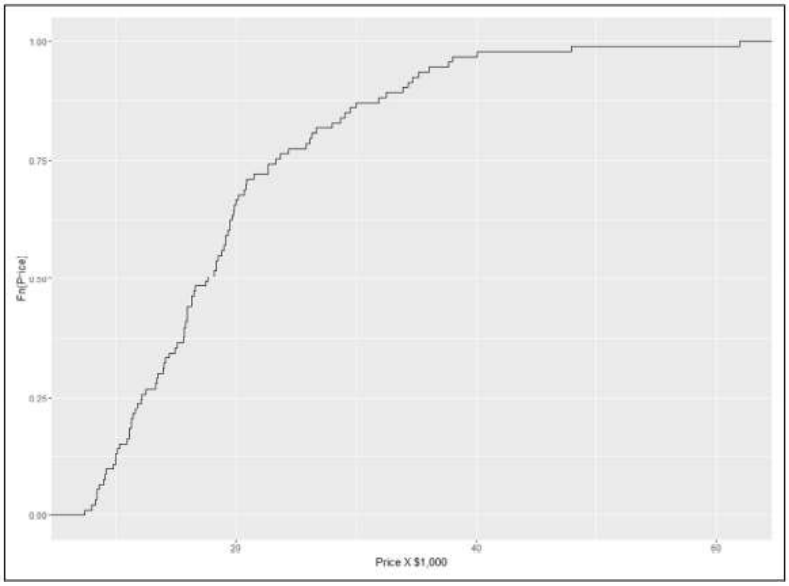
```
geom_step(stat="ecdf")
```

dan saya akan memberi label pada sumbu:

```
labs(x= "Price X $1,000",y = "Fn(Price)")
```

Menyatukan ketiga baris kode itu memberikan Gambar 7.7.

```
ggplot(NULL, aes(x=Cars93$Price)) +
  geom_step(stat="ecdf") +
  labs(x= "Price X $1,000",y = "Fn(Price)")
```



Gambar 7.7 Ecdf untuk data harga di Cars93, diplot dengan ggplot().

Untuk menempatkan sedikit pizzazz di grafik, saya menambahkan garis vertikal putus-putus di setiap kuartil. Sebelum saya menambahkan fungsi geom untuk garis vertikal, saya menempatkan informasi kuartil dalam vektor:

```
price.q <-quantile(Cars93$Price)
```

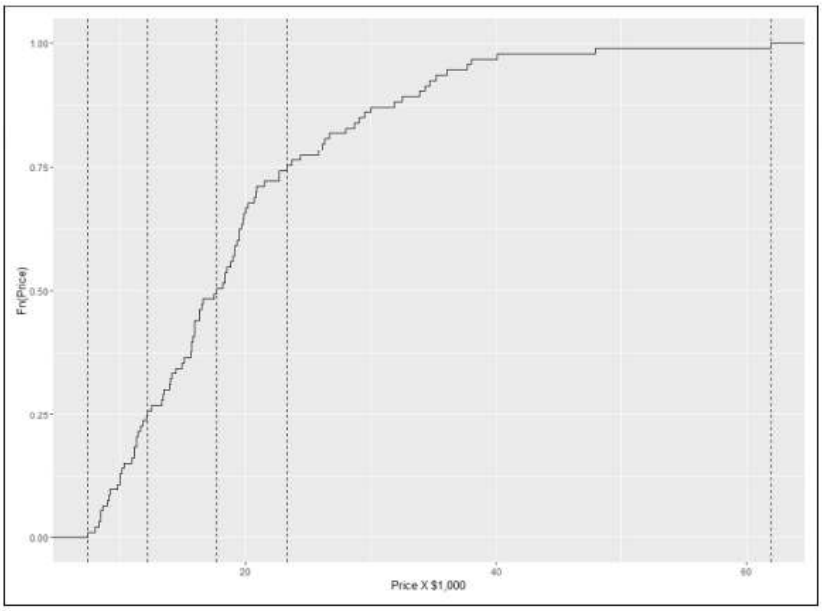
Dan sekarang

```
geom_vline(aes(xintercept=price.q),linetype = "dashed")
```

menambahkan garis vertikal. Pemetaan estetika menetapkan x-intercept dari setiap baris pada nilai kuartil.
Jadi baris kode ini

```
ggplot(NULL, aes(x=Cars93$Price)) +
  geom_step(stat="ecdf") +
  labs(x= "Price X $1,000",y = "Fn(Price)") +
  geom_vline(aes(xintercept=price.q),linetype = "dashed")
```

hasil pada Gambar 7.8.



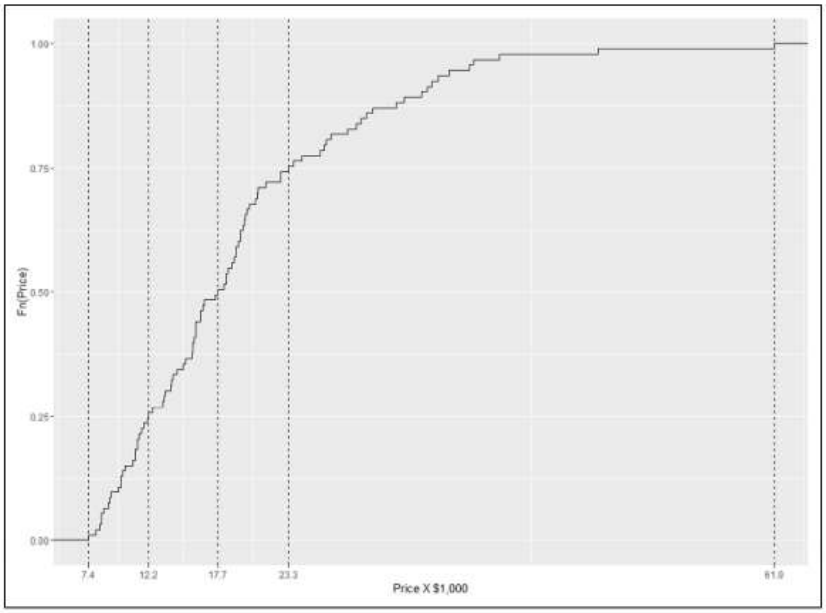
Gambar 7.8 Ecdf untuk data harga, dengan garis vertikal putus-putus di setiap kuartil.

Sentuhan akhir yang bagus adalah dengan menempatkan nilai kuartil pada sumbu x. Fungsi scale_x_continuous() menyelesaikannya. Ini menggunakan satu argumen yang disebut break (yang menetapkan lokasi nilai untuk diletakkan pada sumbu) dan yang lain disebut label (yang menempatkan nilai pada lokasi tersebut). Di sinilah vektor price.q berguna:

```
scale_x_continuous(breaks = price.q,labels = price.q)
```

Dan inilah kode R yang membuat Gambar 7.9:

```
ggplot(NULL, aes(x=Cars93$Price)) +
  geom_step(stat="ecdf") +
  labs(x= "Price X $1,000",y = "Fn(Price)") +
  geom_vline(aes(xintercept=price.q),linetype = "dashed")+
  scale_x_continuous(breaks = price.q,labels = price.q)
```



Gambar 7.9 Ecdf untuk data harga, dengan nilai kuartil pada sumbu x.

Variabel numerik: batang()

Pencipta plot kotak John Tukey mempopulerkan plot batang dan daun sebagai cara untuk memvisualisasikan distribusi angka dengan cepat. Ini bukan "plot" dalam arti grafik biasa di jendela Plot. Sebaliknya, ini adalah susunan angka di jendela Konsol. Dengan setiap skor dibulatkan ke bilangan bulat terdekat, setiap "daun" adalah digit paling kanan skor. Setiap "batang" terdiri dari semua digit lainnya.

Sebuah contoh akan membantu. Berikut adalah harga mobil di Cars93, disusun dalam urutan menaik dan dibulatkan ke bilangan bulat terdekat (ingat bahwa setiap harga dalam ribuan dolar):

```
> rounded <- (round(sort(Cars93$Price),0))
```

Saya menggunakan cat() untuk menampilkan nilai yang dibulatkan pada halaman ini. (Jika tidak, akan terlihat berantakan.) Nilai argumen isiannya membatasi jumlah karakter (termasuk spasi) pada setiap baris:

```
> cat(rounded, fill = 50)
7 8 8 8 8 9 9 9 9 10 10 10 10 10 11 11 11 11 11
11 12 12 12 12 12 13 13 14 14 14 14 14 15 15 16
16 16 16 16 16 16 16 16 16 17 18 18 18 18 18 18
19 19 19 19 19 20 20 20 20 20 20 20 21 21 21 22
23 23 23 24 24 26 26 26 27 28 29 29 30 30 32 32
34 34 35 35 36 38 38 40 48 62
```

Fungsi stem() menghasilkan plot batang-dan-daun dari nilai-nilai ini:

```
> stem(Cars93$Price)

  The decimal point is 1 digit(s) to the right of the |

  0 | 788889999
  1 | 000001111112222333444445566666666677888889999999
  2 | 00000001112333446667899
  3 | 00234455688
  4 | 08
  5 |
  6 | 2
```

Di setiap baris, nomor di sebelah kiri garis vertikal adalah batang. Angka yang tersisa adalah daun untuk baris itu. Pesan tentang titik desimal berarti "kalikan setiap batang dengan 10." Kemudian tambahkan setiap daun ke batang itu. Jadi, baris bawah memberitahu Anda bahwa satu skor yang dibulatkan dalam data adalah 62. Baris berikutnya ke atas menunjukkan bahwa tidak ada skor yang dibulatkan antara 50 dan 59. Baris di atas yang satu menunjukkan bahwa satu skor adalah 40 dan yang lainnya adalah 48. Saya akan serahkan kepada Anda untuk mencari tahu (dan memverifikasi) sisanya.

Saat saya meninjau daun, saya perhatikan bahwa plot batang menunjukkan satu skor 32 dan lainnya 33. Sebaliknya, skor bulat menunjukkan dua 32 dan tidak 33. Tampaknya, putaran stem() berbeda dengan putaran().

7.4 MERINGKAS BINGKAI DATA

Jika Anda mencari statistik deskriptif untuk variabel dalam bingkai data, fungsi summary() akan menemukannya untuk Anda. Saya mengilustrasikan dengan subset dari kerangka data Cars93:

```
> autos <- subset(Cars93, select = c(MPG.city,Type, Cylinders,
  Price, Horsepower))
> summary(autos)
    MPG.city          Type     Cylinders      Price
 Min.   :15.00   Compact:16   3     : 3   Min.   : 7.40
 1st Qu.:18.00   Large  :11   4     :49   1st Qu.:12.20
 Median :21.00   Midsize:22   5     : 2   Median :17.70
 Mean   :22.37   Small  :21   6     :31   Mean   :19.51
 3rd Qu.:25.00   Sporty :14   8     : 7   3rd Qu.:23.30
 Max.   :46.00   Van    : 9   rotary: 1   Max.   :61.90
   Horsepower
```

```
 Min.   : 55.0
 1st Qu.:103.0
 Median :140.0
 Mean   :143.8
 3rd Qu.:170.0
 Max.   :300.0
```

Perhatikan maksima, minima, dan kuartil untuk variabel numerik dan tabel frekuensi untuk Tipe dan Silinder. Dua fungsi dari paket Hmisc juga meringkas bingkai data. Untuk menggunakan fungsi ini, Anda memerlukan Hmisc di perpustakaan Anda. (Pada tab Paket, klik Instal dan ketik Hmisc ke dalam kotak Paket di kotak dialog Instal. Kemudian klik Instal).

Satu fungsi, deskripsikan.data.frame(), memberikan output yang sedikit lebih luas daripada yang Anda dapatkan dari ringkasan():

```
> describe.data.frame(autos)
autos

 5  Variables      93  Observations
---------------------------------------------------------------

MPG.city
      n missing  unique    Info    Mean     .05     .10
     93       0      21    0.99   22.37    16.6    17.0
    .25     .50     .75     .90     .95
   18.0    21.0    25.0    29.0    31.4

lowest : 15 16 17 18 19, highest: 32 33 39 42 46
---------------------------------------------------------------

Type
      n missing  unique
     93       0       6

          Compact Large Midsize Small Sporty Van
Frequency      16    11      22    21     14   9
%              17    12      24    23     15  10
---------------------------------------------------------------
```

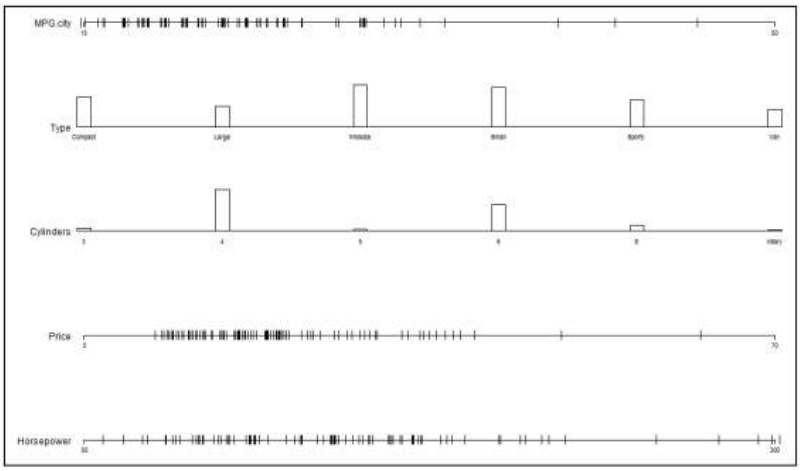
```
Cylinders
      n missing  unique
     93       0       6
          3  4 5  6 8 rotary
Frequency 3 49 2 31 7      1
%         3 53 2 33 8      1
---------------------------------------------------------------

Price
      n missing  unique    Info    Mean     .05     .10
     93       0      81       1   19.51    8.52    9.84
    .25     .50     .75     .90     .95
  12.20   17.70   23.30   33.62   36.74

lowest :  7.4  8.0  8.3  8.4  8.6
highest: 37.7 38.0 40.1 47.9 61.9
---------------------------------------------------------------

Horsepower
      n missing  unique    Info    Mean     .05     .10
     93       0      57       1   143.8    78.2    86.0
    .25     .50     .75     .90     .95
  103.0   140.0   170.0   206.8   237.0

lowest :  55  63  70  73  74, highest: 225 255 278 295 300
---------------------------------------------------------------
```

Nilai berlabel Info muncul di ringkasan variabel numerik. Nilai tersebut terkait dengan jumlah skor seri — semakin besar jumlah seri, semakin rendah nilai Info. (Perhitungan nilainya cukup rumit).

Fungsi Hmisc lainnya, datadensity(), memberikan ringkasan grafis, seperti pada Gambar 7.10:

```
> datadensity(autos)
```

Jika Anda berencana untuk menggunakan fungsi datadensity(), atur variabel bingkai data pertama menjadi numerik. Jika variabel pertama bersifat kategoris (dan dengan demikian muncul di bagian atas bagan), batang yang lebih panjang di plotnya terpotong di bagian atas.



Gambar 7.10 Bagan yang dibuat oleh kepadatan data (otomatis).

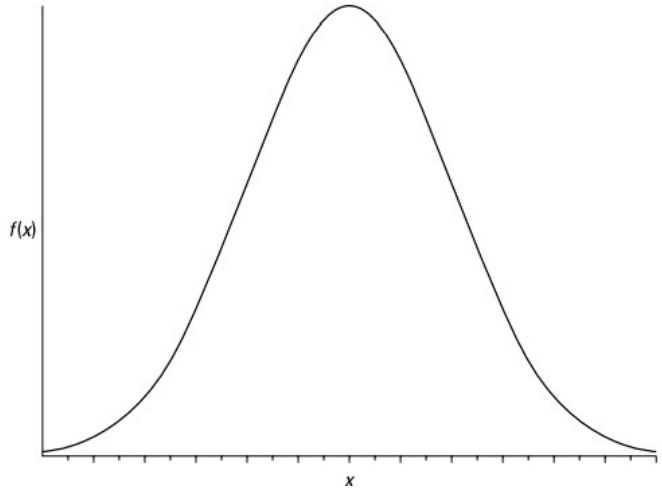
BAB 8
APA YANG NORMAL?

Salah satu tugas utama seorang ahli statistik adalah memperkirakan karakteristik suatu populasi. Pekerjaan menjadi lebih mudah jika ahli statistik dapat membuat beberapa asumsi tentang populasi yang dipelajarinya.

Berikut adalah asumsi yang bekerja berulang-ulang: Sebuah atribut, kemampuan, atau sifat tertentu didistribusikan ke seluruh populasi sehingga (1) kebanyakan orang memiliki jumlah rata-rata atau mendekati rata-rata dari atribut tersebut, dan (2) semakin sedikit orang. memiliki jumlah atribut yang semakin ekstrem. Dalam bab ini, saya membahas asumsi ini dan implikasinya terhadap statistik. Saya juga membahas fungsi R yang terkait dengan asumsi ini.

8.1 MEMUKUL KURVA

Atribut di dunia fisik, seperti panjang atau berat, semuanya tentang objek yang dapat Anda lihat dan sentuh. Tidak semudah itu di dunia ilmuwan sosial, ahli statistik, peneliti pasar, dan pebisnis. Mereka harus kreatif ketika mengukur sifat-sifat yang tidak dapat mereka tangani — seperti “kecerdasan”, “kemampuan bermusik”, atau “kesediaan untuk membeli produk baru”. Asumsi yang saya sebutkan dalam pendahuluan bab ini — bahwa kebanyakan orang berada di sekitar rata-rata dan semakin sedikit orang yang menuju ekstrem — tampaknya berhasil dengan baik untuk sifat-sifat tak berwujud itu. Karena ini sering terjadi, itu menjadi asumsi tentang bagaimana sebagian besar sifat didistribusikan.



Gambar 8.1 Kurva lonceng.

Dimungkinkan untuk menangkap asumsi ini dengan cara grafis. Gambar 8-1 menunjukkan kurva lonceng terkenal yang menggambarkan distribusi berbagai atribut. Sumbu horizontal mewakili pengukuran kemampuan yang sedang dipertimbangkan. Garis vertikal yang ditarik di tengah kurva akan sesuai dengan rata-rata pengukuran.

Asumsikan bahwa mungkin untuk mengukur sifat seperti kecerdasan dan berasumsi bahwa kurva ini mewakili distribusi kecerdasan dalam populasi: Kurva lonceng menunjukkan bahwa kebanyakan orang memiliki kecerdasan rata-rata, hanya sedikit yang memiliki sedikit kecerdasan, dan hanya sedikit yang jenius. Itu sepertinya cocok dengan apa yang kita ketahui tentang orang, bukan?

Menggali lebih dalam

Pada sumbu horizontal Gambar 8-1 Anda melihat x, dan pada sumbu vertikal, f(x). Apakah arti simbol ini? Sumbu horizontal, seperti yang saya sebutkan, mewakili pengukuran, jadi pikirkan setiap pengukuran sebagai x.

Penjelasan f(x) sedikit lebih rumit. Hubungan matematis antara x dan f(x) menciptakan kurva lonceng dan memungkinkan Anda untuk memvisualisasikannya. Hubungannya agak rumit, dan saya tidak akan membebani Anda dengan itu sekarang. (Saya akan membahasnya sebentar lagi.) Cukup pahami bahwa f(x) mewakili ketinggian kurva untuk nilai x tertentu. Ini berarti bahwa Anda memberikan nilai untuk x (dan untuk beberapa hal lainnya), dan kemudian hubungan kompleks itu mengembalikan nilai f(x).

Biarkan saya masuk ke spesifik. Nama formal untuk "kurva lonceng" adalah distribusi normal. Istilah f(x) disebut kerapatan peluang, jadi distribusi normal adalah contoh fungsi kerapatan peluang. Daripada memberi Anda definisi teknis tentang kepadatan probabilitas, saya meminta Anda untuk memikirkan kepadatan probabilitas sebagai sesuatu yang memungkinkan Anda untuk memikirkan area di bawah kurva sebagai probabilitas. Probabilitas dari . . . apa? Itu akan muncul di subbagian berikutnya.

8.2 PARAMETER DARI DISTRIBUSI NORMAL

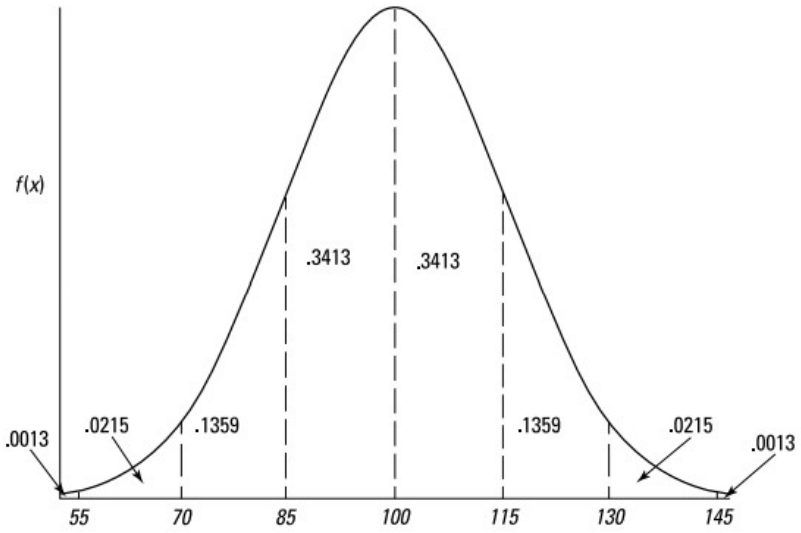
Anda sering mendengar orang berbicara tentang “distribusi normal”. Itu keliru. Ini benar-benar keluarga distribusi. Anggota keluarga berbeda satu sama lain dalam dua parameter — ya, parameter karena saya berbicara tentang populasi. Kedua parameter tersebut adalah mean (μ) dan standar deviasi (σ). Rata-rata memberi tahu Anda di mana pusat distribusi berada, dan standar deviasi memberi tahu Anda seberapa menyebar distribusi di sekitar rata-rata. Mean berada di tengah distribusi. Setiap anggota keluarga distribusi normal adalah simetris — sisi kiri dari distribusi adalah bayangan cermin dari kanan. (Ingat kemiringan, dari Bab 7? "Simetris" berarti bahwa kemiringan distribusi normal adalah nol).

Karakteristik dari keluarga distribusi normal diketahui oleh ahli statistik. Lebih penting lagi, Anda dapat menerapkan karakteristik tersebut pada pekerjaan Anda. Bagaimana? Ini membawa saya kembali ke probabilitas. Anda dapat menemukan beberapa probabilitas yang berguna jika Anda:

- Dapat membuat garis yang mewakili skala atribut yang Anda ukur (sumbu x, dengan kata lain)
- Dapat menunjukkan pada garis di mana rata-rata pengukuran adalah
- Mengetahui simpangan baku
- Dapat mengasumsikan bahwa atribut terdistribusi secara normal ke seluruh populasi

Saya akan bekerja dengan skor IQ untuk menunjukkan kepada Anda apa yang saya maksud. Skor pada tes IQ mengikuti distribusi normal. Rata-rata distribusi skor ini adalah 100, dan

standar deviasinya adalah 15. Gambar 8.2 menunjukkan kepadatan probabilitas untuk distribusi ini. Anda mungkin pernah membaca di tempat lain bahwa standar deviasi untuk IQ adalah 16 daripada 15. Itulah kasus tes IQ versi Stanford-Binet. Untuk versi lain, standar deviasi adalah 15.



Gambar 8.2 Distribusi normal IQ, dibagi menjadi standar deviasi.

Seperti yang ditunjukkan Gambar 8.2, saya telah membuat garis untuk skala IQ (sumbu x). Setiap titik pada garis mewakili skor IQ. Dengan mean (100) sebagai titik referensi, saya telah menandai setiap 15 poin (standar deviasi). Saya telah menggambar garis putus-putus dari rata-rata hingga f(100) (ketinggian distribusi di mana x = 100) dan menggambar garis putus-putus dari setiap titik simpangan baku.

Gambar tersebut juga menunjukkan proporsi daerah yang dibatasi oleh kurva dan sumbu horizontal, dan oleh pasangan simpangan baku yang berurutan. Ini juga menunjukkan proporsi di luar tiga standar deviasi di kedua sisi (55 dan 145). Perhatikan bahwa kurva tidak pernah menyentuh horizontal. Itu semakin dekat dan dekat, tetapi tidak pernah menyentuh. (Para matematikawan mengatakan bahwa kurva asimtotik terhadap horizontal).

Jadi antara mean dan satu standar deviasi — antara 100 dan 115 — adalah 0,3413 (atau 34,13 persen) dari skor dalam populasi. Cara lain untuk mengatakan ini: Probabilitas skor IQ antara 100 dan 115 adalah 0,3413. Pada ekstrem, di bagian ekor distribusi, 0,0013 (0,13 persen) dari skor berada di setiap sisi (kurang dari 55 atau lebih besar dari 145).

Proporsi pada Gambar 8-2 berlaku untuk setiap anggota keluarga dengan distribusi normal, tidak hanya untuk skor IQ. Sebagai contoh, di sidebar “Caching Some z” di Bab 6, saya menyebutkan skor SAT, yang memiliki rata-rata 500 dan standar deviasi 100. Mereka juga terdistribusi secara normal. Itu berarti 34,13 persen nilai SAT antara 500 dan 600, 34,13 persen antara 400 dan 500, dan . . . baik, Anda dapat menggunakan Gambar 8-2 sebagai panduan untuk proporsi lainnya.

Bekerja dengan Distribusi Normal

Hubungan kompleks yang saya katakan tentang antara x dan f(x) adalah:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[\frac{(x-\mu)^2}{2\sigma^2}\right]}$$

Jika Anda memberikan nilai untuk (rata-rata), (standar deviasi), dan x (skor), persamaan tersebut mengembalikan nilai f(x), tinggi dari distribusi normal di x. dan e adalah konstanta penting dalam matematika: kira-kira 3,1416 (perbandingan keliling lingkaran dengan diameternya); e adalah sekitar 2,71828. Ini terkait dengan sesuatu yang disebut logaritma natural (dijelaskan dalam Bab 16) dan banyak konsep matematika lainnya.

Distribusi dalam R

Keluarga distribusi normal adalah salah satu dari banyak keluarga distribusi yang dimasukkan ke dalam R. Berurusan dengan keluarga ini adalah intuitif. Ikuti panduan ini:

- Mulailah dengan nama keluarga distribusi di R (norma untuk keluarga normal, misalnya).
- Di awal nama keluarga, tambahkan d untuk bekerja dengan fungsi kepadatan probabilitas. Untuk fungsi kepadatan probabilitas untuk keluarga normal, maka, ini adalah dnorm() — yang setara dengan persamaan yang baru saja saya tunjukkan kepada Anda.
- Untuk fungsi kepadatan kumulatif (cdf), tambahkan p (pnorm(), misalnya).
- Untuk kuantil, tambahkan q (qnorm(), yang dalam istilah matematika adalah kebalikan dari cdf).
- Untuk membangkitkan bilangan acak dari suatu distribusi, tambahkan r. Jadi rnorm() menghasilkan angka acak dari anggota keluarga distribusi normal.

Fungsi kepadatan normal

Saat bekerja dengan fungsi distribusi normal apa pun, Anda harus memberi tahu fungsi tersebut anggota keluarga distribusi normal mana yang Anda minati. Anda melakukannya dengan menentukan mean dan standar deviasi.

Jadi, jika Anda membutuhkan tinggi distribusi IQ untuk IQ = 100, berikut cara menemukannya:

```
> dnorm(100,m=100,s=15)
[1] 0.02659615
```

Ini tidak berarti bahwa peluang menemukan skor IQ 100 adalah 0,027. Kerapatan probabilitas tidak sama dengan probabilitas. Dengan fungsi kepadatan probabilitas, masuk akal untuk berbicara tentang probabilitas skor antara dua batas — seperti probabilitas skor antara 100 dan 115.

Memplot kurva normal

dnorm() berguna sebagai alat untuk merencanakan distribusi normal. Saya menggunakannya bersama dengan ggplot() untuk menggambar grafik IQ yang sangat mirip dengan Gambar 8.2.

Sebelum saya menyiapkan pernyataan ggplot(), saya membuat tiga vektor yang berguna. Pertama:

```
x.values <- seq(40,160,1)
```

adalah vektor yang akan saya berikan ke ggplot() sebagai pemetaan estetika untuk sumbu x. Pernyataan ini membuat urutan 121 angka, dimulai dengan 40 (4 standar deviasi di bawah mean) sampai 160 (4 standar deviasi di atas mean).
Kedua:

```
sd.values <- seq(40,160,15)
```

adalah vektor dari sembilan nilai standar deviasi dari 40 hingga 160. Angka ini menggambarkan penciptaan garis putus-putus vertikal pada setiap standar deviasi pada Gambar 8.2.
Vektor ketiga:

```
zeros9 <- rep(0,9)
```

juga akan menjadi bagian dari pembuatan garis putus-putus vertikal. Itu hanya vektor sembilan nol.

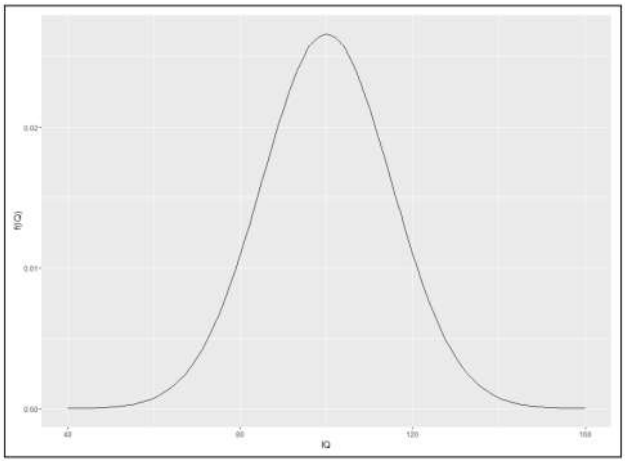
Ke ggplot(). Karena data adalah vektor, argumen pertama adalah NULL. Pemetaan estetis untuk sumbu-x adalah, seperti yang saya sebutkan sebelumnya, vektor nilai-x. Bagaimana dengan pemetaan sumbu y? Nah, ini adalah plot dari fungsi kerapatan normal untuk mean = 100 dan sd =15, jadi Anda akan mengharapkan pemetaan sumbu y menjadi dnorm(x.values, m=100, s=15), bukan Anda? Dan Anda benar! Ini dia ggplot() pernyataan:

```
ggplot(NULL,aes(x=x.values,y=dnorm(x.values,m=100,s=15)))
```

Tambahkan fungsi garis geom untuk plot dan label untuk sumbu, dan inilah yang saya miliki:

```
ggplot(NULL,aes(x=x.values,y=dnorm(x.values,m=100,s=15))) +
  geom_line() +
  labs(x="IQ",y="f(IQ)")
```

Dan itu menarik Gambar 8.3.



Gambar 8.3 Plot awal fungsi kerapatan normal untuk IQ.

Seperti yang Anda lihat, ggplot() memiliki idenya sendiri tentang nilai yang akan diplot pada sumbu x. Alih-alih bertahan dengan default, saya ingin menempatkan sd.values pada sumbu x. Untuk mengubah nilai tersebut, saya menggunakan scale_x_continuous() untuk mengubah skala sumbu x. Salah satu argumennya, memecah, menetapkan titik pada sumbu x

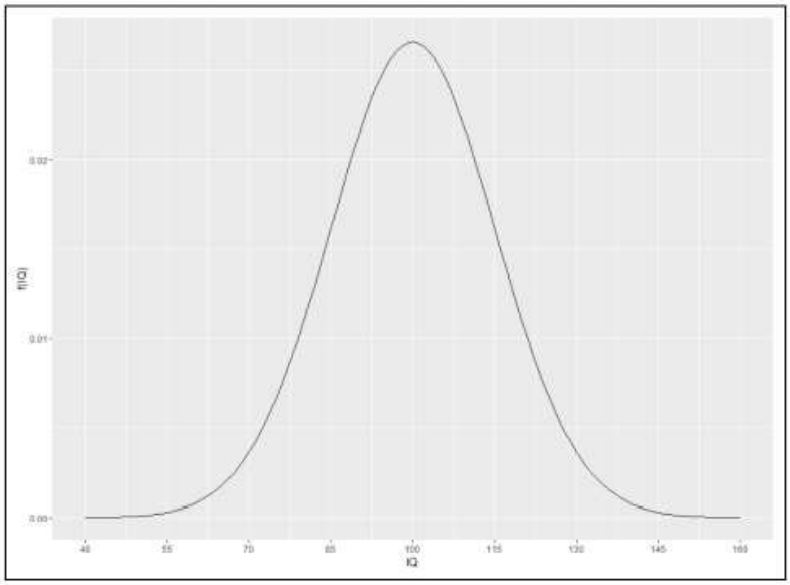
untuk nilai, dan yang lainnya, memberi label, memasok nilai. Untuk masing-masing, saya menyediakan sd.values:

```
scale_x_continuous(breaks=sd.values,labels = sd.values)
```

Sekarang kodenya adalah:

```
ggplot(NULL,aes(x=x.values,y=dnorm(x.values,m=100,s=15))) +
  geom_line() +
  labs(x="IQ",y="f(IQ)")+
  scale_x_continuous(breaks=sd.values,labels = sd.values)
```

dan hasilnya adalah Gambar 8.4.



Gambar 8.4 Fungsi kerapatan normal untuk IQ dengan standar deviasi pada sumbu x.

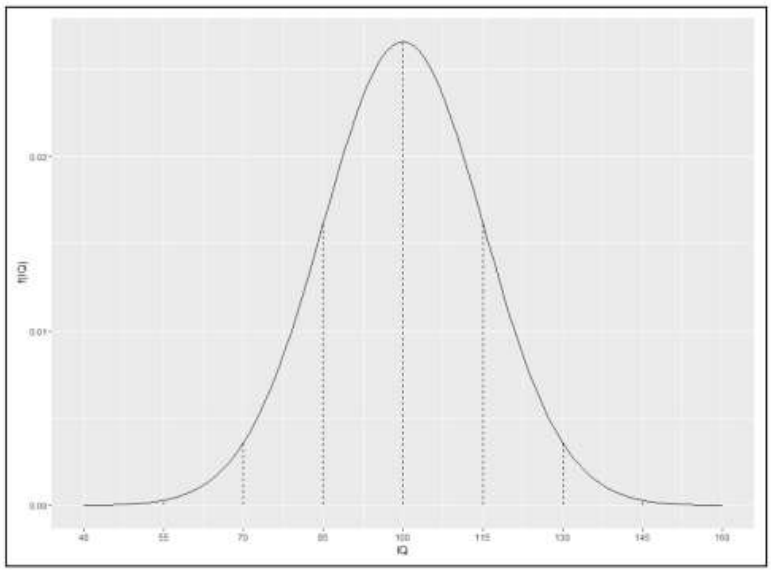
Dalam dunia ggplot, garis vertikal yang dimulai pada sumbu x dan berakhir pada kurva disebut segmen. Jadi fungsi geom yang sesuai untuk menggambarnya adalah geom_ segment(). Fungsi ini membutuhkan titik awal untuk setiap segmen dan titik akhir untuk setiap segmen. Saya menentukan titik-titik itu dalam pemetaan estetika di dalam geom. Koordinat x untuk titik awal untuk sembilan segmen berada di sd. nilai-nilai. Segmen dimulai dari sumbu x, sehingga kesembilan koordinat y semuanya nol — yang kebetulan merupakan isi dari vektor nol9. Segmen berakhir pada kurva, sehingga koordinat x untuk titik akhir sekali lagi, sd.values. Koordinat y? Itu akan menjadi dnorm(sd.values, m=100,s=15). Menambahkan pernyataan tentang garis putus-putus, pernyataan geom_segment() yang agak sibuk adalah:

```
geom_segment((aes(x=sd.values,y=zeros9,xend =
          sd.values,yend=dnorm(sd.values,m=100,s=15))),
          linetype = "dashed")
```

Kodenya sekarang menjadi:

```
ggplot(NULL,aes(x=x.values,y=dnorm(x.values,m=100,s=15))) +
  geom_line() +
  labs(x="IQ",y="f(IQ)")+
  scale_x_continuous(breaks=sd.values,labels = sd.values) +
  geom_segment((aes(x=sd.values,y=zeros9,xend =
          sd.values,yend=dnorm(sd.values,m=100,s=15))),
          linetype = "dashed")
```

yang menghasilkan Gambar 8.5.



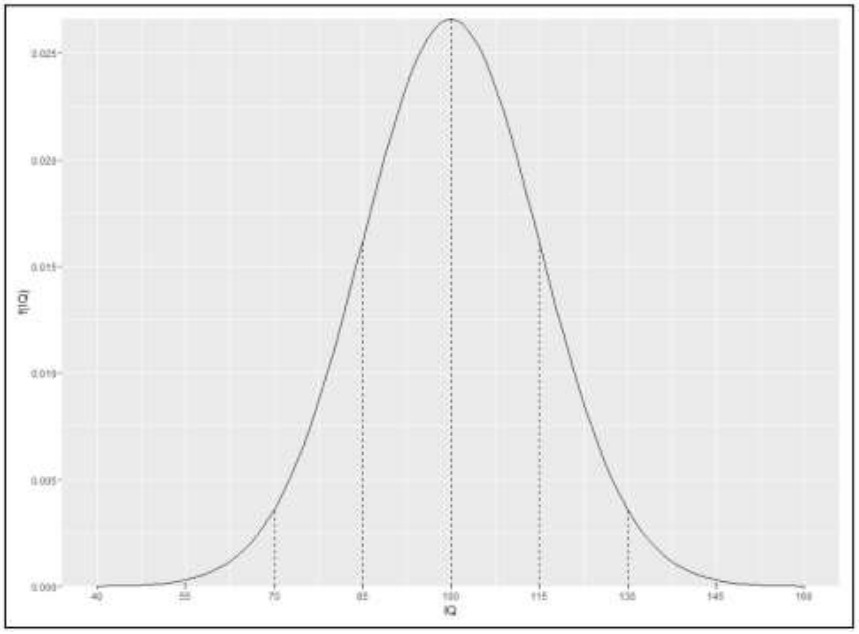
Gambar 8-5 Plot IQ dengan segmen garis putus-putus vertikal pada standar deviasi.

Satu sentuhan kecil lagi dan saya selesai menunjukkan kepada Anda bagaimana hal itu dilakukan. Saya tidak terlalu tergila-gila dengan ruang antara nilai-x dan sumbu-x. Saya ingin menghapus potongan kecil grafik itu dan memindahkan nilainya lebih dekat ke tempat (setidaknya menurut saya) seharusnya.

Untuk melakukan itu, saya menggunakan scale_y_continuous(), yang argumen perluasannya mengontrol ruang antara nilai-x dan sumbu-x. Ini adalah vektor dua elemen dengan default yang mengatur jumlah ruang yang Anda lihat di Gambar 8.5. Tanpa masuk terlalu dalam, menyetel vektor itu ke c(0,0) akan menghilangkan spasi.

Baris-baris kode ini menggambar Gambar 8.6 yang estetis:

```
ggplot(NULL,aes(x=x.values,y=dnorm(x.values,m=100,s=15))) +
  geom_line() +
  labs(x="IQ",y="f(IQ)")+
  scale_x_continuous(breaks=sd.values,labels = sd.values) +
  geom_segment((aes(x=sd.values,y=zeros9,xend =
          sd.values,yend=dnorm(sd.values,m=100,s=15))),
          linetype = "dashed")+
  scale_y_continuous(expand = c(0,0))
```



Gambar 8.6 Produk jadi: Plot IQ tanpa spasi antara nilai-x dan sumbu-x.

Fungsi kepadatan kumulatif

Fungsi kepadatan kumulatif pnorm(x,m,s) mengembalikan probabilitas skor kurang dari x dalam distribusi normal dengan mean m dan standar deviasi s.

Seperti yang Anda harapkan dari Gambar 8.2 (dan plot berikutnya yang saya buat):

```
> pnorm(100,m=100,s=15)
[1] 0.5
```

Bagaimana dengan peluang kurang dari 85?

```
> pnorm(85,m=100,s=15)
[1] 0.1586553
```

Jika Anda ingin menemukan probabilitas skor lebih besar dari 85, pnorm() dapat menanganinya juga. Ini memiliki argumen yang disebut lower.tail yang nilai defaultnya, TRUE, mengembalikan probabilitas “kurang dari.” Untuk “lebih besar dari”, atur nilainya ke FALSE:

```
> pnorm(85,m=100,s=15, lower.tail = FALSE)
[1] 0.8413447
```

Sering kali Anda menginginkan probabilitas skor antara batas bawah dan batas atas — seperti probabilitas skor IQ antara 85 dan 100. Beberapa panggilan ke pnorm() dikombinasikan dengan sedikit aritmatika akan menyelesaikannya.

Namun, itu tidak perlu. Sebuah fungsi yang disebut pnormGC() dalam paket hebat yang disebut tigerstats melakukan itu dan banyak lagi. Huruf GC berarti kalkulator grafis, tetapi bisa juga berarti Georgetown College (di Georgetown, Kentucky), sekolah tempat asal paket ini. (Pada tab Paket, klik Instal, lalu di kotak dialog Instal Paket, ketikkan tigerstats dan klik Instal. Saat Anda melihat tigerstats pada tab Paket, pilih kotak centangnya).

Sekarang perhatikan baik-baik:

```
>pnormGC(c(85,100),region="between",m=100,s=15,graph=TRUE)
[1] 0.3413447
```

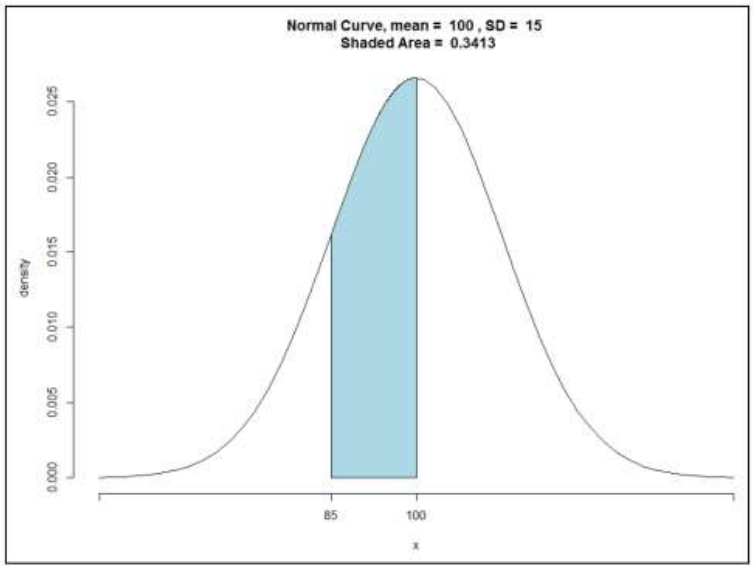
Selain jawaban, argumen graph=TRUE menghasilkan Gambar 8.7.

Merencanakan cdf

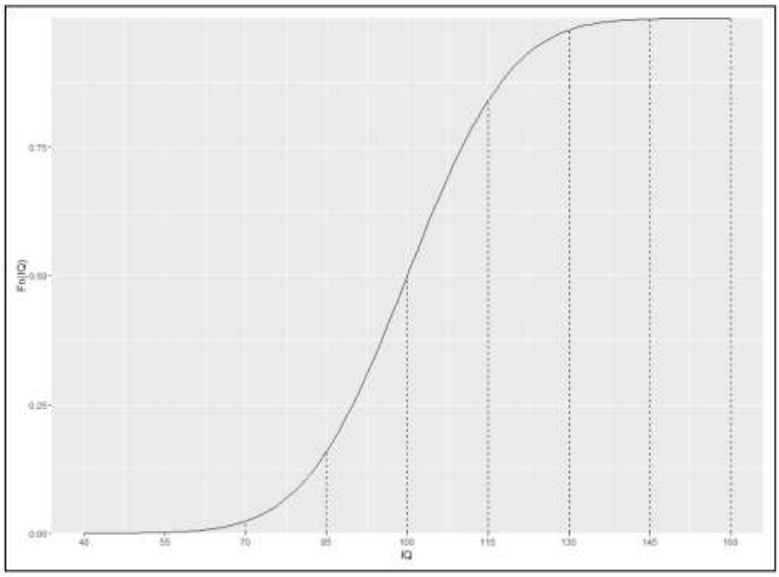
Mengingat bahwa saya sudah melakukan angkat berat ketika saya menunjukkan kepada Anda bagaimana memplot fungsi kepadatan, kode R untuk fungsi kepadatan kumulatif adalah sekejap:

```
ggplot(NULL,aes(x=x.values,y=pnorm(x.values,m=100,s=15))) +
  geom_line() +
  labs(x="IQ",y="Fn(IQ)")+
  scale_x_continuous(breaks=sd.values,labels = sd.values) +
  geom_segment((aes(x=sd.values,y=zeros9,xend =
          sd.values,yend=pnorm(sd.values,mean=100,sd=15))),
          linetype = "dashed")+
  scale_y_continuous(expand=c(0,0))
```

Ya, yang Anda lakukan hanyalah mengubah dnorm menjadi pnorm dan mengedit label sumbu y. Penggunaan kembali kode. Jadi (saya harap Anda setuju) adalah Gambar 8.8.



Gambar 8.7 Memvisualisasikan probabilitas skor IQ antara 85 dan 100 (dalam paket tigerstats)



Gambar 8.8 Fungsi kepadatan kumulatif dari distribusi IQ.

Segmen garis yang terangkat dari sumbu x dengan jelas menunjukkan bahwa 100 berada pada persentil ke-50 (0,50 dari skor di bawah 100). Yang membawa saya ke kuantil distribusi normal, topik bagian berikutnya.

Kuantitas distribusi normal

Fungsi qnorm() adalah kebalikan dari pnorm(). Berikan qnorm() sebuah area, dan itu mengembalikan skor yang memotong area itu (ke kiri) dalam distribusi normal yang ditentukan:

```
> qnorm(0.1586553,m=100,s=15)
[1] 85
```

Area (ke kiri), tentu saja, adalah persentil (dijelaskan dalam Bab 6).

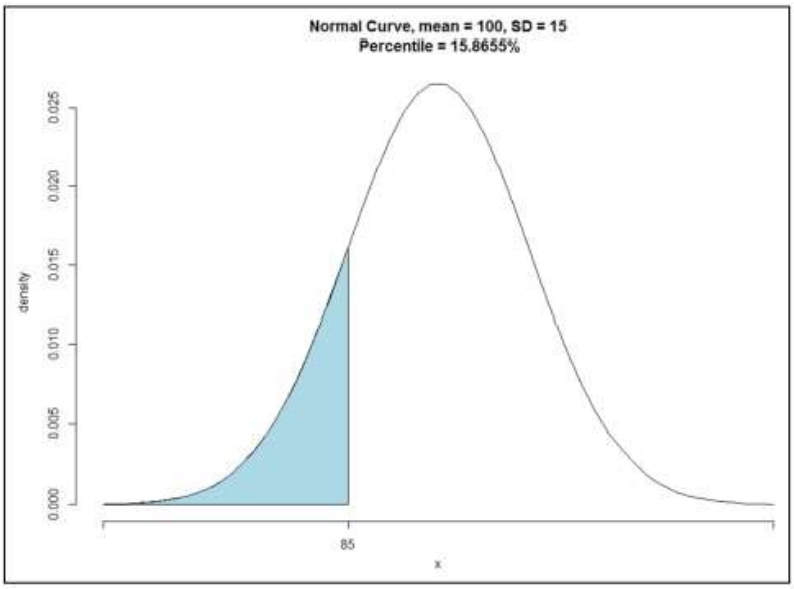
Untuk menemukan skor yang memotong area yang ditunjukkan ke kanan:

```
> qnorm(0.1586553,m=100,s=15, lower.tail = FALSE)
[1] 115
```

Begini cara qnormGC() (dalam paket tigerstats) menanganinya:

```
> qnormGC(.1586553, region = "below",m=100,s=15, graph=TRUE)
[1] 85
```

Fungsi ini juga membuat Gambar 8.9.



Gambar 8.9 Plot dibuat oleh qnormGC().

Anda biasanya tidak peduli dengan persentil ke-15.86553. Biasanya, kuartillah yang menarik perhatian Anda:

```
> qnorm(c(0,.25,.50,.75,1.00),m=100,s=15)
[1]      -Inf  89.88265 100.00000 110.11735       Inf
```

Persentil ke-0 dan ke-100 (— Tak hingga dan Tak Terhingga) menunjukkan bahwa cdf tidak pernah sepenuhnya menyentuh sumbu x atau mencapai maksimum yang tepat. Kuartil tengah adalah yang paling menarik, dan paling baik jika dibulatkan:

```
> round(qnorm(c(.25,.50,.75),m=100,s=15))
[1]  90 100 110
```

Merencanakan cdf dengan kuartil

Untuk mengganti nilai simpangan baku pada Gambar 8-8 dengan tiga nilai kuartil, Anda mulai dengan membuat dua vektor baru:

```
> q.values <-round(qnorm(c(.25,.50,.75),m=100,s=15))
> zeros3 <- c(0,0,0)
```

Sekarang yang harus Anda lakukan adalah meletakkan vektor-vektor tersebut di tempat yang sesuai di scale_x_continuous() dan di geom_segment():

```
ggplot(NULL,aes(x=x.values,y=pnorm(x.values,m=100,s=15))) +
  geom_line() +
  labs(x="IQ",y="Fn(IQ)")+
  scale_x_continuous(breaks=q.values,labels = q.values) +
  geom_segment((aes(x=q.values,y=zeros3,xend =
          q.values,yend=pnorm(q.values,mean=100,sd=15))),
          linetype = "dashed")+
  scale_y_continuous(expand=c(0,0))
```

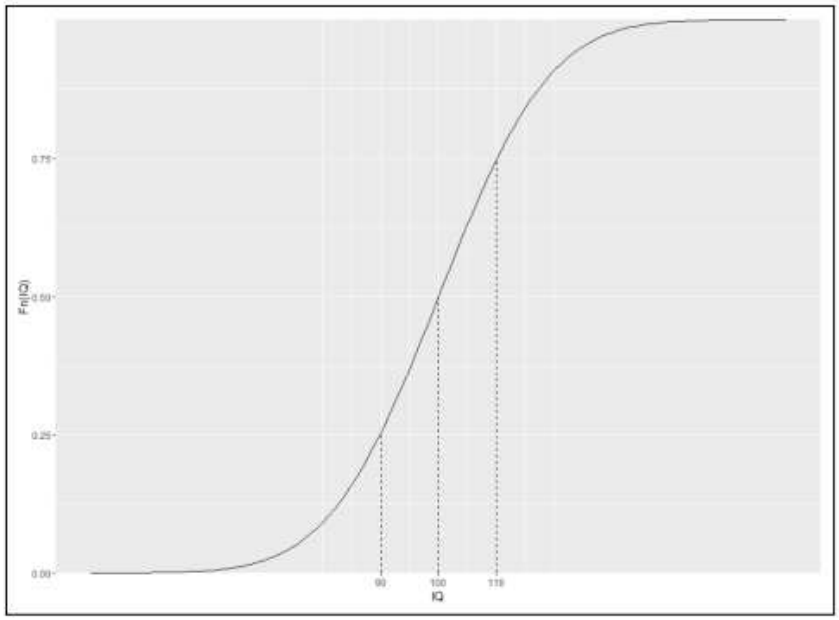
Kode menghasilkan Gambar 8.10.

Pengambilan sampel acak

Fungsi rnorm() menghasilkan angka acak dari distribusi normal. Berikut adalah lima angka acak dari distribusi IQ:

```
> rnorm(5,m=100,s=15)
[1] 127.02944  75.18125  66.49264 113.98305 103.39766
```

Inilah yang terjadi ketika Anda menjalankannya lagi:



Gambar 8.10 Fungsi kerapatan kumulatif normal dengan kuartil nilai-nilai

```
> rnorm(5,m=100,s=15)
[1] 73.73596 91.79841 82.33299 81.59029 73.40033
```

Ya, angkanya berbeda-beda. (Bahkan, ketika Anda menjalankan rnorm(), saya hampir dapat menjamin bahwa nomor Anda akan berbeda dari nomor saya.) Setiap kali Anda menjalankan fungsi tersebut, ia menghasilkan serangkaian nomor acak baru. Proses pengacakan dimulai dengan angka yang disebut seed. Jika Anda ingin mereproduksi hasil pengacakan, gunakan fungsi set.seed() untuk menyetel benih ke nomor tertentu sebelum mengacak:

```
> set.seed(7637060)
> rnorm(5,m=100,s=15)
[1]  71.99120  98.67231  92.68848 103.42207  99.61904
```

Jika Anda mengatur benih ke nomor yang sama saat Anda mengacak berikutnya, Anda mendapatkan hasil yang sama:

```
> set.seed(7637060)
> rnorm(5,m=100,s=15)
[1]  71.99120  98.67231  92.68848 103.42207  99.61904
```

Jika tidak, Anda tidak akan melakukannya.

Pengacakan adalah dasar dari simulasi, yang muncul di Bab selanjutnya. Ingatlah bahwa R (atau sebagian besar perangkat lunak lainnya) tidak menghasilkan angka acak

"benar". R menghasilkan angka "pseudo-random" yang cukup tidak terduga untuk sebagian besar tugas yang memerlukan pengacakan — seperti simulasi yang akan saya bahas nanti.

8.3 MENAMBAH KUMPULAN SKOR

Untuk membakukan sekumpulan skor sehingga Anda dapat membandingkannya dengan kumpulan skor lainnya, Anda mengubah masing-masing skor menjadi skor-z. (Saya membahas skor-z di Bab 6.) Rumus untuk mengubah skor menjadi skor-z (juga dikenal sebagai skor standar) adalah:

$$z = \frac{x - \mu}{\sigma}$$

Idenya adalah untuk menggunakan standar deviasi sebagai satuan ukuran. Misalnya, tes IQ versi Wechsler (antara lain) memiliki rata-rata 100 dan standar deviasi 15. Versi Stanford-Binet memiliki rata-rata 100 dan standar deviasi 16. Bagaimana skor Wechsler dari, katakanlah, 110, dibandingkan dengan skor Stanford-Binet 110?

Salah satu cara untuk menjawab pertanyaan ini adalah dengan menempatkan kedua versi pada level yang sama dengan menstandarkan kedua skor. Untuk Wechsler:

$$z = \frac{110 - 100}{15} = .667$$

Untuk Stanford-Binet:

$$z = \frac{110 - 100}{16} = .625$$

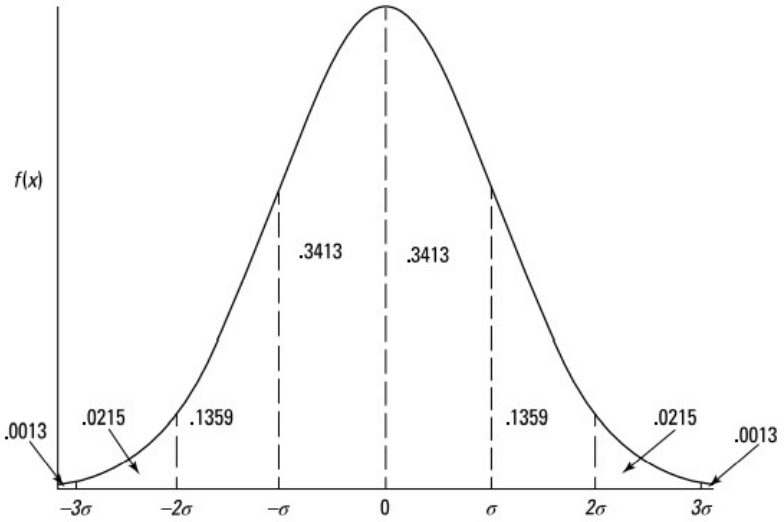
Jadi 110 di Wechsler adalah skor yang sedikit lebih tinggi daripada 110 di Stanford-Binet.

Sekarang, jika Anda membakukan semua skor dalam distribusi normal (seperti salah satu versi IQ), Anda memiliki distribusi skor-z yang normal. Setiap set skor-z (terdistribusi normal atau tidak) memiliki rata-rata 0 dan standar deviasi 1. Jika distribusi normal memiliki parameter tersebut, itu adalah distribusi normal standar — distribusi normal dari skor standar. persamaannya adalah:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{\left[\frac{-z^2}{2}\right]}$$

Gambar 8.11 menunjukkan distribusi normal standar. Sepertinya Gambar 8.2, kecuali bahwa saya telah mengganti 0 untuk mean dan saya telah memasukkan unit standar deviasi di tempat yang sesuai.

Ini adalah anggota keluarga distribusi normal yang dikenal kebanyakan orang. Itu yang paling mereka ingat dari kursus statistik, dan itu yang ada di benak kebanyakan orang ketika mereka (secara keliru) mengatakan distribusi normal. Itu juga yang orang pikirkan ketika mereka mendengar tentang "skor-z." Distribusi ini membawa banyak orang ke ide yang salah bahwa mengkonversi ke z-skor entah bagaimana mengubah satu set skor menjadi distribusi normal.



Gambar 8.11 Distribusi normal standar, dibagi dengan simpangan baku.

Distribusi normal standar dalam R

Bekerja dengan distribusi normal standar di R tidak bisa lebih mudah. Satu-satunya perubahan yang Anda buat pada empat fungsi norma adalah tidak menentukan mean dan standar deviasi — defaultnya adalah 0 dan 1.

Berikut beberapa contohnya:

```
> dnorm(0)
[1] 0.3989423
> pnorm(0)
[1] 0.5
> qnorm(c(.25,.50,.75))
[1] -0.6744898  0.0000000  0.6744898
> rnorm(5)
[1] -0.4280188 -0.9085506  0.6746574  1.0728058 -1.2646055
```

Ini juga berlaku untuk fungsi tigerstats:

```
> pnormGC(c(-1,0),region="between")
[1] 0.3413447
> qnormGC(.50, region = "below")
[1] 0
```

Merencanakan distribusi normal standar

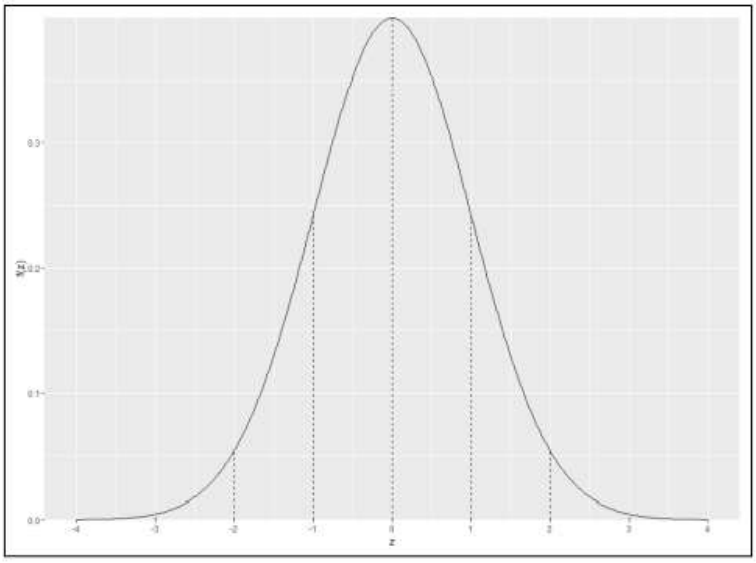
Untuk memplot distribusi normal standar, Anda membuat beberapa vektor baru

```
z.values <-seq(-4,4,.01)
z.sd.values <- seq(-4,4,1)
```

dan buat beberapa perubahan pada kode yang Anda gunakan sebelumnya untuk memplot distribusi IQ:

```
ggplot(NULL,aes(x=z.values,y=dnorm(z.values))) +
  geom_line() +
  labs(x="z",y="f(z)")+
  scale_x_continuous(breaks=z.sd.values,labels=z.sd.values) +
  geom_segment((aes(x=z.sd.values,y=zeros9,xend =
         z.sd.values,yend=dnorm(z.sd.values))),linetype =
         "dashed")+
  scale_y_continuous(expand=c(0,0))
```

Selain menempatkan vektor baru ke dalam scale_x_continuous() dan geom_ segment(), perubahan penting adalah menghilangkan argumen mean dan standar deviasi dari dnorm(). Kode menciptakan Gambar 8.12.



Gambar 8.12 Distribusi normal standar, dibagi dengan simpangan baku dan diplot dalam ggplot().

Saya serahkan kepada Anda sebagai latihan untuk memplot fungsi kepadatan kumulatif untuk distribusi normal standar.

BAGIAN 3
MENARIK KESIMPULAN DARI DATA
BAB 9
ESTIMASI

“Populasi" dan "sampel" adalah konsep yang cukup mudah dipahami. Populasi adalah kumpulan individu yang sangat besar, dan sampel adalah sekelompok individu yang Anda ambil dari suatu populasi. Ukur anggota sampel pada beberapa sifat atau atribut, hitung statistik yang merangkum sampel, dan Anda siap. Selain statistik ringkasan tersebut, Anda dapat menggunakan statistik untuk memperkirakan parameter populasi. Ini adalah masalah besar: Hanya berdasarkan persentase kecil individu dari populasi, Anda dapat menggambar seluruh populasi.

Seberapa definitif gambaran itu? Dengan kata lain, seberapa besar keyakinan Anda terhadap perkiraan Anda? Untuk menjawab pertanyaan ini, Anda harus memiliki konteks untuk perkiraan Anda. Seberapa besar kemungkinan mereka? Seberapa besar kemungkinan nilai sebenarnya dari suatu parameter berada dalam batas bawah dan batas atas tertentu? Dalam bab ini, saya memperkenalkan konteks untuk perkiraan, menunjukkan bagaimana konteks itu berperan dalam keyakinan dalam perkiraan tersebut, dan menunjukkan kepada Anda bagaimana menggunakan R untuk menghitung tingkat kepercayaan.

9.1 MEMAHAMI DISTRIBUSI SAMPLING

Jadi Anda memiliki populasi, dan Anda mengambil sampel dari populasi ini. Anda mengukur anggota sampel pada beberapa atribut dan menghitung rata-rata sampel. Kembalikan anggota sampel ke populasi. Gambarlah sampel lain, nilai anggota sampel baru, dan kemudian hitung rata-ratanya. Ulangi proses ini lagi dan lagi, selalu dengan jumlah individu yang sama seperti pada sampel asli. Jika Anda dapat melakukan ini dalam jumlah tak terbatas (dengan ukuran sampel yang sama setiap kali), Anda akan memiliki jumlah sarana yang tak terbatas. Sampel tersebut berarti membentuk distribusi mereka sendiri. Distribusi ini disebut distribusi sampling dari mean.

Sebagai contoh rata-rata, ini adalah "konteks" yang saya sebutkan di awal bab ini. Seperti nomor lainnya, statistik tidak masuk akal dengan sendirinya. Anda harus tahu dari mana asalnya untuk memahaminya. Tentu saja, statistik berasal dari perhitungan yang dilakukan pada data sampel. Dalam pengertian lain, statistik adalah bagian dari distribusi sampling.

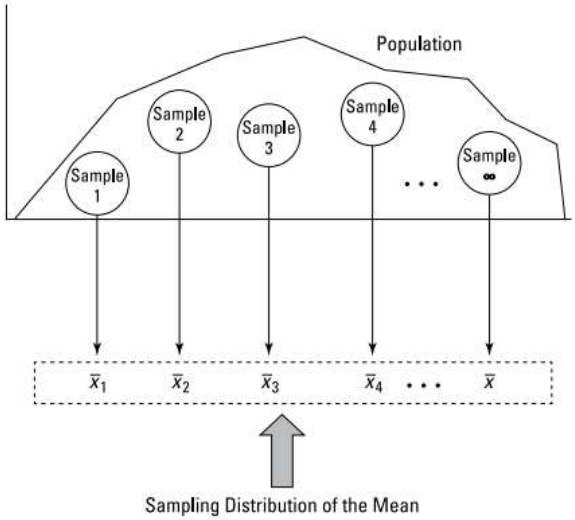
Secara umum, distribusi sampling adalah distribusi semua nilai statistik yang mungkin untuk ukuran sampel tertentu. Saya telah memiringkan definisi karena suatu alasan: Ini sangat penting. Setelah bertahun-tahun mengajar statistika, saya dapat memberi tahu Anda bahwa konsep ini biasanya menetapkan garis batas antara orang yang memahami statistik dan orang yang tidak.

Jadi . . . jika Anda memahami apa itu distribusi sampling, Anda akan memahami apa itu bidang statistik. Jika tidak, Anda tidak akan melakukannya. Ini hampir sesederhana itu.

Jika Anda tidak tahu apa itu distribusi sampling, statistik akan menjadi jenis subjek buku masak untuk Anda: Kapan pun Anda harus menerapkan statistik, Anda akan menemukan diri Anda memasukkan angka ke dalam rumus dan berharap yang terbaik. Di sisi lain, jika Anda merasa nyaman dengan ide distribusi sampling, Anda akan memahami gambaran besar statistik inferensial.

Untuk membantu memperjelas ide distribusi sampling, lihat Gambar 9-1. Ini merangkum langkah-langkah dalam menciptakan distribusi sampling mean. Distribusi sampling — seperti kelompok skor lainnya — memiliki mean dan standar deviasi. Simbol mean dari distribusi sampling mean (ya, saya tahu itu seteguk) adalah $\mu_{\bar{X}} = \mu$.

Standar deviasi dari distribusi sampling adalah item yang cukup panas. Ini memiliki nama khusus: kesalahan standar. Untuk distribusi sampling rata-rata, standar deviasi disebut kesalahan standar rata-rata. Simbolnya adalah $\sigma_{\bar{X}} = {}^{\sigma}/_{\sqrt{N}}$.



Gambar 9.1: Membuat distribusi sampling dari mean.

9.2 IDE YANG SANGAT PENTING: TEOREMA LIMIT PUSAT

Situasi yang saya minta Anda bayangkan tidak pernah terjadi di dunia nyata. Anda tidak pernah mengambil sampel dalam jumlah tak terbatas dan menghitung rata-ratanya, dan Anda tidak pernah benar-benar membuat distribusi sampel rata-rata. Biasanya, Anda menggambar satu sampel dan menghitung statistiknya. Jadi jika Anda hanya memiliki satu sampel, bagaimana Anda bisa mengetahui sesuatu tentang distribusi pengambilan sampel — distribusi teoretis yang mencakup jumlah sampel yang tak terbatas? Apakah ini semua hanya pengejaran angsa liar? Tidak. Anda dapat mengetahui banyak tentang distribusi sampling karena hadiah besar dari matematikawan ke bidang statistik: teorema limit pusat.

Menurut teorema limit pusat:

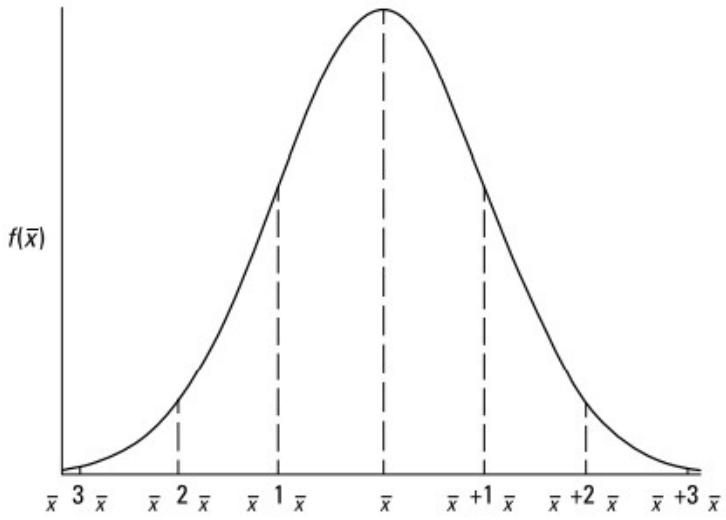
- Distribusi sampling rata-rata mendekati distribusi normal jika ukuran sampel cukup besar.

Cukup besar berarti sekitar 30 atau lebih.

- Rerata distribusi sampling dari mean sama dengan mean populasi.
  Dalam bentuk persamaan, yaitu
  $\mu_{\bar{X}} = \mu$.
- Standar deviasi dari distribusi sampling rata-rata (juga dikenal sebagai kesalahan standar mean) sama dengan standar deviasi populasi dibagi dengan akar kuadrat dari ukuran sampel.
  Persamaan untuk kesalahan standar rata-rata adalah
  $\sigma_{\bar{X}} = {}^{\sigma}/_{\sqrt{N}}$.

Perhatikan bahwa teorema limit pusat tidak mengatakan apa-apa tentang populasi. Yang dikatakan adalah bahwa jika ukuran sampel cukup besar, distribusi sampling rata-rata adalah distribusi normal, dengan parameter yang ditunjukkan. Populasi yang menyediakan sampel tidak harus berdistribusi normal agar teorema limit pusat dapat dipegang. Bagaimana jika populasi berdistribusi normal? Dalam hal ini, distribusi sampling dari mean adalah distribusi normal, terlepas dari ukuran sampel.

Gambar 9.2 menunjukkan gambaran umum dari distribusi sampling rata-rata, dipartisi menjadi unit kesalahan standar.



Gambar 9.2 Distribusi sampling dari mean, dipartisi.

(Perkiraan) Mensimulasikan teorema limit pusat

Kedengarannya hampir tidak benar: Bagaimana populasi yang tidak terdistribusi normal dapat menghasilkan distribusi sampling yang terdistribusi normal? Untuk memberi Anda gambaran tentang cara kerja teorema limit pusat, saya memandu Anda melalui simulasi. Simulasi ini menciptakan sesuatu seperti distribusi sampling rata-rata untuk sampel yang sangat kecil, berdasarkan populasi yang tidak terdistribusi secara normal. Seperti yang akan Anda lihat, meskipun populasinya bukan distribusi normal, dan meskipun sampelnya kecil, distribusi sampling rata-rata terlihat agak mirip dengan distribusi normal.

Bayangkan sebuah populasi besar yang hanya terdiri dari tiga skor — 1, 2, dan 3, dan masing-masing memiliki kemungkinan yang sama untuk muncul dalam sampel. Populasi seperti itu jelas bukan distribusi normal. Bayangkan juga bahwa Anda dapat secara acak memilih sampel tiga skor dari populasi ini. Tabel 9.1 menunjukkan semua sampel yang mungkin dan artinya.

Tabel 9.1 SEMUA Kemungkinan Sampel Tiga Skor (dan Mean Mereka) dari Populasi yang Terdiri dari Skor 1, 2, dan 3

Sample	Mean	Sample	Mean	Sample	Mean
1,1,1	1.00	2,1,1	1.33	3,1,1	1.67
1,1,2	1.33	2,1,2	1.67	3,1,2	2.00
1,1,3	1.67	2,1,3	2.00	3,1,3	2.33
1,2,1	1.33	2,2,1	1.67	3,2,1	2.00
1,2,2	1.67	2,2,2	2.00	3,2,2	2.33
1,2,3	2.00	2,2,3	2.33	3,2,3	2.67
1,3,1	1.67	2,3,1	2.00	3,3,1	2.33
1,3,2	2.00	2,3,2	2.33	3,3,2	2.67
1,3,3	2.33	2,3,3	2.67	3,3,3	3.00

Jika Anda melihat lebih dekat pada tabel, Anda hampir dapat melihat apa yang akan terjadi dalam simulasi. Rata-rata sampel yang paling sering muncul adalah 2,00. Berarti sampel yang paling jarang muncul adalah 1,00 dan 3,00. Hmmm. . . .

Dalam simulasi, Anda secara acak memilih skor dari populasi dan kemudian secara acak memilih dua lagi. Kelompok tiga skor itu adalah sampel. Kemudian Anda menghitung rata-rata sampel itu. Anda mengulangi proses ini untuk total 600 sampel, menghasilkan 600 rata-rata sampel. Akhirnya, Anda membuat grafik distribusi rata-rata sampel. Seperti apa distribusi sampling yang disimulasikan dari mean? Saya memandu Anda melaluinya di R. Anda mulai dengan membuat vektor untuk skor yang mungkin, dan satu lagi untuk probabilitas pengambilan sampel setiap skor:

```
values <- c(1,2,3)
probabilities <- c(1/3,1/3,1/3)
```

Satu vektor lagi akan menampung 600 sampel mean:

```
smpl.means <- NULL
```

Untuk menggambar sampel, Anda menggunakan fungsi sample() :

```
smpl <-sample(x=values,prob = probabilities,
  size=3,replace=TRUE)
```

Dua argumen pertama, tentu saja, memberikan skor untuk sampel dan probabilitas setiap skor. Yang ketiga adalah ukuran sampel. Yang keempat menunjukkan bahwa setelah Anda memilih skor untuk sampel, Anda menggantinya. (Dengan kata lain, Anda memasukkannya

kembali ke dalam populasi.) Prosedur ini (tidak mengherankan disebut "pengambilan sampel dengan penggantian") mensimulasikan populasi yang sangat besar dari mana Anda dapat memilih skor apa pun kapan saja.

Setiap kali Anda menggambar sampel, Anda mengambil meannya dan menambahkannya (tambahkan ke akhir) vektor smpl.means:

```
smpl.means <- append(smpl.means, mean(smpl))
```

Saya tidak ingin Anda harus mengulangi seluruh proses ini secara manual 600 kali. Untungnya, seperti semua bahasa komputer, R memiliki cara untuk menangani ini: for-loopnya melakukan semua pekerjaan. Untuk melakukan sampling, kalkulasi, dan penambahan 600 kali, for-loop terlihat seperti ini:

```
for(i in 1:600){
    smpl <-sample(x = values,prob = probabilities,
                  size = 3,replace=TRUE)
    smpl.means <- append(smpl.means, mean(smpl))
      }
```

Seperti yang Anda lihat, tanda kurung kurawal mengapit apa yang terjadi di setiap iterasi perulangan, dan i adalah penghitung untuk berapa kali perulangan terjadi.

Jika Anda ingin menjalankan ini, berikut semua kode yang mendahului for-loop, termasuk seed sehingga Anda dapat mereplikasi hasil saya:

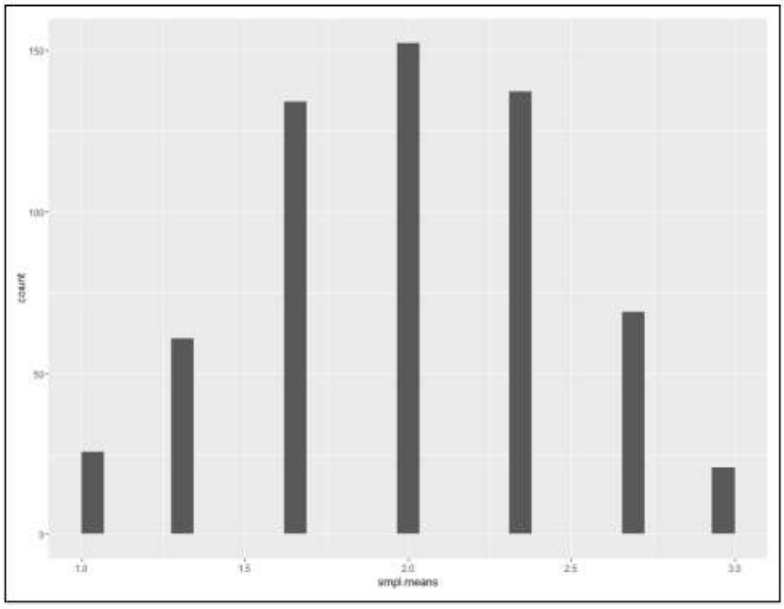
```
> values <- c(1,2,3)
> probabilities <- c(1/3,1/3,1/3)
> smpl.means <- NULL
> set.seed(7637060)
```

Kemudian jalankan for-loop. Jika Anda ingin menjalankan loop berulang kali, pastikan Anda mereset smpl.means ke NULL setiap kali. Jika Anda ingin mendapatkan hasil yang berbeda setiap kali, jangan set benih ke nomor yang sama (atau jangan setel sama sekali).

Seperti apa distribusi samplingnya? Gunakan ggplot() untuk melakukan penghargaan. Nilai data (rata-rata 600 sampel) berada dalam vektor, jadi argumen pertama adalah NULL. smpl.means peta vektor ke sumbu x. Dan Anda membuat histogram, jadi fungsi geomnya adalah geom_histogram():

```
ggplot(NULL,aes(x=smpl.means)) +
  geom_histogram()
```

Gambar 9.3 menunjukkan histogram untuk distribusi sampling mean.



Gambar 9.3 Distribusi sampling mean berdasarkan 600 sampel ukuran 3 dari suatu populasi yang terdiri dari skor kemungkinan yang sama 1, 2, dan 3.

Terlihat sangat mirip dengan awal dari distribusi normal, bukan? Saya akan menjelajahi distribusi lebih jauh dalam beberapa saat, tetapi pertama-tama saya akan menunjukkan kepada Anda bagaimana membuat grafik sedikit lebih informatif. Misalkan Anda ingin titik berlabel pada sumbu x mencerminkan nilai rata-rata dalam vektor smpl.means. Anda tidak bisa hanya menentukan nilai vektor untuk sumbu x, karena vektor memiliki 600 nilai. Sebagai gantinya, Anda mencantumkan nilai unik:

```
> unique(smpl.means)
[1] 2.333333 1.666667 1.333333 2.000000 2.666667 3.000000
[7] 1.000000
```

Mereka terlihat lebih baik jika Anda membulatkannya menjadi dua tempat desimal:

```
> round(unique(smpl.means),2)
[1] 2.33 1.67 1.33 2.00 2.67 3.00 1.00
```

Terakhir, Anda menyimpan nilai-nilai ini dalam vektor yang disebut m.values, yang akan Anda gunakan untuk mengubah skala sumbu x:

```
> m.values <-round(unique(smpl.means),2)
```

Untuk rescaling, gunakan trik yang saya tunjukkan di Bab 8:

```
scale_x_continuous(breaks=m.values,label=m.values)
```

Trik lain dari Bab 8 menghilangkan ruang antara nilai sumbu x dan sumbu x:

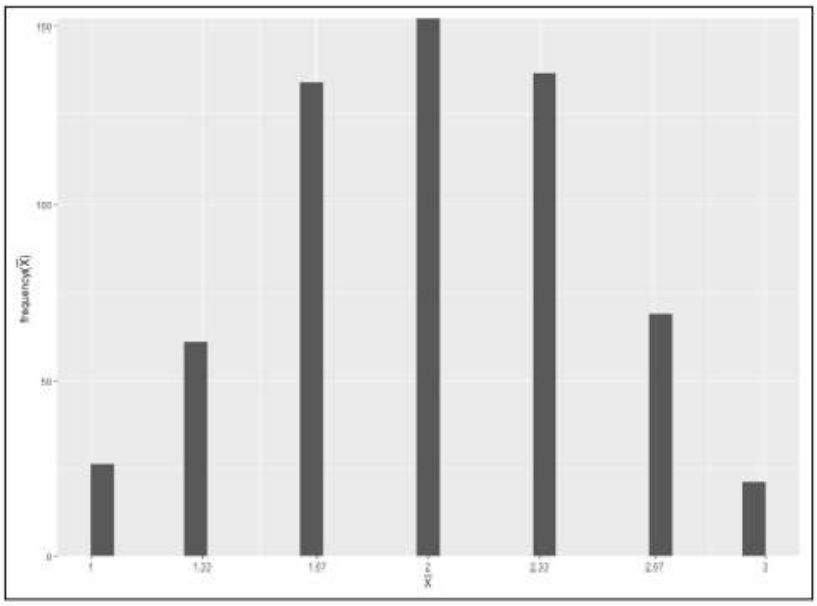
```
scale_y_continuous(expand = c(0,0))
```

Satu trik lagi menggunakan sintaks ekspresi R untuk menampilkan $\bar{X}$ sebagai label sumbu x dan frekuensi $\bar{X}$ sebagai label sumbu y:

```
labs(x=expression(bar(X)),y=expression(frequency(bar(X))))
```

Menempatkan semuanya bersama-sama memberikan distribusi sampling pada Gambar 9.4:

```
ggplot(NULL,aes(x=smpl.means)) +
  geom_histogram()+
  scale_x_continuous(breaks=m.values,label=m.values)+
  scale_y_continuous(expand = c(0,0)) +

          labs(x=expression(bar(X)),y=expression
          (frequency(bar(X))))
```



GAMBAR 9.4 Distribusi sampling rata-rata dengan sumbu-x yang diskalakan ulang dan label sumbu dingin.

Prediksi teorema limit pusat

Bagaimana karakteristik distribusi sampling cocok dengan apa yang diprediksi oleh teorema limit pusat? Untuk mendapatkan prediksi, Anda harus mulai dengan populasi. Pikirkan setiap nilai populasi (1, 2, atau 3) sebagai X, dan pikirkan setiap probabilitas sebagai pr(X). Matematikawan akan mengacu pada X sebagai variabel acak diskrit.

Rata-rata dari variabel acak diskrit disebut nilai harapannya. Notasi untuk nilai harapan dari X adalah E(X). Untuk menemukan E(X), Anda mengalikan setiap X dengan probabilitasnya dan kemudian menjumlahkan semua produk tersebut. Untuk contoh ini, itu:

$$E(X)=\sum X(pr(X))=1\left(\frac{1}{3}\right)+2\left(\frac{1}{3}\right)+3\left(\frac{1}{3}\right)=2$$

Atau, jika Anda lebih suka R:

```
> E.values<-sum(values*probabilities)
> E.values
[1] 2
```

Untuk menemukan varians X, kurangi E(X) dari setiap X, kuadratkan setiap deviasi, kalikan setiap kuadrat deviasi dengan probabilitas X, dan jumlahkan produknya. Untuk contoh ini:

$$\text{var}(X) = \sum (X - E(X))^2 pr(x) = (1-2)^2\left(\frac{1}{3}\right) + (2-2)^2\left(\frac{1}{3}\right) + (3-2)^2\left(\frac{1}{3}\right) = .67$$

Di R:

```
> var.values <- sum((values-E.values)^2*probabilities)
> var.values
[1] 0.6666667
```

Seperti biasa, deviasi standar adalah akar kuadrat dari varians:

$$\sigma = \sqrt{\text{var}(X)} = \sqrt{.67} = .82$$

Sekali lagi, di R:

```
> sd.values<-sqrt(var.values)
> sd.values
[1] 0.8164966
```

Jadi populasi memiliki mean 2 dan standar deviasi 0,82.
Menurut teorema limit pusat, rata-rata distribusi sampling harus:

$$\mu_{\bar{x}} = \mu = 2$$

dan simpangan bakunya adalah:

$$\sigma_{\bar{x}} = \sigma / \sqrt{N} = .82 / \sqrt{3} = .4714$$

Bagaimana nilai prediksi ini cocok dengan karakteristik distribusi sampling?

```
> mean(smpl.means)
[1] 2.002222
> sd(smpl.means)
[1] 0.4745368
```

Cukup dekat! Bahkan dengan populasi yang tidak terdistribusi normal dan ukuran sampel yang kecil, teorema limit pusat memberikan gambaran yang akurat tentang distribusi sampling dari mean.

9.3 KEYAKINAN: ITU ADA BATASNYA!

Saya memberi tahu Anda tentang distribusi sampling karena mereka membantu menjawab pertanyaan yang saya ajukan di awal bab ini: Seberapa besar keyakinan Anda terhadap estimasi yang Anda buat?

Prosedurnya adalah menghitung statistik dan kemudian menggunakan statistik itu untuk menetapkan batas atas dan bawah untuk parameter populasi dengan, katakanlah, 95 persen keyakinan. (Penafsiran batas kepercayaan sedikit lebih terlibat dari itu, seperti yang

akan Anda lihat.) Anda dapat melakukan ini hanya jika Anda mengetahui distribusi sampling statistik dan kesalahan standar statistik. Di bagian berikutnya, saya menunjukkan bagaimana melakukan ini untuk mean.

Menemukan batas kepercayaan untuk mean

FarBlonJet Corporation memproduksi sistem navigasi. (Motto perusahaan: "Melakukan perjalanan? Dapatkan FarBlonJet.") Perusahaan telah mengembangkan baterai baru untuk memberi daya pada model portabelnya. Untuk membantu memasarkan sistem ini, FarBlonJet ingin mengetahui berapa lama, rata-rata, setiap baterai bertahan sebelum habis.

Karyawan FarBlonJet suka memperkirakan rata-rata itu dengan keyakinan 95 persen. Mereka menguji sampel 100 baterai dan menemukan bahwa rata-rata sampel adalah 60 jam, dengan standar deviasi 20 jam. Teorema limit pusat, ingat, mengatakan bahwa dengan sampel yang cukup besar (30 atau lebih), distribusi sampling rata-rata mendekati distribusi normal. Kesalahan standar rata-rata (standar deviasi dari distribusi sampling rata-rata) adalah:

$$\sigma_{\bar{x}} = \sigma / \sqrt{N}$$

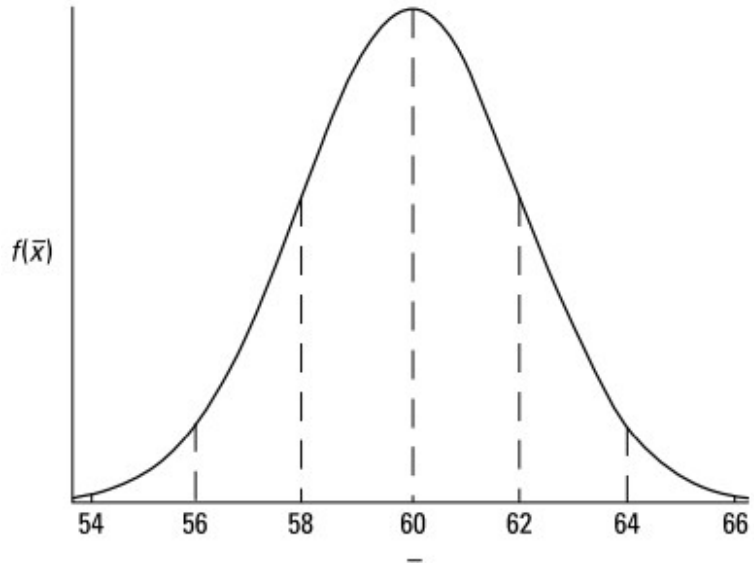
Ukuran sampel, N, adalah 100. Bagaimana dengan ? Itu tidak diketahui, jadi Anda harus memperkirakannya. Jika Anda tahu , itu berarti Anda tahu , dan menetapkan batas kepercayaan tidak diperlukan.

Estimasi terbaik dari adalah standar deviasi sampel. Dalam hal ini, itu adalah 20. Ini mengarah pada perkiraan kesalahan standar rata-rata.

$$s_{\bar{x}} = s / \sqrt{N} = 20 / \sqrt{100} = 20 / 10 = 2$$

Estimasi terbaik dari mean populasi adalah mean sampel: 60. Berbekal informasi ini — estimasi mean, estimasi standar error mean, distribusi normal — Anda dapat membayangkan distribusi sampling mean, yang ditunjukkan pada Gambar 9.5 . Konsisten dengan Gambar 9.2, setiap standar deviasi adalah kesalahan standar mean.

Sekarang setelah Anda memiliki distribusi sampling, Anda dapat menetapkan batas kepercayaan 95 persen untuk mean. Mulai dari pusat distribusi, seberapa jauh ke samping yang harus Anda perpanjang sampai Anda memiliki 95 persen luas di bawah kurva? (Untuk lebih lanjut tentang area di bawah distribusi normal dan apa artinya, lihat Bab 8).



Gambar 9.5 Distribusi sampling rata-rata untuk baterai FarBlonJet.

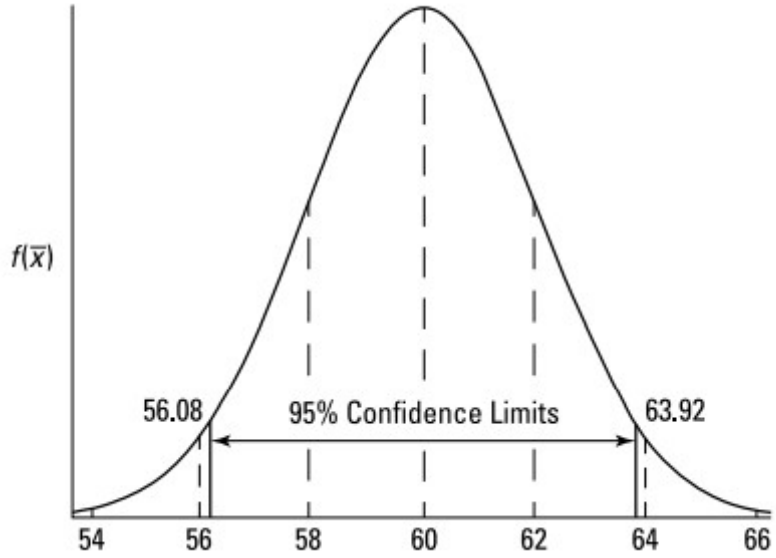
Salah satu cara untuk menjawab pertanyaan ini adalah dengan bekerja dengan distribusi normal standar dan menemukan skor-z yang memotong 2,5 persen dari area di ekor atas. Kemudian kalikan skor-z itu dengan kesalahan standar. Tambahkan hasilnya ke mean sampel untuk mendapatkan batas kepercayaan atas; kurangi hasil dari rata-rata untuk mendapatkan batas kepercayaan bawah.
Inilah cara melakukan semua itu di R. Pertama, pengaturannya:

```
> mean.battery <- 60
> sd.battery <- 20
> N <- 100
> error <- qnorm(.025,lower.tail=FALSE)*sd.battery/sqrt(N)
```

Maka batasannya:

```
> lower <- mean.battery - error
> upper <- mean.battery + error
> lower
[1] 56.08007
> upper
[1] 63.91993
```

Gambar 9.6 menunjukkan batas-batas ini pada distribusi sampling.



Gambar 9.6 Batas kepercayaan 95 persen pada FarBlonJet distribusi sampel.

Apa ini memberitahu Anda, tepatnya? Salah satu interpretasinya adalah jika Anda mengulangi prosedur pengambilan sampel dan estimasi ini berkali-kali, interval kepercayaan yang Anda hitung (yang akan berbeda setiap kali Anda melakukannya) akan mencakup rata-rata populasi 95 persen dari waktu.

Cocok untuk t

Teorema limit pusat menentukan (kurang lebih) distribusi normal untuk sampel besar. Di dunia nyata, bagaimanapun, Anda berurusan dengan sampel yang lebih kecil, dan distribusi normal tidak sesuai. Apa yang kamu kerjakan?

Pertama-tama, Anda membayar harga untuk menggunakan sampel yang lebih kecil — Anda memiliki kesalahan standar yang lebih besar. Misalkan FarBlonJet Corporation menemukan rata-rata 60 dan standar deviasi 20 dalam sampel 25 baterai. Estimasi kesalahan standar adalah

$$s_{\bar{x}} = s/\sqrt{N} = 20/\sqrt{25} = 20/5 = 4$$

yang dua kali lebih besar dari kesalahan standar untuk N=100.

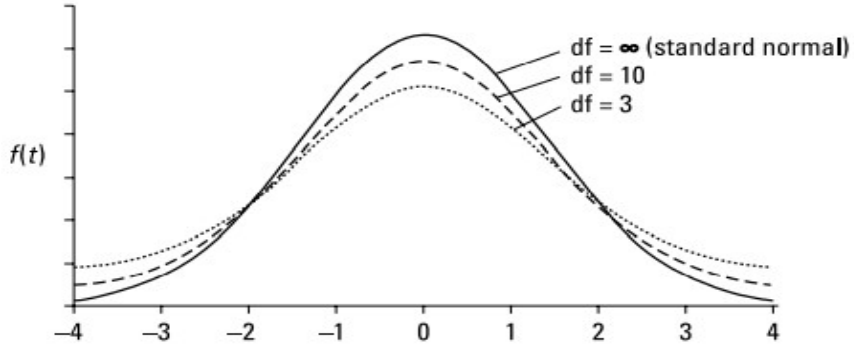
Kedua, Anda tidak bisa menggunakan distribusi normal standar untuk mengkarakterisasi distribusi sampling mean. Untuk sampel kecil, distribusi sampling dari mean adalah anggota dari keluarga distribusi yang disebut distribusi-t. Parameter yang membedakan anggota keluarga ini satu sama lain disebut derajat kebebasan.

Seperti yang saya katakan di Bab 5, pikirkan "derajat kebebasan" sebagai penyebut estimasi varians Anda. Misalnya, jika sampel Anda terdiri dari 25 individu, varians sampel yang memperkirakan varians populasi adalah:

$$s^2 = \frac{\sum(x-\bar{x})^2}{N-1} = \frac{\sum(x-\bar{x})^2}{25-1} = \frac{\sum(x-\bar{x})^2}{24}$$

Bilangan pada penyebut adalah 24, dan itulah nilai parameter derajat kebebasan. Secara umum, derajat kebebasan (df) = N-1 (N adalah ukuran sampel) ketika Anda menggunakan distribusi-t seperti yang saya tunjukkan di bagian ini.

Gambar 9.7 menunjukkan dua anggota keluarga distribusi-t (df = 3 dan df = 10), bersama dengan distribusi normal untuk perbandingan. Seperti yang ditunjukkan gambar, semakin besar df, semakin dekat t mendekati distribusi normal.



Gambar 9.7 Beberapa anggota keluarga distribusi-t.

Untuk menentukan batas bawah dan batas atas untuk tingkat kepercayaan 95 persen untuk sampel kecil, bekerjalah dengan anggota keluarga distribusi-t yang memiliki df yang sesuai. Temukan nilai yang memotong 2,5 persen atas area di ekor atas distribusi. Kemudian kalikan nilai itu dengan kesalahan standar. Tambahkan hasilnya ke mean untuk mendapatkan batas kepercayaan atas; kurangi hasil dari rata-rata untuk mendapatkan batas kepercayaan bawah.

R menyediakan dt() (fungsi kepadatan), pt() (fungsi kepadatan kumulatif), qt() (kuantil), dan rt() (pembuatan angka acak) untuk bekerja dengan distribusi-t. Untuk interval kepercayaan, saya menggunakan qt().

Dalam contoh baterai FarBlonJet:

```
> mean.battery <- 60
> sd.battery <- 20
> N <- 25
> error <- qt(.025,N-1,lower.tail=FALSE)*sd.battery/sqrt(N)
> lower <- mean.battery - error
> upper <- mean.battery + error
> lower
[1] 51.74441
> upper
[1] 68.25559
```

Batas bawah dan atas adalah 51,74 dan 68,26. Perhatikan bahwa dengan sampel yang lebih kecil, jangkauannya lebih lebar dari pada contoh sebelumnya. Jika Anda memiliki data mentah, Anda dapat menggunakan t.test() untuk menghasilkan interval kepercayaan:

```
> battery.data <- c(82,64,68,44,54,47,50,85,51,41,61,84,
          53,83,91,43,35,36,33,87,90,86,49,37,48)
```

Berikut ini cara menggunakan t.test() untuk menghasilkan batas bawah dan atas untuk kepercayaan 90 persen — nilai defaultnya adalah 0,95:

```
> t.test(battery.data, conf.level=.90)

            One Sample t-test

data:  c(82, 64, 68, 44, 54, 47, 50, 85, 51, 41, 61, 84,
         53, 83, 91,  ...
t = 15, df = 24, p-value = 1.086e-13
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 53.22727 66.93273
sample estimates:
mean of x
    60.08
```

Fungsi t.test() benar-benar lebih sesuai untuk bab berikutnya.

BAB 10
PENGUJIAN HIPOTESIS SATU SAMPEL

Apa pun pekerjaan Anda, Anda sering kali harus menilai apakah sesuatu yang baru dan berbeda telah terjadi. Kadang-kadang Anda mulai dengan populasi yang Anda tahu banyak (seperti mean dan simpangan bakunya) dan Anda mengambil sampel. Apakah sampel itu seperti populasi lainnya, atau apakah itu mewakili sesuatu yang luar biasa?

Untuk menjawab pertanyaan itu, Anda mengukur setiap individu dalam sampel dan menghitung statistik sampel. Kemudian Anda membandingkan statistik tersebut dengan parameter populasi. Apakah mereka sama? Apakah mereka berbeda? Apakah sampelnya luar biasa dalam beberapa hal? Penggunaan statistik yang tepat membantu Anda membuat keputusan. Namun, terkadang Anda tidak mengetahui parameter populasi dari mana sampel itu berasal. Apa yang terjadi kemudian? Dalam bab ini, saya membahas teknik statistik dan fungsi R untuk menangani kedua kasus.

10.1 HIPOTESIS, PENGUJIAN, DAN KESALAHAN

Hipotesis adalah tebakan tentang cara dunia bekerja. Ini adalah penjelasan tentatif dari beberapa proses, apakah proses itu terjadi di alam atau di laboratorium.

Sebelum mempelajari dan mengukur individu dalam sampel, seorang peneliti merumuskan hipotesis yang memprediksi seperti apa data itu. Umumnya, satu hipotesis memprediksi bahwa data tidak akan menunjukkan sesuatu yang baru atau luar biasa. Ini disebut hipotesis nol (disingkat H0). Menurut hipotesis nol, jika data menyimpang dari norma dengan cara apa pun, penyimpangan itu semata-mata karena kebetulan. Hipotesis lain, hipotesis alternatif (disingkat H1), menjelaskan hal-hal secara berbeda. Menurut hipotesis alternatif, data menunjukkan sesuatu yang penting.

Setelah mengumpulkan data, terserah peneliti untuk membuat keputusan. Cara logika bekerja, keputusan berpusat di sekitar hipotesis nol. Peneliti harus memutuskan untuk menolak hipotesis nol atau tidak menolak hipotesis nol.

Dalam pengujian hipotesis, Anda

- Merumuskan hipotesis nol dan alternatif
- Kumpulkan data
- Putuskan apakah akan menolak atau tidak menolak hipotesis nol.

Tidak ada logika yang melibatkan penerimaan salah satu hipotesis. Logikanya juga tidak melibatkan pengambilan keputusan tentang hipotesis alternatif. Ini semua tentang menolak atau tidak menolak H0.

Terlepas dari keputusan tolak-jangan-tolak, kesalahan mungkin terjadi. Salah satu jenis kesalahan terjadi ketika Anda yakin bahwa data menunjukkan sesuatu yang penting dan Anda menolak H0, tetapi pada kenyataannya data tersebut hanya karena kebetulan. Ini disebut kesalahan Tipe I. Pada awal penelitian, Anda menetapkan kriteria untuk menolak H0. Dengan

melakukan itu, Anda mengatur kemungkinan kesalahan Tipe I. Probabilitas ini disebut alpha (α).

Jenis kesalahan lainnya terjadi ketika Anda tidak menolak H0 dan data tersebut benar-benar disebabkan oleh sesuatu yang tidak biasa. Untuk satu dan lain alasan, Anda kebetulan melewatkannya. Ini disebut kesalahan Tipe II. Probabilitasnya disebut beta (β). Tabel 10-1 merangkum kemungkinan keputusan dan kesalahan.

Tabel 10.1 Keputusan dan Kesalahan dalam Pengujian Hipotesis

		"Keadaan Sejati" Dunia	
		H0 Benar	H1 Benar
	Tolak H_0	Kesalahan Tipe I	Keputusan yang Benar
Keputusan			
	Jangan Tolak H_0	Keputusan yang Benar	Kesalahan Tipe II

Perhatikan bahwa Anda tidak pernah tahu keadaan dunia yang sebenarnya. (Jika ya, tidak perlu melakukan penelitian!) Yang dapat Anda lakukan hanyalah mengukur individu dalam sampel, menghitung statistik, dan membuat keputusan tentang H0. (Saya membahas hipotesis dan pengujian hipotesis di Bab 1.)

10.2 UJI HIPOTESIS DAN DISTRIBUSI SAMPLING

Dalam Bab 9, saya membahas distribusi sampling. Ingat, distribusi sampling adalah himpunan semua nilai statistik yang mungkin untuk ukuran sampel tertentu.

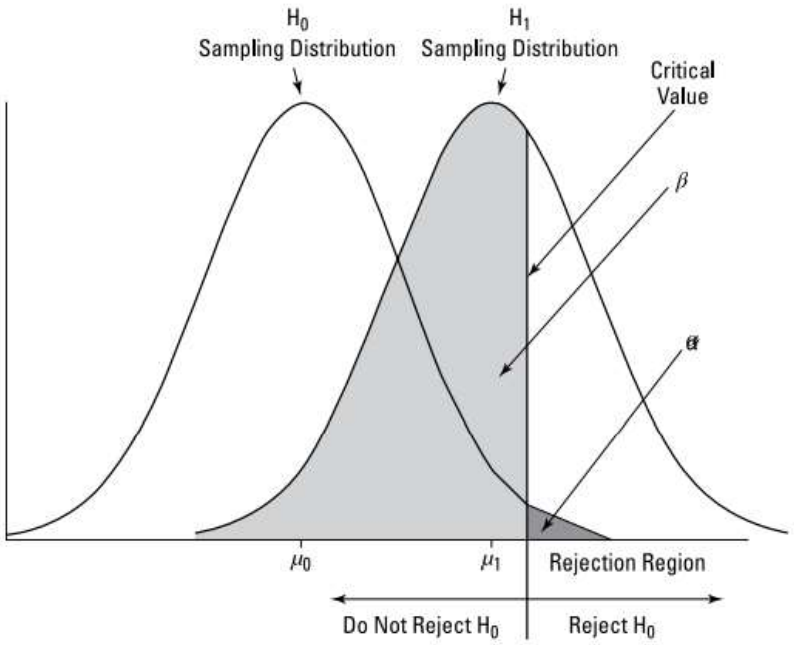
Juga dalam Bab 9, saya membahas teorema limit pusat. Teorema ini memberitahu Anda bahwa distribusi sampling rata-rata mendekati distribusi normal jika ukuran sampel besar (untuk tujuan praktis, setidaknya 30). Ini berfungsi apakah populasi terdistribusi normal atau tidak. Jika populasinya berdistribusi normal, maka distribusi samplingnya normal untuk semua ukuran sampel. Berikut adalah dua poin lain dari teorema limit pusat:

- Rata-rata distribusi sampling rata-rata sama dengan rata-rata populasi.
  Persamaan untuk ini adalah
  $\mu_{\bar{X}} = \mu$
- Kesalahan standar rata-rata (standar deviasi dari distribusi sampling) sama dengan standar deviasi populasi dibagi dengan akar kuadrat dari ukuran sampel.
  Persamaan ini adalah
  $\sigma_{\bar{X}} = {}^{\sigma}/_{\sqrt{N}}$

Distribusi sampling dari angka rata-rata menonjol ke dalam jenis pengujian hipotesis yang saya bahas dalam bab ini. Secara teoritis, ketika Anda menguji hipotesis nol versus hipotesis alternatif, setiap hipotesis sesuai dengan distribusi pengambilan sampel yang terpisah.

Gambar 10.1 menunjukkan apa yang saya maksud. Gambar tersebut menunjukkan dua distribusi normal. Saya menempatkan mereka secara sewenang-wenang. Setiap distribusi

normal mewakili distribusi sampling dari mean. Yang di sebelah kiri mewakili distribusi sampel yang mungkin berarti jika hipotesis nol benar-benar bagaimana dunia bekerja. Yang di sebelah kanan mewakili distribusi kemungkinan sampel berarti jika hipotesis alternatif benar-benar bagaimana dunia bekerja.



Gambar 10.1 H_0 dan H_1 masing-masing sesuai dengan distribusi sampling.

Tentu saja, ketika Anda melakukan uji hipotesis, Anda tidak pernah tahu distribusi mana yang menghasilkan hasil. Anda bekerja dengan mean sampel — sebuah titik pada sumbu horizontal. Keputusan menolak-atau-tidak menolak bermuara pada memutuskan distribusi mana yang menjadi bagian dari rata-rata sampel. Anda menetapkan nilai kritis — kriteria keputusan. Jika mean sampel berada di satu sisi nilai kritis, Anda menolak H0. Jika tidak, Anda tidak. Dalam nada ini, gambar juga menunjukkan dan . Ini, seperti yang saya sebutkan sebelumnya dalam bab ini, adalah probabilitas kesalahan keputusan. Daerah yang bersesuaian dengan berada dalam distribusi H_0. Saya telah menaungi itu dalam abu-abu gelap. Ini mewakili probabilitas bahwa rata-rata sampel berasal dari distribusi H0, tetapi sangat ekstrem sehingga Anda menolak H_0.

Di mana Anda mengatur nilai kritis menentukan . Dalam sebagian besar pengujian hipotesis, Anda menetapkan pada 0,05. Ini berarti bahwa Anda bersedia mentolerir kesalahan Tipe I (menolak H_0 padahal seharusnya tidak) 5 persen dari waktu. Secara grafis, nilai kritis memotong 5 persen dari area distribusi sampling. Omong-omong, jika Anda berbicara tentang 5 persen area yang berada di ekor kanan distribusi (lihat Gambar 10.1), Anda berbicara tentang 5 persen teratas. Jika 5 persen di ekor kiri yang Anda minati, itu adalah 5 persen yang lebih rendah.

Daerah yang sesuai dengan berada dalam distribusi H1. Saya telah menaungi itu dalam warna abu-abu muda. Area ini mewakili probabilitas bahwa rata-rata sampel berasal dari distribusi H1, tetapi cukup dekat dengan pusat distribusi H0 sehingga Anda tidak menolak H0

(tetapi seharusnya Anda memilikinya). Anda tidak dapat menyetel . Ukuran area ini tergantung pada pemisahan antara sarana dua distribusi, dan itu terserah dunia tempat kita tinggal — bukan terserah Anda. Distribusi pengambilan sampel ini sesuai ketika pekerjaan Anda sesuai dengan kondisi teorema limit pusat: jika Anda tahu bahwa populasi yang Anda kerjakan adalah distribusi normal atau jika Anda memiliki sampel yang besar.

10.3 MENGHITUNG BEBERAPA Z

Berikut adalah contoh uji hipotesis yang melibatkan sampel dari populasi yang terdistribusi normal. Karena populasi terdistribusi normal, setiap ukuran sampel menghasilkan distribusi sampling yang terdistribusi normal. Karena ini adalah distribusi normal, Anda menggunakan skor-z dalam uji hipotesis:

$$z = \frac{\bar{x} - \mu}{\sigma / \sqrt{N}}$$

Satu lagi "karena": Karena Anda menggunakan skor-z dalam uji hipotesis, skor-z di sini disebut statistik uji.

Misalkan Anda berpikir bahwa orang yang tinggal di kode pos tertentu memiliki IQ di atas rata-rata. Anda mengambil sampel sembilan orang dari kode pos itu, memberi mereka tes IQ, membuat tabulasi hasilnya, dan menghitung statistiknya. Untuk populasi skor IQ, = 100 dan = 15.

Hipotesisnya adalah:

$$H_0: \mu_{ZIP\ code} \leq 100$$

$$H_1: \mu_{ZIP\ code} > 100$$

Asumsikan bahwa α = 0,05. Itulah daerah yang diarsir di bagian ekor distribusi H_0 pada Gambar 10-1. Mengapa di H_0? Anda menggunakan simbol itu karena Anda akan menolak H_0 hanya jika rata-rata sampel lebih besar dari nilai yang dihipotesiskan. Hal lain adalah bukti yang mendukung untuk tidak menolak H_0. Misalkan mean sampel adalah 108,67. Bisakah Anda menolak H_0?

Tes ini melibatkan pengubahan 108,67 menjadi skor standar dalam distribusi sampling mean:

$$z = \frac{\bar{x} - \mu}{\sigma / \sqrt{N}} = \frac{108.67 - 100}{\left(15 / \sqrt{9}\right)} = \frac{8.67}{\left(15/3\right)} = \frac{8.67}{5} = 1.73$$

Apakah nilai statistik uji cukup besar untuk memungkinkan Anda menolak H_0 dengan α = 0,05? Dia. Nilai kritis — nilai z yang memotong 5 persen area dalam distribusi normal standar — adalah 1,645. (Setelah bertahun-tahun bekerja dengan distribusi normal standar, kebetulan saya mengetahui hal ini. Baca Bab 8, cari tahu tentang fungsi qnorm() R, dan Anda juga dapat memperoleh informasi seperti itu di ujung jari Anda.) Nilai yang dihitung, 1,73, melebihi 1,645, sehingga berada di wilayah penolakan. Keputusannya adalah menolak H_0.

Ini berarti bahwa jika H_0 benar, probabilitas mendapatkan nilai statistik uji yang paling tidak sebesar ini lebih kecil dari 0,05. Itu bukti kuat yang mendukung penolakan H_0. Dalam bahasa statistik, setiap kali Anda menolak H_0, hasilnya dikatakan signifikan secara statistik. Jenis pengujian hipotesis ini disebut satu arah karena daerah penolakan berada dalam satu ekor dari distribusi sampling.

Sebuah uji hipotesis bisa satu arah ke arah lain. Misalkan Anda memiliki alasan untuk percaya bahwa orang-orang dalam kode pos itu memiliki IQ di bawah rata-rata. Dalam hal ini, hipotesisnya adalah:

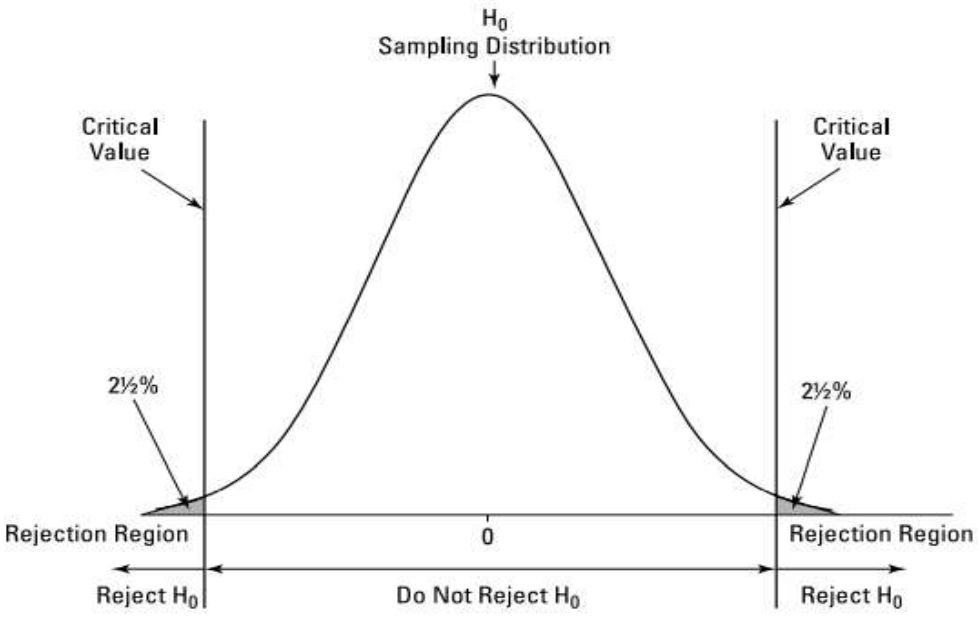
$$H_0: \mu_{ZIP\ code} \geq 100$$
$$H_1: \mu_{ZIP\ code} < 100$$

Untuk uji hipotesis ini, nilai kritis dari statistik uji adalah –1,645 jika α = 0,05.

Uji hipotesis dapat bersifat two-tailed, artinya daerah penolakan berada pada kedua ekor dari distribusi sampling H_0. Itu terjadi ketika hipotesis terlihat seperti ini:

$$H_0: \mu_{ZIP\ code} = 100$$
$$H_1: \mu_{ZIP\ code} \neq 100$$

Dalam hal ini, hipotesis alternatif hanya menetapkan bahwa mean berbeda dari nilai hipotesis nol, tanpa mengatakan apakah itu lebih besar atau lebih kecil. Gambar 10.2 menunjukkan seperti apa daerah penolakan dua sisi untuk α = .05. 5 persen dibagi rata antara ekor kiri (juga disebut ekor bawah) dan ekor kanan (ekor atas).



Gambar 10.2 Daerah penolakan dua sisi untuk α = .05.

Untuk distribusi normal standar, kebetulan, skor-z yang memotong 2,5 persen di ekor kanan adalah 1,96. Skor-z yang memotong 2,5 persen di ekor kiri adalah -1,96. (Sekali lagi,

saya kebetulan mengetahui nilai-nilai ini setelah bertahun-tahun bekerja dengan distribusi normal standar.) Nilai-z pada contoh sebelumnya, 1,73, tidak melebihi 1,96. Keputusan, dalam kasus dua sisi, adalah untuk tidak menolak H_0.

Ini memunculkan poin penting. Uji hipotesis satu sisi dapat menolak H_0, sedangkan uji dua sisi pada data yang sama mungkin tidak. Tes dua sisi menunjukkan bahwa Anda mencari perbedaan antara mean sampel dan mean hipotesis nol, tetapi Anda tidak tahu ke arah mana. Tes satu sisi menunjukkan bahwa Anda memiliki gagasan yang cukup bagus tentang bagaimana perbedaannya akan terlihat. Untuk tujuan praktis, ini berarti Anda harus mencoba memiliki pengetahuan yang cukup untuk dapat menentukan uji satu sisi: Itu memberi Anda peluang yang lebih baik untuk menolak H_0 ketika Anda seharusnya melakukannya.

10.4 PENGUJIAN Z DI R

Fungsi R yang disebut z.test() akan sangat bagus untuk melakukan jenis pengujian yang saya diskusikan di bagian sebelumnya. Satu masalah: Fungsi itu tidak ada di basis R. Meskipun Anda dapat menemukannya di paket lain, cukup mudah untuk membuatnya dan belajar sedikit tentang pemrograman R dalam prosesnya.

Fungsinya akan bekerja seperti ini:

```
> IQ.data <- c(100,101,104,109,125,116,105,108,110)
> z.test(IQ.data,100,15)
 z = 1.733
 one-tailed probability = 0.042
 two-tailed probability = 0.084
```

Mulailah dengan membuat nama fungsi dan argumennya:

```
z.test = function(x,mu,popvar){
```

Argumen pertama adalah vektor data, yang kedua adalah mean populasi, dan yang ketiga adalah varians populasi. Tanda kurung kurawal kiri menandakan bahwa sisa kode adalah apa yang terjadi di dalam fungsi.

Selanjutnya, buat vektor yang akan menampung probabilitas satu sisi dari skor-z yang akan Anda hitung:

```
one.tail.p <- NULL
```

Kemudian Anda menghitung skor-z dan membulatkannya ke tiga tempat desimal:

```
z.score <- round((mean(x)-mu)/(popvar/sqrt(length(x))),3)
```

Tanpa pembulatan, R mungkin menghitung banyak tempat desimal, dan hasilnya akan terlihat berantakan. Terakhir, Anda menghitung probabilitas satu sisi (proporsi area di luar skor-z yang dihitung), dan kembali membulatkan ke tiga tempat desimal:

```
one.tail.p <- round(pnorm(abs(z.score),lower.tail = FALSE),3)
```

Mengapa menempatkan abs() (nilai absolut) dalam argumen ke pnorm? Ingat bahwa hipotesis alternatif dapat menentukan nilai di bawah rata-rata, dan data mungkin menghasilkan skor-z negatif.

Urutan bisnis selanjutnya adalah mengatur tampilan output. Untuk ini, Anda menggunakan fungsi cat() . Saya menggunakan fungsi ini di Bab 7 untuk menampilkan serangkaian angka yang cukup besar dengan cara yang terorganisir. Nama cat adalah kependekan dari concatenate and print, yang persis seperti yang saya ingin Anda lakukan di sini: Menggabungkan (menggabungkan) string (seperti probabilitas satu sisi =) dengan ekspresi (seperti one.tail.p), lalu tunjukkan itu semuanya di layar. Saya juga ingin Anda memulai baris baru untuk setiap rangkaian, dan \n adalah cara R untuk mewujudkannya.
Inilah pernyataan kucing:

```
cat(" z =",z.score,"\n",
      "one-tailed probability =", one.tail.p,"\n",
      "two-tailed probability =", 2*one.tail.p )}
```

Spasi antara kutipan kiri dan z berbaris di baris pertama dengan dua berikutnya di layar. Tanda kurung kurawal kanan menutup fungsi. Ini dia, semuanya bersama-sama:

```
z.test = function(x,mu,popvar){
  one.tail.p <- NULL
  z.score <- round((mean(x)-mu)/(popvar/sqrt(length(x))),3)
  one.tail.p <- round(pnorm(abs(z.score),lower.tail
          = FALSE),3)
  cat(" z =",z.score,"\n",
      "one-tailed probability =", one.tail.p,"\n",
      "two-tailed probability =", 2*one.tail.p )}
```

Menjalankan fungsi ini menghasilkan apa yang Anda lihat di awal bagian ini.

10.5 T UNTUK SATU

Dalam contoh sebelumnya, Anda bekerja dengan skor IQ. Populasi skor IQ adalah distribusi normal dengan mean dan standar deviasi yang terkenal. Dengan demikian, Anda dapat bekerja dengan teorema limit pusat dan menggambarkan distribusi sampling mean sebagai distribusi normal. Anda kemudian dapat menggunakan z sebagai statistik uji.

Namun, di dunia nyata, Anda biasanya tidak memiliki kemewahan bekerja dengan populasi yang terdefinisi dengan baik. Anda biasanya memiliki sampel kecil, dan Anda biasanya mengukur sesuatu yang tidak begitu dikenal seperti IQ. Intinya adalah Anda sering tidak mengetahui parameter populasi, juga tidak tahu apakah populasi terdistribusi normal.

Jika itu masalahnya, Anda menggunakan data sampel untuk memperkirakan simpangan baku populasi, dan Anda memperlakukan distribusi sampel rata-rata sebagai anggota keluarga distribusi yang disebut distribusi-t. Anda menggunakan t sebagai statistik uji. Dalam Bab 9, saya memperkenalkan distribusi ini dan menyebutkan bahwa Anda membedakan anggota keluarga ini dengan parameter yang disebut derajat kebebasan (df).
Rumus untuk statistik uji adalah:

$$t = \frac{\bar{x} - \mu}{s / \sqrt{N}}$$

Pikirkan df sebagai penyebut estimasi varians populasi. Untuk pengujian hipotesis di bagian ini, yaitu N-1, di mana N adalah jumlah skor dalam sampel. Semakin tinggi df, semakin dekat distribusi t menyerupai distribusi normal.

Berikut ini contohnya. FarKlempt Robotics, Inc., memasarkan robot mikro. Perusahaan mengklaim bahwa produknya rata-rata empat cacat per unit. Sebuah kelompok konsumen percaya rata-rata ini lebih tinggi. Kelompok konsumen mengambil sampel sembilan robot mikro FarKlempt dan menemukan rata-rata tujuh cacat, dengan standar deviasi 3,12. Uji hipotesisnya adalah:

$$H_0: \mu \leq 4$$
$$H_1: \mu > 4$$
$$\alpha = .05$$

Rumusnya adalah:

$$t = \frac{\bar{x} - \mu}{s/\sqrt{N}} = \frac{7-4}{\left(3.12/\sqrt{9}\right)} = \frac{3}{\left(3.12/3\right)} = 2.88$$

Bisakah Anda menolak H0? Fungsi R di bagian berikutnya memberitahu Anda.

t Pengujian di R

Saya melihat pratinjau fungsi t.test() di Bab 2 dan membicarakannya lebih detail di Bab 9. Di sini, Anda menggunakannya untuk menguji hipotesis.

Mulailah dengan data untuk FarKlemp Robotics:

```
> FarKlempt.data <- c(3,6,9,9,4,10,6,4,12)
```

Kemudian terapkan t.test(). Sebagai contoh, terlihat seperti ini:

```
t.test(FarKlempt.data,mu=4, alternative="greater")
```

Argumen kedua menentukan bahwa Anda menguji rata-rata yang dihipotesiskan 4, dan argumen ketiga menunjukkan bahwa hipotesis alternatif adalah bahwa rata-rata sebenarnya lebih besar dari 4.

```
> t.test(FarKlempt.data,mu=4, alternative="greater")

            One Sample t-test

data:  c(3, 6, 9, 9, 4, 10, 6, 4, 12)
t = 2.8823, df = 8, p-value = 0.01022
alternative hypothesis: true mean is greater than 4
95 percent confidence interval:
 5.064521      Inf
sample estimates:
mean of x
        7
```

Keluaran memberikan nilai-t dan nilai-p rendah menunjukkan bahwa Anda dapat menolak hipotesis nol dengan α = 0,05. Fungsi t.test() ini serbaguna. Saya bekerja dengannya lagi di Bab 11 ketika saya menguji hipotesis tentang dua sampel.

Bekerja dengan t-Distributions

Sama seperti Anda dapat menggunakan awalan d, p, q, dan r untuk keluarga distribusi normal, Anda dapat menggunakan dt() (fungsi densitas), pt() (fungsi densitas kumulatif), qt() (kuantil), dan rt() (pembuatan angka acak) untuk keluarga distribusi-t.

Berikut adalah dt() dan rt() yang bekerja untuk distribusi-t dengan 12 df:

```
> t.values <- seq(-4,4,1)
> round(dt(t.values,12),2)
[1] 0.00 0.01 0.06 0.23 0.39 0.23 0.06 0.01 0.00
> round(pt(t.values,12),2)
[1] 0.00 0.01 0.03 0.17 0.50 0.83 0.97 0.99 1.00
```

Saya tunjukkan cara menggunakan dt() lebih banyak di bagian selanjutnya. (Jauh lebih. Percayalah padaku).

Untuk informasi kuantil tentang distribusi-t dengan 12 df:

```
> quartiles <- c(0,.25,.50,.75,1)
> qt(quartiles,12)
[1]       -Inf -0.6954829  0.0000000  0.6954829        Inf
```

The –Inf dan Inf memberitahu Anda bahwa kurva tidak pernah menyentuh sumbu x di kedua ekor. Untuk membangkitkan delapan (pembulatan) bilangan acak dari distribusi-t dengan 12 df:

```
> round(rt(8,12),2)
[1]  0.73  0.13 -1.32  1.33 -1.27  0.91 -0.48 -0.83
```

Semua fungsi ini memberi Anda pilihan untuk bekerja dengan distribusi-t yang tidak berpusat di sekitar nol. Anda melakukan ini dengan memasukkan nilai untuk ncp (parameter noncentrality). Di sebagian besar aplikasi distribusi-t, noncentrality tidak muncul. Untuk kelengkapan, saya menjelaskan konsep ini secara lebih rinci dalam Lampiran 3 online.

Memvisualisasikan Distribusi-t

Memvisualisasikan distribusi sering kali membantu Anda memahaminya. Prosesnya bisa sedikit terlibat dalam R, tetapi itu sepadan dengan usaha. Gambar 9-7 menunjukkan tiga anggota keluarga distribusi-t pada grafik yang sama. Yang pertama memiliki df=3, yang kedua memiliki df=10, dan yang ketiga adalah distribusi normal standar (df=tak terhingga). Di bagian ini, saya menunjukkan cara membuat grafik itu di grafik dasar R dan di ggplot2.

Dengan salah satu metode, langkah pertama adalah menyiapkan vektor nilai yang akan digunakan oleh fungsi densitas:

```
t.values <- seq(-4,4,.1)
```

Satu hal lagi dan saya akan membantu Anda memulai. Setelah grafik selesai, Anda akan meletakkan simbol tak terhingga, pada legenda untuk menunjukkan df untuk distribusi normal standar. Untuk melakukannya, Anda harus menginstal paket yang disebut grDevices: Pada tab Packages, klik Install, lalu di kotak dialog Install Packages, ketik grDevices dan klik Install. Ketika grDevices muncul di tab Packages, pilih kotak centangnya.

Dengan grDevices terinstal, ini menambahkan simbol infinity ke legenda:

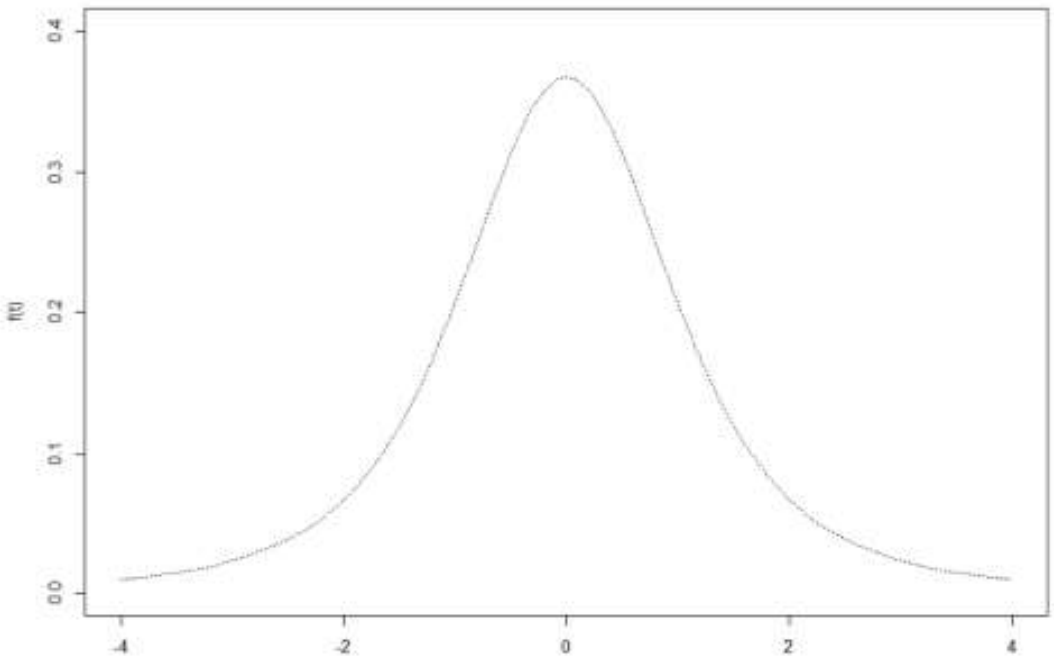
```
expression(infinity)
```

Merencanakan t dalam grafik R dasar

Mulailah dengan fungsi plot(), dan plot distribusi-t dengan 3 df:

```
plot(x = t.values,y = dt(t.values,3),  type = "l", lty =
          "dotted", ylim = c(0,.4), xlab = "t", ylab = "f(t)")
```

Dua argumen pertama cukup jelas. Dua berikutnya menentukan jenis plot — type = "l" berarti plot garis (itu huruf kecil "L" bukan angka 1), dan lty = "dotted" menunjukkan jenis garis. Argumen ylim menetapkan batas bawah dan atas sumbu y — ylim = c(0,.4). Sedikit mengutak-atik menunjukkan bahwa jika Anda tidak melakukan ini, kurva berikutnya akan terpotong di bagian atas. Dua argumen terakhir memberi label sumbu. Gambar 10.3 menunjukkan grafik sejauh ini:

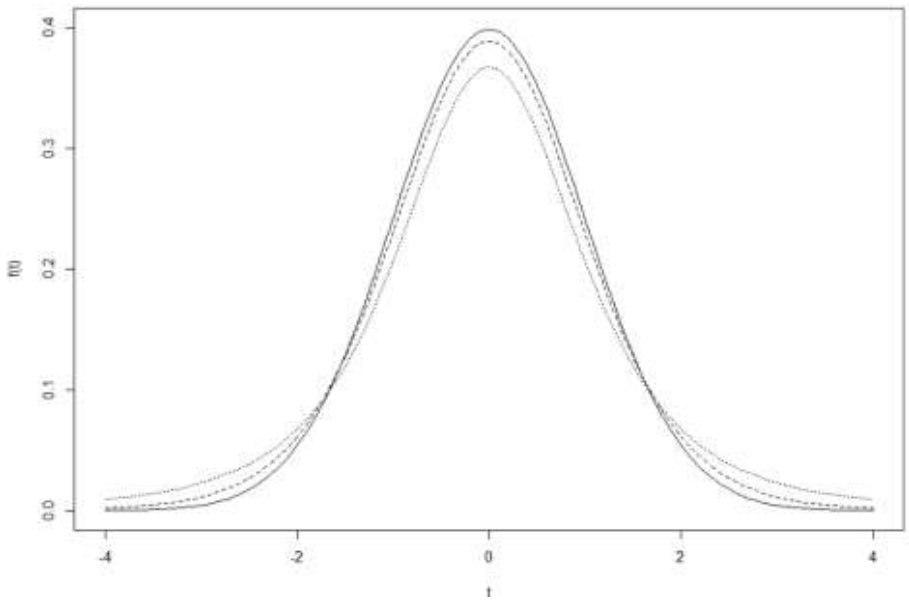


Gambar 10.3 distribusi t dengan 3 df, basis R.

Dua baris berikutnya menambahkan distribusi-t untuk df=10, dan untuk normal standar (df = tak terhingga):

```
lines(t.values,dt(t.values,10),lty = "dashed")
lines(t.values,dnorm(t.values))
```

Garis untuk standar normal adalah solid (nilai default untuk lty). Gambar 10.4 menunjukkan kemajuan. Yang hilang hanyalah legenda yang menjelaskan kurva mana.



Gambar 10.4 Tiga distribusi untuk mencari legenda.

Salah satu keuntungan dari basis R adalah memposisikan dan mengisi legenda tidak sulit:

```
legend("topright", title = "df",legend =
          c(expression(infinity),"10","3"), lty =
          c("solid","dashed","dotted"), bty = "n")
```

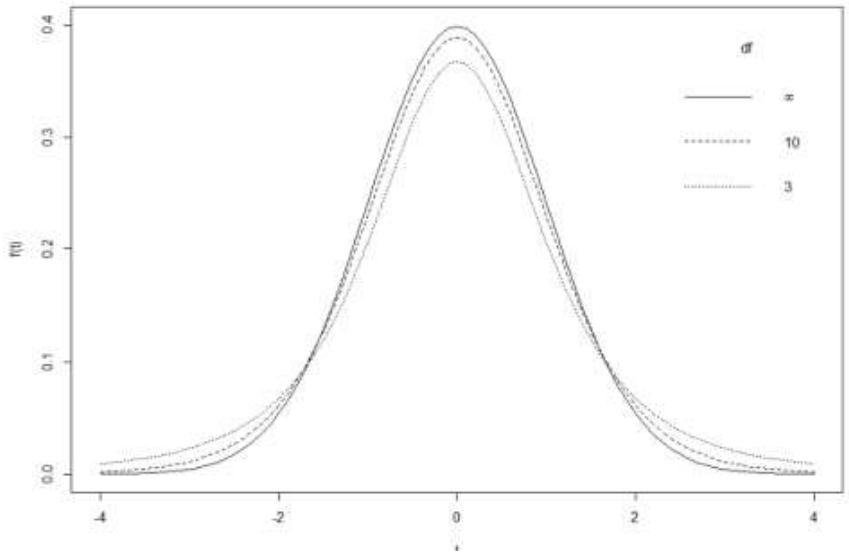
Argumen pertama menempatkan legenda di sudut kanan atas. Yang kedua memberi legenda itu judulnya. Argumen ketiga adalah vektor yang menentukan apa yang ada di legenda. Seperti yang Anda lihat, elemen pertama adalah ekspresi tak terhingga yang saya tunjukkan sebelumnya, sesuai dengan df untuk normal standar. Elemen kedua dan ketiga adalah df untuk dua distribusi t yang tersisa. Anda memesannya dengan cara ini karena itulah urutan kurva yang muncul di tengahnya. Argumen lty adalah vektor yang menentukan urutan linetypes (mereka sesuai dengan df). Argumen terakhir bty="n" menghapus batas dari legenda. Dan ini menghasilkan Gambar 10.5.

Merencanakan t di ggplot2

Pendekatan tata bahasa-grafis membutuhkan lebih banyak upaya daripada basis R. Tapi ikuti terus dan Anda akan belajar banyak tentang ggplot2.

Anda mulai dengan memasukkan angka-angka yang relevan ke dalam bingkai data:

```
t.frame = data.frame(t.values,
                     df3 = dt(t.values,3),
                     df10 = dt(t.values,10),
                     std_normal = dnorm(t.values))
```



Gambar 10.5 Grafik terakhir, termasuk legenda.

Enam baris pertama dari bingkai data terlihat seperti ini:

```
> head(t.frame)
  t.values         df3         df10   std_normal
1     -4.0 0.009163361 0.002031034 0.0001338302
2     -3.9 0.009975671 0.002406689 0.0001986555
3     -3.8 0.010875996 0.002854394 0.0002919469
4     -3.7 0.011875430f 0.003388151 0.0004247803
5     -3.6 0.012986623 0.004024623 0.0006119019
6     -3.5 0.014224019 0.004783607 0.0008726827
```

Itu adalah bingkai data yang cukup bagus, tetapi dalam format lebar. Seperti yang saya tunjukkan di Bab 3, ggplot() lebih memilih format panjang — yang merupakan tiga kolom kepadatan- angka yang ditumpuk menjadi satu kolom. Untuk mendapatkan format itu — disebut membentuk kembali data — pastikan Anda telah menginstal paket reshape2. Pilih kotak centangnya pada tab Paket dan Anda siap untuk pergi.

Mengubah dari format lebar ke format panjang disebut melebur data, jadi fungsinya adalah:

```
t.frame.melt <- melt(t.frame,id="t.values")
```

Argumen id menetapkan bahwa t.values adalah variabel yang jumlahnya tidak ditumpuk dengan yang lain. Anggap saja sebagai variabel yang menyimpan data. Enam baris pertama t.frame.melt adalah:

```
> head(t.frame.melt)
  t.values variable       value
1     -4.0      df3 0.009163361
2     -3.9      df3 0.009975671
3     -3.8      df3 0.010875996
4     -3.7      df3 0.011875430
5     -3.6      df3 0.012986623
6     -3.5      df3 0.014224019
```

Itu selalu merupakan ide yang baik untuk memiliki nama kolom yang bermakna, jadi . . .

```
> colnames(t.frame.melt)= c("t","df","density")
> head(t.frame.melt)
     t  df     density
1 -4.0 df3 0.009163361
2 -3.9 df3 0.009975671
3 -3.8 df3 0.010875996
4 -3.7 df3 0.011875430
5 -3.6 df3 0.012986623
6 -3.5 df3 0.014224019
```

Sekarang untuk satu hal lagi sebelum saya mulai membuat grafik. Ini adalah vektor yang akan berguna saat Anda meletakkan sumbu x:

```
x.axis.values <- seq(-4,4,2)
```

Mulailah dengan ggplot():

```
ggplot(t.frame.melt, aes(x=t,y=f(t),group =df))
```

Argumen pertama adalah bingkai data. Pemetaan estetika memberi tahu Anda bahwa t berada pada sumbu x, kepadatan berada pada sumbu y, dan data dikelompokkan ke dalam kelompok yang ditentukan oleh variabel df.

Ini adalah plot garis, jadi fungsi geom yang tepat untuk ditambahkan adalah geom_line:

```
geom_line(aes(linetype=df))
```

Fungsi geom dapat bekerja dengan pemetaan estetika. Pemetaan estetika di sini memetakan df ke jenis garis.

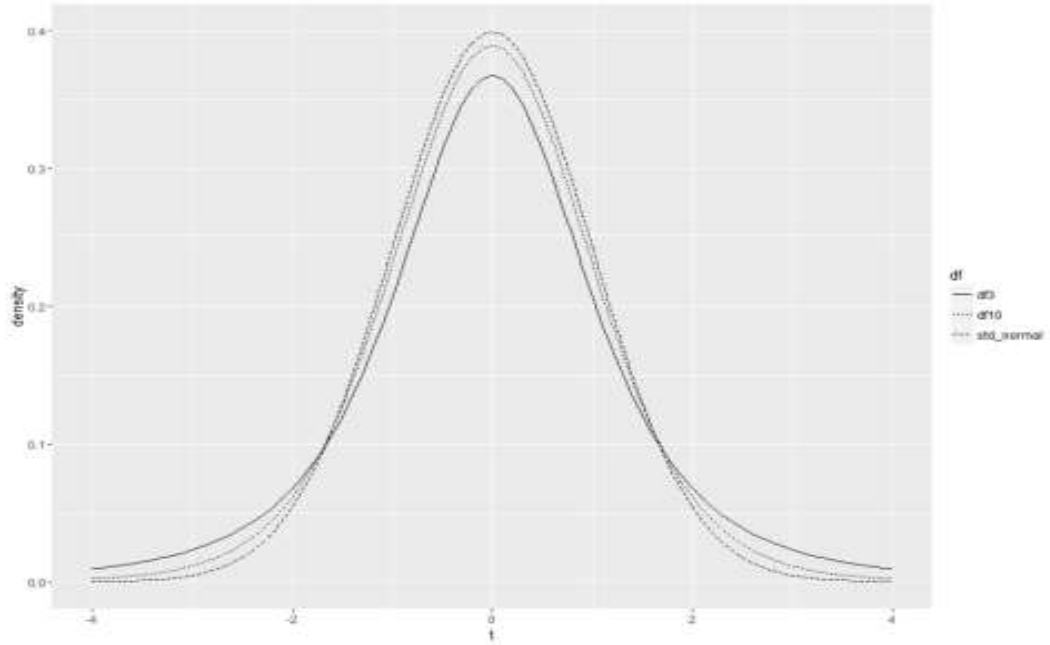
Ubah skala sumbu x sehingga berubah dari -4 ke 4, dua kali. Di sinilah untuk menggunakan vektor x.axis.values itu:

```
scale_x_continuous(breaks=x.axis.values,labels=x.axis.values)
```

Argumen pertama menetapkan titik putus untuk sumbu x, dan argumen kedua memberikan label untuk titik tersebut. Menempatkan tiga pernyataan ini bersama-sama.

```
ggplot(t.frame.melt, aes(x=t,y=density,group =df)) +
  geom_line(aes(linetype=df)) +
  scale_x_continuous(breaks = x.axis.values,labels =
          x.axis.values)
```

Hasil pada Gambar 10.6. Salah satu keunggulan ggplot2 adalah kode tersebut secara otomatis menghasilkan legenda.



GAMBAR 10.6 Tiga kurva distribusi-t, diplot dalam ggplot2.

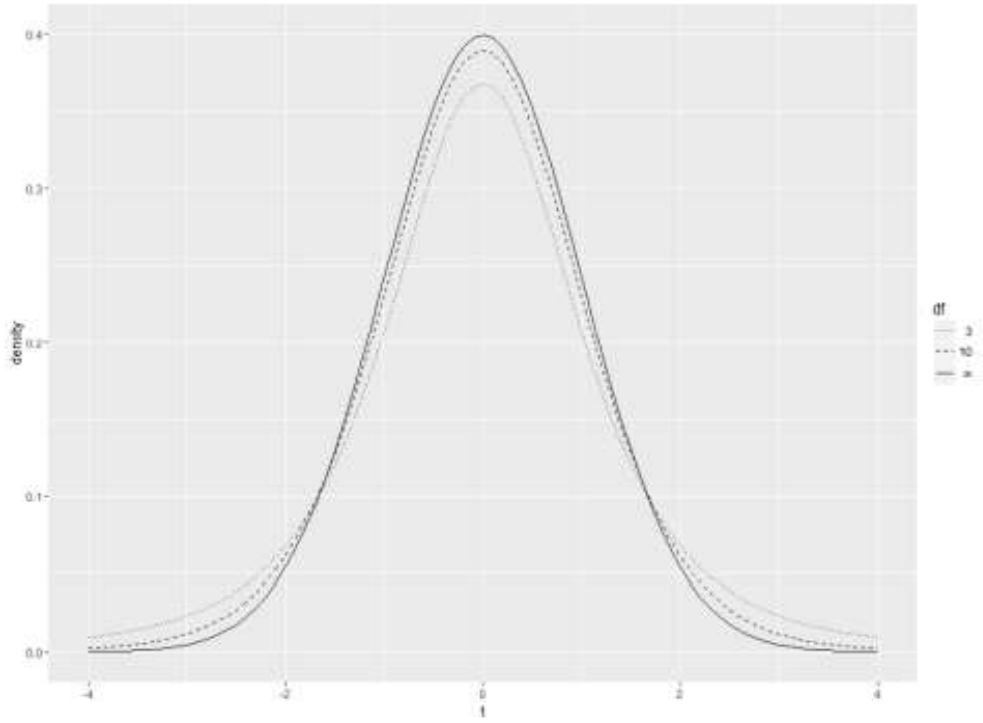
Anda masih memiliki beberapa pekerjaan yang harus dilakukan. Pertama-tama, penetapan linetype default bukanlah yang Anda inginkan, jadi Anda harus mengulanginya:

```
scale_linetype_manual(values =
          c("dotted","dashed","solid"),
labels = c("3","10", expression(infinity)))
```

Empat pernyataan:

```
ggplot(t.frame.melt, aes(x=t,y=density,group =df)) +
  geom_line(aes(linetype=df)) +
  scale_x_continuous(breaks = x.axis.values,labels =
          x.axis.values)+
  scale_linetype_manual(values =
           c("dotted","dashed","solid"),
            labels = c("3","10", expression(infinity)))
```

menghasilkan Gambar 10.7.



Gambar 10.7 Tiga kurva distribusi-t, dengan tipe garis dipindahkan.

Seperti yang Anda lihat, item dalam legenda tidak sesuai dengan urutan kurva yang muncul di tengahnya. Saya seorang yang ngotot untuk itu. Saya pikir itu membuat grafik lebih mudah dipahami ketika elemen grafik dan elemen legenda disinkronkan. ggplot2 menyediakan fungsi panduan yang memungkinkan Anda mengontrol detail legenda. Untuk membalik urutan linetypes dalam legenda, inilah yang Anda lakukan:

```
  guides(linetype=guide_legend(reverse = TRUE))
```

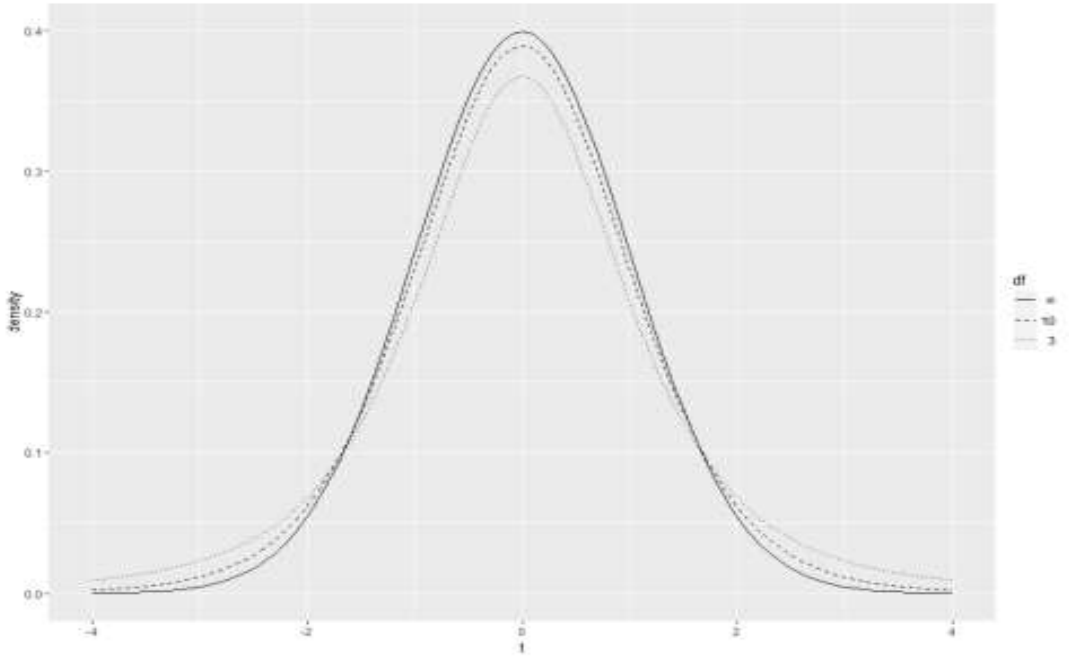
Menempatkan semua kode bersama-sama, akhirnya, menghasilkan Gambar 10.8.

```
ggplot(t.frame.melt, aes(x=t,y=density,group =df)) +
  geom_line(aes(linetype=df)) +
```

```
  scale_x_continuous(breaks = x.axis.values,labels =
          x.axis.values)+
  scale_linetype_manual(values =
           c("dotted","dashed","solid"),
           labels = c("3","10", expression(infinity)))+
  guides(linetype=guide_legend(reverse = TRUE))
```

Saya serahkan kepada Anda sebagai latihan untuk menamai kembali sumbu y f(t).

Grafik Base R versus ggplot2: Ini seperti mengendarai mobil dengan transmisi standar versus mengemudi dengan transmisi otomatis — tapi saya tidak selalu yakin yang mana!



Gambar 10.8 Produk akhir, dengan legenda yang disusun ulang.

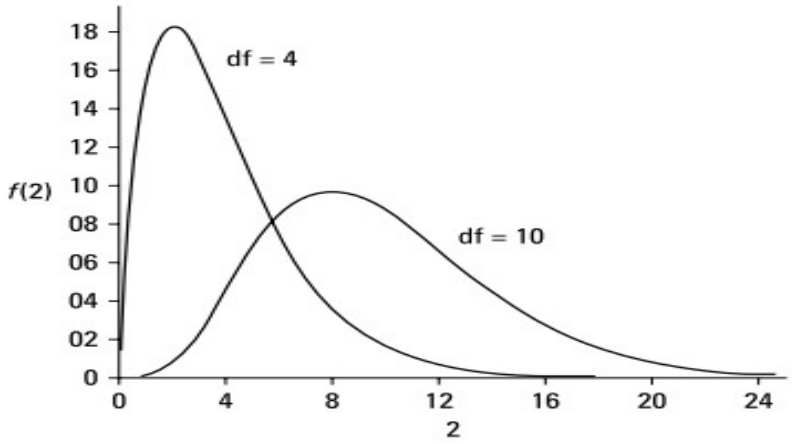
Satu hal lagi tentang ggplot2

Saya bisa meminta Anda merencanakan semua ini tanpa membuat dan membentuk kembali bingkai data. Pendekatan alternatif adalah menetapkan NULL sebagai sumber data, memetakan t.values ke sumbu x, lalu menambahkan tiga pernyataan geom_line. Masing-masing pernyataan tersebut akan memetakan vektor kepadatan (dibuat dengan cepat) ke sumbu y, dan masing-masing akan memiliki tipe garis sendiri.

Masalah dengan pendekatan itu? Ketika Anda melakukannya dengan cara itu, tata bahasa tidak secara otomatis membuat legenda. Tanpa bingkai data, tidak ada yang bisa membuat legenda. Ini seperti menggunakan ggplot() untuk membuat grafik R dasar. Apakah pernah ide yang baik untuk menggunakan pendekatan ini? Ya, benar — ketika Anda tidak ingin menyertakan legenda tetapi Anda ingin memberi anotasi pada grafik dengan cara lain. Saya memberikan contoh di bagian selanjutnya “Memvisualisasikan Distribusi Chi-Kuadrat.”

10.6 MENGUJI VARIANS

Sejauh ini, saya membahas pengujian hipotesis satu sampel untuk sarana. Anda juga dapat menguji hipotesis tentang varians. Topik ini terkadang muncul dalam konteks manufaktur. Misalkan FarKlempt Robotics, Inc., menghasilkan bagian yang harus memiliki panjang tertentu dengan variabilitas yang sangat kecil. Anda dapat mengambil sampel bagian, mengukurnya, menemukan variabilitas sampel, dan melakukan uji hipotesis terhadap variabilitas yang diinginkan.



Gambar 10.9 Dua anggota keluarga chi-kuadrat.

Keluarga distribusi untuk tes ini disebut chi-kuadrat. Simbolnya adalah 2. Saya tidak akan membahas semua matematika. Saya hanya akan memberitahu Anda bahwa, sekali lagi, df adalah parameter yang membedakan satu anggota keluarga dari yang lain. Gambar 10-9 menunjukkan dua anggota keluarga chi-kuadrat.

Seperti yang ditunjukkan gambar, chi-kuadrat tidak seperti keluarga distribusi sebelumnya yang saya tunjukkan kepada Anda. Anggota keluarga ini dapat dimiringkan, dan tidak satu pun dari mereka dapat mengambil nilai kurang dari nol.

Rumus untuk statistik uji adalah:

$$\chi^2 = \frac{(N-1)s^2}{\sigma^2}$$

N adalah jumlah skor dalam sampel, s2 adalah varians sampel, dan 2 adalah varians populasi yang ditentukan dalam H_0. Dengan tes ini, Anda harus mengasumsikan bahwa apa yang Anda ukur memiliki distribusi normal.

Misalkan proses untuk bagian FarKlempt harus memiliki, paling banyak, standar deviasi 1,5 inci untuk panjangnya. (Perhatikan bahwa saya menggunakan standar deviasi. Ini memungkinkan saya untuk berbicara dalam inci. Jika saya menggunakan varians, unit akan menjadi inci persegi.) Setelah mengukur sampel 10 bagian, Anda menemukan standar deviasi 1,80 inci.

Hipotesisnya adalah:

$$H_0: \sigma^2 \leq 2.25$$

$$H_1: \sigma^2 > 2.25$$

$$\alpha = .05$$

(ingat untuk mengkuadratkan standar deviasi "paling banyak" 1,5 inci)

Bekerja dengan rumus,

$$\chi^2 = \frac{(N-1)s^2}{\sigma^2} = \frac{(10-1)(1.80)^2}{(1.5)^2} = \frac{(9)(3.25)}{2.25} = 12.96$$

dapatkah kamu menolak H_0? Baca terus.

Pengujian di R

Pada titik ini, Anda mungkin berpikir bahwa fungsi chisq.test() akan menjawab pertanyaan tersebut. Meskipun basis R menyediakan fungsi ini, itu tidak sesuai di sini. Seperti yang Anda lihat di Bab 18 dan 20, ahli statistik menggunakan fungsi ini untuk menguji jenis hipotesis lainnya.

Alih-alih, beralihlah ke fungsi yang disebut varTest, yang ada dalam paket EnvStats. Pada tab Paket, klik Instal. Kemudian ketik EnvStats ke dalam kotak dialog Instal Paket dan klik Instal. Ketika EnvStats muncul di tab Paket, pilih kotak centangnya.

Sebelum Anda menggunakan pengujian, Anda membuat vektor untuk menampung sepuluh pengukuran yang dijelaskan dalam contoh di bagian sebelumnya:

```
FarKlempt.data2 <- c(12.43, 11.71, 14.41, 11.05, 9.53,
          11.66, 9.33,11.71,14.35,13.81)
```

Dan sekarang, tesnya:

```
varTest(FarKlempt.data2,alternative="greater",conf.level
          = 0.95,sigma.squared = 2.25)
```

Argumen pertama adalah vektor data. Yang kedua menentukan hipotesis alternatif bahwa varians sebenarnya lebih besar dari varians yang dihipotesiskan, yang ketiga memberikan tingkat kepercayaan (1-α), dan yang keempat adalah varians yang dihipotesiskan.
Menjalankan baris kode itu menghasilkan hasil berikut:

```
Results of Hypothesis Test
--------------------------

Null Hypothesis:          variance = 2.25

Alternative Hypothesis:   True variance is greater than 2.25

Test Name:                Chi-Squared Test on Variance

Estimated Parameter(s):   variance = 3.245299

Data:                     FarKlempt.data2

Test Statistic:           Chi-Squared = 12.9812

Test Statistic Parameter: df = 9

P-value:                  0.163459

95% Confidence Interval:  LCL = 1.726327
                          UCL =      Inf
```

Di antara statistik lainnya, output menunjukkan chi-kuadrat (12,9812) dan nilai-p (0,163459). (Nilai chi-kuadrat di bagian sebelumnya sedikit lebih rendah karena pembulatan.) Nilai p lebih besar dari 0,05. Oleh karena itu, Anda tidak dapat menolak hipotesis nol.
Berapa tinggi chi-kuadrat (dengan df=9) yang harus dimiliki untuk menolak? Hmmm. . . .

10.7 BEKERJA DENGAN DISTRIBUSI CHI-SQUARE

Seperti halnya untuk keluarga distribusi yang telah saya diskusikan dalam bab ini, R menyediakan fungsi untuk bekerja dengan keluarga distribusi chi-kuadrat: dchisq() (untuk fungsi kerapatan), phisq() (untuk fungsi kerapatan kumulatif), qchisq() (untuk kuantil), dan rhisq() (untuk pembuatan angka acak).

Untuk menjawab pertanyaan yang saya ajukan di akhir bagian sebelumnya, saya menggunakan qchisq():

```
> qchisq(.05,df=9,lower.tail = FALSE)
[1] 16.91898
```

Nilai yang diamati sedikit meleset dari nilai kritis itu.

Berikut adalah contoh fungsi chisq lainnya dengan df=9. Untuk kumpulan nilai ini,

```
> chisq.values <- seq(0,16,2)
```

berikut adalah kepadatannya

```
> round(dchisq(chisq.values,9),3)
[1] 0.000 0.016 0.066 0.100 0.101 0.081 0.056 0.036 0.021
```

dan di sini adalah kepadatan kumulatif

```
> round(pchisq(chisq.values,9),3)
[1] 0.000 0.009 0.089 0.260 0.466 0.650 0.787 0.878 0.933
```

Berikut adalah enam angka acak yang dipilih dari distribusi chi-kuadrat ini:

```
> round(rchisq(n=6,df=9),3)
[1] 13.231  5.674  7.396  6.170 11.806  7.068
```

Memvisualisasikan Distribusi Chi-Square

Gambar 10.9 dengan baik menunjukkan beberapa anggota keluarga chi-kuadrat, dengan masing-masing anggota dijelaskan dengan derajat kebebasannya. Di bagian ini, saya menunjukkan cara menggunakan grafik R dasar dan ggplot2 untuk membuat ulang gambar itu. Anda akan belajar lebih banyak tentang grafik, dan Anda akan tahu cara memvisualisasikan setiap anggota keluarga ini.

Merencanakan chi-kuadrat dalam grafik R dasar

Untuk memulai, Anda membuat vektor nilai dari mana dchisq() menghitung kepadatan:

```
chi.values <- seq(0,25,.1)
```

Mulai grafik dengan pernyataan plot:

```
plot(x=chi.values,
     y=dchisq(chi.values,df=4),
     type = "l",
     xlab=expression(chi^2),
     ylab="")
```

Dua argumen pertama menunjukkan apa yang Anda rencanakan — distribusi kai kuadrat dengan empat derajat kebebasan versus vektor nilai kai. Argumen ketiga menentukan garis (itu huruf kecil "L", bukan angka 1). Argumen ketiga melabeli sumbu-x dengan huruf Yunani chi (χ) yang dipangkatkan kedua. Argumen keempat memberi sumbu y label kosong.

Kenapa aku menyuruhmu melakukan itu? Ketika saya pertama kali membuat grafik, saya menemukan bahwa ylab menempatkan label sumbu y terlalu jauh ke kiri, dan labelnya terpotong sedikit. Untuk memperbaikinya, saya mengosongkan ylab dan kemudian menggunakan mtext():

```
mtext(side = 2, text = expression(f(chi^2)), line = 2.5)
```

Argumen samping menentukan sisi grafik untuk menyisipkan label: bawah = 1, kiri = 2, atas = 3, dan kanan = 4. Argumen teks menetapkan f (X^2) sebagai label untuk sumbu. Argumen garis menentukan jarak dari label ke sumbu y: Jarak bertambah dengan nilai.

Selanjutnya, Anda menambahkan kurva untuk chi-kuadrat dengan sepuluh derajat kebebasan:

```
lines(x=chi.values,y=dchisq(chi.values,df= 10))
```

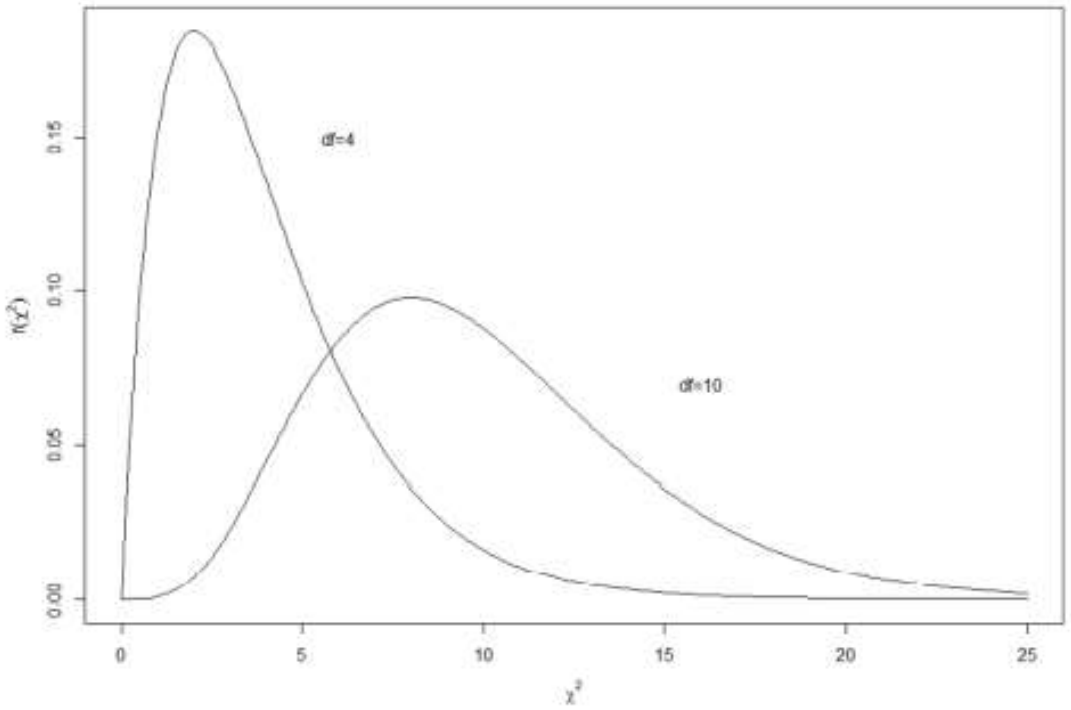
Daripada menambahkan legenda, ikuti Gambar 10-9 dan tambahkan anotasi untuk setiap kurva. Begini caranya:

```
text(x=6,y=.15, label="df=4")
text(x=16, y=.07, label = "df=10")
```

Dua argumen pertama menemukan anotasi, dan yang ketiga menyediakan konten. Menyatukan semuanya:

```
plot(x=chi.values,
     y=dchisq(chi.values,df=4),
     type = "l",
     xlab=expression(chi^2),
     ylab="")
mtext(side = 2, expression(f(chi^2)), line = 2.5)
lines(x=chi.values,y=dchisq(chi.values,df= 10))
text(x=6,y=.15, label="df=4")
text(x=16, y=.07, label = "df=10")
```

menciptakan Gambar 10.10.



GAMBAR 10.10 Dua anggota keluarga chi-kuadrat, diplot dalam grafik R dasar.

Merencanakan chi-kuadrat di ggplot2

Dalam plot ini, saya sekali lagi meminta Anda menggunakan anotasi daripada legenda, jadi Anda menetapkan NULL sebagai sumber data dan bekerja dengan vektor untuk setiap baris. Estetika pertama memetakan chi.values ke sumbu x:

```
ggplot(NULL, aes(x=chi.values))
```

Kemudian Anda menambahkan geom_line untuk setiap kurva chi-kuadrat, dengan pemetaan ke sumbu mereka seperti yang ditunjukkan:

```
geom_line(aes(y=dchisq(chi.values,4)))
geom_line(aes(y=dchisq(chi.values,10)))
```

Seperti yang saya tunjukkan sebelumnya di bab ini, ini seperti menggunakan ggplot2 untuk membuat grafik R dasar, tetapi dalam kasus ini berhasil (karena tidak membuat legenda yang tidak diinginkan).

Selanjutnya, Anda memberi label sumbu:

```
labs(x=expression(chi^2),y=expression(f(chi^2)))
```

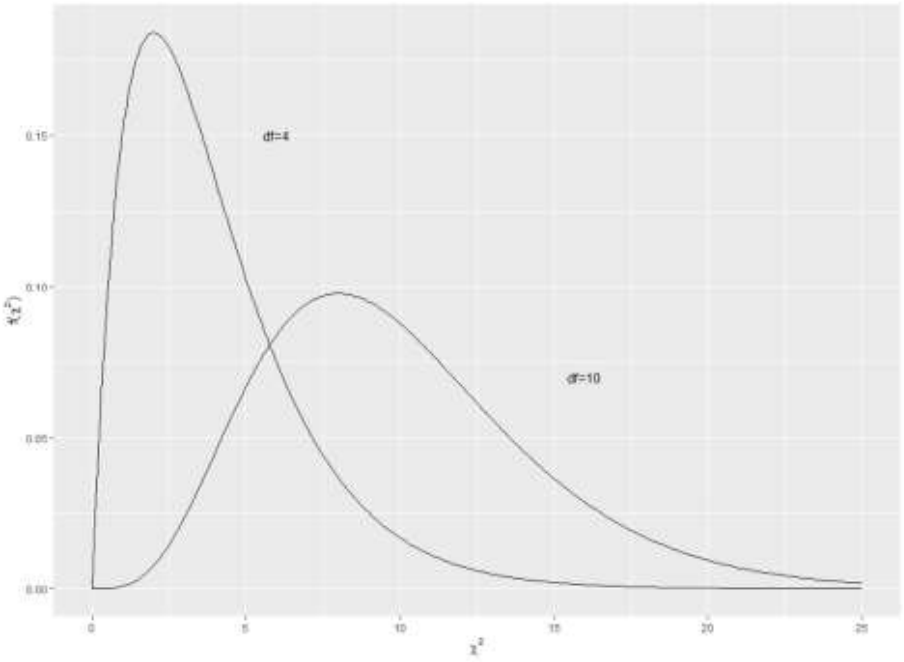
Dan akhirnya, fungsi annotate() yang bernama tepat menambahkan anotasi:

```
annotate(geom = "text",x=6,y=.15,label="df=4")
annotate(geom = "text",x=16,y=.07,label="df=10")
```

Argumen pertama menetapkan bahwa anotasi adalah objek teks. Dua berikutnya menemukan anotasi dalam grafik, dan yang keempat memberikan label.
Jadi semua ini:

```
ggplot(NULL, aes(x=chi.values))+
  geom_line(aes(y=dchisq(chi.values,4))) +
  geom_line(aes(y=dchisq(chi.values,10))) +
  labs(x=expression(chi^2),y=expression(f(chi^2)))+
  annotate(geom = "text",x=6,y=.15,label = "df=4")+
  annotate(geom = "text",x=16,y=.07,label = "df=10")
```

menggambar Gambar 10.11.



Gambar 10.11 Dua anggota keluarga chi-kuadrat, diplot dalam ggplot2.

BAB 11
PENGUJIAN HIPOTESIS DUA SAMPEL

Dalam berbagai bidang, seringkali muncul kebutuhan untuk membandingkan satu sampel dengan sampel lainnya. Terkadang sampelnya independen, dan terkadang mereka cocok dalam beberapa hal. Setiap sampel berasal dari populasi yang terpisah. Tujuannya adalah untuk memutuskan apakah populasi ini berbeda satu sama lain.

Biasanya, ini melibatkan pengujian hipotesis tentang rata-rata populasi. Anda juga dapat menguji hipotesis tentang varians populasi. Dalam bab ini, saya menunjukkan kepada Anda bagaimana melakukan tes ini, dan bagaimana menggunakan R untuk menyelesaikan pekerjaan.

11.1 HIPOTESIS DIBANGUN UNTUK DUA

Seperti dalam kasus satu sampel (lihat Bab 10), pengujian hipotesis dengan dua sampel dimulai dengan hipotesis nol (H_0) dan hipotesis alternatif (H_1). Hipotesis nol menetapkan bahwa perbedaan apa pun yang Anda lihat di antara kedua sampel itu semata-mata karena kebetulan. Hipotesis alternatif mengatakan, pada dasarnya, bahwa setiap perbedaan yang Anda lihat adalah nyata dan bukan karena kebetulan.

Dimungkinkan untuk memiliki uji satu sisi, di mana hipotesis alternatif menentukan arah perbedaan antara dua cara, atau uji dua sisi di mana hipotesis alternatif tidak menentukan arah perbedaan.

Untuk uji satu sisi, hipotesisnya terlihat seperti ini:

$$H_0: \mu_1 - \mu_2 = 0$$

$$H_1: \mu_1 - \mu_2 > 0$$

atau seperti ini:

$$H_0: \mu_1 - \mu_2 = 0$$

$$H_1: \mu_1 - \mu_2 < 0$$

Untuk uji dua sisi, hipotesisnya adalah:

$$H_0: \mu_1 - \mu_2 = 0$$

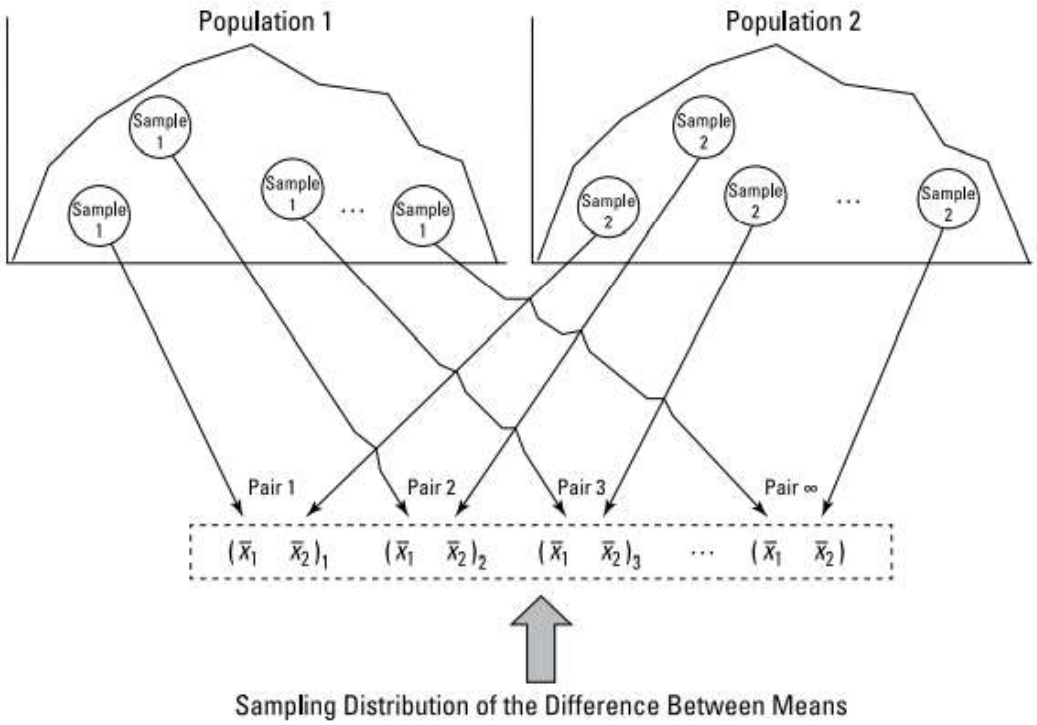
$$H_1: \mu_1 - \mu_2 \neq 0$$

Nol dalam hipotesis ini adalah kasus yang khas. Namun, dimungkinkan untuk menguji nilai apa pun — cukup ganti nilai itu dengan nol. Untuk melakukan pengujian, Anda terlebih dahulu menetapkan , probabilitas kesalahan Tipe I yang ingin Anda tanggung. (Lihat Bab 10.) Kemudian Anda menghitung rata-rata dan simpangan baku setiap sampel, kurangi satu rata-rata dari yang lain, dan gunakan rumus untuk mengubah hasilnya menjadi statistik uji.

Bandingkan statistik uji dengan distribusi sampel statistik uji. Jika di daerah penolakan yang ditentukan (sekali lagi, lihat Bab 10), tolak H0. Jika tidak, jangan tolak H_0.

11.2 PENINJAUAN KEMBALI DISTRIBUSI SAMPLING

Dalam Bab 9, saya memperkenalkan ide distribusi sampling — distribusi semua nilai statistik yang mungkin untuk ukuran sampel tertentu. Dalam bab itu, saya menjelaskan distribusi sampling mean. Dalam Bab 10, saya menunjukkan hubungannya dengan pengujian hipotesis satu sampel. Untuk pengujian hipotesis dua sampel, diperlukan distribusi sampel lain. Yang ini adalah distribusi sampling dari perbedaan antara rata-rata.



Gambar 11.1 Membuat distribusi sampling dari perbedaan antara rata-rata.

Distribusi sampling dari perbedaan antara rata-rata adalah distribusi dari semua kemungkinan nilai perbedaan antara pasangan rata-rata sampel dengan ukuran sampel tetap konstan dari pasangan ke pasangan. (Ya, itu seteguk.) Diadakan konstan dari pasangan ke pasangan berarti sampel pertama dalam pasangan selalu memiliki ukuran yang sama, dan sampel kedua dalam pasangan selalu memiliki ukuran yang sama. Kedua ukuran sampel tidak harus sama. Dalam setiap pasangan, setiap sampel berasal dari populasi yang berbeda. Semua sampel independen satu sama lain sehingga memilih individu untuk satu sampel tidak berpengaruh pada memilih individu untuk yang lain.

Gambar 11.1 menunjukkan langkah-langkah dalam membuat distribusi sampling ini. Ini adalah sesuatu yang tidak pernah Anda lakukan dalam praktik. Semuanya teoretis. Seperti yang ditunjukkan gambar, idenya adalah untuk mengambil sampel dari satu populasi dan

sampel dari yang lain, menghitung rata-rata mereka, dan mengurangi satu rata-rata dari yang lain. Kembalikan sampel ke populasi, dan ulangi berulang-ulang. Hasil dari proses tersebut adalah sekumpulan perbedaan antar sarana. Himpunan perbedaan ini adalah distribusi sampling.

11.3 MENERAPKAN TEOREMA LIMIT PUSAT

Seperti kumpulan angka lainnya, distribusi sampling ini memiliki mean dan standar deviasi. Seperti halnya dengan distribusi sampling mean (lihat Bab 9 dan 10), teorema limit pusat berlaku di sini. Menurut teorema limit pusat, jika sampelnya besar, distribusi sampling dari selisih antara rata-rata mendekati distribusi normal. Jika populasi berdistribusi normal, maka distribusi samplingnya adalah distribusi normal walaupun sampelnya kecil.

Teorema limit pusat juga memiliki sesuatu untuk dikatakan tentang mean dan standar deviasi dari distribusi sampling ini. Misalkan parameter untuk populasi pertama adalah 1 dan 1, dan parameter untuk populasi kedua adalah $\mu 2$ dan $\sigma 2$. Rata-rata dari distribusi sampling adalah

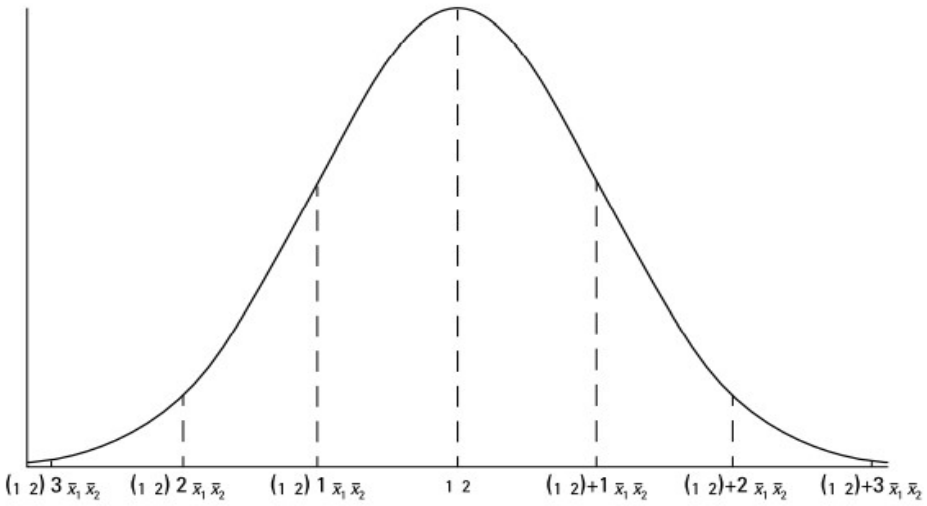
$$\mu_{\bar{x}_1 - \bar{x}_2} = \mu_1 - \mu_2$$

Standar deviasi dari distribusi sampling adalah:

$$\sigma_{\bar{x}_1 - \bar{x}_2} = \sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}$$

N_1 adalah jumlah individu dalam sampel dari populasi pertama, dan N_2 adalah jumlah individu dalam sampel dari populasi kedua. Standar deviasi ini disebut standar error selisih rata-rata.

Gambar 11.2 menunjukkan distribusi sampling beserta parameternya, sebagaimana ditentukan oleh teorema limit pusat.



Gambar 11.2: Distribusi sampling dari perbedaan antara rata-rata, menurut teorema limit pusat.

Uji Z

Karena teorema limit pusat mengatakan bahwa distribusi pengambilan sampel mendekati normal untuk sampel besar (atau untuk sampel kecil dari populasi yang terdistribusi normal), Anda menggunakan skor-z sebagai statistik uji Anda. Cara lain untuk mengatakan "gunakan skor-z sebagai statistik pengujian Anda" adalah "lakukan uji-z". Berikut rumusnya:

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sigma_{\bar{x}_1 - \bar{x}_2}}$$

Suku (μ_1-μ_2) menunjukkan selisih antara rata-rata di H_0.

Rumus ini mengubah perbedaan antara rata-rata sampel menjadi skor standar. Bandingkan skor standar dengan distribusi normal standar — distribusi normal dengan μ= 0 dan σ = 1. Jika skor berada di daerah penolakan yang ditentukan oleh σ, tolak H_0. Jika tidak, jangan tolak H_0.
Anda menggunakan rumus ini jika Anda mengetahui nilai σ_1^2 dan σ_2^2 .

Berikut ini contohnya. Bayangkan sebuah teknik pelatihan baru yang dirancang untuk meningkatkan IQ. Ambil sampel sembilan orang dan latih mereka dengan teknik baru. Ambil sampel sembilan orang lagi dan jangan beri mereka pelatihan khusus. Misalkan mean sampel untuk sampel teknik baru adalah 110,222, dan untuk sampel tanpa pelatihan adalah 101. Uji hipotesisnya adalah:

$$H_0: \mu_1 - \mu_2 \leq 0$$

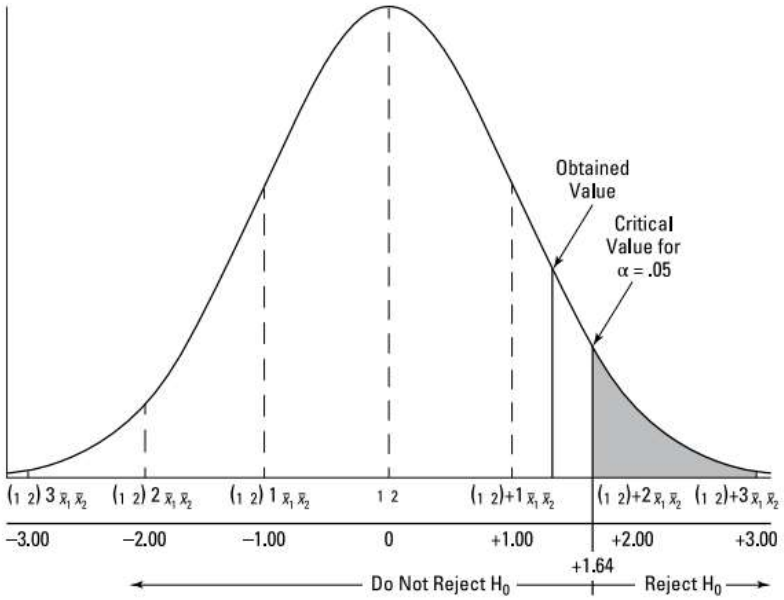
$$H_1: \mu_1 - \mu_2 > 0$$

Saya akan menyetel α pada 0,05.

IQ diketahui memiliki standar deviasi 15, dan saya berasumsi bahwa standar deviasi akan sama pada populasi orang yang terlatih dengan teknik baru. Tentu saja, populasi itu tidak ada. Asumsinya adalah bahwa jika demikian, ia harus memiliki nilai standar deviasi yang sama dengan populasi reguler dari skor IQ. Apakah mean populasi (teoretis) itu memiliki nilai yang sama dengan populasi reguler? H_0 mengatakan itu. H_1 mengatakan itu lebih besar.
Statistik uji adalah:

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sigma_{\bar{x}_1 - \bar{x}_2}} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^{\,2}}{N_1} + \frac{\sigma_2^{\,2}}{N_2}}} = \frac{(107 - 101.2)}{\sqrt{\frac{16^2}{25} + \frac{16^2}{25}}} = \frac{5.8}{4.53} = 1.28$$

Dengan = 0,05, nilai kritis z — nilai yang memotong 5 persen teratas dari area di bawah distribusi normal standar — adalah 1,645. (Anda dapat menggunakan fungsi qnorm() dari Bab 8 untuk memverifikasi ini.) Nilai statistik uji yang dihitung kurang dari nilai kritis, jadi keputusannya adalah untuk tidak menolak H0. Gambar 11.3 merangkum hal ini.



Gambar 11.3 Distribusi sampling dari perbedaan antara rata-rata, bersama dengan nilai kritis untuk = 0,05 dan nilai statistik uji yang diperoleh dalam contoh IQ.

11.4 PENGUJIAN Z UNTUK DUA SAMPEL DI R

Seperti halnya pengujian satu sampel (dijelaskan dalam Bab 10), basis R tidak menyediakan fungsi untuk uji-z dua sampel. Jika fungsi ini ada, Anda mungkin ingin berfungsi seperti ini misalnya:

```
> sample1 <-c(100,118,97,92,118,125,136,95,111)
> sample2 <-c(91,109,83,88,115,108,127,102,86)
> z.test2(sample1,sample2,15,15)
 mean1 = 110.2222    mean2 = 101
 standard error = 7.071068
 z = 1.304
 one-tailed probability = 0.096
 two-tailed probability = 0.192
```

Karena fungsi ini tidak tersedia, saya akan menunjukkan cara membuatnya. Mulailah dengan nama fungsi dan argumen:

```
z.test2 = function(x,y,popsd1,popsd2){
```

Dua argumen pertama adalah vektor data, dan dua argumen kedua adalah simpangan baku populasi. Tanda kurung kurawal kiri menunjukkan bahwa pernyataan berikutnya adalah apa yang terjadi di dalam fungsi.

Selanjutnya, Anda menginisialisasi vektor yang memiliki probabilitas satu arah:

```
one.tail.p <- NULL
```

Kemudian Anda menghitung kesalahan standar dari perbedaan antara rata-rata

```
std.error <- sqrt((popsd1^2/length(x) + popsd2^2/length(y)))
```

dan kemudian skor-z (dibulatkan)

```
z.score <- round((mean(x)-mean(y))/std.error,3)
```

Akhirnya, Anda menghitung probabilitas satu sisi yang dibulatkan:

```
one.tail.p <- round(pnorm(abs(z.score),lower.tail = FALSE),3)
```

Fungsi abs() (nilai absolut) memastikan penghitungan yang tepat untuk nilai-z negatif.
Last but not least, pernyataan cat() (concatenate-and-print) menampilkan output:

```
cat(" mean1 =", mean(x),"  ", "mean2 =", mean(y), "\n",
      "standard error =", std.error, "\n",
      "z =", z.score,"\n",
      "one-tailed probability =", one.tail.p,"\n",
      "two-tailed probability =", 2*one.tail.p )}
```

Saya menggunakan fungsi cat() seperti ini untuk kasus satu contoh di Bab 10. Tanda kurung kurawal kanan menutup fungsi tersebut.
Inilah fungsi yang baru didefinisikan:

```
z.test2 = function(x,y,popsd1,popsd2){
  one.tail.p <- NULL
  std.error <- sqrt((popsd1^2/length(x) + popsd2^2/length(y)))
  z.score <- round((mean(x)-mean(y))/std.error,3)
  one.tail.p <- round(pnorm(abs(z.score),lower.tail = FALSE),3)
  cat(" mean1 =", mean(x),"  ", "mean2 =", mean(y), "\n",
      "standard error =", std.error, "\n",
      "z =", z.score,"\n",
      "one-tailed probability =", one.tail.p,"\n",
      "two-tailed probability =", 2*one.tail.p )}
```

11.5 T UNTUK DUA

Contoh di bagian sebelumnya melibatkan situasi yang jarang Anda temui — varians populasi yang diketahui. Jika Anda mengetahui varians populasi, kemungkinan besar Anda akan mengetahui mean populasi. Jika Anda tahu artinya, Anda mungkin tidak perlu melakukan tes hipotesis tentang hal itu.

Tidak mengetahui varians menghilangkan teorema limit pusat. Ini berarti bahwa Anda tidak dapat menggunakan distribusi normal sebagai perkiraan distribusi sampling dari perbedaan antara rata-rata. Sebagai gantinya, Anda menggunakan distribusi-t, keluarga distribusi yang saya perkenalkan di Bab 9 dan berlaku untuk pengujian hipotesis satu sampel di Bab 10. Anggota dari keluarga distribusi ini berbeda satu sama lain dalam hal parameter yang disebut derajat kebebasan (df). Pikirkan df sebagai penyebut dari estimasi varians yang Anda gunakan saat menghitung nilai t sebagai statistik uji. Cara lain untuk mengatakan "hitung nilai t sebagai statistik uji" adalah "Lakukan uji-t".

Varians populasi yang tidak diketahui menyebabkan dua kemungkinan untuk pengujian hipotesis. Satu kemungkinan adalah bahwa meskipun varians tidak diketahui, Anda

memiliki alasan untuk menganggap mereka sama. Kemungkinan lainnya adalah Anda tidak dapat menganggap mereka sama. Pada bagian berikut, saya membahas kemungkinan ini.

11.6 VARIANS YANG SAMA

Bila Anda tidak mengetahui varians populasi, Anda menggunakan varians sampel untuk memperkirakannya. Jika Anda memiliki dua sampel, Anda rata-rata (mengurutkan) dua varian sampel untuk sampai pada perkiraan.

Menempatkan varians sampel bersama-sama untuk memperkirakan varians populasi disebut pooling. Dengan dua varian sampel, inilah cara Anda melakukannya:

$$s_p^{\,2} = \frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2}{(N_1 - 1) + (N_2 - 1)}$$

Dalam rumus ini, sp2 adalah singkatan dari perkiraan gabungan. Perhatikan bahwa penyebut dari taksiran ini adalah (N1–1) + (N2–1). Ini dfnya? Sangat!

Rumus untuk menghitung t adalah:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{N_1} + \frac{1}{N_2}}}$$

Ke sebuah contoh. FarKlempt Robotics mencoba memilih antara dua mesin untuk memproduksi komponen untuk mikrorobot barunya. Kecepatan adalah esensinya, sehingga perusahaan memiliki setiap mesin memproduksi sepuluh salinan komponen, dan waktu setiap produksi berjalan. Hipotesisnya adalah:

$$H_0: \mu_1 - \mu_2 = 0$$

$$H_1: \mu_1 - \mu_2 \neq 0$$

Mereka menetapkan pada 0,05. Ini adalah tes dua sisi karena mereka tidak tahu sebelumnya mesin mana yang mungkin lebih cepat. Tabel 11.1 menyajikan data untuk waktu produksi dalam menit.

Tabel 11.1 Contoh Statistik dari Studi Mesin FarKlempt

	Mesin 1	Mesin 2
Rata-rata Waktu Produksi	23.00	20.00
Standar Deviasi	2.71	2.79
Ukuran sampel	10	10

Estimasi gabungan dari 2 adalah

$$s_p^{\,2} = \frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2}{(N_1 - 1) + (N_2 - 1)} = \frac{(10-1)(2.71)^2 + (10-1)(2.79)^2}{(10-1) + (10-1)}$$

$$= \frac{(9)(2.71)^2 + (9)(2.79)^2}{(9) + (9)} = \frac{66 + 70}{18} = 7.56$$

Perkiraan adalah 2,75, akar kuadrat dari 7,56.
Statistik uji adalah:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{N_1} + \frac{1}{N_2}}} = \frac{(23-20)}{2.75\sqrt{\frac{1}{10} + \frac{1}{10}}} = \frac{3}{1.23} = 2.44$$

Untuk statistik uji ini, df = 18, penyebut estimasi varians. Pada distribusi-t dengan 18 df, nilai kritisnya adalah 2,10 untuk ekor sisi kanan (atas) dan –2,10 untuk ekor sisi kiri (bawah). Jika Anda tidak percaya, terapkan qt(). (Lihat Bab 10.) Nilai statistik uji yang dihitung lebih besar dari 2,10, sehingga keputusannya adalah menolak H_0. Data memberikan bukti bahwa Mesin 2 secara signifikan lebih cepat daripada Mesin 1. (Anda dapat menggunakan kata signifikan setiap kali Anda menolak H_0).

11.7 UJI-T DI R

Berikut adalah beberapa vektor untuk data sampel dalam contoh di bagian sebelumnya:

```
machine1 <-c(24.58, 22.09, 23.70, 18.89, 22.02, 28.71, 24.44,
          20.91, 23.83, 20.83)
```

```
machine2 <- c(21.61, 19.06, 20.72, 15.77, 19, 25.88, 21.48,
          17.85, 20.86, 17.77)
```

R menyediakan dua cara untuk melakukan uji-t. Keduanya melibatkan t.test(), yang saya gunakan di Bab 9 dan 10.

Bekerja dengan dua vektor

Berikut cara menguji hipotesis dengan dua vektor dan asumsi varians yang sama:

```
t.test(machine1,machine2,var.equal = TRUE, alternative="two.
          sided", mu=0)
```

Argumen alternatif=dua sisi mencerminkan jenis hipotesis alternatif yang ditentukan dalam contoh, dan argumen terakhir menunjukkan perbedaan yang dihipotesiskan antara rata-rata. Menjalankan fungsi itu menghasilkan output ini:

```
Two Sample t-test
data:  c(24.58, 22.09, 23.7, 18.89, 22.02, 28.71, 24.44, 20.91,
         23.83,  ... and c(21.61, 19.06, 20.72, 15.77, 19,
         25.88, 21.48, 17.85, 20.86,  ...
t = 2.4396, df = 18, p-value = 0.02528
alternative hypothesis: true difference in means is not
         equal to 0
```

```
95 percent confidence interval:
 0.4164695 5.5835305
sample estimates:
mean of x mean of y
       23        20
```

Nilai t dan nilai p yang rendah menunjukkan bahwa Anda dapat menolak hipotesis nol. Mesin 2 secara signifikan lebih cepat daripada Mesin 1.

Bekerja dengan bingkai data dan formula

Cara lain untuk melakukan pengujian ini adalah dengan membuat kerangka data dan kemudian menggunakan rumus yang terlihat seperti ini:

```
prod.time ~ machine
```

Rumus mengungkapkan gagasan bahwa waktu produksi tergantung pada mesin yang Anda gunakan. Meskipun tidak perlu melakukan tes dengan cara ini, ada baiknya membiasakan diri dengan formula. Saya menggunakannya sedikit di bab-bab selanjutnya.

Hal pertama yang harus dilakukan adalah membuat bingkai data dalam format panjang. Pertama, Anda membuat vektor untuk 20 kali produksi — pertama kali mesin1 dan kemudian mesin2:

```
prod.time <- c(machine1,machine2)
```

Selanjutnya, Anda membuat vektor dari dua nama mesin:

```
machine <-c("machine1","machine2")
```

Kemudian Anda mengubah vektor itu menjadi vektor sepuluh pengulangan "mesin1" diikuti dengan sepuluh pengulangan "mesin2". Ini sedikit rumit, tapi begini caranya:

```
machine <- rep(machine, times = c(10,10))
```

Dan bingkai datanya adalah:

```
FarKlempt.frame <-data.frame(machine,prod.time)
```

Enam baris pertamanya adalah:

```
> head(FarKlempt.frame)
   machine prod.time
1 machine1     24.58
2 machine1     22.09
3 machine1     23.70
4 machine1     18.89
5 machine1     22.02
6 machine1     28.71
```

Ini menghasilkan output yang sama dengan versi dua vektor.

Memvisualisasikan hasilnya

Dalam penelitian seperti pada bagian sebelumnya, dua cara untuk menyajikan hasil adalah diagram kotak dan grafik batang.

Plot kotak

Boxplot menggambarkan data di setiap sampel bersama dengan median sampel (seperti yang dijelaskan dalam Bab 3). Mereka mudah dibuat di basis R dan di ggplot2. Untuk grafik R dasar, kodenya agak mirip dengan metode rumus untuk t.test():

```
with (FarKlempt.frame,boxplot(prod.time~machine, xlab =
         "Machine", ylab="Production Time (minutes)"))
```

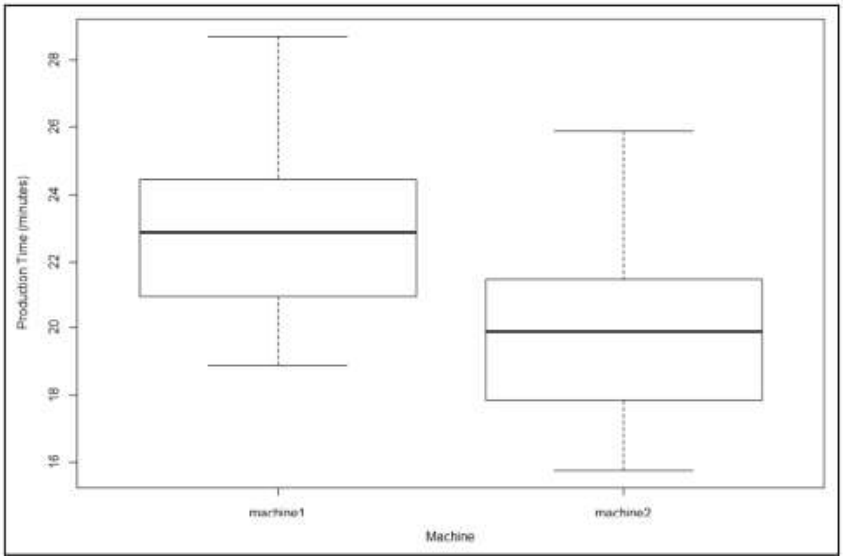
Plotnya terlihat seperti Gambar 11.4.

Gambar 11.5 menunjukkan boxplot yang dirender dalam ggplot2. Kode yang menghasilkan boxplot itu adalah:

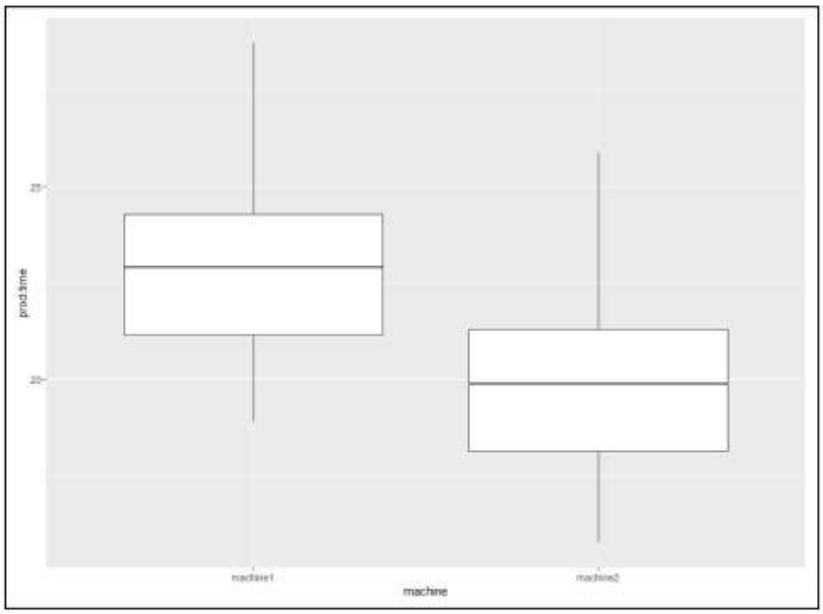
```
ggplot(FarKlempt.frame, aes(x=machine, y=prod.time))+
  stat_boxplot(geom="errorbar", width =.5) +
  geom_boxplot()
```

Satu-satunya fungsi baru adalah stat_boxplot(), yang menambahkan garis tegak lurus ke ujung setiap whisker. Lebar default dari garis tersebut adalah lebar kotak. Saya menambahkan lebar =.5 untuk memotong lebar itu menjadi dua.

Di ggplot2, stat adalah cara meringkas data sehingga fungsi geom dapat menggunakannya. Fungsi stat yang digunakan di sini menghitung komponen untuk boxplot. Anda menggunakannya untuk mengganti tampilan default boxplot — yang tanpa garis tegak lurus di akhir setiap whisker. Pada contoh sebelumnya (dan contoh berikutnya), Anda menggunakan stat= "identity" untuk menginstruksikan geom_bar() untuk menggunakan data tabel daripada menghitung.



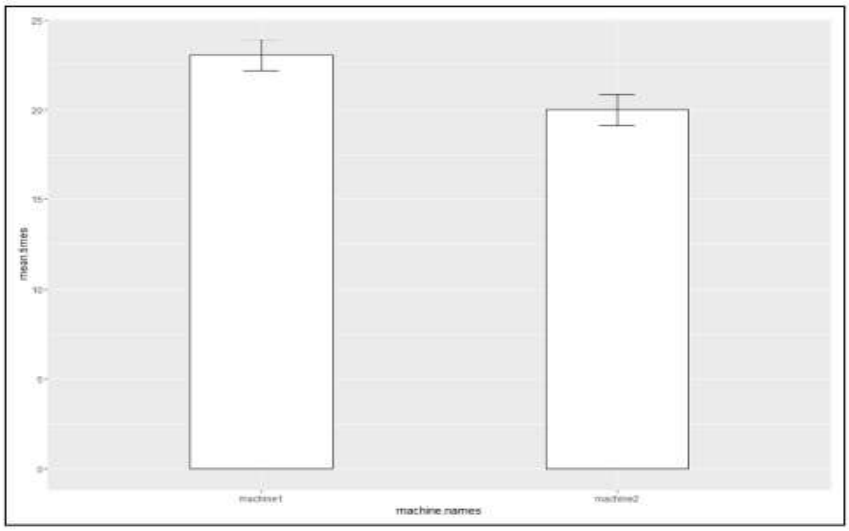
Gambar 11.4 Boxplot data FarKlempt Machines di basis R.



Gambar 11.5 Boxplot data FarKlempt Machines di ggplot2.

Grafik batang

Secara tradisional, peneliti melaporkan dan memplot sarana sampel dan kesalahan standar. Sangat mudah untuk melakukannya di ggplot2. Gambar 11.6 menunjukkan apa yang saya maksud.



Gambar 11.6 Cara Mesin FarKlempt dan kesalahan standar.

Batang berbentuk t yang memanjang di atas dan di bawah bagian atas setiap batang adalah batang kesalahan yang menunjukkan kesalahan standar mean.

Untuk menggunakan ggplot2, Anda harus membuat kerangka data nama mesin, waktu rata-rata, dan kesalahan standar. Tiga vektor yang akan membentuk kerangka data adalah:

```
machine.names <-c("machine1","machine2")
mean.times <- c(mean(machine1),mean(machine2))
se.times <- c(sd(machine1)/sqrt(length(machine1)),
          sd(machine2)/sqrt(length(machine2)))
```

Bingkai data kemudian:

```
FKmeans.frame <-data.frame(machine.names,mean.times,se.times)
```

Ini terlihat seperti ini:

```
> FKmeans.frame
  machine.names mean.times  se.times
1      machine1         23 0.8570661
2      machine2         20 0.8818339
```

Kode untuk membuat Gambar 11.6 adalah:

```
ggplot(FKmeans.frame, aes(x=machine.names, y=mean.
          times))+
```

```
 geom_bar(stat="identity", width=.4,color="black",
         fill="white")+
 geom_errorbar(aes(ymin=mean.times-se.times, ymax=mean.
         times+se.times),width=.1)
```

Fungsi pertama mengatur panggung dengan pemetaan estetika, dan yang kedua memplot bar. Argumen stat = identity menginstruksikan geom_bar untuk menggunakan statistik tabel daripada menghitung instance machine1 dan machine2. Argumen lain mengatur tampilan bar. Fungsi ketiga adalah geom yang memplot bilah kesalahan. Pemetaan estetika mengatur titik minimum dan titik maksimum untuk setiap bilah kesalahan. Argumen lebar menetapkan lebar untuk garis tegak lurus di akhir setiap bilah kesalahan.

Di sebagian besar publikasi ilmiah, Anda melihat grafik seperti ini hanya dengan bilah kesalahan positif — yang memanjang di atas rata-rata. Untuk membuat grafik seperti itu dalam contoh ini, setel ymin=mean.times daripada ymin=mean.times-se.times.

Seperti p dan q: varians yang tidak sama

Kasus varians yang tidak sama menghadirkan tantangan. Seperti yang terjadi, ketika varians tidak sama, distribusi t dengan (N1-1) + (N2-1) derajat kebebasan tidak sedekat pendekatan dengan distribusi sampling seperti yang diinginkan ahli statistik. Ahli statistik memenuhi tantangan ini dengan mengurangi derajat kebebasan. Untuk mencapai pengurangan, mereka menggunakan rumus yang cukup terlibat yang tergantung pada standar deviasi sampel dan ukuran sampel.

Karena variansnya tidak sama, estimasi gabungan tidak sesuai. Jadi, Anda menghitung uji-t dengan cara yang berbeda:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

Anda mengevaluasi statistik uji terhadap anggota keluarga distribusi-t yang memiliki derajat kebebasan tereduksi.

Inilah yang dihasilkan t.test() untuk contoh FarKlempt jika saya menganggap variansnya tidak sama:

```
with (FarKlempt.frame,t.test(prod.time~machine,
                             var.equal = FALSE,
                             alternative="two.sided",
                             mu=0))
```

```
Welch Two Sample t-test
data:  prod.time by machine
t = 2.4396, df = 17.985, p-value = 0.02529
alternative hypothesis: true difference in means is not
          equal to 0
95 percent confidence interval:
 0.4163193 5.5836807
sample estimates:
mean in group machine1 mean in group machine2
                    23                     20
```

Anda dapat melihat sedikit pengurangan derajat kebebasan. Variansnya sangat dekat sehingga hanya sedikit yang berubah.

11.8 PENGUJIAN HIPOTESIS UNTUK SAMPEL YANG DIPASANGKAN

Dalam uji hipotesis yang saya uraikan sejauh ini, sampel-sampelnya tidak bergantung satu sama lain. Memilih individu untuk satu sampel tidak berpengaruh pada pilihan individu untuk sampel lainnya.

Terkadang, sampelnya cocok. Kasus yang paling jelas adalah ketika individu yang sama memberikan skor di bawah masing-masing dari dua kondisi — seperti dalam studi sebelum-sesudah. Misalkan sepuluh orang berpartisipasi dalam program penurunan berat badan. Mereka menimbang sebelum memulai program dan sekali lagi setelah satu bulan mengikuti program. Data yang penting adalah himpunan perbedaan sebelum-sesudah. Tabel 11-2 menunjukkan data:

Idenya adalah memikirkan perbedaan ini sebagai sampel skor dan memperlakukannya seperti yang Anda lakukan dalam uji-t satu sampel. (Lihat Bab 10.)

Anda melakukan tes pada hipotesis ini:

$$H_0: \mu_d \leq 0$$

$$H_1: \mu_d > 0$$

Huruf d dalam subskrip berarti "perbedaan". Tetapkan α= 0,05.

Tabel 11.2 Data untuk Contoh Penurunan Berat Badan

Orang	Berat Sebelum Program	Berat Setelah Satu Bulan	Perbedaan
1	198	194	**4**
2	201	203	**-2**
3	210	200	**10**

4	185	183	**2**
5	204	200	**4**
6	156	153	**3**
7	167	166	**1**
8	197	197	**0**
9	220	215	**5**
10	186	184	**2**
Mean			**2.9**
Standar Deviasi			**3.25**

Rumus untuk uji-t semacam ini adalah:

$$t = \frac{\bar{d} - \mu_d}{s_d}$$

Dalam rumus ini, $\bar{d}$ adalah rata-rata dari perbedaan. Untuk menemukan $s_{\bar{d}}$, Anda menghitung simpangan baku dari perbedaan dan membagi dengan akar kuadrat dari jumlah pasangan:

$$s_{\bar{d}} = \frac{s}{\sqrt{N}}$$

Df adalah N 1 (di mana N adalah jumlah pasangan).
Dari Tabel 11.2,

$$t = \frac{\bar{d} - \mu_d}{s_d} = \frac{2.9}{\left(3.25 / \sqrt{10}\right)} = 2.82$$

Dengan df = 9 (Jumlah pasangan – 1), nilai kritis untuk α = .05 adalah 1,83. (Gunakan qt() untuk memverifikasi.) Nilai yang dihitung melebihi nilai ini, jadi keputusannya adalah menolak H_0.

11.9 UJI-T SAMPEL BERPASANGAN DALAM R

Untuk uji-t sampel berpasangan, rumusnya sama dengan uji-t sampel independen. Seperti yang akan Anda lihat, Anda menambahkan argumen. Berikut data dari Tabel 11.2:

```
before <-c(198,201,210,185,204,156,167,197,220,186)
after <- c(194,203,200,183,200,153,166,197,215,184)
```

Dan uji-t:

```
t.test(before,after,alternative = "greater",paired=TRUE)
```

Argumen terakhir itu, tentu saja, menentukan uji sampel berpasangan. Nilai default untuk yang itu adalah FALSE.
Menjalankan tes itu menghasilkan:

```
            Paired t-test

data:  before and after
t = 2.8241, df = 9, p-value = 0.009956
alternative hypothesis: true difference in means is greater
          than 0
95 percent confidence interval:
 1.017647      Inf
sample estimates:
mean of the differences
                   2.9
```

Karena nilai p yang sangat rendah, Anda menolak hipotesis nol.

11.10 MENGUJI DUA VARIANS

Pengujian hipotesis dua sampel yang saya uraikan dalam bab ini berkaitan dengan sarana. Dimungkinkan juga untuk menguji hipotesis tentang varians.

Di bagian ini, saya memperluas contoh manufaktur satu varian yang saya gunakan di Bab 10. FarKlempt Robotics, Inc., menghasilkan bagian yang harus memiliki panjang tertentu dengan variabilitas yang sangat kecil. Perusahaan sedang mempertimbangkan dua mesin untuk memproduksi bagian ini, dan ingin memilih salah satu yang menghasilkan variabilitas paling sedikit. FarKlempt Robotics mengambil sampel suku cadang dari setiap mesin, mengukurnya, menemukan varians untuk setiap sampel, dan melakukan uji hipotesis untuk melihat apakah varians satu mesin secara signifikan lebih besar daripada varian lainnya.
Hipotesisnya adalah:

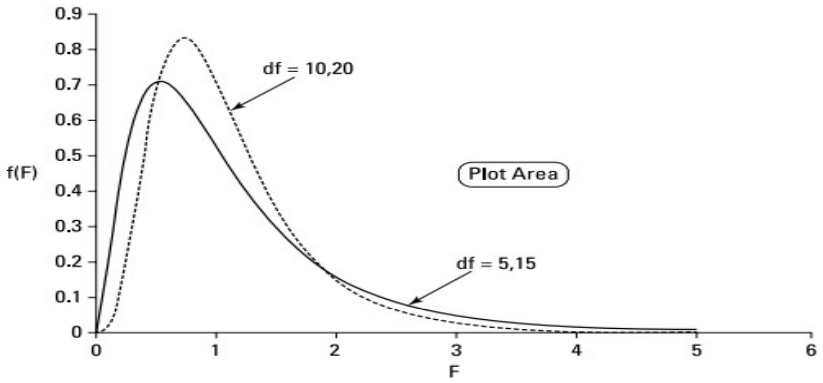
$$H_0: \sigma_1^2 = \sigma_2^2$$

$$H_1: \sigma_1^2 \neq \sigma_2^2$$

Seperti biasa, α adalah item yang harus dimiliki. Seperti biasa, saya set ke .05.

Saat Anda menguji dua varians, Anda tidak mengurangi satu dari yang lain. Sebagai gantinya, Anda membagi satu dengan yang lain untuk menghitung statistik uji. Sir Ronald Fisher adalah ahli statistik terkenal yang mengerjakan matematika dan keluarga distribusi untuk bekerja dengan varians dengan cara ini. Statistik uji dinamai untuk menghormatinya. Ini disebut F-ratio dan tesnya adalah F-test. Keluarga distribusi untuk pengujian ini disebut distribusi-F.

Tanpa membahas semua matematika, saya hanya akan memberi tahu Anda bahwa, sekali lagi, df adalah parameter yang membedakan satu anggota keluarga dari yang lain. Apa yang berbeda tentang keluarga ini adalah bahwa dua estimasi varians terlibat, sehingga setiap anggota keluarga dikaitkan dengan dua nilai df, bukan satu seperti pada uji-t. Perbedaan lain antara distribusi-F dan yang lain yang telah Anda lihat adalah bahwa F tidak dapat memiliki nilai negatif. Gambar 11.7 menunjukkan dua anggota keluarga distribusi-F.



Gambar 11.7 Dua anggota keluarga distribusi-F.

Statistik uji adalah

$$F = \frac{\text{larger } s^2}{\text{smaller } s^2}$$

Misalkan FarKlempt Robotics menghasilkan 10 bagian dengan Mesin 1 dan menemukan varians sampel 0,81 inci persegi. Ini menghasilkan 15 bagian dengan Mesin 2 dan menemukan varians sampel 0,64 inci persegi. Dapatkah perusahaan menolak H_0?
Menghitung statistik uji,

$$F = \frac{.81}{.64} = 1.27$$

Dfnya adalah 9 dan 14: Estimasi varians pada pembilang rasio-F didasarkan pada 10 kasus, dan estimasi varians pada penyebut didasarkan pada 15 kasus. Ketika df adalah 9 dan 14 dan merupakan uji dua sisi pada = .05, nilai kritis F adalah 3,21. (Sebentar, saya tunjukkan fungsi R yang menghitung ini.) Nilai yang dihitung kurang dari nilai kritis, jadi keputusannya adalah untuk tidak menolak H_0.

Itu membuat perbedaan df mana yang ada di pembilang dan mana df yang ada di penyebut. Distribusi F untuk df = 9 dan df = 14 berbeda dengan distribusi F untuk df = 14 dan df = 9. Misalnya, nilai kritis dalam kasus terakhir adalah 3,80, bukan 3,21.

F-pengujian di R

R menyediakan fungsi untuk menguji hipotesis seperti yang ada pada contoh dua mesin FarKlempt Robotics. Ini disebut var.test(). Haruskah itu disebut F.test()? Ya, mungkin. Poin penting adalah untuk tidak mengacaukan fungsi ini dengan varTest(), yang saya gunakan di Bab 10 untuk menguji hipotesis tentang varians sampel tunggal (dengan chi-kuadrat). Fungsi itu ada dalam paket EnvStats.

Untuk menerapkan var.test(), Anda terlebih dahulu membuat vektor yang menyimpan data untuk bagian yang dihasilkan oleh mesin 1 dan mesin 2:

```
> var.test(m1.parts,m2.parts,ratio=1,alternative="two.sided")

Results of Hypothesis Test

--------------------------
Null Hypothesis:                 ratio of variances = 1

Alternative Hypothesis:          True ratio of variances is not equal to 1

Test Name:                       F test to compare two variances

Estimated Parameter(s):          ratio of variances = 1.26482

Data:                            m1.parts and m2.parts

Test Statistic:                  F = 1.26482

Test Statistic Parameters:       num df   =  9
                                 denom df = 14
P-value:                         0.6690808

95% Confidence Interval:         LCL = 0.3941108
                                 UCL = 4.8037262
```

Rasio-F yang rendah dan nilai-p yang tinggi menunjukkan bahwa Anda tidak dapat menolak hipotesis nol. (Sedikit perbedaan antara rasio-F ini dan yang dihitung dalam contoh adalah karena pembulatan).

F dalam hubungannya dengan t

Salah satu penggunaan distribusi-F adalah dalam hubungannya dengan uji-t untuk sampel independen. Sebelum Anda melakukan uji-t, Anda menggunakan F untuk membantu memutuskan apakah akan mengasumsikan varians yang sama atau varians yang tidak sama dalam sampel.

Dalam contoh uji-t varians yang sama yang saya tunjukkan sebelumnya, standar deviasinya adalah 2,71 dan 2,79. Variansnya adalah 7,34 dan 7,78. F-rasio varians ini adalah

$$F = \frac{7.78}{7.34} = 1.06$$

Setiap sampel didasarkan pada sepuluh pengamatan, sehingga df = 9 untuk setiap varians sampel. Rasio-F 1,06 memotong 47 persen teratas dari distribusi-F yang df-nya 9 dan 9, jadi aman untuk menggunakan versi uji-t varian yang sama untuk data ini. Bagaimana semua ini terjadi dalam konteks pengujian hipotesis? Pada kesempatan langka, H_0 adalah hasil yang diinginkan dan Anda lebih suka tidak menolaknya. Dalam hal ini, Anda menumpuk tumpukan agar tidak menolak dengan menetapkan pada level tinggi sehingga perbedaan kecil menyebabkan Anda menolak H_0.

Ini adalah salah satu kesempatan langka. Lebih baik menggunakan uji t varians yang sama, yang biasanya memberikan lebih banyak derajat kebebasan daripada uji-t varians yang tidak sama. Menetapkan nilai yang tinggi (.20 adalah nilai yang baik) untuk uji-F memungkinkan Anda untuk percaya diri saat mengasumsikan varians yang sama.

11.11 BEKERJA DENGAN F-DISTRIBUTION

Sama seperti keluarga distribusi lain yang saya bahas sebelumnya (normal, t, chi-kuadrat), R menyediakan fungsi untuk menangani distribusi-F: qf() memberikan informasi kuantil, df() menyediakan fungsi kerapatan, pf() menyediakan fungsi kepadatan kumulatif, dan rf() menghasilkan angka acak. Perhatikan bahwa di seluruh bagian ini, saya mengeja "derajat kebebasan" daripada menggunakan singkatan "df" seperti yang saya lakukan di tempat lain. Itu untuk menghindari kebingungan dengan fungsi kepadatan df().

Nilai kritis yang saya rujuk sebelumnya untuk uji F dua arah dengan 9 dan 14 derajat kebebasan adalah:

```
> qf(.025,9,14,lower.tail = FALSE)
[1] 3.2093
```

Ini adalah tes dua sisi pada = 0,05, jadi 0,025 ada di setiap ekor.
Untuk menonton df() dan pf() beraksi, Anda membuat vektor untuk mereka operasikan:

```
F.scores <-seq(0,5,1)
```

Dengan 9 dan 14 derajat kebebasan, kerapatan (pembulatan) untuk nilai-nilai ini adalah:

```
> round(df(F.scores,9,14),3)
[1] 0.000 0.645 0.164 0.039 0.011 0.004
```

Kepadatan kumulatif (dibulatkan) adalah:

```
> round(pf(F.scores,9,14),3)
[1] 0.000 0.518 0.882 0.968 0.990 0.996
```

Untuk menghasilkan lima angka acak dari anggota keluarga-F ini:

```
> rf(5,9,14)
[1] 0.6409125 0.4015354 1.1601984 0.6552502 0.8652722
```

11.2 MEMVISUALISASIKAN DISTRIBUSI-F

Seperti yang saya katakan, memvisualisasikan distribusi membantu Anda mempelajarinya. Distribusi-F tidak terkecuali, dan dengan fungsi kepadatan dan ggplot2, mudah untuk memplotnya. Tujuan saya di bagian ini adalah untuk menunjukkan kepada Anda bagaimana menggunakan ggplot2 untuk membuat grafik yang terlihat seperti Gambar 11-7, yang menggambarkan distribusi-F dengan 5 dan 15 derajat kebebasan dan lainnya dengan 10 dan 20 derajat kebebasan. Untuk membuat grafik terlihat seperti gambar, saya harus menambahkan anotasi dengan panah yang menunjuk ke kurva yang sesuai.
Mulailah dengan vektor nilai untuk df() untuk melakukan pekerjaannya pada:

```
F.values <-seq(0,5,.05)
```

Kemudian buat vektor kerapatan untuk distribusi-F dengan 5 dan 15 derajat kebebasan:

```
F5.15 <- df(F.values,5,15)
```

dan satu lagi untuk distribusi-F dengan 10 dan 20 derajat kebebasan:

```
F10.20 <- df(F.values,10,20)
```

Sekarang untuk bingkai data untuk ggplot2:

```
F.frame <- data.frame(F.values,F5.15,F10.20)
```

Seperti inilah enam baris pertama F.frame:

```
> head(F.frame)
  F.values      F5.15      F10.20
1     0.00 0.00000000 0.000000000
2     0.05 0.08868702 0.001349914
3     0.10 0.21319965 0.015046816
4     0.15 0.33376038 0.053520748
5     0.20 0.43898395 0.119815721
6     0.25 0.52538762 0.208812406
```

Ini dalam format lebar. Seperti yang saya tunjukkan sebelumnya, ggplot() lebih menyukai format panjang, di mana nilai data ditumpuk di atas satu sama lain dalam satu kolom. Ini disebut mencairkan data dan merupakan bagian tak terpisahkan dari paket reshape2. (Pada tab Packages, temukan kotak centang di sebelah reshape2. Jika tidak dicentang, klik di atasnya).

Untuk membentuk kembali data dengan tepat,

```
F.frame.melt <- melt(F.frame,id="F.values")
```

Argumen id memberi tahu melt() apa yang tidak boleh disertakan dalam tumpukan. (F.values adalah "pengidentifikasi," dengan kata lain.) Selanjutnya, tetapkan nama kolom yang bermakna:

```
colnames(F.frame.melt)=c("F","deg.fr","density")
```

Enam baris pertama dari bingkai data yang dilebur adalah:

```
> head(F.frame.melt)
     F deg.fr    density
1 0.00  F5.15 0.00000000
2 0.05  F5.15 0.08868702
3 0.10  F5.15 0.21319965
4 0.15  F5.15 0.33376038
5 0.20  F5.15 0.43898395
6 0.25  F5.15 0.52538762
```

Untuk memulai visualisasi, pernyataan pertama, seperti biasa, adalah ggplot():

```
ggplot(F.frame.melt,aes(x=F,y=density,group=deg.fr))
```

Argumen pertama adalah bingkai data. Dua pemetaan estetika pertama mengaitkan F dengan sumbu x, dan kepadatan dengan sumbu y. Pemetaan ketiga membentuk grup berdasarkan variabel deg.fr.
Selanjutnya, Anda menambahkan geom_line:

```
geom_line(stat="identity",aes(linetype=deg.fr))
```

Argumen stat memberi tahu fungsi geom untuk menggunakan data tabel. Pemetaan estetika mengaitkan linetype ("padat" dan "bertitik" adalah nilai default) dengan deg.fr.

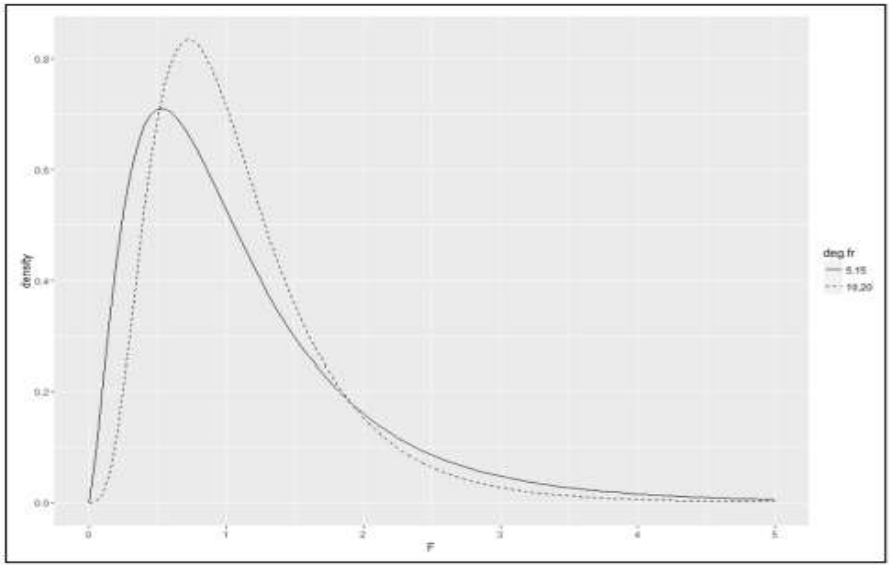
Jika Anda lebih suka "padat" dan "putus-putus", seperti pada Gambar 11.7, Anda harus mengubahnya secara manual:

```
scale_linetype_manual(values = c("solid","dashed"),
          labels = c("5,15","10,20"))
```

Nilai dan label akan muncul dalam legenda yang secara otomatis dibuat oleh tata bahasa. Berikut kode sejauh ini:

```
ggplot(F.frame.melt,aes(x=F,y=density,group=deg.fr)) +
  geom_line(stat="identity",aes(linetype=deg.fr))+
  scale_linetype_manual(values = c("solid","dashed"),
          labels = c("5,15","10,20"))
```

Gambar 11.8 menunjukkan kemajuan.



Gambar 11.8 Dua anggota keluarga distribusi-F dalam ggplot2 — graf perantara.

Tetapi tujuannya adalah untuk membuat grafik tanpa legenda, seperti Gambar 11-7. Anda menggunakan guides() untuk memanipulasi legenda, dan legenda didasarkan pada linetype. Jadi, inilah cara menghapus legenda:

```
guides(linetype=FALSE)
```

Terakhir, tambahkan beberapa anotasi yang menunjukkan derajat kebebasan untuk setiap kurva. Penjelasan untuk kurva dengan 10 dan 20 derajat kebebasan adalah:

```
annotate(geom="text",x=1.98,y=.78,label="df=10,20")
```

Argumen pertama menentukan geom teks, dua berikutnya memposisikan geom teks dalam grafik (berpusat pada koordinat yang ditunjukkan), dan argumen keempat menetapkan apa yang dikatakan anotasi. Sekarang untuk panah yang menunjuk dari anotasi ke kurva. Ini terdiri dari segmen garis dan panah. Bagian segmen garis dari panah adalah geom segmen. Bagian kepala panah dari panah adalah produk dari fungsi yang disebut arrow(), yang ada dalam paket grid. Pada tab Paket, temukan kotak centang di sebelah kisi, dan klik.
Fungsi annotate() lain mengatur panah:

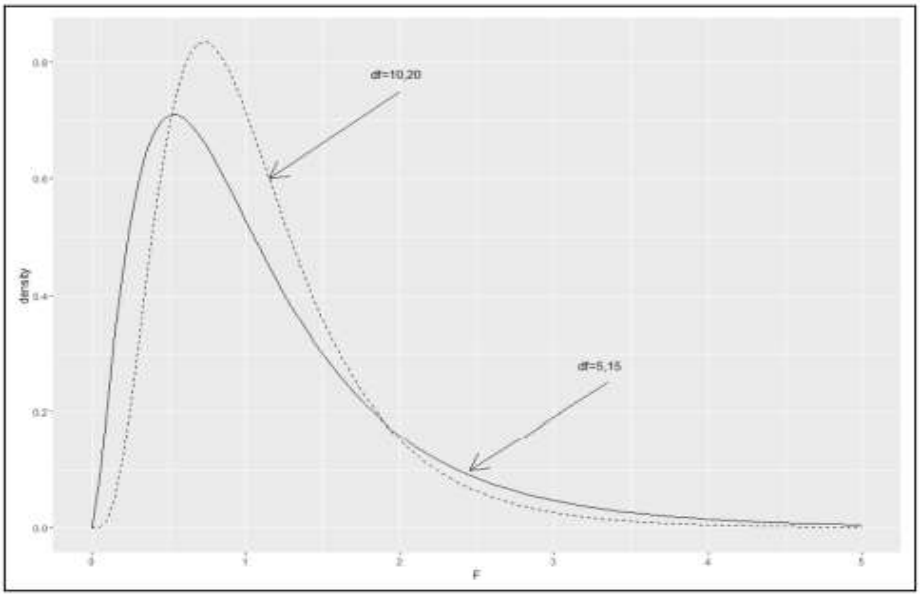
```
annotate(geom="segment",x=2.0,xend=1.15,y=0.75,yend = .6,
        arrow=arrow())
```

Empat argumen setelah fungsi geom menemukan titik awal dan titik akhir untuk segmen. Argumen terakhir memplot panah. Menemukan nilai untuk titik awal dan titik akhir dapat melibatkan beberapa percobaan dan kesalahan. Bukan ide yang buruk untuk memplot panah terlebih dahulu dan kemudian teks.

Berikut kode untuk semuanya, termasuk dua fungsi annotate() untuk kurva lainnya:

```
ggplot(F.frame.melt,aes(x=F,y=density,group=deg.fr)) +
  geom_line(stat="identity",aes(linetype=deg.fr))+
  scale_linetype_manual(values = c("solid","dashed"),
          labels = c("5,15","10,20")) +
  guides(linetype=FALSE) +
  annotate(geom="text",x=1.98,y=.78,label="df=10,20")+
  annotate(geom="segment",y=0.75,yend=.6,
  arrow=arrow())+
  annotate(geom="text",x=3.3,y=.28,label="df=5,15")+
  annotate(geom="segment",x = 3.35, xend=2.45,y =0.25,
          yend=.1,arrow=arrow())
```

Dan Gambar 11.9 adalah hasilnya.



Gambar 11.9 Dua anggota keluarga distribusi-F di ggplot2 — produk akhir.

Bereksperimenlah dengan nilai derajat kebebasan lain dan lihat seperti apa kurva itu.

BAB 12
MENGUJI LEBIH DARI DUA SAMPEL

Statistik akan terbatas jika Anda hanya dapat membuat kesimpulan tentang satu atau dua sampel. Dalam bab ini, saya membahas prosedur untuk menguji hipotesis tentang tiga atau lebih sampel. Saya menunjukkan apa yang harus dilakukan ketika sampel tidak tergantung satu sama lain, dan apa yang harus dilakukan ketika tidak. Dalam kedua kasus, saya membahas apa yang harus dilakukan setelah Anda menguji hipotesis. Saya juga membahas fungsi R yang bekerja untuk Anda.

12.1 MENGUJI LEBIH DARI DUA

Bayangkan situasi ini. Perusahaan Anda meminta Anda untuk mengevaluasi tiga metode berbeda untuk melatih karyawannya melakukan pekerjaan tertentu. Anda secara acak menetapkan 30 karyawan ke salah satu dari tiga metode. Rencana Anda adalah melatihnya, mengujinya, mentabulasi hasilnya, dan membuat beberapa kesimpulan. Sebelum Anda dapat menyelesaikan studi, tiga orang meninggalkan perusahaan — satu dari kelompok Metode 1 dan dua dari kelompok Metode 3. Tabel 12.1 menunjukkan data.

Tabel 12.1 Data dari Tiga Metode Pelatihan

	Method 1	Method 2	Method 3
	95	83	68
	91	89	75
	89	85	79
	90	89	74
	99	81	75
	88	89	81
	96	90	73
	98	82	77
	95	84	
		80	
Mean	93.44	85.20	75.25
Perbedaan	16.28	14.18	15.64
Standar Deviasi	4.03	3.77	3.96

Apakah ketiga metode tersebut memberikan hasil yang berbeda, atau sangat mirip sehingga Anda tidak dapat membedakannya? Untuk memutuskan, Anda harus melakukan uji hipotesis.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

$$H_1: \text{Not } H_0$$

Dengan α = 0,05.

Masalah pelik

Menemukan perbedaan di antara tiga kelompok terdengar cukup mudah, terutama jika Anda telah membaca Bab 11. Ambil rata-rata skor dari Metode 1, rata-rata skor dari Metode 2, dan lakukan uji-t untuk melihat apakah mereka berbeda. Ikuti prosedur yang sama untuk Metode 1 versus Metode 3, dan untuk Metode 2 versus Metode 3. Jika setidaknya satu dari uji-t menunjukkan perbedaan yang signifikan, tolak H0. Tidak ada untuk itu, kan? Salah. Jika Anda adalah 0,05 untuk setiap uji-t, Anda menyiapkan diri Anda untuk kesalahan Tipe I dengan probabilitas lebih tinggi dari yang Anda rencanakan. Probabilitas bahwa setidaknya satu dari tiga uji-t menghasilkan perbedaan yang signifikan jauh di atas 0,05. Faktanya, ini .14, yang sangat jauh dari yang dapat diterima. (Matematika di balik penghitungan angka itu sedikit terlibat, jadi saya tidak akan menguraikannya).

Dengan lebih dari tiga sampel, situasinya menjadi lebih buruk. Empat kelompok memerlukan enam uji-t, dan probabilitas bahwa setidaknya salah satu dari mereka signifikan adalah 0,26. Tabel 12.2 menunjukkan apa yang terjadi dengan bertambahnya jumlah sampel.

Tabel 12.2 Alpha yang Meningkat Luar Biasa

Jumlah Sampel t	Jumlah Tes	Pr (Setidaknya Satu Signifikan t)
3	3	.14
4	6	.26
5	10	.40
6	15	.54
7	21	.66
8	28	.76
9	36	.84
10	45	.90

Melakukan beberapa t-test jelas bukan jawabannya. Apa yang kamu kerjakan?

Sebuah solusi

Ini perlu untuk mengambil pendekatan yang berbeda. Idenya adalah untuk berpikir dalam istilah varians daripada berarti. Saya ingin Anda memikirkan varians dengan cara yang sedikit berbeda. Rumus untuk memperkirakan varians populasi, ingat, adalah:

$$s^2 = \frac{\sum (x - \bar{x})^2}{N - 1}$$

Karena varians hampir merupakan mean dari kuadrat deviasi dari mean, ahli statistik juga menyebutnya sebagai mean-square. Di satu sisi, itu adalah nama panggilan yang tidak menguntungkan: Ini meninggalkan "penyimpangan dari rata-rata," tetapi begitulah.

Pembilang varians — permisi, kuadrat rata-rata — adalah jumlah deviasi kuadrat dari mean. Ini mengarah ke nama panggilan lain, jumlah kuadrat. Penyebutnya, seperti yang saya katakan di Bab 10, adalah derajat kebebasan (df). Jadi, cara yang sedikit berbeda untuk memikirkan varians adalah:

$$\text{Mean Square} = \frac{\text{Sum of Squares}}{\text{df}}$$

Anda dapat menyingkat ini sebagai:

$$\text{MS} = \frac{\text{SS}}{\text{df}}$$

Sekarang, untuk memecahkan masalah pelik. Salah satu langkah penting adalah menemukan kuadrat tengah yang bersembunyi di dalam data. Cara lainnya adalah memahami bahwa Anda menggunakan kuadrat rata-rata ini untuk memperkirakan varians dari populasi yang menghasilkan sampel-sampel ini. Dalam kasus ini, asumsikan bahwa varians tersebut sama, jadi Anda benar-benar memperkirakan satu varians. Langkah terakhir adalah memahami bahwa Anda menggunakan perkiraan ini untuk menguji hipotesis yang saya tunjukkan di awal bab ini.

Tiga kuadrat rata-rata yang berbeda ada di dalam data pada Tabel 12-1. Mulailah dengan seluruh rangkaian 27 skor, lupakan sejenak bahwa mereka dibagi menjadi tiga kelompok. Misalkan Anda ingin menggunakan 27 skor tersebut untuk menghitung perkiraan varians populasi. (Ide yang tidak pasti, tapi buat saya lucu.) Rata-rata dari 27 skor itu adalah 85. Saya akan menyebutnya rata-rata besar karena itu rata-rata dari semuanya.
Jadi kuadrat rata-ratanya adalah:

$$\frac{(95-85)^2+(91-85)^2+\ldots+(73-85)^2+(77-85)^2}{(27-1)} = 68.08$$

Penyebutnya memiliki 26 (27 – 1) derajat kebebasan. Saya menyebut varians itu sebagai varians total, atau dalam cara berpikir baru tentang ini, MSTotal. Ini sering disingkat MS_T. Inilah varian lain yang perlu dipertimbangkan. Dalam Bab 11, saya menjelaskan uji-t untuk dua sampel dengan varians yang sama. Untuk pengujian itu, Anda mengumpulkan dua varian sampel untuk membuat perkiraan gabungan dari varians populasi. Data pada Tabel 12-1 memberikan tiga varian sampel untuk perkiraan gabungan: 16,28, 14,18, dan 15,64.

Dengan asumsi bahwa angka-angka ini mewakili varians populasi yang sama, perkiraan yang dikumpulkan adalah:

$$s_p^2 = \frac{(N_1-1)s_1^2+(N_2-1)s_2^2+(N_3-1)s_3^2}{(N_1-1)+(N_2-1)+(N_3-1)}$$

$$= \frac{(9-1)(16.28)+(10-1)(14.18)+(8-1)(15.64)}{(9-1)+(10-1)+(8-1)} = 15.31$$

Karena perkiraan gabungan ini berasal dari varians dalam grup, ini disebut MS_{Within}, atau MS_W.

Satu lagi mean-square untuk pergi — varians sampel berarti di sekitar mean besar. Dalam contoh ini, itu berarti varians dalam angka-angka ini 93,44, 85,20, dan 75,25 — semacam. Saya katakan "semacam" karena ini berarti, bukan skor. Ketika Anda berurusan

dengan rata-rata, Anda harus memperhitungkan jumlah skor yang menghasilkan setiap rata-rata. Untuk melakukannya, Anda mengalikan setiap deviasi kuadrat dengan jumlah skor dalam sampel itu.

Jadi varians ini adalah:

$$\frac{(9)(93.44-85)^2+(10)(85.20-85)^2+(8)(75.25-85)^2}{3-1}=701.34$$

Df untuk varians ini adalah 2 (jumlah sampel – 1).

Ahli statistik, tidak dikenal karena kerenyahan penggunaannya, menyebut ini sebagai varians antara rata-rata sampel. (Di antara adalah kata yang tepat ketika Anda berbicara tentang lebih dari dua item.) Varians ini dikenal sebagai $MS_{Between}$, atau MS_B. Jadi Anda sekarang memiliki tiga perkiraan varians populasi: MS_T, MS_W, dan MS_B. Apa yang Anda lakukan dengan mereka?

Ingatlah bahwa tujuan awalnya adalah untuk menguji hipotesis tentang tiga cara. Menurut H_0, setiap perbedaan yang Anda lihat di antara tiga rata-rata sampel hanya disebabkan oleh kebetulan. Implikasinya adalah bahwa varians di antara rata-rata tersebut sama dengan varians dari setiap tiga angka yang dipilih secara acak dari populasi. Jika Anda entah bagaimana bisa membandingkan varians di antara rata-rata (itu MS_B, ingat) dengan varians populasi, Anda bisa melihat apakah itu bertahan. Jika saja Anda memiliki perkiraan varians populasi yang tidak bergantung pada perbedaan di antara kelompok, Anda akan berada dalam bisnis.

Ah . . . tetapi Anda memiliki perkiraan itu. Anda memiliki MSW, perkiraan berdasarkan penyatuan varians dalam sampel. Dengan asumsi bahwa varians tersebut mewakili varians populasi yang sama, ini adalah perkiraan yang solid. Dalam contoh ini, ini didasarkan pada 24 derajat kebebasan.

Alasannya sekarang menjadi: Jika MSB hampir sama dengan MSW, Anda memiliki bukti yang konsisten dengan H0. Jika MSB secara signifikan lebih besar dari MSW, Anda memiliki bukti yang tidak konsisten dengan H0. Akibatnya, Anda mengubah hipotesis ini:

$$H_0: \mu_1 = \mu_2 = \mu_3$$

$$H_1: \text{Not } H_0$$

ke dalam ini:

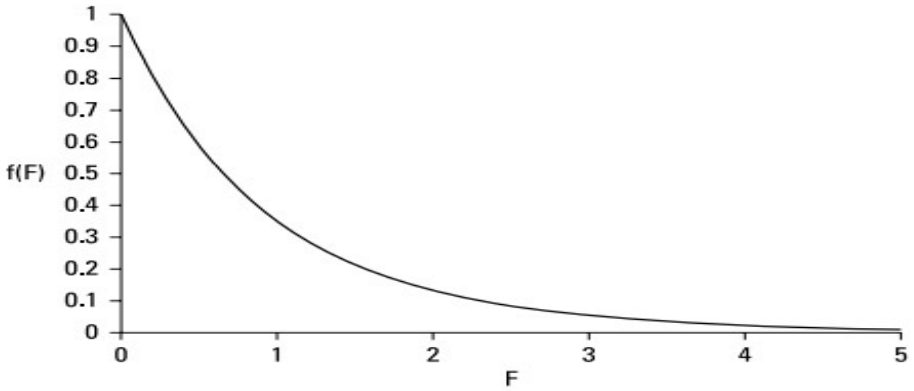
$$H_0: \sigma_B^2 \leq \sigma_W^2$$

$$H_1: \sigma_B^2 > \sigma_W^2$$

Daripada melakukan beberapa uji-t di antara rata-rata sampel, Anda melakukan uji perbedaan antara dua varians. Apa itu tes? Dalam Bab 11, saya menunjukkan pengujian hipotesis tentang dua varians. Ini disebut uji-F. Untuk melakukan tes ini, Anda membagi satu varians dengan yang lain. Anda mengevaluasi hasil terhadap keluarga distribusi yang disebut distribusi-F. Karena dua varians terlibat, dua nilai derajat kebebasan menentukan setiap anggota keluarga.

Untuk contoh ini, F memiliki df = 2 (untuk MS_B) dan df = 24 (untuk MS_W). Gambar 12.1 menunjukkan seperti apa anggota keluarga F ini. Untuk tujuan kita, ini adalah distribusi

kemungkinan nilai-f jika H_0 benar. (Lihat bagian di Bab 11 tentang memvisualisasikan distribusi-F).



Gambar 12.1 Distribusi-F dengan 2 dan 24 derajat kebebasan.

Statistik uji untuk contohnya adalah:

$$F = \frac{701.34}{15.31} = 45.82$$

Berapa proporsi area yang dipotong oleh nilai ini di bagian atas distribusi F? Dari Gambar 12.1, Anda dapat melihat bahwa proporsi ini mikroskopis, karena nilai pada sumbu horizontal hanya mencapai 5. (Dan proporsi area di luar 5 sangat kecil.) Jauh lebih kecil dari 0,05. Ini berarti bahwa sangat tidak mungkin bahwa perbedaan di antara rata-rata disebabkan oleh kebetulan. Ini berarti Anda menolak H_0.

Keseluruhan prosedur untuk menguji lebih dari dua sampel ini disebut analisis varians, sering disingkat ANOVA. Dalam konteks ANOVA, penyebut rasio-F memiliki istilah kesalahan nama generik. Variabel bebas kadang-kadang disebut faktor. Jadi ini adalah ANOVA faktor tunggal atau (1 faktor). Dalam contoh ini, faktornya adalah Metode Pelatihan. Setiap instance dari variabel independen disebut level. Variabel bebas dalam contoh ini memiliki tiga tingkatan. Studi yang lebih kompleks memiliki lebih dari satu faktor, dan setiap faktor dapat memiliki banyak tingkatan.

Hubungan yang bermakna

Perhatikan lagi kuadrat rata-rata dalam contoh ini, masing-masing dengan jumlah kuadrat dan derajat kebebasannya. Sebelumnya, ketika saya menghitung setiap kuadrat rata-rata untuk Anda, saya tidak secara eksplisit menunjukkan kepada Anda setiap jumlah kuadrat, tetapi di sini saya menyertakannya:

$$MS_B = \frac{SS_B}{df_B} = \frac{1402.68}{2} = 701.34$$

$$MS_W = \frac{SS_W}{df_W} = \frac{367.32}{24} = 15.31$$

$$MS_T = \frac{SS_T}{df_T} = \frac{1770}{26} = 68.08$$

Mulailah dengan derajat kebebasan: df_B = 2, df_W = 24, dan df_T = 26. Apakah kebetulan keduanya dijumlahkan? Hampir tidak. Itu selalu terjadi

$$df_B + df_W = df_T$$

Bagaimana dengan jumlah kuadrat itu?

$$1402.68 + 367.32 = 1770$$

Sekali lagi, ini bukan kebetulan. Dalam analisis varians, ini selalu terjadi:

$$SS_B + SS_W = SS_T$$

Faktanya, ahli statistik yang bekerja dengan analisis varians berbicara tentang mempartisi (baca "mengurai menjadi bagian yang tidak tumpang tindih") SS_T menjadi satu bagian untuk SS_B dan satu lagi untuk SS_W, dan mempartisi df_T menjadi satu jumlah untuk df_B dan satu lagi untuk df_W.

12.2 ANOVA DALAM R

Di bagian ini, saya memandu Anda melalui contoh bagian sebelumnya dan menunjukkan kepada Anda betapa mudahnya analisis varians di R. Sebenarnya, saya mulai dari garis finish sehingga Anda dapat melihat ke mana saya menuju.

Fungsi R untuk ANOVA adalah aov(). Berikut tampilannya secara umum:

```
aov(Dependent_variable ~ Independent_variable, data)
```

Dalam contoh, skor adalah variabel dependen dan metode adalah variabel independen. Jadi, Anda memerlukan bingkai data 2 kolom dengan Metode di kolom pertama dan Skor di kolom kedua. (Ini setara dengan format data "bentuk panjang", yang saya bahas di Bab 10 dan 11).

Mulailah dengan sebuah vektor untuk setiap kolom pada Tabel 12-1:

```
method1.scores <- c(95,91,89,90,99,88,96,98,95)
method2.scores <- c(83,89,85,89,81,89,90,82,84,80)
method3.scores <- c(68,75,79,74,75,81,73,77)
```

Kemudian buat satu vektor yang terdiri dari semua skor ini:

```
Score <- c(method1.scores, method2.scores, method3.scores)
```

Selanjutnya, buat vektor yang terdiri dari nama-nama metode, dicocokkan dengan skor. Dengan kata lain, vektor ini harus terdiri dari “metode1” yang diulang sembilan kali, diikuti oleh “metode2” yang diulang sepuluh kali, diikuti oleh “metode3” yang diulang delapan kali:

```
Method <- rep(c("method1", "method2", "method3"),
          times=c(length(method1.scores),
          length(method2.scores), length(method3.scores)))
```

Bingkai data kemudian:

```
Training.frame <- data.frame(Method,Score)
```

Dan ANOVAnya adalah:

```
analysis <-aov(Score ~ Method,data = Training.frame)
```

Untuk tabel analisis, gunakan ringkasan().

```
> summary(analysis)
            Df Sum Sq Mean Sq F value   Pr(>F)
Method       2 1402.7   701.3   45.82 6.38e-09 ***
Residuals   24  367.3    15.3
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

pertama terdiri dari Metode dan Residual, yang dipetakan ke Antara dan Dalam dari bagian sebelumnya. Residual, dalam konteks ini, adalah deviasi skor dari mean kelompoknya. (Lebih banyak yang ingin saya katakan tentang residu di Bab 14.) Kolom berikutnya memberikan derajat kebebasan, SS, MS, F, dan p. Nilai F yang tinggi dan nilai p yang kecil (tercantum di sini sebagai Pr(>F)) memberitahu Anda untuk menolak hipotesis nol. Kode signifikansi memberitahu Anda bahwa F sangat tinggi sehingga Anda dapat menolak hipotesis nol bahkan jika adalah 0,0001.

Memvisualisasikan hasilnya

Salah satu cara untuk memplot temuan adalah dengan menunjukkannya sebagai boxplot. Berikut cara memplot satu di ggplot2.

Pernyataan pertama memetakan variabel ke sumbu:

```
ggplot(Training.frame, aes(x=Method, y=Score))
```

Selanjutnya mengatur palang untuk kumis:

```
stat_boxplot(geom="errorbar", width =.5)
```

Dan yang terakhir memplot fungsi geom yang sesuai:

```
geom_boxplot()
```

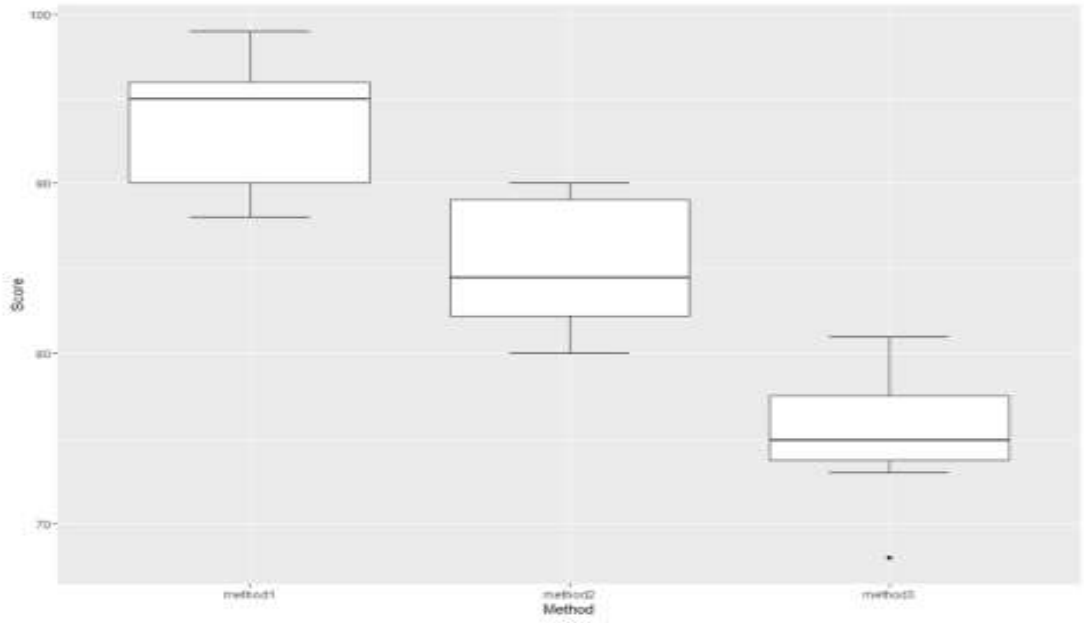
Jadi baris kode R ini

```
ggplot(Training.frame, aes(x=Method, y=Score))+
  stat_boxplot(geom="errorbar", width =.5) +
  geom_boxplot()
```

menghasilkan Gambar 12.2.

Setelah ANOVA

Analisis ANOVA memungkinkan Anda untuk memutuskan apakah akan menolak H0 atau tidak. Setelah Anda memutuskan untuk menolak, lalu apa? Yang bisa Anda katakan adalah bahwa di suatu tempat dalam kumpulan sarana, ada sesuatu yang berbeda dari sesuatu yang lain. Analisis tidak menentukan apa "sesuatu" itu.



Gambar 12.2 Boxplot hasil sampel.

Perbandingan yang direncanakan

Untuk mendapatkan lebih spesifik, Anda harus melakukan beberapa tes lebih lanjut. Tidak hanya itu, Anda harus merencanakan tes tersebut sebelum melakukan ANOVA.

Tes pasca-ANOVA ini disebut perbandingan terencana. Beberapa ahli statistik menyebutnya sebagai tes apriori atau kontras. Saya mengilustrasikannya dengan mengikuti contoh. Misalkan sebelum Anda mengumpulkan data, Anda memiliki alasan untuk percaya bahwa Metode 2 akan menghasilkan skor yang lebih tinggi daripada Metode 3 dan Metode 1 akan menghasilkan skor yang lebih tinggi daripada Metode 2 dan Metode 3 yang dirata-ratakan secara bersamaan. Dalam hal ini, Anda berencana terlebih dahulu untuk membandingkan rata-rata sampel tersebut jika keputusan berbasis ANOVA Anda adalah untuk menolak H_0. Seperti yang saya sebutkan sebelumnya, analisis keseluruhan mempartisi SS_T menjadi SS_B dan SS_W, dan df_T menjadi df_B dan df_W. Perbandingan yang direncanakan lebih lanjut mempartisi SS_B dan df_B. Setiap kontras (ingat, itu nama lain untuk "perbandingan terencana") memiliki SS sendiri bersama dengan 1 df. Saya merujuk ke Metode 2 versus Metode 3 sebagai Kontras1 dan Metode 1 versus rata-rata Metode 2 dan 3 sebagai Kontras2. Untuk contoh ini:

$$SS_{Contrast1} + SS_{Contrast2} = SS_B \quad \text{dan} \quad df_{Contrast1} + df_{Contrast2} = df_B$$

Karena setiap SS memiliki 1 df, itu sama dengan MS yang sesuai. Membagi SS untuk kontras dengan MSW menghasilkan rasio-F untuk kontras. F memiliki df=1 dan dfW. Jika F itu memotong kurang dari 0,05 di ekor atas distribusi F-nya, tolak hipotesis nol untuk kontras itu (dan Anda dapat merujuk ke kontras sebagai "signifikan secara statistik").

Dimungkinkan untuk mengatur kontras antara dua cara sebagai ekspresi yang melibatkan ketiga cara sampel. Misalnya, untuk membandingkan Metode 2 versus Metode 3, saya dapat menulis perbedaan di antara keduanya sebagai:

$$(0)\bar{x}_1 + (+1)\bar{x}_2 + (-1)\bar{x}_3$$

0, +1, dan -1 adalah koefisien perbandingan. Saya menyebut mereka, secara umum, sebagai c_1, c_2, dan c_3. Untuk membandingkan Metode 1 versus rata-rata Metode 2 dan Metode 3, ini

$$(+2)\bar{x}_1 + (-1)\bar{x}_2 + (-1)\bar{x}_3$$

Poin penting adalah bahwa koefisien bertambah hingga 0. Bagaimana Anda menggunakan koefisien perbandingan dan cara menghitung SS untuk kontras? Untuk contoh ini, inilah $SS_{Contrast1}$:

$$SS_{Contrast1} = \frac{\left((0)(93.44) + (+1)(85.20) + (-1)(75.25)\right)^2}{\frac{(0)^2}{9} + \frac{(+1)^2}{10} + \frac{(-1)^2}{8}} = 358.5$$

Dan inilah $SS_{Contrast2}$:

$$SS_{Contrast2} = \frac{\left((+2)(93.44) + (-1)(85.20) + (-1)(75.25)\right)^2}{\frac{(2)^2}{9} + \frac{(-1)^2}{10} + \frac{(-1)^2}{8}} = 1044.2$$

Secara umum, rumusnya adalah

$$SS_{Contrast} = \frac{\sum c_j \bar{x}_j}{\sum\left(\frac{c_j^2}{n_j}\right)}$$

di mana subskrip j adalah singkatan dari "tingkat variabel independen" (untuk Metode 1, j=1, misalnya).

Untuk Kontras 1

$$F_{1,24} = \frac{SS_{Contrast1}}{MS_{Within}} = \frac{358.5}{15.3} = 23.42$$

dan untuk Kontras 2

$$F_{1,24} = \frac{SS_{Contrast2}}{MS_{Within}} = \frac{1044.2}{15.3} = 68.22$$

Apakah kontras ini signifikan? Ya benar — artinya Metode 2 menghasilkan pembelajaran yang jauh lebih tinggi daripada Metode 3, dan Metode 1 menghasilkan pembelajaran yang jauh lebih tinggi daripada rata-rata Metode 2 dan 3. Anda dapat menggunakan pf() untuk memverifikasi (atau menunggu hingga ayat “Kontras dalam R”).

Kata lain tentang kontras

Sebelumnya, saya mengatakan bahwa hal penting tentang kontras adalah bahwa koefisiennya berjumlah 0. Hal penting lainnya adalah hubungan antara koefisien dalam satu set kontras. Dalam dua kontras yang saya tunjukkan, jumlah produk dari koefisien yang sesuai adalah 0:

$$((0)(+2))+((+1)(-1))+((-1)(-1))=0$$

Ketika ini terjadi, kontrasnya ortogonal. Ini berarti mereka tidak memiliki informasi yang tumpang tindih. Itu tidak berarti bahwa kontras lainnya tidak mungkin. Hanya saja kontras lain akan menjadi bagian dari set (atau set) kontras ortogonal yang berbeda. Dua set kontras ortogonal lainnya untuk contoh ini adalah: (1) Metode 1 versus Metode 2, dan Metode 3 versus rata-rata Metode 1 dan Metode 2; (2) Metode 1 versus Metode 3, dan Metode 2 versus rata-rata Metode 1 dan Metode 3.

Kontras dalam R

Tujuannya di sini adalah untuk membuat tabel ANOVA yang menunjukkan kontras mempartisi SSB dan akan menunjukkan rasio-F dan nilai-p terkait. Ini akan terlihat seperti ini:

```
                   Df Sum Sq Mean Sq F value   Pr(>F)
Method              2 1402.7   701.3   45.82 6.38e-09 ***
  Method: 2 vs 3    1  358.5   358.5   23.42 6.24e-05 ***
  Method: 1 vs 2 & 3  1 1044.2  1044.2   68.22 1.78e-08 ***
Residuals          24  367.3    15.3
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Untuk menyiapkan kontras, Anda terlebih dahulu membuat matriks koefisien dalam himpunan kontras ortogonal:

```
contrasts(Training.frame$Method) <- matrix(c(0,1,-1,2,-1,-1),3,2)
```

Di sebelah kiri, istilah di dalam tanda kurung menentukan apa yang akan dikontraskan — level dari variabel independen Method dalam Training.frame. Di sebelah kanan, fungsi matrix() membuat matriks dengan koefisien di kolom:

```
> contrasts(Training.frame$Method)
        [,1] [,2]
method1    0    2
method2    1   -1
method3   -1   -1
```

Selanjutnya, Anda menjalankan analisis varians, tetapi kali ini dengan argumen kontras:

```
Anova.w.Contrasts <-aov(Score ~ Method,data=Training.frame,
        contrasts = contrasts(Training.frame$Method))
```

Bagaimana Anda membuat tabel di awal subbagian ini? Dengan pernyataan summary() yang menambahkan sedikit twist:

```
summary(Anova.w.Contrasts,split=list(Method=list("2 vs 3"= 1,
  "1 vs 2 & 3" = 2)))
```

Sentuhan kecil (sebenarnya sedikit "terpisah,") ada di argumen kedua. Tujuannya adalah untuk mempartisi Metode menjadi dua bagian — satu yang sesuai dengan kontras pertama dan satu yang sesuai dengan yang kedua. Anda melakukannya dengan split, yang membagi

daftar ke dalam jumlah komponen yang ditunjukkan dan menyusun kembali daftar dengan nama yang ditetapkan untuk setiap komponen. Dalam hal ini, daftar adalah Metode yang dipecah menjadi daftar dengan dua komponen. Nama setiap komponen sesuai dengan apa yang ada di kontras. Menjalankan pernyataan ringkasan itu menghasilkan tabel di bagian atas subbagian ini.

Perbandingan yang tidak direncanakan

Hal-hal akan menjadi membosankan jika pengujian pasca-ANOVA Anda terbatas pada perbandingan yang harus Anda rencanakan sebelumnya. Terkadang Anda ingin mengintip data Anda dan melihat apakah sesuatu yang menarik muncul dengan sendirinya. Terkadang, sesuatu melompat keluar pada Anda yang tidak Anda antisipasi. Ketika ini terjadi, Anda dapat membuat perbandingan yang tidak Anda rencanakan. Perbandingan ini disebut tes posteriori, tes post hoc, atau perbandingan yang tidak direncanakan. Ahli statistik telah datang dengan berbagai macam tes ini, banyak dari mereka dengan nama eksotis dan banyak dari mereka bergantung pada distribusi sampling khusus.

Gagasan di balik tes ini adalah Anda membayar harga karena tidak merencanakannya sebelumnya. Harga itu ada hubungannya dengan menumpuk dek terhadap menolak H0 untuk perbandingan tertentu. Salah satu anggota paling terkenal di dunia post-hoc adalah tes HSD (Honest Significant Difference) Tukey. Tes ini melakukan semua kemungkinan perbandingan berpasangan di antara rata-rata sampel. Tunggu. Apa? Di bagian sebelumnya "Masalah pelik," saya membahas mengapa beberapa uji-t berpasangan tidak bekerja — jika setiap pengujian memiliki = 0,05, probabilitas keseluruhan dari kesalahan Tipe I meningkat dengan jumlah rata-rata.

Jadi bagaimana ceritanya? Ceritanya adalah bahwa uji Tukey menyesuaikan jumlah rata-rata sampel dan membandingkan perbedaannya bukan dengan distribusi-t tetapi dengan distribusi Rentang Terpelajar. Efek keseluruhannya adalah membuat lebih sulit untuk menolak hipotesis nol tentang perbandingan berpasangan apa pun daripada jika Anda membandingkan perbedaan terhadap distribusi-t. (Saya belum pernah mendengar beberapa uji-t yang disebut sebagai "Perbedaan Signifikan yang Tidak Jujur", tetapi mungkin suatu hari nanti).

Tes ini mudah dilakukan di R:

```
> TukeyHSD(analysis)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Score ~ Method, data = Training.frame)

$Method
                     diff       lwr        upr     p adj
method2-method1  -8.244444 -12.73337  -3.755523 0.0003383
method3-method1 -18.194444 -22.94172 -13.447166 0.0000000
method3-method2  -9.950000 -14.58423  -5.315769 0.0000481
```

Tabel menunjukkan setiap perbandingan berpasangan beserta perbedaannya, batas kepercayaan 95 persen bawah dan atas, dan probabilitas yang disesuaikan. Setiap probabilitas jauh lebih rendah dari 0,05, jadi kesimpulannya adalah bahwa setiap perbedaan signifikan secara statistik.

12.3 JENIS HIPOTESIS LAIN, JENIS TES LAIN

ANOVA sebelumnya bekerja dengan sampel independen. Seperti yang dijelaskan Bab 11, terkadang Anda bekerja dengan sampel yang cocok. Misalnya, terkadang seseorang memberikan data dalam sejumlah kondisi yang berbeda. Di bagian ini, saya memperkenalkan ANOVA yang Anda gunakan ketika Anda memiliki lebih dari dua sampel yang cocok. Jenis ANOVA ini disebut tindakan berulang. Anda akan melihatnya disebut nama lain juga, seperti blok acak atau di dalam subjek.

Bekerja dengan tindakan berulang ANOVA

Untuk menunjukkan cara kerjanya, saya memperluas contoh dari Bab 11. Dalam contoh itu, sepuluh pria berpartisipasi dalam program penurunan berat badan. Tabel 12.3 menunjukkan data mereka selama periode tiga bulan.

Tabel 12.3 Data untuk Contoh Penurunan Berat Badan

Orang	Sebelum	Satu Bulan	Dua Bulan	Tiga Bulan	Mean
Al	198	194	191	188	192.75
Bill	201	203	200	196	200.00
Charlie	210	200	192	188	197.50
Dan	185	183	180	178	181.50
Ed	204	200	195	191	197.50
Fred	156	153	150	145	151.00
Gary	167	166	167	166	166.50
Harry	197	197	195	192	195.25
Irv	220	215	209	205	212.25
Jon	186	184	179	175	181.00
Mean	192.4	189.5	185.8	182.4	187.525

Apakah program tersebut efektif? Pertanyaan ini memerlukan uji hipotesis:

$$H_0: \mu_{Before} = \mu_1 = \mu_2 = \mu_3$$

$$H_1: \text{Not } H_0$$

Sekali lagi, Anda menetapkan α = .05

Seperti pada ANOVA sebelumnya, mulailah dengan varians dalam data. MS_T adalah varians di semua 40 skor dari grand mean, yaitu 187,525:

$$MS_T = \frac{(198-187.525)^2 + (201-187.525)^2 + \ldots + (175-187.525)^2}{(40-1)} = 318.20$$

Orang-orang yang berpartisipasi dalam program penurunan berat badan juga menyediakan varians. Rata-rata keseluruhan masing-masing (rata-ratanya selama empat pengukuran) bervariasi dari rata-rata besar. Karena data ini ada di baris, saya menyebutnya MS_{rows}:

$$MS_{Rows} = \frac{(192.75-187.525)^2+(200-187.525)^2+\ldots+(181-187.525)^2}{(10-1)} = 1292.41$$

Sarana kolom juga bervariasi dari mean besar:

$$MS_{Columns} = \frac{(192.4-187.525)^2+(189.5-187.525)^2+(185.8-187.525)^2+(182.4-187.525)^2}{(4-1)}$$

$$= 189.69$$

Satu lagi sumber varians ada di data. Anggap saja sebagai varians yang tersisa setelah Anda menarik varians di baris dan varians di kolom dari total varians. Sebenarnya, lebih tepat untuk mengatakan bahwa itu adalah jumlah kuadrat yang tersisa ketika Anda mengurangi SS_{rows} dan $SS_{Columns}$ dari SS_T. Varians ini disebut MS_{Error}. Seperti yang saya katakan sebelumnya, dalam ANOVA penyebut dari F disebut istilah kesalahan. Jadi kesalahan kata di sini memberi Anda petunjuk bahwa MS ini adalah penyebut untuk F.

Untuk menghitung MS_{Error}, Anda menggunakan hubungan di antara jumlah kuadrat dan di antara df.

$$MS_{Error} = \frac{SS_{Error}}{df_{Error}} = \frac{SS_T - SS_{Rows} - SS_{Columns}}{df_T - df_{Rows} - df_{Columns}} = \frac{209.175}{27} = 7.75$$

Berikut cara lain untuk menghitung df_{Error}:

$$df_{Error} = (\text{number of rows - 1})(\text{number of columns - 1})$$

Untuk melakukan uji hipotesis, Anda menghitung F:

$$F = \frac{MS_{Columns}}{MS_{Error}} = \frac{189.69}{7.75} = 24.49$$

Dengan 3 dan 27 derajat kebebasan, F kritis untuk α = 0,05 adalah 2,96. (Gunakan qf() untuk memverifikasi.) F yang dihitung lebih besar dari F kritis, jadi keputusannya adalah menolak H_0. Bagaimana dengan F yang melibatkan MS_{rows}? Yang itu tidak masuk ke dalam H_0 untuk contoh ini. Jika Anda menemukan F yang signifikan, semua itu menunjukkan bahwa orang berbeda satu sama lain dalam hal berat dan itu tidak banyak memberi tahu Anda.

Pengukuran berulang ANOVA dalam R

Untuk mengatur tahapan analisis pengukuran berulang, masukkan kolom-kolom Tabel 12-3 ke dalam vektor-vektor:

```
Person <-c("Al", "Bill", "Charlie", "Dan", "Ed", "Fred",
           "Gary","Harry","Irv","Jon")
Before <- c(198,201,210,185,204,156,167,197,220,186)
OneMonth <- c(194,203,200,183,200,153,166,197,215,184)
TwoMonths <- c(191,200,192,180,195,150,167,195,209,179)
ThreeMonths <- c(188,196,188,178,191,145,166,192,205,175)
```

Kemudian buat bingkai data:

```
Weight.frame <- data.frame(Person, Before, OneMonth,
          TwoMonths, ThreeMonths)
```

Bingkai data terlihat seperti ini:

```
> Weight.frame
    Person Before OneMonth TwoMonths ThreeMonths
1       Al    198      194       191         188
2     Bill    201      203       200         196
3  Charlie    210      200       192         188
4      Dan    185      183       180         178
5       Ed    204      200       195         191
6     Fred    156      153       150         145
7     Gary    167      166       167         166
8    Harry    197      197       195         192
9      Irv    220      215       209         205
10     Jon    186      184       179         175
```

Itu dalam format lebar, dan Anda harus membentuknya kembali. Dengan paket reshape2 terinstal (pada tab Packages, pilih kotak centang di sebelah reshape2), lelehkan data ke dalam format panjang:

```
Weight.frame.melt <- melt(Weight.frame,id="Person")
```

Selanjutnya, tetapkan nama kolom ke bingkai data yang dilebur:

```
colnames(Weight.frame.melt) = c("Person","Time","Weight")
```

Dan sekarang, enam baris pertama dari bingkai data baru adalah:

```
> head(Weight.frame.melt)
   Person   Time Weight
1      Al Before    198
2    Bill Before    201
3 Charlie Before    210
4     Dan Before    185
5      Ed Before    204
6    Fred Before    156
```

Selain Person, Anda sekarang memiliki Time sebagai variabel independen.

Saya akan menggunakan R sebagai alat pengajaran: Untuk memberi Anda gambaran tentang cara kerja analisis ini, saya akan mulai dengan berpura-pura bahwa ini adalah analisis sampel independen, seperti yang pertama dalam bab ini. Kemudian saya akan menjalankannya sebagai analisis tindakan berulang sehingga Anda dapat melihat perbedaannya dan mungkin lebih memahami apa yang dilakukan analisis tindakan berulang. Sebagai sampel independen:

```
> ind.anova <- aov(Weight ~ Time, data=Weight.frame.melt)
> summary(ind.anova)
            Df Sum Sq Mean Sq F value Pr(>F)
Time         3    569   189.7   0.577  0.634
Residuals   36  11841   328.9
```

Analisis ini tidak menunjukkan perbedaan yang signifikan antara tingkat Waktu. Kuncinya adalah untuk menghilangkan efek setiap baris mewakili data dari satu orang. Itu akan memecah SS untuk Residual menjadi dua komponen — satu SS untuk Orang (yang memiliki sembilan derajat kebebasan) dan SS lain yang memiliki 27 derajat kebebasan tersisa. Bagi SS kedua itu dengan derajat kebebasannya, dan Anda memiliki MS_{Error} yang saya sebutkan sebelumnya (walaupun R tidak merujuknya seperti itu).

Berikut cara menyelesaikannya:

```
rm.anova <- aov(Weight ~ Time + Error(Person/Time),
                data = Weight.frame.melt)
```

Istilah baru menunjukkan bahwa Berat tidak hanya bergantung pada Waktu tetapi juga pada Orang, dan bahwa setiap Orang mengalami semua tingkat Waktu. Pengaruh Waktu — penurunan berat badan selama empat tingkat Waktu — terbukti dalam setiap Orang. (Lebih mudah untuk melihatnya dalam format lebar daripada format panjang.) Di beberapa bidang, kata subjek berarti orang: Itu sebabnya analisis tindakan berulang juga disebut analisis dalam subjek, seperti yang saya tunjukkan sebelumnya. Dan sekarang untuk tabel

```
> summary(rm.anova)

Error: Person
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  9  11632    1292

Error: Person:Time
          Df Sum Sq Mean Sq F value  Pr(>F)
Time       3  569.1  189.69   24.48 7.3e-08 ***
Residuals 27  209.2    7.75
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sekarang analisis menunjukkan pengaruh signifikan dari Waktu.

Memvisualisasikan hasilnya

Salah satu cara untuk memvisualisasikan hasilnya adalah dengan memplot rata-rata penurunan berat badan pada sumbu y dan bulan (0, 1, 2, 3) pada sumbu x. Perhatikan saya menggunakan 0-3 untuk mewakili tingkat Waktu (Sebelum, Satu Bulan, Dua Bulan, Tiga Bulan). Gambar 12.3 menunjukkan plot, bersama dengan kesalahan standar rata-rata (tercermin pada bilah kesalahan).

Landasan untuk plot adalah kerangka data yang menahan waktu (untuk kenyamanan, sebagai variabel numerik), bobot rata-rata, dan kesalahan standar:

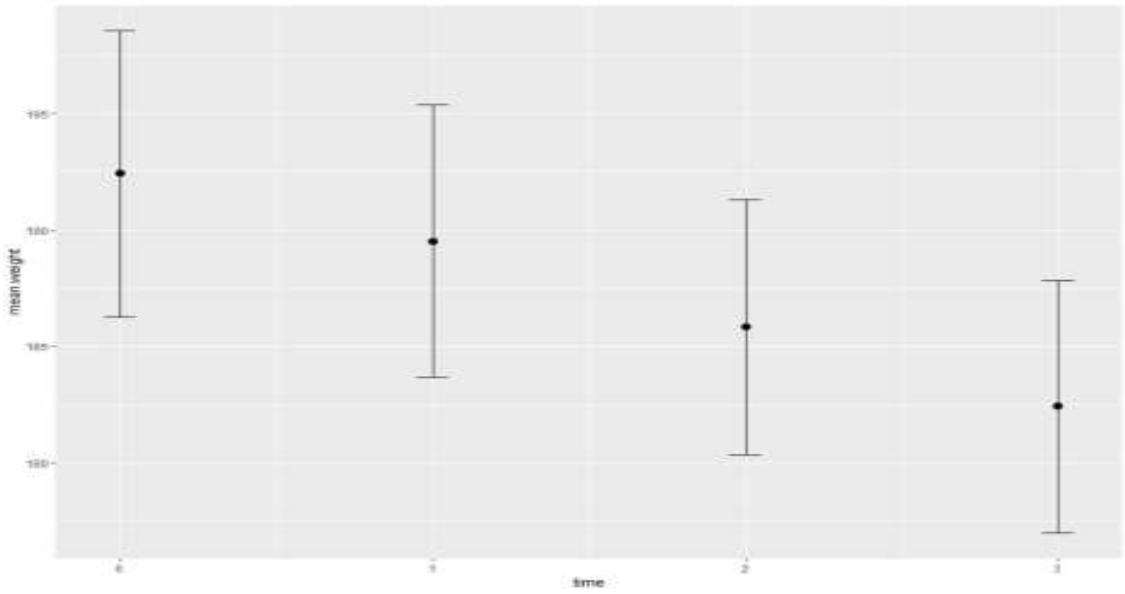
```
time <- c(0,1,2,3)

mean.weight <- c(mean(Before),mean(OneTime),
          mean(TwoTimes),mean(ThreeTimes))
```

```
se.weight <- c(sd(Before), sd(OneTime), sd(TwoTimes),
          sd(ThreeTimes))/sqrt(length(Person))
```

```
wt.means.frame <- data.frame(time,mean.weight,se.weight)
```

```
> wt.means.frame
  time mean.weight se.weight
```

```
1    0       192.4  6.144917
2    1       189.5  5.856146
3    2       185.8  5.466667
4    3       182.4  5.443038
```



Gambar 12.3 Cara dan kesalahan standar untuk contoh penurunan berat badan.

Merencanakan di ggplot2:

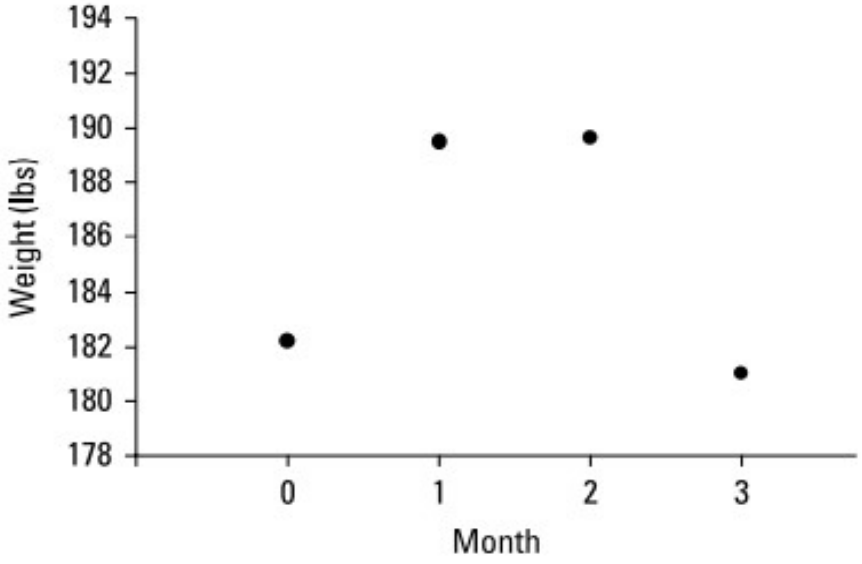
```
ggplot(wt.means.frame,aes(x=time,y=mean.weight)) +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=mean.weight-se.weight,
          ymax=mean.weight+se.weight),width=.1)
```

Pernyataan pertama memetakan variabel independen ke dalam sumbu x, dan variabel dependen ke dalam sumbu y. Pernyataan kedua menentukan titik sebagai objek geometris dan menetapkan ukurannya. Pernyataan ketiga memberikan batasan dan ukuran untuk bilah kesalahan.

12.4 MENJADI TRENDI

Dalam situasi seperti contoh penurunan berat badan, Anda memiliki variabel independen yang kuantitatif — levelnya adalah angka (0 bulan, 1 bulan, 2 bulan, 3 bulan). Tidak hanya itu, tetapi dalam hal ini, intervalnya sama. Dengan variabel independen semacam itu, sering kali merupakan ide yang baik untuk mencari tren dalam data daripada hanya merencanakan perbandingan di antara sarana. Seperti yang diperlihatkan Gambar 12-3, rata-rata dalam contoh penurunan berat badan tampaknya mengikuti garis.

Analisis tren adalah prosedur statistik yang memeriksa pola itu. Tujuannya adalah untuk melihat apakah pola tersebut berkontribusi terhadap perbedaan yang signifikan di antara rata-rata. Tren bisa linier, seperti yang terlihat dalam contoh ini, atau nonlinier (di mana rata-rata jatuh pada kurva). Dua jenis kurva nonlinier untuk empat rata-rata disebut kuadrat dan kubik. Jika rata-rata menunjukkan tren kuadrat, mereka sejajar dalam pola yang menunjukkan satu perubahan arah. Gambar 12.4 menunjukkan apa yang saya maksud.



Gambar 12.4 Tren kuadrat dengan empat rata-rata.

Jika rata-rata menunjukkan tren kubik, mereka sejajar dalam pola yang menunjukkan dua perubahan arah. Gambar 12.5 menunjukkan seperti apa tren kubik.
Ketiga komponen tersebut ortogonal, jadi

$$SS_{Linear} + SS_{Quadratic} + SS_{Cubic} = SS_{Time}$$

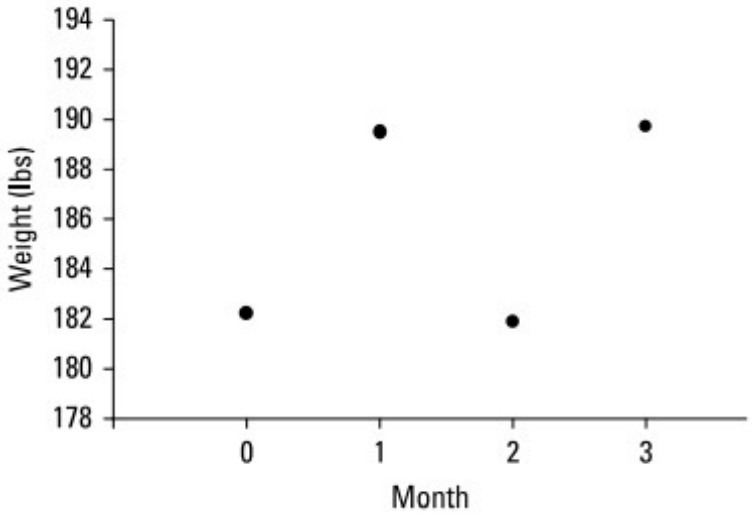
dan

$$df_{Linear} + df_{Quadratic} + df_{Cubic} = df_{Time}$$

Untuk menganalisis tren, Anda menggunakan koefisien perbandingan — angka-angka yang Anda gunakan dalam kontras. Anda menggunakannya dengan cara yang sedikit berbeda dari sebelumnya. Rumus untuk menghitung SS untuk komponen tren adalah:

$$SS_{Component} = \frac{N\left(\sum c\bar{x}\right)^2}{\sum c^2}$$

Dalam rumus ini, N adalah jumlah orang dan c mewakili koefisien.



Gambar 12.5 Tren kubik dengan empat cara.

Jadi, Anda mulai dengan menggunakan koefisien perbandingan untuk menemukan jumlah kuadrat untuk tren linier. Saya menyingkatnya sebagai SS_{Linear}. Koefisien perbandingan berbeda untuk jumlah sampel yang berbeda. Untuk empat sampel, koefisien liniernya adalah –3, -1, 1, dan 3. Cara termudah untuk mendapatkan koefisien adalah dengan mencarinya di buku teks statistik atau di Internet!

Untuk contoh ini, SSLlinearnya adalah:

$$SS_{Linear} = \frac{N\left(\sum c\bar{x}\right)^2}{\sum c^2} = \frac{10\left[(-3)(192.4)+(-1)(189.5)+(1)(185.8)+(3)(182.4)\right]^2}{(-3)^2+(-1)^2+(3)^2+(1)^2} = 567.845$$

Setelah Anda menghitung SS_{Linear}, Anda membaginya dengan dfLinear untuk menghasilkan MS_{Linear}. Ini sangat mudah karena dfLinear = 1. Bagi MS_{Linear} dengan MS_{Error} dan Anda memiliki F. Jika F itu lebih tinggi dari nilai kritis F dengan df = 1 dan dfError pada Anda, maka bobot berkurang secara linier selama periode program penurunan berat badan. F-rasio di sini adalah:

$$F = \frac{MS_{Linear}}{MS_{Error}} = \frac{567.85}{7.75} = 73.30$$

Nilai kritis untuk F dengan 1 dan 27 derajat kebebasan dan = .05 adalah 4,21. Karena nilai yang dihitung lebih besar dari nilai kritis, ahli statistik akan mengatakan bahwa data menunjukkan komponen linier yang signifikan. Ini, tentu saja, memverifikasi apa yang Anda lihat pada Gambar 12.3. Komponen linier SSTime sangat besar sehingga dua komponen lainnya sangat kecil. Saya akan memandu Anda melalui perhitungan.

Koefisien untuk komponen kuadrat adalah 1,-1,-1,1. Jadi $SS_{Quadratic}$ adalah:

$$SS_{Quadratic} = \frac{N\left(\sum c\bar{x}\right)^2}{\sum c^2} = \frac{10\left[(1)(192.4)+(-1)(189.5)+(-1)(185.8)+(1)(182.4)\right]^2}{(1)^2+(-1)^2+(-1)^2+(1)^2} = 0.6$$

Koefisien untuk komponen kubik adalah -1,3,-3,1, dan SS_{Cubic} adalah:

$$SS_{Cubic} = \frac{N\left(\sum c\bar{x}\right)^2}{\sum c^2} = \frac{10\left[(-1)(192.4)+(3)(189.5)+(-3)(185.8)+(1)(182.4)\right]^2}{(-1)^2+(3)^2+(-3)^2+(1)^2} = 0.6$$

Daripada menyelesaikan perhitungan akhir untuk mendapatkan rasio-F mikroskopis, saya akan membiarkan R melakukan pekerjaan untuk Anda di subbagian berikutnya.

Sedikit Lagi Trend

Linear, kuadrat, dan kubik sejauh yang Anda bisa lakukan dengan empat cara. Dengan lima cara, Anda dapat mencari ketiganya ditambah komponen quartic (perubahan tiga arah), dan dengan enam Anda dapat mencoba untuk melihat semua komponen sebelumnya ditambah komponen quintic (perubahan empat arah). Seperti apa bentuk koefisiennya?

Untuk lima cara, mereka adalah:
Linier: –2, -1, 0, 1, 2
Kuadrat: 2, -1, –2, -1, 2
Kubik: -1, 2, 0, –2, 1
Kuartik: 1, –4, 6, –4, 1

Dan untuk enam cara. Mereka adalah:
Linier: –5, –3, -1, 1, 3, 5
Kuadrat: 5, -1, –4, –4, -1, 5
Kubik: –5, 7, 4, –4, –7, 5
Kuartik: 1, –3, 2, 2, –3, 1
Kuintik: -1, 5, -10, 10, -5, 1
Saya bisa melanjutkan dengan lebih banyak cara, koefisien, dan nama komponen eksotis (hextic? septic?), tapi sudah cukup. Ini akan menahan Anda untuk sementara waktu.

12.5 ANALISIS TREN DI R

Saya memperlakukan analisis ini dengan cara yang hampir sama dengan kontras untuk contoh sampel independen. Saya mulai dengan membuat matriks koefisien untuk tiga komponen tren:

```
contrasts(Weight.frame.melt$Time) <- matrix(c(-3,-1,1,3,1,-1,
        -1,1,-1,3,-3,1), 4, 3)
```

Kemudian saya menjalankan ANOVA, menambahkan argumen kontras:

```
rm.anova <- aov(Weight ~ Time + Error(factor(Person)/Time),
         data=Weight.frame.melt,
         contrasts = contrasts(Weight.frame.melt$Time))
```

Akhirnya, saya menerapkan ringkasan() (termasuk pembagian Waktu menjadi tiga komponen) untuk mencetak tabel analisis:

```
summary(rm.anova,split=list(Time=list("Linear" =1,
          "Quadratic"=2,"Cubic" =3)))
```

Menjalankan pernyataan ini menghasilkan tabel ini:

```
Error: factor(Person)
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  9  11632    1292

Error: factor(Person):Time
                 Df Sum Sq Mean Sq F value   Pr(>F)
Time              3  569.1   189.7  24.485 7.30e-08 ***
  Time: Linear    1  567.8   567.8  73.297 3.56e-09 ***
  Time: Quadratic 1    0.6     0.6   0.081    0.779
  Time: Cubic     1    0.6     0.6   0.078    0.782
Residuals        27  209.2     7.7
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Sekali lagi, Anda dapat melihat linieritas tren yang luar biasa — seperti yang kita harapkan dari Gambar 12.3.

BAB 13
PENGUJIAN LEBIH RUMIT

Dalam Bab 11, saya menunjukkan cara menguji hipotesis dengan dua sampel. Dalam Bab 12, saya menunjukkan kepada Anda bagaimana menguji hipotesis ketika Anda memiliki lebih dari dua sampel. Benang merah di kedua bab adalah satu variabel independen (juga disebut faktor). Sering kali, Anda harus menguji efek lebih dari satu faktor. Dalam bab ini, saya menunjukkan bagaimana menganalisis dua faktor dalam kumpulan data yang sama. Beberapa jenis situasi yang mungkin, dan saya menjelaskan fungsi R yang berhubungan dengan masing-masing.

13.1 MEMECAHKAN KOMBINASI

Bayangkan sebuah perusahaan memiliki dua metode untuk menyajikan informasi pelatihannya: Satu melalui orang yang menyajikan informasi secara lisan, dan yang lainnya melalui dokumen teks. Bayangkan juga bahwa informasi itu disajikan dengan cara yang lucu atau teknis. Saya mengacu pada faktor pertama sebagai Metode Presentasi dan yang kedua sebagai Gaya Presentasi.

Menggabungkan dua tingkat Metode Presentasi dengan dua tingkat Gaya Presentasi memberikan empat kombinasi. Perusahaan secara acak menetapkan 4 orang untuk setiap kombinasi, dengan total 16 orang. Usai memberikan pelatihan, mereka menguji ke-16 orang tersebut dalam pemahaman materi.

Gambar 13.1 menunjukkan kombinasi, empat skor pemahaman dalam setiap kombinasi, dan statistik ringkasan untuk kombinasi, baris, dan kolom.

		Presentation Style: Humorous	Presentation Style: Technical	
Presentation Method	Spoken	Spoken and Humorous: 57, 56, 60, 64 Mean = 59.25 Variance = 12.92	Spoken and Technical: 22, 21, 29, 25 Mean = 24.25 Variance = 12.92	Mean = 41.75
	Text	Text and Humorous: 33, 25, 28, 31 Mean = 29.25 Variance = 12.25	Text and Technical: 66, 65, 71, 72 Mean = 68.50 Variance = 12.33	Mean = 48.88
		Mean = 44.25	Mean = 46.38	Grand Mean = 44.31

Gambar 13.1 Menggabungkan Level Metode Presentasi dengan Level Gaya Presentasi.

Dengan masing-masing dua tingkat dari satu faktor digabungkan dengan masing-masing dari dua tingkat faktor lainnya, jenis penelitian ini disebut desain faktorial 2 X 2.
Berikut hipotesisnya:

$H_0: \mu_{Spoken} = \mu_{Text}$

$H_1: \text{Not } H_0$

dan

$H_0: \mu_{Humorous} = \mu_{Technical}$

$H_1: \text{Not } H_0$

Karena dua metode presentasi (Lisan dan Teks) berada di baris, saya merujuk ke Jenis Presentasi sebagai faktor baris. Dua gaya presentasi (Humor dan Teknis) ada di kolom, jadi Gaya Presentasi adalah faktor kolom.

Interaksi

Saat Anda memiliki baris dan kolom data dan Anda menguji hipotesis tentang faktor baris dan faktor kolom, Anda memiliki pertimbangan tambahan. Yaitu, Anda harus memperhatikan kombinasi baris-kolom. Apakah kombinasi menghasilkan efek yang aneh? Untuk contoh yang saya sajikan, mungkin saja menggabungkan Lisan dan Teks dengan Humor dan Teknis menghasilkan hasil yang tidak terduga. Faktanya, Anda dapat melihat bahwa pada data pada Gambar 13.1: Untuk presentasi Lisan, gaya Humor menghasilkan rata-rata yang lebih tinggi daripada gaya Teknis. Untuk presentasi Teks, gaya Humor menghasilkan rata-rata yang lebih rendah daripada gaya Teknis.

Situasi seperti itu disebut interaksi. Dalam istilah formal, interaksi terjadi ketika tingkat satu faktor mempengaruhi tingkat faktor lain secara berbeda. Label untuk interaksi adalah faktor baris × faktor kolom, jadi untuk contoh ini, itu adalah Metode × Jenis.
Hipotesis untuk ini adalah:
H_0: Metode Presentasi tidak berinteraksi dengan Gaya Presentasi
H_1: Tidak H_0

Analisis

Analisis statistik, sekali lagi, adalah analisis varians (ANOVA). Seperti halnya dengan ANOVA yang saya tunjukkan sebelumnya, itu tergantung pada varians dalam data. Ini disebut ANOVA dua faktor, atau ANOVA dua arah. Varians pertama adalah varians total, berlabel MS_T. Itulah varians dari semua 16 skor di sekitar mean mereka (rata-rata besar), yaitu 44,81:

$$MS_T = \frac{(57-45.31)^2 + (56-45.31)^2 + \ldots + (72-45.31)^2}{16-1} = \frac{5885.43}{15} = 392.36$$

Penyebut memberitahu Anda bahwa df = 15 untuk MS_T.
Varians berikutnya berasal dari faktor baris. Itu adalah MS_{Method}, dan itu adalah varian dari rata-rata baris di sekitar mean besar:

$$MS_{Method} = \frac{(8)(41.75-45.31)^2 + (8)(48.88-45.31)^2}{2-1} = \frac{203.06}{1} = 203.06$$

Angka 8 dalam persamaan mengalikan setiap deviasi kuadrat karena Anda harus memperhitungkan jumlah skor yang menghasilkan rata-rata setiap baris. Df untuk MS_{Method} adalah jumlah baris – 1, yaitu 1.

Demikian pula, varians untuk faktor kolom adalah:

$$MS_{Style} = \frac{(8)(43.25-45.31)^2 + (8)(46.38-45.31)^2}{2-1} = \frac{18.06}{1} = 18.06$$

Df untuk MSStyle adalah 1 (jumlah kolom – 1).

Varians lainnya adalah estimasi gabungan berdasarkan varians dalam empat kombinasi baris-kolom. Ini disebut MS_{Within}, atau MS_W. (Untuk rincian tentang MS_W dan perkiraan gabungan, lihat Bab 12.). Untuk contoh ini,

$$MS_W = \frac{(4-1)(12.92)+(4-1)(12.92)+(4-1)(12.25)+(4-1)(12.33)}{(4-1)+(4-1)+(4-1)+(4-1)}$$

$$= \frac{151.25}{12} = 12.60$$

Ini adalah istilah kesalahan (penyebut) untuk setiap F yang Anda hitung. Penyebutnya memberi tahu Anda bahwa df = 12 untuk MS ini.

Varians terakhir berasal dari interaksi antara faktor baris dan faktor kolom. Dalam contoh ini, ini diberi label $MS_{Method\ X\ Type}$. Anda dapat menghitung ini dalam beberapa cara. Cara termudah adalah dengan memanfaatkan hubungan umum ini:

$$SS_{Row\ X\ Column} = SS_T - SS_{Row\ Factor} - SS_{Column\ Factor} - SS_W$$

Dan yang satu ini:

$$df_{Row\ X\ Column} = df_T - df_{Row\ Factor} - df_{Column\ Factor} - df_W$$

Cara lain untuk menghitung ini adalah:

$$df_{Row\ X\ Column} = (\text{number of rows - 1})(\text{number of columns - 1})$$

MS adalah:

$$MS_{Row\ X\ Column} = \frac{SS_{Row\ X\ Column}}{df_{Row\ X\ Column}}$$

Untuk contoh ini,

$$MS_{Method\ X\ Style} = \frac{SS_{Method\ X\ Style}}{df_{Method\ X\ Style}} = \frac{5885.43-203.06-18.06-151.25}{15-12-1-1}$$

$$= \frac{5513.06}{1} = 5513.06$$

Untuk menguji hipotesis, Anda menghitung tiga Fs:

$$F = \frac{MS_{Style}}{MS_W} = \frac{18.06}{12.60} = 1.43$$

$$F = \frac{MS_{Method}}{MS_W} = \frac{203.06}{12.60} = 16.12$$

$$F = \frac{MS_{Method\ X\ Style}}{MS_W} = \frac{5513.06}{12.60} = 437.54$$

Untuk df = 1 dan 12, F kritis pada α = .05 adalah 4,75. (Anda dapat menggunakan qf() untuk memverifikasi). Keputusannya adalah menolak H_0 untuk interaksi Metode Presentasi dan Gaya Presentasi, dan tidak menolak H_0 untuk Gaya Presentasi. Mungkin saja, tentu saja, memiliki lebih dari dua level untuk setiap faktor. Mungkin juga memiliki lebih dari dua faktor. Dalam hal ini, hal-hal (seperti interaksi) menjadi jauh lebih kompleks.

13.2 ANOVA DUA ARAH DALAM R

Seperti dalam analisis apa pun, langkah pertama adalah mendapatkan data dalam bentuk, dan dalam R itu berarti memasukkan data ke dalam format panjang.
Mulailah dengan vektor untuk skor di setiap kolom pada Gambar 13.1:

```
humorous <- c(57,56,60,64,33,25,28,31)
technical <- c(22,21,29,25,66,65,71,72)
```

Kemudian gabungkan mereka untuk menghasilkan vektor semua skor:

```
Score = c(humorous,technical)
```

Selanjutnya, buat vektor untuk Metode dan Gaya:

```
Method =rep(c("spoken","text"),each=4,2)
Style =rep(c("humorous","technical"),each=8)
```

Dan kemudian masukkan semuanya ke dalam bingkai data:

```
pres.frame <-data.frame(Method,Style,Score)
```

yang terlihat seperti ini:

```
> pres.frame
   Method     Style Score
1  spoken  humorous    57
2  spoken  humorous    56
3  spoken  humorous    60
4  spoken  humorous    64
5    text  humorous    33
6    text  humorous    25
7    text  humorous    28
8    text  humorous    31
9  spoken technical    22
10 spoken technical    21
11 spoken technical    29
12 spoken technical    25
13   text technical    66
14   text technical    65
15   text technical    71
16   text technical    72
```

Dan inilah analisis varians dua arah:

```
> two.way <- aov(Score ~ Style*Method,
                 data = pres.frame)
```

Ekspresi Gaya*Metode menunjukkan bahwa semua tingkat Gaya (lucu dan teknis) digabungkan dengan semua tingkat Metode (lisan dan teks). Berikut tabel ANOVA:

```
> summary(two.way)
             Df Sum Sq Mean Sq F value   Pr(>F)
Style         1     18      18   1.433  0.25438
Method        1    203     203  16.111  0.00172 **
Style:Method  1   5513    5513 437.400 8.27e-11 ***
Residuals    12    151      13
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sekali lagi, nilai-f dan nilai-p menunjukkan penolakan hipotesis nol untuk Metode dan interaksi Metode Gaya X, tetapi tidak untuk Gaya. Dengan hanya dua tingkat dari masing-masing faktor, tidak diperlukan tes pasca-analisis untuk mengeksplorasi hasil yang signifikan.

Memvisualisasikan hasil dua arah

Cara terbaik untuk menunjukkan hasil penelitian seperti ini adalah dengan diagram batang yang dikelompokkan yang menunjukkan rata-rata dan kesalahan standar. Landasan plot adalah kerangka data yang menyimpan statistik ini untuk setiap kombinasi level variabel independen:

```
> mse.frame
  Method     Style  Mean       SE
1 spoken  humorous 59.25 1.796988
2   text  humorous 29.25 1.750000
3 spoken technical 24.25 1.796988
4   text technical 68.50 1.755942
```

Untuk membuat bingkai data ini, mulailah dengan membuat empat vektor:

```
Score.spk.hum <- with(pres.frame, Score[Method=="spoken" &
          Style=="humorous"])
Score.txt.hum <- with(pres.frame, Score[Method=="text" &
          Style=="humorous"])
Score.spk.tec <- with(pres.frame, Score[Method=="spoken" &
          Style=="technical"])
Score.txt.tec <- with(pres.frame, Score[Method=="text" &
          Style=="technical"])
```

Kemudian gabungkan sarana vektor menjadi vektor lain:

```
mean.Scores <- c(mean(Score.spk.hum), mean(Score.txt.hum),
         mean(Score.spk.tec), mean(Score.txt.tec))
```

Dan gabungkan kesalahan standar menjadi vektor lain:

```
se.Scores <- c(sd(Score.spk.hum), sd(Score.txt.hum), sd(Score.
          spk.tec), sd(Score.txt.tec))/2
```

Dalam membagi dengan 2, saya sedikit curang pada yang terakhir itu. Setiap kombinasi terdiri dari empat skor, dan akar kuadrat dari 4 adalah 2.
Buat vektor untuk level Metode dan satu lagi untuk level Gaya:

```
mse.Method =rep(c("spoken","text"),2)
mse.Style =rep(c("humorous","technical"),each=2)
```

Kemudian buat bingkai data:

```
mse.frame <- data.frame(mse.Method,mse.Style,mean.Scores,se.Scores)
```

Terakhir, buat nama kolom sedikit lebih bagus:

```
colnames(mse.frame)=c("Method","Style","Mean","SE")
```

Ke visualisasi. Di ggplot2, mulailah dengan pernyataan ggplot() yang memetakan komponen data ke komponen grafik:

```
ggplot(mse.frame,aes(x=Method,y=Mean,fill=Style))
```

Sekarang gunakan geom_bar yang menggunakan mean yang diberikan sebagai statistiknya:

```
  geom_bar(stat = "identity", position = "dodge",
        color = "black", width = .5)
```

Argumen posisi mengatur plot ini sebagai plot batang yang dikelompokkan, argumen warna menentukan "hitam" sebagai warna batas, dan lebar mengatur ukuran untuk batang yang tampak bagus. Anda dapat bereksperimen sedikit untuk melihat apakah lebar lain lebih sesuai dengan keinginan Anda.

Jika Anda tidak mengubah warna batang, mereka akan tampak sebagai merah muda dan biru muda, yang cukup menyenangkan tetapi tidak dapat dibedakan pada halaman hitam-putih. Berikut cara mengubah warna:

```
scale_fill_grey(start = 0,end = .8)
```

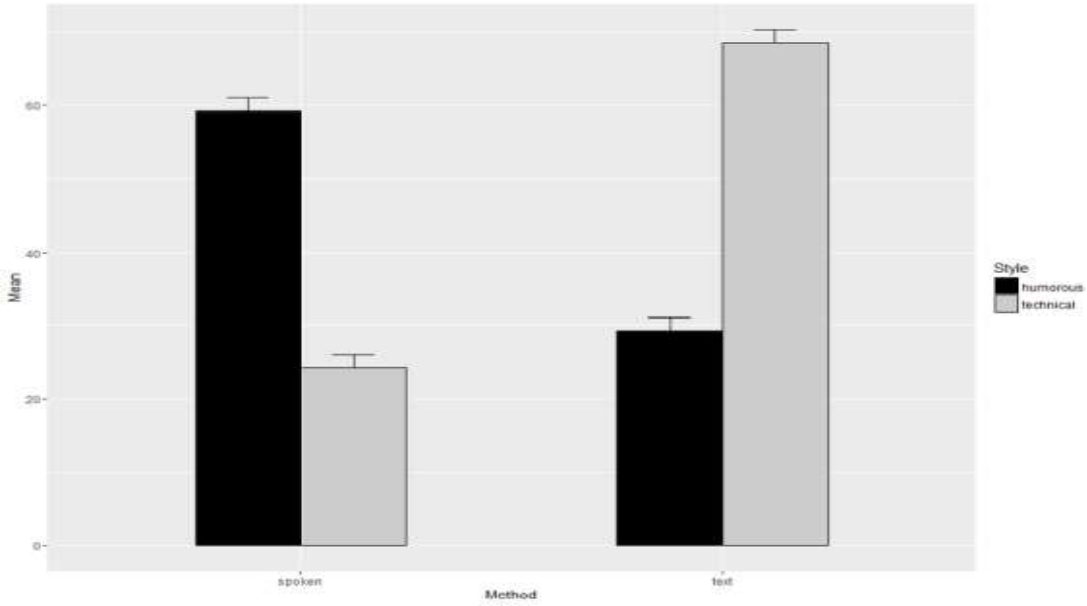
Dalam skala abu-abu, 0 sesuai dengan hitam dan 1 untuk putih. Terakhir, geom_errorbar menambahkan bilah untuk kesalahan standar:

```
geom_errorbar(aes(ymin=Mean,ymax=Mean+SE), width=.2,
              position=position_dodge(width=.5))
```

Menggunakan Mean sebagai nilai ymin memastikan bahwa Anda hanya memplot bilah kesalahan atas, yang biasanya Anda lihat di plot batang yang diterbitkan. Argumen posisi menggunakan fungsi position_dodge() untuk memusatkan bilah kesalahan.
Jadi, baris kode ini:

```
ggplot(mse.frame,aes(x=Method,y=Mean,fill=Style)) +
  geom_bar(stat = "identity", position = "dodge",
           color = "black", width = .5)+
  scale_fill_grey(start = 0,end = .8)+
  geom_errorbar(aes(ymin=Mean,ymax=Mean+SE), width=.2,
           position=position_dodge(width=.5))
```

Menghasilkan Gambar 13.2.



Gambar 13.2 Cara dan kesalahan standar dari studi presentasi.

Grafik ini dengan jelas menunjukkan interaksi Metode X Style. Untuk presentasi lisan, humor lebih efektif daripada teknis, dan sebaliknya untuk presentasi teks.

13.3 DUA MACAM VARIABEL SEKALIGUS

Apa yang terjadi jika Anda memiliki variabel Antara Grup dan variabel Dalam Grup . . . pada waktu bersamaan? Bagaimana itu bisa terjadi? Sangat mudah. Berikut ini contohnya. Misalkan Anda ingin mempelajari pengaruh media presentasi terhadap kecepatan membaca siswa kelas empat. Anda secara acak menugaskan siswa kelas empat (saya akan menyebutnya mata pelajaran) untuk membaca buku atau e-reader. Jadi "Sedang" adalah variabel Antar Grup.

Katakanlah Anda juga tertarik dengan efek font. Jadi, Anda menugaskan setiap subjek untuk membaca setiap font berikut: Haettenschweiler, Arial, dan Calibri. (Saya belum pernah melihat dokumen di Haettenschweiler, tapi itu font favorit saya karena "Haettenschweiler" menyenangkan untuk diucapkan. Cobalah. Benarkah?) Karena setiap subjek membaca semua font, "Font" adalah Grup Dalam variabel. Untuk kelengkapan, Anda harus memesan font secara acak untuk setiap mata pelajaran. Tabel 13.1 menunjukkan data yang mungkin dihasilkan dari penelitian seperti ini. Variabel terikat adalah skor pada tes pemahaman bacaan.

Tabel 13.1 Data Studi Media Presentasi (Variabel Antar Grup) dan Jenis Huruf (Variabel Dalam Grup)

Medium	Subjek	Haettenschweiler	Arial	Calibri
Buku	Alice	48	40	38
	Brad	55	43	45
	Chris	46	45	44
	Donna	61	53	53
Pembaca elektronik	Eddie	43	45	47
	Fran	50	52	54
	Gil	56	57	57
	Harriet	53	53	55

Karena jenis analisis ini mencampurkan variabel Antara Grup dengan variabel Dalam Grup, ini disebut ANOVA Campuran.

Untuk menunjukkan cara kerja analisis, saya menyajikan jenis tabel yang dihasilkan dari ANOVA Campuran. Ini sedikit lebih lengkap daripada output ANOVA di R, tapi bersabarlah. Tabel 13.2 menunjukkannya kepada Anda secara umum. Ini dikategorikan ke dalam satu set sumber yang membentuk variabilitas Antara Grup dan satu set sumber yang membentuk variabilitas Dalam Grup (juga dikenal sebagai Tindakan Berulang).

Dalam kategori Antara, A adalah nama variabel Antara Grup. (Dalam contoh, itu Medium.) Baca “S/A” sebagai “Subjek dalam A.” Ini hanya mengatakan bahwa orang-orang di satu level A berbeda dari orang-orang di level A lainnya. Dalam kategori Dalam, B adalah nama variabel Dalam Grup. (Contohnya adalah Font.) A X B adalah interaksi dua variabel. B X S/A

adalah sesuatu seperti variabel B yang berinteraksi dengan subjek dalam A. Seperti yang Anda lihat, apa pun yang terkait dengan B termasuk dalam kategori Dalam Grup.

Tabel 13.2 Tabel ANOVA untuk ANOVA Campuran

Source	SS	df	MS	F
Between	$SS_{Between}$	$df_{Between}$		
A	SS_A	df_A	SS_A/df_A	$MS_A/MS_{S/A}$
S/A	$SS_{S/A}$	$df_{S/A}$	$SS_{S/A}/df_{S/A}$	
Within	SS_{Within}	df_{Within}		
B	SS_B	df_B	SS_B/df_B	$MS_B/MS_{B\ X\ S/A}$
A X B	$SS_{A\ X\ B}$	$df_{A\ X\ B}$	$SS_{A\ X\ B}/df_{A\ X\ B}$	$MS_{A\ X\ B}/MS_{B\ X\ S/A}$
B X S/A	$SS_{B\ X\ S/A}$	$df_{B\ XS/A}$	$SS_{B\ X\ S/A}/df_{B\ X\ S/A}$	
Total	SS_{Total}	df_{Total}		

Hal pertama yang perlu diperhatikan adalah tiga F-rasio. Yang pertama menguji perbedaan antara tingkat A, yang kedua untuk perbedaan antara tingkat B, dan yang ketiga untuk interaksi keduanya. Perhatikan juga bahwa penyebut untuk rasio-F pertama berbeda dengan penyebut untuk dua lainnya. Ini semakin sering terjadi seiring dengan meningkatnya kompleksitas ANOVA.

Selanjutnya, penting untuk menyadari beberapa hubungan. Di tingkat atas:

$$SS_{Between} + SS_{Within} = SS_{Total}$$

$$df_{Between} + df_{Within} = df_{Total}$$

Komponen Antara rusak lebih lanjut:

$$SS_A + SS_{S/A} = SS_{Between}$$

$$df_A + df_{S/A} = df_{Between}$$

Komponen Dalam juga rusak:

$$SS_B + SS_{A\ X\ B} + SS_{B\ X\ S/A} = SS_{Within}$$

$$df_B + df_{A\ X\ B} + df_{B\ X\ S/A} = df_{Within}$$

Dimungkinkan untuk memiliki lebih dari satu faktor Antar Grup dan lebih dari satu pengukuran berulang dalam sebuah penelitian.

Campuran ANOVA dalam R

Pertama, saya tunjukkan cara menggunakan data dari Tabel 13.1 untuk membangun bingkai data dalam format panjang. Setelah selesai, tampilannya seperti ini:

```
> mixed.frame
     Medium             Font Subject Score
1      Book Haettenschweiler   Alice    48
2      Book Haettenschweiler    Brad    55
3      Book Haettenschweiler   Chris    46
4      Book Haettenschweiler   Donna    61
5      Book            Arial   Alice    40
6      Book            Arial    Brad    43
7      Book            Arial   Chris    45
8      Book            Arial   Donna    53
9      Book          Calibri   Alice    38
10     Book          Calibri    Brad    45
11     Book          Calibri   Chris    44
12     Book          Calibri   Donna    53
13 E-reader Haettenschweiler   Eddie    43
14 E-reader Haettenschweiler    Fran    50
15 E-reader Haettenschweiler     Gil    56
16 E-reader Haettenschweiler Harriet    53
17 E-reader            Arial   Eddie    45
18 E-reader            Arial    Fran    52
19 E-reader            Arial     Gil    57
20 E-reader            Arial Harriet    53
21 E-reader          Calibri   Eddie    47
22 E-reader          Calibri    Fran    54
23 E-reader          Calibri     Gil    57
24 E-reader          Calibri Harriet    55
```

Saya mulai dengan vektor untuk skor Buku dan vektor untuk skor e-reader:

```
BkScores <- c(48,55,46,61,40,43,45,53,38,45,44,53)
ErScores <- c(43,50,56,53,45,52,57,53,47,54,57,55)
```

Lalu saya gabungkan menjadi sebuah vektor:

```
Score <-c(BkScores,ErScores)
```

Saya menyelesaikan proses serupa untuk mata pelajaran: satu vektor untuk mata pelajaran Buku dan satu lagi untuk mata pelajaran e-reader. Perhatikan bahwa saya harus mengulangi setiap daftar tiga kali:

```
BkSubjects <- rep(c("Alice","Brad","Chris","Donna"),3)
ErSubjects <- rep(c("Eddie","Fran","Gil","Harriet"),3)
```

Kemudian saya menggabungkan keduanya:

```
Subject <- c(BkSubjects,ErSubjects)
```

Selanjutnya adalah vektor untuk Buku versus e-reader, dan perhatikan bahwa saya mengulangi daftar itu 12 kali:

```
Medium <- rep(c("Book","E-reader"),each=12)
```

Vektor untuk Font agak rumit. Saya harus mengulangi setiap nama font empat kali dan mengulanginya lagi:

```
Font <- rep(c("Haettenschweiler","Arial","Calibri"),
          each=4,2)
```

Saya sekarang dapat membuat bingkai data:

```
mixed.frame <-data.frame(Medium,Font,Subject,Score)
```

Analisisnya adalah:

```
mixed.anova <- aov(Score ~ Medium*Font + Error(Subject/Font),
          data=mixed.frame)
```

Argumen menunjukkan bahwa Skor bergantung pada Medium dan Font dan Font tersebut diulang di setiap Subjek.

Untuk melihat tabel:

```
> summary(mixed.anova)

Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Medium     1  108.4  108.37   1.227   0.31
Residuals  6  529.9   88.32
```

```
Error: Subject:Font
           Df Sum Sq Mean Sq F value   Pr(>F)
Font        2  40.08   20.04   5.681 0.018366 *
Medium:Font 2 120.25   60.13  17.043 0.000312 ***
Residuals  12  42.33    3.53
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anda dapat menolak hipotesis nol tentang Font dan tentang interaksi Medium dan Font, tetapi tidak tentang Medium.

Memvisualisasikan hasil Mixed ANOVA

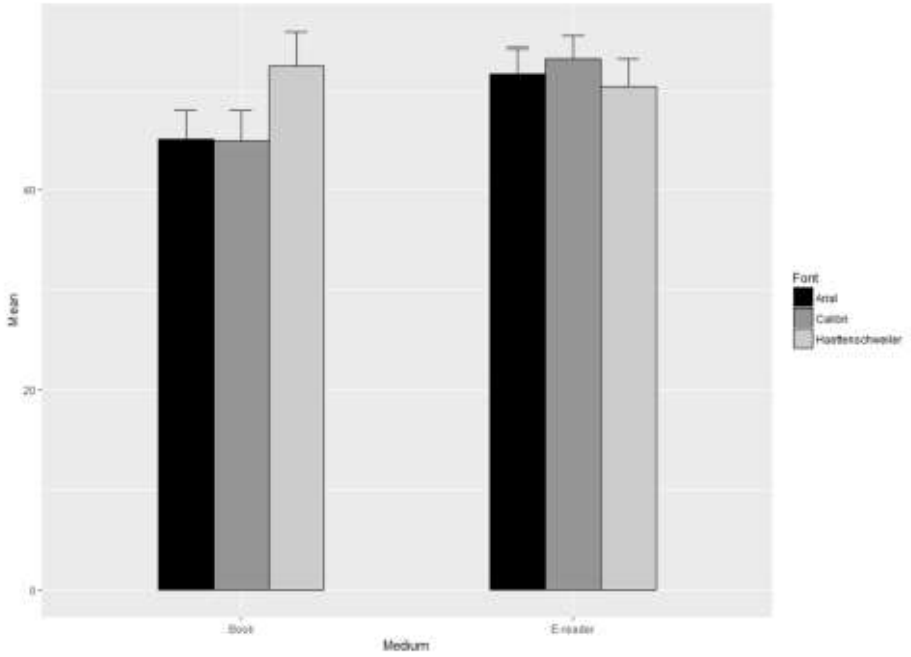
Anda menggunakan ggplot() untuk membuat plot batang rata-rata dan kesalahan standar. Mulailah dengan membuat bingkai data ini, yang berisi informasi yang diperlukan:

```
> mse.frame
    Medium             Font  Mean       SE
1     Book Haettenschweiler 52.50 3.427827
2     Book            Arial 45.25 2.780138
3     Book          Calibri 45.00 3.082207
4 E-reader Haettenschweiler 50.50 2.783882
5 E-reader            Arial 51.75 2.495830
6 E-reader          Calibri 53.25 2.174665
```

Untuk membuat kerangka data ini, ikuti langkah yang sama seperti di bagian "Memvisualisasikan hasil dua arah" sebelumnya, dengan perubahan yang sesuai. Kode ggplot juga sama seperti di bagian sebelumnya, dengan perubahan nama variabel:

```
ggplot(mse.frame,aes(x=Medium,y=Mean,fill=Font)) +
  geom_bar(stat = "identity", position =
           "dodge",color="black",width = .5) +
  scale_fill_grey(start = 0,end = .8) +
  geom_errorbar(aes(ymin=Mean,ymax=Mean+SE),
           width=.2,position=position_dodge(width=.5))
```

Hasilnya adalah Gambar 13.3. Gambar menunjukkan variabel Antara Grup pada sumbu x dan tingkat pengukuran berulang di batang — tetapi itu hanya preferensi saya. Anda mungkin lebih suka sebaliknya. Dalam tata letak ini, urutan ketinggian batang yang berbeda dari Buku ke e-reader mencerminkan interaksi.



Gambar 13.3 Cara dan kesalahan standar untuk studi Book versus e-reader.

13.4 SETELAH ANALISIS

Seperti yang saya tunjukkan di Bab 12, hasil signifikan dalam ANOVA memberi tahu Anda bahwa ada efek yang bersembunyi di suatu tempat dalam data. Tes pasca-analisis memberi tahu Anda di mana. Dua jenis tes yang mungkin: direncanakan atau tidak direncanakan. Bab 12 memberikan rinciannya. Dalam contoh ini, variabel Antar Grup hanya memiliki dua level. Untuk alasan ini, jika hasilnya signifikan secara statistik, tidak diperlukan pengujian lebih lanjut. Variabel Dalam Grup, Font, adalah signifikan. Biasanya, pengujian akan dilanjutkan seperti yang dijelaskan dalam Bab 12. Namun, dalam kasus ini, interaksi antara Media dan Font memerlukan jalur yang berbeda.

Dengan interaksi, tes pasca-analisis dapat dilanjutkan dengan salah satu (atau keduanya) dari dua cara. Anda dapat memeriksa efek dari setiap tingkat variabel A (variabel Antar Kelompok) pada tingkat variabel B (ukuran berulang), atau Anda dapat memeriksa efek dari setiap tingkat variabel B pada tingkat A variabel. Ahli statistik menyebutnya sebagai efek utama yang sederhana. Untuk contoh ini, cara pertama meneliti arti dari tiga font dalam sebuah buku dan arti dari tiga font di e-reader. Cara kedua menguji rata-rata untuk buku versus rata-rata untuk e-reader dengan font Haettenschweiler, dengan Arial, dan dengan Calibri.

Teks statistik memberikan rumus rumit untuk menghitung analisis ini. R membuatnya mudah. Untuk menganalisis ketiga font dalam buku, lakukan ANOVA langkah-langkah berulang untuk Mata Pelajaran 1-4. Untuk menganalisis ketiga font dalam e-reader, lakukan pengukuran ANOVA berulang untuk Subjek 5–8. Untuk analisis buku versus e-reader dalam font Haettenschweiler, itu adalah ANOVA faktor tunggal untuk data Haettenschweiler. Anda akan menyelesaikan prosedur serupa untuk setiap font lainnya.

13.5 ANALISIS VARIANS MULTIVARIAT

Contoh-contoh sejauh ini dalam bab ini melibatkan variabel dependen dan lebih dari satu variabel independen. Apakah mungkin untuk memiliki lebih dari satu variabel terikat? Sangat! Itu memberi Anda MANOVA — singkatan untuk judul bagian ini.

Kapan Anda mungkin menghadapi situasi seperti ini? Misalkan Anda berpikir untuk mengadopsi salah satu dari tiga buku teks untuk kursus sains dasar. Anda memiliki 12 siswa, dan Anda secara acak menugaskan empat dari mereka untuk membaca Buku 1, empat lainnya untuk membaca Buku 2, dan empat sisanya untuk membaca Buku 3. Anda tertarik pada bagaimana setiap buku meningkatkan pengetahuan dalam fisika, kimia, dan biologi, jadi setelah siswa membaca buku, mereka mengikuti tes pengetahuan dasar di masing-masing dari ketiga ilmu tersebut. Variabel independen adalah Buku, dan variabel dependen adalah multivariat — itu adalah vektor yang terdiri dari skor Fisika, skor Kimia, dan skor Biologi. Tabel 13.3 menunjukkan data.

Tabel 13.3 Data untuk Buku Ajar IPA Studi MANOVA

Siswa	Buku	Fisika	Kimia	Biologi
Art	Buku 1	50	66	71
Brenda	Buku 1	53	45	56
Cal	Buku 1	52	48	65
Dan	Buku 1	54	51	68
Eva	Buku 2	75	55	88
Frank	Buku 2	72	58	85
Greg	Buku 2	64	59	79
Hank	Buku 2	76	59	82
Iris	Buku 3	68	67	55
Jim	Buku 3	61	56	59

Kendra	Buku 3	62	66	63
Lee	Buku 3	64	78	61

Variabel terikat untuk siswa pertama dalam sampel Buku 1 adalah vektor yang terdiri dari 50, 66, dan 71. Apa hipotesis dalam kasus seperti ini? Hipotesis nol harus memperhitungkan semua komponen vektor, jadi inilah nol dan alternatifnya:

$$H_0 : \begin{pmatrix} \mu_{Book1,Phys} \\ \mu_{Book1,Chem} \\ \mu_{Book1,Bio} \end{pmatrix} = \begin{pmatrix} \mu_{Book2,Phys} \\ \mu_{Book2,Chem} \\ \mu_{Book2,Bio} \end{pmatrix} = \begin{pmatrix} \mu_{Book3,Phys} \\ \mu_{Book3,Chem} \\ \mu_{Book3,Bio} \end{pmatrix}$$

$$H_1 : Not\ H_0$$

Saya tidak masuk ke kedalaman yang sama tentang MANOVA dalam bab ini seperti yang saya lakukan di ANOVA. Saya tidak membahas SS, MS, dan df. Itu akan membutuhkan pengetahuan matematika (aljabar matriks) dan materi lain yang berada di luar cakupan bab ini. Sebagai gantinya, saya langsung masuk dan menunjukkan kepada Anda bagaimana menyelesaikan analisis.

MANOVA di R

Kerangka data untuk MANOVA terlihat seperti Tabel 13.3:

```
> Textbooks.frame
   Student  Book Physics Chemistry Biology
1      Art Book1      50        66      71
2   Brenda Book1      53        45      56
3      Cal Book1      52        48      65
4      Dan Book1      54        51      68
5      Eva Book2      75        55      88
6    Frank Book2      72        58      85
7     Greg Book2      64        59      79
8     Hank Book2      76        59      82
9     Iris Book3      68        67      55
10     Jim Book3      61        56      59
11  Kendra Book3      62        66      63
12     Lee Book3      64        78      61
```

Dalam ANOVA, variabel dependen untuk analisis adalah kolom tunggal. Dalam MANOVA, variabel dependen untuk analisis adalah matriks. Dalam hal ini, ini adalah matriks dengan 12 baris (satu untuk setiap siswa) dan tiga kolom (Fisika, Kimia, dan Biologi).

Untuk membuat matriks, gunakan fungsi cbind() untuk mengikat kolom yang sesuai. Anda dapat melakukan ini di dalam fungsi manova() yang melakukan analisis:

```
m.analysis <- manova(cbind(Physics,Chemistry,Biology) ~ Book,
          data = Textbooks.frame)
```

Rumus di dalam tanda kurung menunjukkan matriks 12 X 3 (hasil dari cbind()) tergantung pada Buku, dengan Textbooks.frame sebagai sumber datanya.
Seperti biasa, terapkan ringkasan() untuk melihat tabel:

```
> summary(m.analysis)
          Df Pillai approx F num Df den Df    Pr(>F)
Book       2 1.7293   17.036      6     16 3.922e-06 ***
Residuals  9
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Satu-satunya item baru adalah Pillai, statistik uji yang dihasilkan dari MANOVA. Ini sedikit rumit, jadi saya akan membiarkannya sendiri. Cukuplah untuk mengatakan bahwa R mengubah Pillai menjadi rasio-F (dengan 6 dan 16 df) dan itulah yang Anda gunakan sebagai statistik uji. F tinggi dan nilai p yang sangat rendah menunjukkan penolakan hipotesis nol.
Pillai adalah tes default. Dalam pernyataan ringkasan, Anda dapat menentukan statistik uji MANOVA lainnya. Mereka disebut "Wilks", "Hotelling-Lawley", dan "Roy". Sebagai contoh:

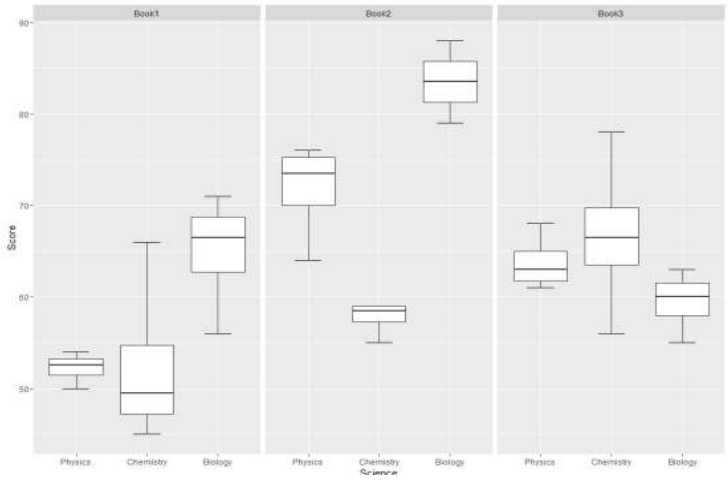
```
> summary(m.analysis, test = "Roy")
          Df    Roy approx F num Df den Df    Pr(>F)
Book       2 10.926   29.137      3      8 0.0001175 ***
Residuals  9
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pengujian yang berbeda menghasilkan nilai yang berbeda untuk F dan df, tetapi keputusan keseluruhannya sama.

Contoh ini adalah perpanjangan MANOVA dari ANOVA hanya dengan satu faktor. Ada kemungkinan untuk memiliki beberapa variabel dependen dengan desain yang lebih kompleks (seperti yang saya diskusikan sebelumnya dalam bab ini).

Memvisualisasikan hasil MANOVA

Tujuan dari penelitian ini adalah untuk menunjukkan bagaimana distribusi skor Fisika, Kimia, dan Biologi berbeda dari buku ke buku. Satu set plot kotak terpisah untuk setiap buku memvisualisasikan perbedaannya. Gambar 13-4 menunjukkan apa yang saya bicarakan.



Gambar 13.4 Tiga boxplot menunjukkan distribusi nilai Fisika, Kimia, dan Biologi untuk setiap buku.

Kemampuan faceting ggplot2 membagi data berdasarkan Buku dan membuat tiga grafik berdampingan. Setiap graf disebut faset. (Lihat bagian “Menjelajahi data” di Bab 4).

Untuk mengatur ini semua, Anda harus membentuk kembali Textbooks.frame menjadi format panjang. Dengan paket reshape2 terinstal (pada tab Packages, pilih kotak centang di sebelah reshape2), terapkan fungsi melt():

```
Textbooks.frame.melt = melt(Textbooks.frame)
```

Setelah menetapkan nama kolom:

```
colnames(Textbooks.frame.melt) = c("Student", "Book", "Science",
          "Score")
```

enam baris pertama dari bingkai yang dilebur adalah:

```
> head(Textbooks.frame.melt)
  Student  Book Science Score
1     Art Book1 Physics    50
2  Brenda Book1 Physics    53
3     Cal Book1 Physics    52
4     Dan Book1 Physics    54
5     Eva Book2 Physics    75
6   Frank Book2 Physics    72
```

Untuk membuat Gambar 13.4 di ggplot2, mulailah dengan

```
ggplot(Textbooks.frame.melt,(aes(x=Science,y=Score)))
```

yang menunjukkan kerangka data dan secara estetis memetakan Sains ke sumbu x dan Skor ke sumbu y.

Selanjutnya, gunakan stat_boxplot() untuk menghitung garis tegak lurus untuk kumis:

```
stat_boxplot(geom="errorbar", width =.5)
```

Kemudian, fungsi geom untuk boxplot:

```
geom_boxplot()
```

Dan, akhirnya, pernyataan yang membagi data berdasarkan Buku dan membuat deretan tiga grafik (permisi — segi):

```
facet_grid(. ~ Book)
```

Titik yang diikuti dengan tilde (~) diikuti oleh Book mengatur segi-segi secara berdampingan. Untuk menempatkan tiga grafik dalam kolom, itu

```
facet_grid(Book ~ .)
```

Menyatukan semuanya, kode untuk membuat Gambar 13.4 adalah

```
ggplot(Textbooks.frame.melt,(aes(x=Science,y=Score)))+
  stat_boxplot(geom="errorbar", width =.5) +
  geom_boxplot() +
  facet_grid(. ~ Book)
```

Ketika MANOVA menghasilkan penolakan hipotesis nol, salah satu cara untuk melanjutkan adalah dengan melakukan ANOVA pada setiap komponen variabel dependen. Hasilnya memberi tahu Anda komponen mana yang berkontribusi pada MANOVA yang signifikan.

Fungsi summary.aov() melakukan ini untuk Anda. Ingat bahwa m.analysis menyimpan hasil MANOVA dalam contoh bagian ini:

```
> summary.aov(m.analysis)
 Response Physics :
            Df Sum Sq Mean Sq F value    Pr(>F)
Book         2 768.67  384.33  27.398 0.0001488 ***
Residuals    9 126.25   14.03
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Response Chemistry :
            Df Sum Sq Mean Sq F value  Pr(>F)
Book         2  415.5 207.750  3.6341 0.06967 .
Residuals    9  514.5  57.167
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Response Biology :
            Df Sum Sq Mean Sq F value    Pr(>F)
Book         2 1264.7  632.33  27.626 0.0001441 ***
Residuals    9  206.0   22.89
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analisis ini menunjukkan bahwa Fisika dan Biologi berkontribusi pada efek keseluruhan, dan Kimia hanya kehilangan signifikansi.

Perhatikan kata Response dalam tabel ini. Ini adalah R-terminologi untuk "variabel dependen". Prosedur ANOVA terpisah ini tidak mempertimbangkan hubungan antar pasangan komponen. Hubungan itu disebut korelasi, yang saya bahas di Bab 15.

BAB 14
REGRESI: MODEL LINIER, KELIPATAN, DAN UMUM

Salah satu hal utama yang Anda lakukan saat bekerja dengan statistik adalah membuat prediksi. Idenya adalah menggunakan data dari satu atau lebih variabel untuk memprediksi nilai variabel lain. Untuk melakukan ini, Anda harus memahami bagaimana meringkas hubungan antar variabel, dan menguji hipotesis tentang hubungan tersebut.

Dalam bab ini, saya memperkenalkan regresi, cara statistik untuk melakukan hal itu. Regresi juga memungkinkan Anda menggunakan detail hubungan untuk membuat prediksi. Pertama, saya tunjukkan cara menganalisis hubungan antara satu variabel dengan variabel lainnya. Kemudian saya menunjukkan kepada Anda bagaimana menganalisis hubungan antara variabel dan dua lainnya. Akhirnya, saya memberi tahu Anda tentang hubungan antara regresi dan ANOVA.

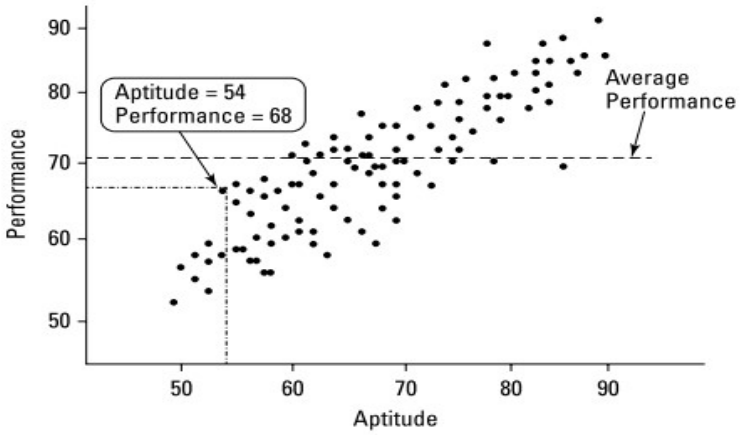
14.1 PLOT PENCAR

FarMisht Consulting, Inc., adalah perusahaan konsultan dengan berbagai spesialisasi. Ini menerima banyak aplikasi dari orang-orang yang tertarik untuk menjadi konsultan FarMisht. Dengan demikian, Sumber Daya Manusia FarMisht harus dapat memprediksi pelamar mana yang akan berhasil dan mana yang tidak. Mereka telah mengembangkan ukuran Kinerja yang mereka gunakan untuk menilai karyawan mereka saat ini. Skalanya adalah 0–100, di mana 100 menunjukkan kinerja terbaik.

Apa prediksi terbaik untuk pelamar baru? Tanpa mengetahui apa pun tentang pelamar, dan hanya mengetahui skor Kinerja karyawan mereka sendiri, jawabannya jelas: Ini adalah skor Kinerja rata-rata di antara karyawan mereka. Terlepas dari siapa pelamarnya, hanya itu yang bisa dikatakan tim Sumber Daya Manusia jika pengetahuan anggotanya terbatas.

Dengan lebih banyak pengetahuan tentang karyawan dan pelamar, prediksi yang lebih akurat menjadi mungkin. Misalnya, jika FarMisht mengembangkan tes bakat dan menilai karyawannya, Sumber Daya Manusia dapat mencocokkan skor Kinerja setiap karyawan dengan skor Aptitude mereka dan melihat apakah kedua bagian data tersebut terkait. Jika ya, pelamar dapat mengikuti tes bakat FarMisht, dan Sumber Daya Manusia dapat menggunakan skor itu (dan hubungan antara Bakat dan Kinerja) untuk membantu membuat prediksi.

Gambar 14.1 menunjukkan pertarungan Aptitude-Performance secara grafis. Karena titik-titiknya tersebar, maka disebut scatter plot. Berdasarkan konvensi, sumbu vertikal (sumbu y) mewakili apa yang Anda coba prediksi. Itu juga disebut variabel dependen, atau variabel y. Dalam hal ini, itulah Performa. Juga menurut konvensi, sumbu horizontal (sumbu x) mewakili apa yang Anda gunakan untuk membuat prediksi. Itu juga disebut variabel bebas, atau variabel-x. Di sini, itu adalah Aptitude.



Gambar 14.1 Bakat dan Kinerja di FarMisht Consulting.

Setiap titik dalam grafik mewakili Performa dan Bakat individu. Dalam plot pencar untuk perusahaan kehidupan nyata, Anda akan melihat lebih banyak poin daripada yang saya tunjukkan di sini. Kecenderungan umum dari kumpulan poin tampaknya adalah bahwa skor Bakat yang tinggi dikaitkan dengan skor Kinerja yang tinggi dan bahwa skor Bakat yang rendah dikaitkan dengan skor Kinerja yang rendah.

Saya telah memilih salah satu poin. Ini menunjukkan seorang karyawan FarMisht dengan skor Aptitude 54 dan skor Kinerja 58. Saya juga menunjukkan skor Kinerja rata-rata, untuk memberi Anda perasaan bahwa mengetahui hubungan Aptitude-Kinerja memberikan keuntungan daripada hanya mengetahui rata-ratanya. Bagaimana Anda membuat keuntungan itu bekerja untuk Anda? Anda mulai dengan meringkas hubungan antara Aptitude dan Performance. Rangkuman adalah garis yang melalui titik-titik. Bagaimana dan di mana Anda menggambar garis?

Saya mendapatkan itu dalam satu menit. Pertama, saya harus memberi tahu Anda tentang garis secara umum.

14.2 GARIS GRAFIK

Dalam dunia matematika, garis adalah cara untuk menggambarkan hubungan antara variabel bebas (x) dan variabel terikat (y). Dalam hubungan ini,

$y = 4 + 2x$

Jika Anda memberikan nilai untuk x, Anda dapat mengetahui nilai yang sesuai untuk y. Persamaan mengatakan untuk mengalikan nilai x dengan 2 dan kemudian menambahkan 3.

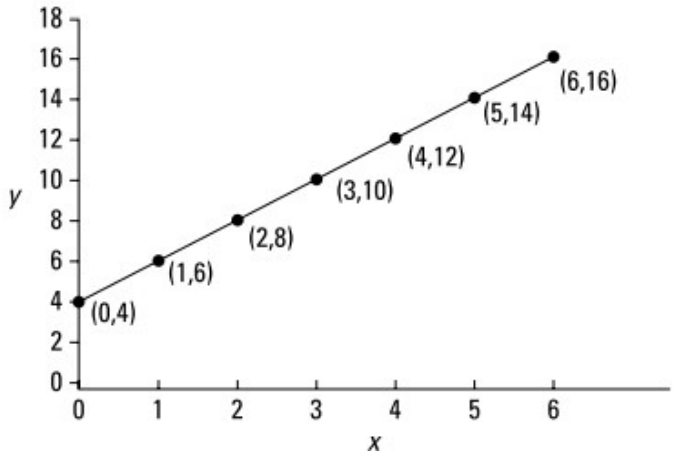
Jika x = 1, misalnya, y = 6. Jika x = 2, y = 8. Tabel 14-1 menunjukkan jumlah pasangan x - y pada hubungan ini, termasuk pasangan di mana x = 0.

Tabel 14-1 x-y Pair dalam y = 4 + 2x

x	y
0	4

1	6
2	8
3	10
4	12
5	14
6	16

Gambar 14.2 menunjukkan pasangan-pasangan ini sebagai titik-titik pada himpunan sumbu xy, bersama dengan garis yang melalui titik-titik tersebut. Setiap kali saya mencantumkan pasangan xy dalam tanda kurung, nilai x adalah yang pertama.



Gambar 14.2 Grafik untuk y 4 2x.

Seperti yang ditunjukkan gambar, titik-titik jatuh dengan rapi ke garis. Garis tersebut menggambarkan persamaan y = 4 + 2x. Faktanya, setiap kali Anda memiliki persamaan seperti ini, di mana x tidak dikuadratkan atau pangkat tiga atau dipangkatkan lebih tinggi dari 1, Anda memiliki apa yang disebut matematikawan sebagai persamaan linier. (Jika x dinaikkan ke pangkat yang lebih tinggi dari 1, Anda menghubungkan titik-titik dengan kurva, bukan garis.) Beberapa hal yang perlu diingat tentang sebuah garis: Anda dapat menggambarkan sebuah garis dalam hal kemiringannya, dan di mana garis itu menuju sumbu y.

Bagian bagaimana-miringnya adalah lereng. Kemiringan memberi tahu Anda berapa banyak y berubah ketika x berubah satu unit. Pada garis yang ditunjukkan pada Gambar 14.2, ketika x berubah satu (dari 4 menjadi 5, misalnya), y berubah dua (dari 12 menjadi 14). Bagian di mana ia berjalan ke sumbu y disebut perpotongan y (atau kadang-kadang hanya penyadapan). Itulah nilai y saat x = 0. Pada Gambar 14.2, perpotongan y adalah 4.

Anda dapat melihat angka-angka ini dalam persamaan. Kemiringan adalah angka yang mengalikan x, dan intersep adalah angka yang Anda tambahkan ke x. Secara umum,

y = a + bx

di mana a mewakili intersep dan b mewakili kemiringan.

Kemiringan dapat berupa bilangan positif, bilangan negatif, atau 0. Pada Gambar 14.2, kemiringannya positif. Jika kemiringannya negatif, garis miring ke arah yang berlawanan dengan apa yang Anda lihat pada Gambar 14.2. Kemiringan negatif berarti bahwa y berkurang sebagai x meningkat. Jika kemiringannya 0, garisnya sejajar dengan sumbu horizontal. Jika kemiringannya 0, y tidak berubah saat x berubah.

Hal yang sama berlaku untuk intersep — dapat berupa bilangan positif, bilangan negatif, atau 0. Jika intersep positif, garis memotong sumbu y di atas sumbu x. Jika intersepnya negatif, garis memotong sumbu y di bawah sumbu x. Jika intersep adalah 0, ia berpotongan dengan sumbu y dan sumbu x, pada titik yang disebut titik asal.

Dan sekarang, kembali ke apa yang saya bicarakan sebelumnya.

14.3 REGRESI GARIS

Saya menyebutkan sebelumnya bahwa garis adalah cara terbaik untuk meringkas hubungan dalam plot pencar pada Gambar 14-1. Dimungkinkan untuk menggambar garis lurus dalam jumlah tak terbatas melalui plot pencar. Manakah yang paling tepat meringkas hubungan tersebut?

Secara intuitif, garis yang “paling pas” seharusnya merupakan garis yang melewati jumlah titik maksimum dan tidak terlalu jauh dari titik yang tidak dilewatinya. Untuk ahli statistik, garis itu memiliki sifat khusus: Jika Anda menggambar garis itu melalui sebar plot, kemudian menggambar jarak (dalam arah vertikal) antara titik dan garis, dan kemudian kuadratkan jarak dan menjumlahkannya, jumlah dari kuadrat jarak adalah minimum.

Ahli statistik menyebut garis ini sebagai garis regresi, dan mereka menunjukkannya sebagai

$$y' = a + bx$$

Setiap y' adalah titik pada garis. Ini mewakili prediksi terbaik dari y untuk nilai x yang diberikan. Untuk mengetahui dengan tepat di mana garis ini, Anda menghitung kemiringan dan intersepnya. Untuk garis regresi, kemiringan dan intersep disebut koefisien regresi.

Rumus untuk koefisien regresi cukup sederhana. Untuk kemiringan, rumusnya adalah:

$$b = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

Rumus intersepnya adalah:

$$a = \bar{y} - b\bar{x}$$

Saya ilustrasikan dengan sebuah contoh. Untuk menjaga agar angka-angka tersebut dapat dikelola dan dipahami, saya menggunakan sampel kecil alih-alih ratusan (atau mungkin ribuan) karyawan yang Anda temukan di plot pencar untuk sebuah perusahaan. Tabel 14.2 menunjukkan contoh data dari 16 konsultan FarMisht.

Tabel 14.2 Skor Bakat dan Skor Kinerja untuk 16 Konsultan FarMisht

Konsultan	Bakat	Pertunjukan
1	45	56
2	81	74
3	65	56
4	87	81
5	68	75
6	91	84
7	77	68
8	61	52
9	55	57
10	66	82
11	82	73
12	93	90
13	76	67
14	83	79
15	61	70
16	74	66
Mean	72.81	70.63
Variance	181.63	126.65
Standar Deviasi	13.48	11.25

Untuk kumpulan data ini, kemiringan garis regresi adalah:

$$b = \frac{(45-72.81)(56-70.63)+(81-72.81)(74-70.63)+\ldots+(74-72.81)(66-70.63)}{(45-72.81)^2+(81-72.81)^2+\ldots+(74-72.81)^2}$$

$$= 0.654$$

Intersep adalah:

$$a = \bar{y} - b\bar{x} = 70.63 - 0.654(72.81) = 23.03$$

Jadi persamaan garis yang paling tepat melalui 16 titik ini adalah:

$$y' = 23.03 + 0.654x$$

Atau, dalam hal Performa dan Bakat, itu

$$Predicted\ Performance = 23.03 + 0.654(Aptitude)$$

Kemiringan dan perpotongan garis regresi secara umum disebut koefisien regresi.

Menggunakan regresi untuk peramalan

Berdasarkan sampel ini dan garis regresi ini, Anda dapat mengambil skor Bakat pelamar — katakanlah, 85 — dan memprediksi Kinerja pelamar:

$$Predicted\ Performance = 23.03 + 0.654(85) = 78.59$$

Tanpa garis regresi ini, satu-satunya prediksi adalah Kinerja rata-rata, 70,63.

Variasi di sekitar garis regresi

Dalam Bab 5, saya menjelaskan bagaimana mean tidak menceritakan keseluruhan cerita tentang sekumpulan data. Anda harus menunjukkan bagaimana skor bervariasi di sekitar rata-rata. Untuk alasan itu, saya memperkenalkan varians dan standar deviasi. Anda memiliki situasi yang sama di sini. Untuk mendapatkan gambaran lengkap tentang hubungan dalam sebar plot, Anda harus menunjukkan bagaimana skor bervariasi di sekitar garis regresi. Di sini, saya memperkenalkan varians residual dan kesalahan standar estimasi, yang analog dengan varians dan standar deviasi.

Varians residual adalah semacam rata-rata deviasi kuadrat dari nilai-y yang diamati di sekitar nilai-y yang diprediksi. Setiap penyimpangan titik data dari titik prediksi (y - y′) disebut residual; maka nama. Rumusnya adalah:

$$s_{yx}^2 = \frac{\sum (y - y')^2}{N - 2}$$

Saya katakan “semacam” karena penyebutnya adalah N-2 dan bukan N. Memberi tahu Anda alasan untuk –2 berada di luar cakupan diskusi ini. Seperti yang saya sebutkan sebelumnya, penyebut dari estimasi varians adalah derajat kebebasan (df), dan konsep itu akan berguna sebentar lagi.

Kesalahan standar pendugaan adalah:

$$s_{yx} = \sqrt{s_{yx}^2} = \sqrt{\frac{\sum (y - y')^2}{N \quad 2}}$$

Untuk menunjukkan kepada Anda bagaimana kesalahan residual dan kesalahan standar estimasi berlaku untuk data dalam contoh, berikut adalah Tabel 14.3. Tabel ini memperluas Tabel 14.2 dengan menunjukkan skor Performa yang diprediksi untuk setiap skor Aptitude yang diberikan:

Tabel 14.3 Skor Bakat, Skor Performa, dan Prediksi Skor Performa untuk 16 Konsultan FarMisht

Konsultan	Bakat	Pertunjukan	Kinerja yang diprediksi
1	45	56	52.44
2	81	74	75.98
3	65	56	65.52
4	87	81	79.90
5	68	75	67.48
6	91	84	82.51

7	77	68	73.36
8	61	52	62.90
9	55	57	58.98
10	66	82	66.17
11	82	73	76.63
12	93	90	83.82
13	76	67	72.71
14	83	79	77.28
15	61	70	62.90
16	74	66	71.40
Mean	72.81	70.63	
Variance	181.63	126.65	
Standar Deviasi	13.48	11.25	

Seperti yang ditunjukkan tabel, terkadang skor Performa yang diprediksi cukup mendekati, dan terkadang tidak.

Untuk data ini, varians residual adalah:

$$s_{yx}^2 = \frac{\sum(y-y')^2}{N-2} = \frac{(56-52.44)^2+(74-75.98)^2+\ldots+(66-71.40)^2}{16-2} = \frac{735.65}{14} = 52.54$$

Kesalahan standar pendugaan adalah:

$$s_{yx} = \sqrt{s_{yx}^2} = \sqrt{52.54} = 7.25$$

Jika varians residual dan kesalahan standar pendugaan kecil, garis regresi cocok dengan data di plot pencar. Jika varians residual dan kesalahan standar pendugaan besar, garis regresi tidak cocok. Apa itu "kecil"? Apa itu "besar"? Apa yang "baik" cocok? Baca terus.

Menguji hipotesis tentang regresi

Persamaan regresi yang Anda kerjakan:

$$y' = a + bx$$

Merangkum hubungan dalam plot pencar sampel. Koefisien regresi a dan b adalah statistik sampel. Anda dapat menggunakan statistik ini untuk menguji hipotesis tentang parameter populasi, dan itulah yang Anda lakukan di bagian ini.

Garis regresi melalui populasi yang menghasilkan sampel (seperti seluruh rangkaian konsultan FarMisht) adalah grafik persamaan yang terdiri dari parameter dan bukan statistik. Dengan konvensi, ingat, huruf Yunani berarti parameter, jadi persamaan regresi untuk populasi adalah:

$$y' = \alpha + \beta x + \varepsilon$$

Dua huruf Yunani pertama di sebelah kanan adalah (alfa) dan (beta), setara dengan a dan b. Bagaimana dengan yang terakhir itu? Itu terlihat seperti padanan bahasa Yunani dari e. Apa yang dilakukannya di sana?

Istilah terakhir itu adalah huruf Yunani epsilon. Ini mewakili "kesalahan" dalam populasi. Di satu sisi, kesalahan adalah istilah yang tidak menguntungkan. Ini adalah jawaban untuk "hal-hal yang tidak Anda ketahui atau hal-hal yang tidak dapat Anda kendalikan." Kesalahan tercermin dalam residual — penyimpangan dari prediksi. Semakin Anda memahami tentang apa yang Anda ukur, semakin Anda mengurangi kesalahan.

Anda tidak dapat mengukur kesalahan dalam hubungan antara Aptitude dan Kinerja, tetapi kesalahan itu tersembunyi di sana. Seseorang mungkin mendapat nilai rendah pada Aptitude, misalnya, dan kemudian melanjutkan karir konsultasi yang luar biasa dengan Kinerja yang lebih tinggi dari yang diperkirakan. Pada plot pencar, titik Aptitude-Performance orang ini terlihat seperti kesalahan dalam prediksi. Saat Anda mengetahui lebih banyak tentang orang itu, Anda mungkin menemukan bahwa dia sakit pada hari Aptitude, dan itu menjelaskan "kesalahan". Anda dapat menguji hipotesis tentang , , dan , dan itulah yang Anda lakukan di subbagian mendatang.

Menguji kecocokan

Anda mulai dengan tes seberapa baik garis regresi cocok dengan plot pencar. Ini adalah sebuah uji , kesalahan dalam hubungan. Tujuannya adalah untuk memutuskan apakah garis benar-benar mewakili hubungan antara variabel atau tidak. Mungkin saja apa yang tampak seperti sebuah hubungan hanya karena kebetulan dan persamaan garis regresi tidak berarti apa-apa (karena jumlah kesalahannya sangat banyak) — atau mungkin saja variabel-variabelnya sangat terkait.

Kemungkinan ini dapat diuji, dan Anda menyiapkan hipotesis untuk mengujinya:

H_0: Tidak ada hubungan nyata

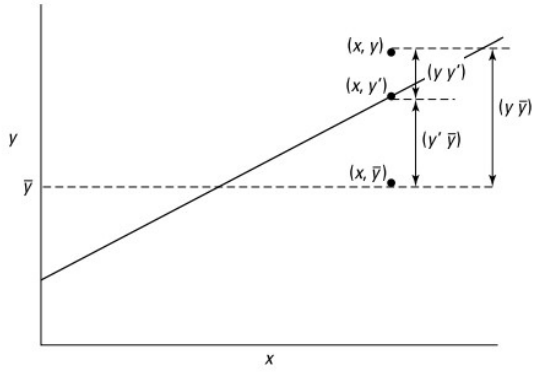
H_1: Tidak H_0

Meskipun hipotesis tersebut membuat bacaan ringan yang bagus, mereka tidak menyiapkan uji statistik. Untuk mengatur tes, Anda harus mempertimbangkan varians. Untuk mempertimbangkan varians, Anda mulai dengan deviasi. Gambar 14.3 berfokus pada satu titik dalam sebar plot dan penyimpangannya dari garis regresi (sisa) dan dari rata-rata variabel y. Ini juga menunjukkan penyimpangan antara garis regresi dan rata-rata.

Seperti yang ditunjukkan gambar, jarak antara titik dan garis regresi dan jarak antara garis regresi dan rata-rata dijumlahkan dengan jarak antara titik dan rata-rata:

$$(y-y')+(y'-\bar{y})=(y-\bar{y})$$

Ini menetapkan panggung untuk beberapa hubungan penting lainnya.



Gambar 14.3 Penyimpangan dalam plot pencar.

Mulailah dengan mengkuadratkan setiap simpangan. Itu memberi Anda $(y-y')^2$, $(y'-\bar{y})^2$, dan $(y-\bar{y})^2$.

Jika Anda menjumlahkan setiap kuadrat deviasi, Anda memiliki

$$\sum(y-y')^2$$

Anda baru saja melihat yang ini. Itu pembilang untuk varians residual. Ini mewakili variabilitas di sekitar garis regresi — "kesalahan" yang saya sebutkan sebelumnya. Dalam terminologi Bab 12, pembilang varians disebut jumlah kuadrat, atau SS. Jadi ini adalah $SS_{Residual}$.

$$\sum(y'-\bar{y})^2$$

Yang ini baru. Deviasi $(y'-\bar{y})$ mewakili keuntungan dalam prediksi karena menggunakan garis regresi daripada mean. Jumlah tersebut mencerminkan keuntungan ini dan disebut $SS_{Regression}$.

$$\sum(y-\bar{y})^2$$

Saya tunjukkan yang ini di Bab 5 — meskipun saya menggunakan x daripada y. Itulah pembilang varians y. Dalam istilah Bab 12, ini adalah pembilang total varians. Yang ini SSTotal. Hubungan ini berlaku di antara tiga jumlah ini:

$$SS_{Residual} + SS_{Regression} = SS_{Total}$$

Masing-masing dikaitkan dengan nilai derajat kebebasan — penyebut dari estimasi varians. Seperti yang saya tunjukkan di bagian sebelumnya, penyebut untuk $SS_{Residual}$ adalah N-2. Df untuk SS_{Total} adalah N-1. (Lihat Bab 5 dan 12.) Seperti halnya SS, derajat kebebasan bertambah:

$$df_{Residual} + df_{Regression} = df_{Total}$$

Ini menyisakan satu derajat kebebasan untuk Regresi.

Ke mana arah semua ini, dan apa hubungannya dengan pengujian hipotesis? Nah, karena Anda bertanya, Anda mendapatkan estimasi varians dengan membagi SS dengan df. Setiap estimasi varians disebut mean-square, disingkat MS (sekali lagi, lihat Bab 12):

$$MS_{Regression} = \frac{SS_{Regression}}{df_{Regression}}$$

$$MS_{Residual} = \frac{SS_{Residual}}{df_{Residual}}$$

$$MS_{Total} = \frac{SS_{Total}}{df_{Total}}$$

Sekarang untuk bagian hipotesis. Jika H_0 benar dan apa yang tampak seperti hubungan antara x dan y sebenarnya bukan masalah besar, bagian yang mewakili keuntungan dalam prediksi karena garis regresi ($MS_{Regression}$) tidak boleh lebih besar dari variabilitas di sekitar regresi baris ($MS_{Residual}$). Jika H_0 tidak benar, dan keuntungan dalam prediksi cukup besar, maka $MS_{Regression}$ harus jauh lebih besar dari $MS_{Residual}$.

Jadi hipotesis sekarang ditetapkan sebagai

$H_0: \sigma^2_{Regression} \leq \sigma^2_{Residual}$

$H_1: \sigma^2_{Regression} > \sigma^2_{Residual}$

Ini adalah hipotesis yang bisa Anda uji. Bagaimana? Untuk menguji hipotesis tentang dua varians, Anda menggunakan uji F. (Lihat Bab 11.) Statistik uji di sini adalah

$$F = \frac{MS_{Regression}}{MS_{Residual}}$$

Untuk menunjukkan cara kerjanya, saya menerapkan rumus ke contoh FarMisht. $MS_{Residual}$ sama dengan syx2 dari bagian sebelumnya, dan nilainya adalah 18,61. $MS_{Regresi}$ adalah

$$MS_{Regression} = \frac{(59.64 - 70.63)^2 + (71.40 - 70.63)^2 + \ldots + (66.17 - 70.63)^2}{1} = 1164.1$$

Ini mengatur F:

$$F = \frac{MS_{Regression}}{MS_{Residual}} = \frac{1164.1}{52.55} = 22.15$$

Dengan 1 dan 14 df dan = 0,05, nilai kritis F adalah 4,60. (Gunakan qf() untuk memverifikasi.) F yang dihitung lebih besar dari F kritis, jadi keputusannya adalah menolak H0. Itu berarti garis regresi memberikan kecocokan yang baik dengan data dalam sampel.

Menguji kemiringan

Pertanyaan lain yang muncul dalam regresi linier adalah apakah kemiringan garis regresi berbeda nyata dengan nol. Jika tidak, mean sama baiknya dengan prediktor garis regresi.

Hipotesis untuk tes ini adalah:

$H_0: \beta \leq 0$

$H_1: \beta > 0$

Uji statistiknya adalah t, yang saya bahas dalam Bab 9, 10, dan 11 sehubungan dengan rata-rata. Uji t untuk kemiringannya adalah:

$$t = \frac{b - \beta}{s_b}$$

Dengan df = N–2. Penyebut memperkirakan kesalahan standar kemiringan. Istilah ini terdengar lebih rumit dari itu. Rumusnya adalah:

$$s_b = \frac{s_{yx}}{s_x\sqrt{(N-1)}}$$

dimana sx adalah simpangan baku dari variabel x. Untuk data dalam contoh,

$$s_b = \frac{s_{yx}}{s_x\sqrt{(N-1)}} = \frac{7.25}{(13.48)\sqrt{(16-1)}} = .139$$

$$t = \frac{b - \beta}{s_b} = \frac{.654 - 0}{.139} = 4.71$$

Ini lebih besar dari nilai kritis t untuk 14 df dan α = 0,05 (2,14), sehingga keputusannya adalah menolak H_0.

Menguji intersep

Akhirnya, inilah uji hipotesis untuk intersep. Hipotesisnya adalah:

$H_0: \alpha = 0$

$H_1: \alpha \neq 0$

Tes, sekali lagi, adalah t-test. Rumusnya adalah

$$t = \frac{a - \alpha}{s_a}$$

Penyebut adalah perkiraan kesalahan standar intersep. Tanpa membahas detailnya, rumus untuk s_a adalah

$$s_a = s_{yx}\sqrt{\left[\frac{1}{N} + \frac{\bar{x}^2}{(N-1)s_x^2}\right]}$$

dimana s_x adalah simpangan baku dari variabel x, s_x^2 adalah variansi dari variabel x, dan $\bar{x}^2$adalah rata-rata kuadrat dari variabel x. Menerapkan rumus ini ke data dalam contoh,

$$s_a = s_{yx}\sqrt{\left[\frac{1}{N} + \frac{\bar{x}^2}{(N-1)s_x^2}\right]} = 10.27$$

uji-t adalah:

$$t = \frac{a - \alpha}{s_a} = \frac{23.03}{10.27} = 2.24$$

Dengan 15 derajat kebebasan, dan probabilitas kesalahan Tipe I pada 0,05, t kritis adalah 2,13 untuk uji dua sisi. Ini adalah tes dua sisi karena H_1 adalah bahwa intersep tidak sama dengan nol — itu tidak menentukan apakah intersep lebih besar dari nol atau kurang dari nol. Karena nilai yang dihitung lebih besar dari nilai kritis, keputusannya adalah menolak H_0.

14.4 REGRESI LINIER DI R

Saatnya melihat bagaimana R menangani regresi linier. Untuk memulai analisis untuk contoh ini, buat sebuah vektor untuk skor Aptitude dan satu lagi untuk skor Performa:

```
Aptitude <- c(45, 81, 65, 87, 68, 91, 77, 61, 55, 66, 82, 93,
          76, 83, 61, 74)
Performance <- c(56, 74, 56, 81, 75, 84, 68, 52, 57, 82, 73, 90,
          67, 79, 70, 66)
```

Kemudian gunakan dua vektor untuk membuat bingkai data

```
FarMisht.frame <- data.frame(Aptitude,Performance)
```

Fungsi lm() (model linier) melakukan analisis:

```
FM.reg <-lm(Performance ~ Aptitude, data=FarMisht.frame)
```

Seperti biasa, operator tilde (~) menandakan "tergantung pada," jadi ini adalah contoh sempurna dari variabel dependen dan variabel independen.
Menerapkan ringkasan() ke FM.reg menghasilkan informasi regresi:

```
> summary(FM.reg)

Call:
lm(formula = Performance ~ Aptitude, data = FarMisht.frame)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9036  -5.3720  -0.4379   4.2111  15.8281

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  23.0299    10.2732   2.242 0.041697 *
Aptitude      0.6537     0.1389   4.707 0.000337 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.249 on 14 degrees of freedom
Multiple R-squared:  0.6128,   Adjusted R-squared:  0.5851
F-statistic: 22.15 on 1 and 14 DF,  p-value: 0.0003368
```

Beberapa baris pertama memberikan informasi ringkasan tentang residu. Tabel koefisien menunjukkan intersep dan kemiringan garis regresi. Jika Anda membagi setiap angka di kolom Perkirakan dengan angka yang bersebelahan di Std. Kolom kesalahan, Anda mendapatkan nomor di kolom nilai t. Nilai-t ini, tentu saja, adalah uji signifikansi yang saya sebutkan sebelumnya untuk intersep dan kemiringan. Nilai p yang sangat rendah menunjukkan penolakan hipotesis nol (bahwa koefisien = 0) untuk setiap koefisien.

Bagian bawah output menunjukkan info tentang seberapa cocok garis tersebut dengan plot pencar. Ini menyajikan kesalahan standar residual, diikuti oleh Multiple R-squared dan Adjusted R-squared. Dua yang terakhir ini berkisar dari 0 hingga 1,00 (semakin tinggi nilainya, semakin baik kecocokannya). Saya membahasnya di Bab 15, tetapi untuk saat ini saya akan membiarkannya sendiri. Statistik-F sesuai dengan rasio-F yang saya tunjukkan sebelumnya. Nilainya yang tinggi dan nilai p terkait yang rendah menunjukkan bahwa garis tersebut sangat cocok dengan plot pencar. Saya menyebut hasil analisis regresi linier sebagai "model linier".

Fitur model linier

Model linier yang dihasilkan oleh lm() adalah objek yang memberikan informasi, jika Anda memintanya dengan cara yang benar. Seperti yang sudah saya tunjukkan, menerapkan ringkasan() memberikan semua informasi yang Anda butuhkan tentang analisis.
Anda juga dapat membidik pada koefisien:

```
> coefficients(FM.reg)
(Intercept)    Aptitude
  23.029869    0.653667
```

dan pada interval kepercayaan mereka:

```
> confint(FM.reg)
                2.5 %     97.5 %
(Intercept) 0.9961369 45.0636002
Aptitude    0.3558034  0.9515307
```

Menerapkan pas(FM.reg) menghasilkan nilai pas, dan residu(FM.reg) memberikan residu.

Membuat prediksi

Nilai regresi linier adalah memberikan Anda kemampuan untuk memprediksi, dan R menyediakan fungsi yang melakukan hal itu: predict() menerapkan sekumpulan nilai-x ke model linier dan mengembalikan nilai prediksi. Bayangkan dua pelamar dengan skor Aptitude 85 dan 62:

```
predict(FM.reg,data.frame(Aptitude=c(85,62)))
```

Argumen pertama adalah model linier, dan argumen kedua membuat kerangka data dari vektor nilai untuk variabel independen. Menjalankan fungsi ini menghasilkan nilai prediksi ini:

```
       1        2
78.59157 63.55723
```

Memvisualisasikan scatter plot dan garis regresi

Dengan paket ggplot2, Anda dapat memvisualisasikan sebar plot dan garis regresinya dalam tiga pernyataan. Pernyataan pertama, seperti biasa, menunjukkan sumber data dan memetakan komponen data ke komponen plot:

```
ggplot(FarMisht.frame,aes(x=Aptitude,y=Performance))
```

Pernyataan kedua memplot poin dalam grafik

```
geom_point()
```

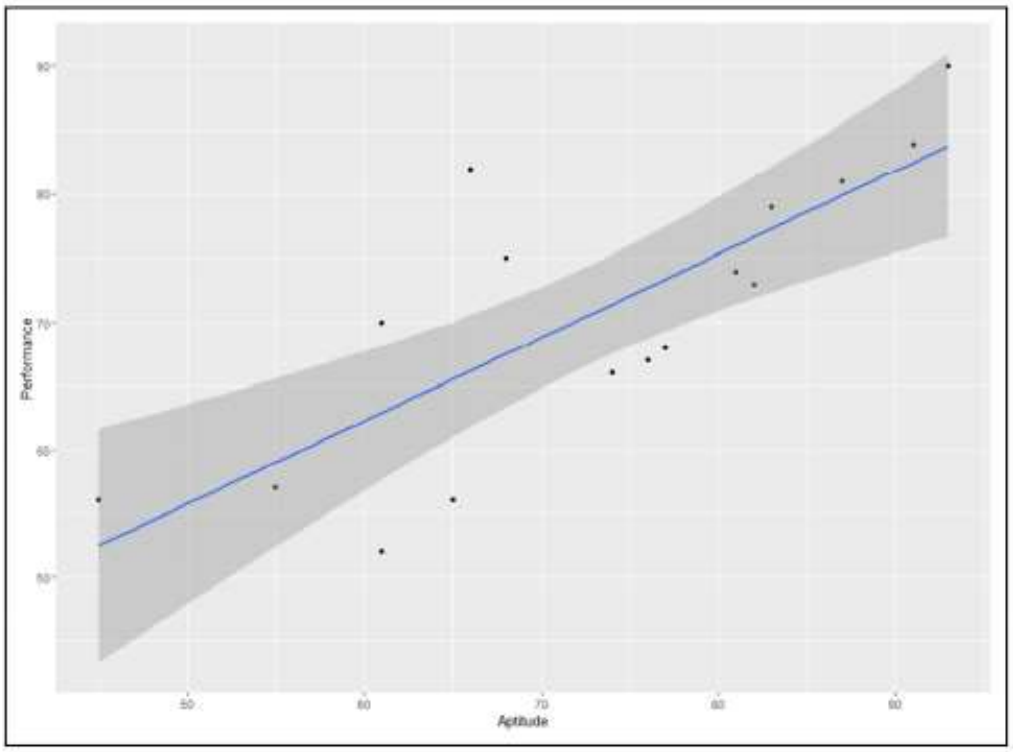
dan yang ketiga menentukan fungsi geom yang menambahkan garis regresi (seperti yang ditunjukkan oleh argumen metode = lm):

```
geom_smooth(method=lm)
```

Menyatukan ketiganya

```
ggplot(FarMisht.frame,aes(x=Aptitude,y=Performance)) +
  geom_point()+
  geom_smooth(method=lm)
```

menghasilkan Gambar 14.4.



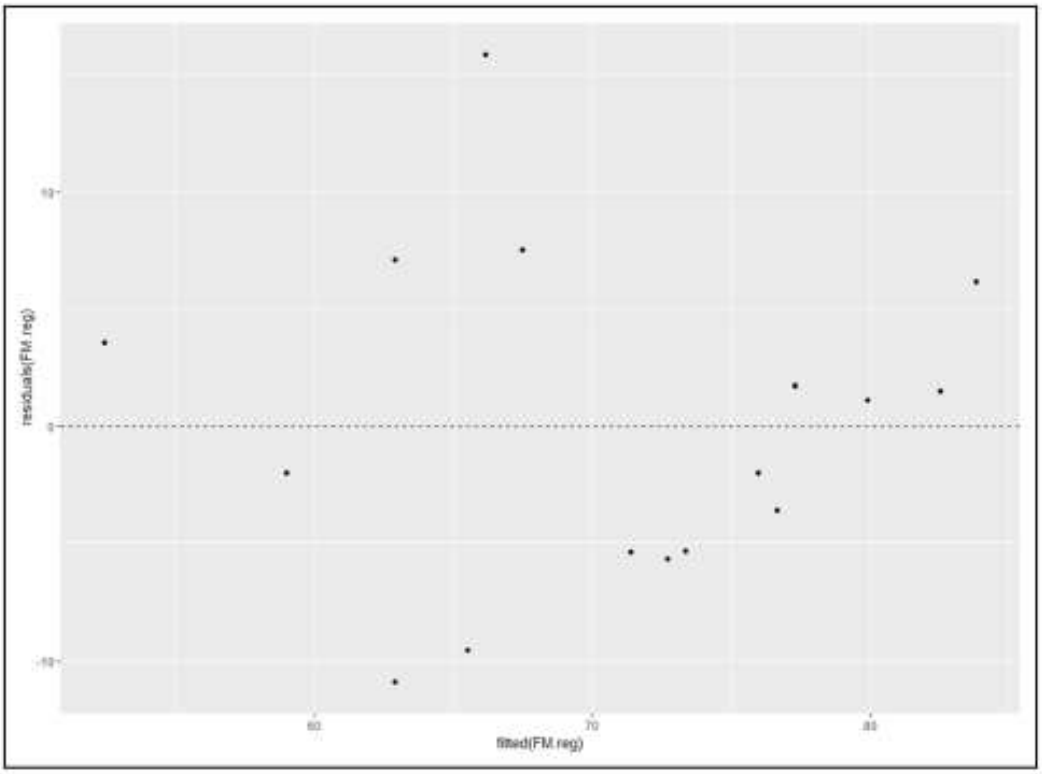
Gambar 14.4 Plot sebar dan garis regresi untuk 16 konsultan FarMisht.

Area yang diarsir mewakili interval kepercayaan 95 persen di sekitar garis regresi.

Merencanakan residu

Setelah analisis regresi, ada baiknya untuk memplot residual terhadap nilai yang diprediksi. Jika residual membentuk pola acak di sekitar garis horizontal di nol, itu bukti yang

mendukung hubungan linier antara variabel independen dan variabel dependen. Gambar 14-5 menunjukkan plot residual untuk contoh. Pola residual di sekitar garis konsisten dengan model linier.



Gambar 14.5 Plot sisa untuk contoh FarMisht.

Plot didasarkan pada FM.reg, model linier. Berikut pernyataan ggplot():

```
ggplot(FM.reg, aes(x=fitted(FM.reg), y=residuals(FM.reg)))
```

Pemetaan x dan y didasarkan pada informasi dari analisis. Seperti yang Anda duga, fitting(FM.reg) mengambil nilai prediksi, dan residuals(FM.reg) mengambil residual.
Untuk memplot titik, tambahkan fungsi geom yang sesuai:

```
geom_point()
```

dan kemudian fungsi geom untuk garis horizontal putus-putus yang perpotongan y-nya adalah 0:

```
geom_hline(yintercept = 0, linetype = "dashed" )
```

Jadi kode untuk Gambar 14.5 adalah:

```
ggplot(FM.reg, aes(x=fitted(FM.reg), y=residuals(FM.reg)))+
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed" )
```

Menyulap Banyak Hubungan Sekaligus: Regresi Berganda

Regresi linier adalah alat yang hebat untuk membuat prediksi. Ketika Anda mengetahui kemiringan dan titik potong garis yang menghubungkan dua variabel, Anda dapat mengambil nilai x baru dan memprediksi nilai y baru. Dalam contoh yang telah Anda kerjakan dalam bab ini, Anda mengambil skor Aptitude dan memprediksi skor Performa untuk pelamar FarMisht.

Bagaimana jika Anda tahu lebih dari sekadar skor Aptitude untuk setiap pelamar? Misalnya, bayangkan tim manajemen FarMisht memutuskan bahwa tipe kepribadian tertentu ideal untuk konsultan mereka. Jadi mereka mengembangkan Inventarisasi Kepribadian FarMisht, skala 20 poin di mana skor yang lebih tinggi menunjukkan kompatibilitas yang lebih besar dengan budaya perusahaan FarMisht dan, mungkin, memprediksi kinerja yang lebih baik. Idenya adalah untuk menggunakan data itu bersama dengan skor Aptitude untuk memprediksi kinerja.

Tabel 14.4 menunjukkan skor Aptitude, Performance, dan Personality untuk 16 konsultan saat ini. Tentu saja, dalam perusahaan kehidupan nyata, Anda mungkin memiliki lebih banyak karyawan sebagai sampel.

Tabel 14.4 Skor Bakat, Kinerja, dan Kepribadian untuk 16 Konsultan FarMisht

Konsultan	Bakat	Pertunjukan	Kepribadian
1	45	56	9
2	81	74	15
3	65	56	11
4	87	81	15
5	68	75	14
6	91	84	19
7	77	68	12
8	61	52	10
9	55	57	9
10	66	82	14
11	82	73	15
12	93	90	14
13	76	67	16
14	83	79	18
15	61	70	15
16	74	66	12
Mean	72.81	70.63	13.63

Variance	181.63	126.65	8.65
Standar Deviasi	13.48	11.25	2.94

Saat Anda bekerja dengan lebih dari satu variabel independen, Anda berada di ranah regresi berganda. Seperti dalam regresi linier, Anda menemukan koefisien regresi. Dalam kasus dua variabel independen, Anda mencari bidang yang paling pas melalui plot sebar tiga dimensi. Sekali lagi, "paling pas" berarti jumlah kuadrat jarak dari titik data ke bidang adalah minimum.

Berikut persamaan bidang regresi:

$$\text{standard residual} = \frac{\text{residual - average residual}}{s_{yx}}$$

Untuk contoh ini, yang diterjemahkan menjadi

$$y' = a + b_1 x_1 + b_2 x_2$$

Anda dapat menguji hipotesis tentang kecocokan keseluruhan, dan tentang ketiga koefisien regresi.

Saya tidak memandu Anda melalui semua rumus untuk menemukan koefisien, karena itu menjadi sangat rumit. Sebagai gantinya, saya langsung ke analisis R. Berikut adalah beberapa hal yang perlu diingat sebelum saya melanjutkan:

- Anda dapat memiliki sejumlah variabel x. (Saya menggunakan dua dalam contoh ini.)
- Harapkan koefisien Aptitude untuk berubah dari regresi linier ke regresi berganda. Harapkan intersep juga berubah.
- Mengharapkan kesalahan standar estimasi menurun dari regresi linier ke regresi berganda. Karena regresi berganda menggunakan lebih banyak informasi daripada regresi linier, ini mengurangi kesalahan.

14.5 REGRESI BERGANDA DALAM R

Saya mulai dengan menambahkan vektor untuk skor kepribadian di Kolom 4 Tabel 14.4:

```
Personality <- c(9, 15, 11, 15, 14, 19, 12, 10, 9, 14, 15, 14,
          16, 18, 15, 12)
```

Dan kemudian saya menambahkan vektor itu ke bingkai data:

```
FarMisht.frame["Personality"] = Personality
```

Menerapkan lm() menghasilkan analisis:

```
FM.multreg <- lm(Performance ~ Aptitude + Personality,
          data = FarMisht.frame)
```

Dan menerapkan ringkasan() memberikan informasi:

```
> summary(FM.multreg)

Call:
lm(formula = Performance ~ Aptitude + Personality, data
          = FarMisht.frame)

Residuals:
   Min     1Q Median     3Q    Max
-8.689 -2.834 -1.840  2.886 13.432

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.2825     9.6595   2.100   0.0558 .
Aptitude      0.3905     0.1949   2.003   0.0664 .
Personality   1.6079     0.8932   1.800   0.0951 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.73 on 13 degrees of freedom
Multiple R-squared:   0.69,    Adjusted R-squared:  0.6423
F-statistic: 14.47 on 2 and 13 DF,  p-value: 0.0004938
```

Jadi persamaan umum untuk bidang regresi adalah:

$$Predicted\ GPA = a + b_1(SAT) + b_2(High\ School\ Average)$$

Atau, dalam hal contoh ini

$$y' = a + .0025x_1 + .043x_2$$

Sekali lagi, nilai F yang tinggi dan nilai p yang rendah menunjukkan bahwa bidang regresi sangat cocok untuk plot pencar.

Membuat prediksi

Sekali lagi, predict() mengaktifkan prediksi Kinerja. Kali ini saya menggunakan model regresi berganda: FM.multreg. Bayangkan dua pelamar: Satu memiliki skor Aptitude dan Personality 85 dan 14, dan yang lainnya memiliki skor Aptitude dan Personality 62 dan 17. Ini membutuhkan dua vektor — satu untuk skor Aptitude dan satu untuk skor Personality:

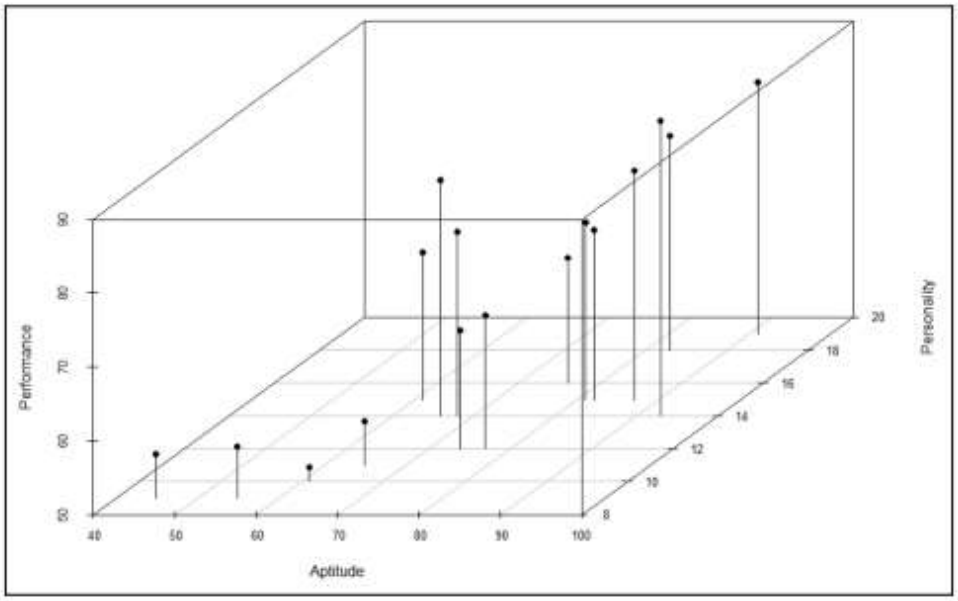
```
> predict(FM.multreg, data.frame(Aptitude = c(85,62),
          Personality=c(14,17)))
       1        2
75.98742 71.82924
```

Memvisualisasikan plot sebar 3D dan bidang regresi

Paket ggplot2, untuk semua fiturnya yang luar biasa, tidak menyediakan cara untuk menggambar grafik 3 dimensi — seperti plot pencar untuk variabel dependen dan dua variabel independen. Namun, jangan pernah takut: R memiliki sejumlah cara lain untuk melakukan ini. Di bagian ini, saya tunjukkan dua di antaranya.

Paket scatterplot3d

Jika Anda ingin membuat plot sebar tiga dimensi yang bagus seperti yang ditunjukkan pada Gambar 14.6 — sosok yang terlihat bagus pada halaman yang dicetak, fungsi scatterplot3d() adalah untuk Anda.



Gambar 14.6 Scatter plot untuk contoh regresi berganda FarMisht, direnderdalam scatter plot3d().

Pertama, instal paket scatterplot3d. Pada tab Packages, temukan scatter-plot3d dan pilih kotak centangnya.

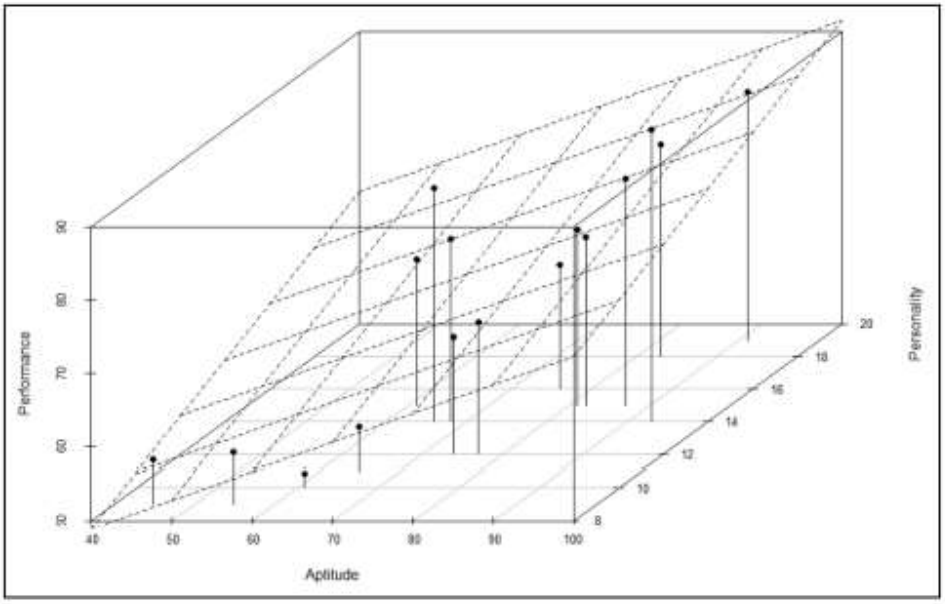
Selanjutnya, tulis pernyataan yang membuat plot:

```
with (FarMisht.frame,
(splot <- scatterplot3d(Performance ~ Aptitude +
          Personality, type = "h", pch = 19)))
```

Jika Anda menggunakan dengan Anda tidak perlu mengulang nama frame data tiga kali. Argumen pertama untuk scatterplot3d() adalah rumus untuk menyiapkan model linier. Argumen kedua menambahkan garis vertikal dari bidang xy ke titik data. Garis-garis vertikal itu tidak mutlak diperlukan, tetapi saya pikir garis-garis itu membantu pemirsa memahami di mana titik-titik dalam plot. Argumen ketiga menentukan seperti apa karakter plot.

Fungsi ini menghasilkan objek yang dapat Anda gunakan untuk memperindah plot. Untuk ujian-
ple, inilah cara menambahkan bidang regresi dan menghasilkan Gambar 14.7:

```
splot$plane3d(FM.multreg,lty="dashed")
```



Gambar 14.7 Scatter plot untuk contoh regresi berganda FarMisht, lengkap dengan bidang regresi.

Mobil dan rgl: Kesepakatan paket

Jika Anda harus menyajikan plot sebar 3D kepada audiens dan Anda ingin mempesona mereka dengan plot interaktif, metode selanjutnya adalah untuk Anda. Fungsi pembuatan plot disebut scatter3d(), dan ia tinggal di dalam paket mobil. Pada tab Paket, klik Instal. Di kotak dialog Instal Paket, ketik mobil dan klik Instal. Saat mobil muncul di tab Paket, pilih kotak centangnya.

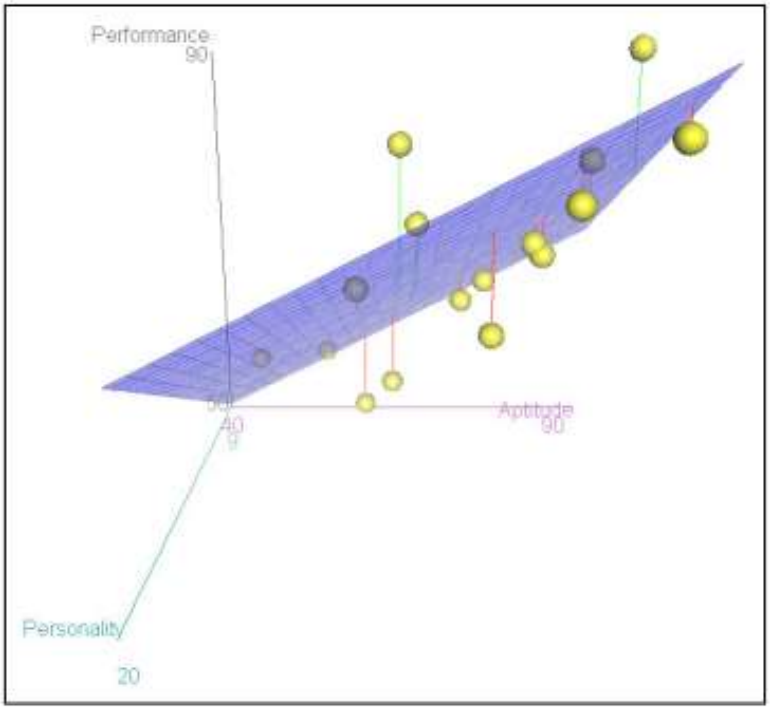
Fungsi ini bekerja dengan paket rgl, yang menggunakan alat dari Open Graphics Library (OpenGL), seperangkat alat untuk membuat grafik 2D dan 3D. Anda akan menemukan alat OpenGL bekerja dalam realitas virtual, desain berbantuan komputer, simulasi penerbangan, dan sejumlah aplikasi lainnya.

Pada tab Paket, temukan rgl dan pilih kotak centangnya. Dengan dua paket yang diinstal, jalankan fungsi ini:

```
scatter3d(Performance ~ Aptitude + Personality,
          data=FarMisht.frame)
```

Ini membuka jendela RGL dengan plot sebar 3D yang ditunjukkan pada Gambar 14.8. Seperti yang Anda lihat, plot pencar menunjukkan bidang regresi dan residu. Anda dapat

menggerakkan mouse di dalam plot ini, menekan tombol kiri mouse, dan memutar plot untuk menyajikan sudut yang berbeda. Anda juga dapat menggunakan roda gulir untuk memperbesar atau memperkecil plot. Cobalah!



Gambar 14.8 Plot pencar untuk contoh regresi berganda FarMisht, dirender dalam scatter3d().

14.6 ANOVA: TAMPILAN LAIN

Berikut pernyataan yang mungkin Anda anggap radikal: Analisis varians dan regresi linier sebenarnya adalah hal yang sama. Keduanya adalah bagian dari apa yang disebut General Linear Model (GLM). Dalam regresi linier, tujuannya adalah untuk memprediksi nilai variabel dependen yang diberikan nilai variabel independen. Dalam ANOVA, tujuannya adalah untuk memutuskan apakah beberapa sampel berarti cukup berbeda satu sama lain untuk memungkinkan Anda menolak hipotesis nol tentang tingkat variabel independen.

Bagaimana mereka mirip? Lebih mudah untuk melihat hubungannya jika Anda memikirkan kembali ANOVA: Mengingat datanya, bayangkan bahwa tujuannya adalah untuk memprediksi variabel dependen dengan mempertimbangkan tingkat variabel independen. Apa yang akan menjadi prediksi terbaik? Untuk setiap tingkat variabel independen, itu akan menjadi rata-rata sampel untuk tingkat itu — juga dikenal sebagai "rata-rata kelompok." Ini berarti bahwa penyimpangan dari mean grup (nilai prediksi terbaik) adalah residual, dan inilah mengapa, dalam R ANOVA, MSError disebut MSResiduals. Ini lebih dalam dari itu. Untuk menunjukkan caranya, saya meninjau kembali contoh ANOVA dari Bab 12. Untuk kenyamanan, inilah Tabel 12.1 yang direproduksi sebagai Tabel 14.5.

Tabel 14.5 Data dari Tiga Metode Pelatihan (Contoh ANOVA dari Bab 12)

	Method 1	Method 2	Method 3
	95	83	68
	91	89	75
	89	85	79
	90	89	74
	99	81	75
	88	89	81
	96	90	73
	98	82	77
	95	84	
		80	
Mean	93.44	85.20	75.25
Variance	16.28	14.18	15.64
Standar Deviasi	4.03	3.77	3.96

Anda harus menguji

$H_0: \mu_1 = \mu_2 = \mu_3$

H_1: Not H_0

Untuk menggunakan fungsi aov() untuk menghasilkan analisis varians, siapkan data dalam format panjang. Berikut adalah enam baris pertama:

```
> head(Training.frame)
   Method Score
1 method1    95
2 method1    91
3 method1    89
4 method1    90
5 method1    99
6 method1    88
```

Hasil analisisnya adalah:

```
> analysis <-aov(Score~Method,data = Training.frame)
> summary(analysis)
            Df Sum Sq Mean Sq F value   Pr(>F)
Method       2 1402.7   701.3   45.82 6.38e-09 ***
Residuals   24  367.3    15.3
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bagaimana jika Anda mencoba analisis regresi linier pada data?

```
> reg.analysis <-lm(Score~Method,data = Training.frame)
> summary(reg.analysis)

Call:
lm(formula = Score ~ Method, data = Training.frame)

Residuals:
   Min     1Q Median     3Q    Max
-7.250 -2.822 -0.250  3.775  5.750

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     93.444      1.304  71.657  < 2e-16 ***
Methodmethod2   -8.244      1.798  -4.587 0.000119 ***
Methodmethod3  -18.194      1.901  -9.571 1.15e-09 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.912 on 24 degrees of freedom
Multiple R-squared:  0.7925,   Adjusted R-squared:  0.7752
F-statistic: 45.82 on 2 and 24 DF,  p-value: 6.381e-09
```

Anda melihat sedikit lebih banyak informasi daripada di tabel ANOVA, tetapi intinya menunjukkan rasio-F yang sama dan informasi terkait sebagai analisis varians. Juga, koefisien memberikan mean grup: Intersep (93,444) adalah rata-rata Metode 1, intersep ditambah koefisien kedua (–8,244) adalah rata-rata Metode 2 (85,20), dan intersep ditambah koefisien ketiga (– 18.194) adalah rata-rata Metode 3 (75.25). Periksa Cara di Tabel 14-1, jika Anda tidak percaya.

Sedikit lebih banyak tentang koefisien: Intercept mewakili Metode 1, yang merupakan garis dasar untuk membandingkan satu sama lain. Nilai t untuk Metode 2 (bersama dengan probabilitas yang terkait, yang jauh lebih kecil dari 0,05) menunjukkan bahwa Metode 2 berbeda secara signifikan dari Metode 1. Ini adalah cerita yang sama untuk Metode 3, yang juga berbeda secara signifikan dari Metode 1.

Inilah pertanyaan yang harus terbentuk di benak Anda: Bagaimana Anda bisa melakukan regresi linier ketika variabel independen (Metode) adalah kategorikal daripada numerik? Senang Anda bertanya. Untuk membentuk analisis regresi dengan data kategorikal, R (dan paket perangkat lunak statistik lainnya) mengkode ulang level variabel seperti Metode ke dalam kombinasi variabel dummy numerik. Satu-satunya nilai yang dapat diambil oleh variabel dummy adalah 0 atau 1: 0 menunjukkan tidak adanya nilai kategoris; 1 menunjukkan adanya nilai kategoris.

Saya akan melakukan ini secara manual. Untuk tiga level Metode (Metode 1, Metode 2, dan Metode 3), saya memerlukan dua variabel dummy. Saya akan menyebutnya D1 dan D2. Inilah cara saya (secara sewenang-wenang) menetapkan nilai:

- Untuk Metode 1, D1 = 0 dan D2 = 0
- Untuk Metode 2, D1 = 1, dan D2 = 0
- Untuk Metode 3, D1 = 0, dan D2 = 1

Untuk mengilustrasikan lebih lanjut, inilah kerangka data yang disebut Training.frame.w.Dummies. Biasanya, saya tidak akan menunjukkan kepada Anda semua 27 baris bingkai data, tetapi di sini saya pikir itu instruktif:

```
> Training.frame.w.Dummies
    Method D1 D2 Score
1  method1  0  0    95
2  method1  0  0    91
3  method1  0  0    89
4  method1  0  0    90
5  method1  0  0    99
6  method1  0  0    88
7  method1  0  0    96
8  method1  0  0    98
9  method1  0  0    95
10 method2  1  0    83
11 method2  1  0    89
12 method2  1  0    85
13 method2  1  0    89
14 method2  1  0    81
15 method2  1  0    89
16 method2  1  0    90
17 method2  1  0    82
18 method2  1  0    84
19 method2  1  0    80
20 method3  0  1    68
21 method3  0  1    75
22 method3  0  1    79
23 method3  0  1    74
24 method3  0  1    75
25 method3  0  1    81
26 method3  0  1    73
27 method3  0  1    77
```

Baris kode ini

```
model.w.Dummies <- lm(Score ~ D1 + D2,
                      data= Training.frame.w.Dummies)
summary(model.w.Dummies)
```

Hasilkan hasil yang sama seperti analisis varians dan regresi linier yang saya tunjukkan sebelumnya. Satu-satunya perbedaan adalah bahwa koefisien dinyatakan dalam variabel dummy:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   93.444      1.304  71.657  < 2e-16 ***
D1            -8.244      1.798  -4.587 0.000119 ***
D2           -18.194      1.901  -9.571 1.15e-09 ***
```

Jadi, variabel dummy memungkinkan model regresi linier dengan variabel bebas kategoris. Faktanya, regresi linier dengan variabel bebas kategoris adalah analisis varians.

14.7 ANALISIS KOVARIANS: KOMPONEN AKHIR GLM

Dalam bab ini, saya telah menunjukkan kepada Anda bagaimana regresi linier bekerja dengan variabel independen numerik (prediktor), dan dengan variabel independen kategoris (prediktor). Apakah mungkin untuk mengadakan penelitian dengan variabel prediktor numerik dan variabel prediktor kategoris? Sangat! Alat analisis untuk jenis studi ini disebut Analisis Kovarians (ANCOVA). Ini adalah komponen ketiga dan terakhir dari Model Linier Umum. (Regresi linier dan ANOVA adalah dua yang pertama.) Cara termudah untuk menggambarkannya adalah dengan sebuah contoh.

Pastikan Anda telah menginstal paket MASS. Pada tab Paket, temukan kotak centangnya dan pilih, jika belum. Dalam paket MASS terdapat kerangka data yang disebut anoreksia. (Saya menggunakannya di Bab 2.) Kerangka data ini berisi data untuk 72 wanita muda yang dipilih secara acak untuk salah satu dari tiga jenis pengobatan untuk anoreksia: Lanjutan (kondisi kontrol tanpa terapi), CBT (terapi perilaku kognitif), atau FT (pengobatan keluarga).

Berikut adalah enam baris pertama:

```
> head(anorexia)
  Treat Prewt Postwt
1  Cont  80.7   80.2
2  Cont  89.4   80.1
3  Cont  91.8   86.4
4  Cont  74.0   86.3
5  Cont  78.1   76.1
6  Cont  88.3   78.1
```

Prewt adalah berat sebelum perawatan, dan Postwt adalah berat setelah perawatan. Yang Anda butuhkan, tentu saja, adalah variabel yang menunjukkan jumlah berat badan yang diperoleh selama perawatan. Saya akan menyebutnya WtGain, dan inilah cara menambahkannya ke bingkai data:

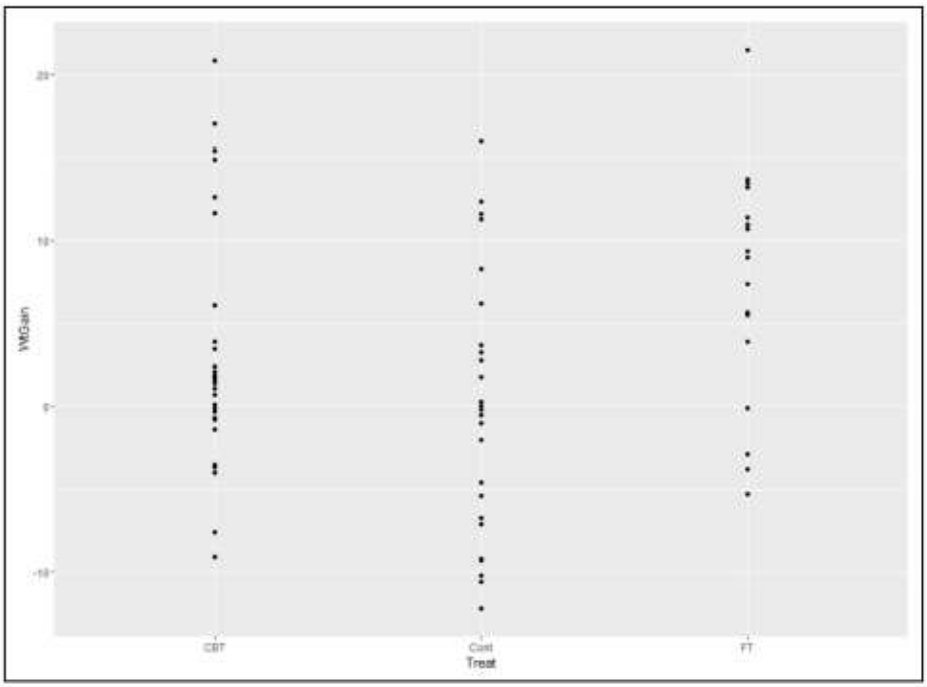
```
anorexia["WtGain"]=anorexia["Postwt"]-anorexia["Prewt"]
```

Sekarang:

```
> head(anorexia)
  Treat Prewt Postwt WtGain
1  Cont  80.7   80.2   -0.5
2  Cont  89.4   80.1   -9.3
3  Cont  91.8   86.4    5.4
4  Cont  74.0   86.3   12.3
5  Cont  78.1   76.1   -2.0
6  Cont  88.3   78.1  -10.2
```

Gambar 14.9 memplot titik data untuk kerangka data ini.
Berikut kode untuk plot ini, jika Anda penasaran:

```
ggplot(anorexia,aes(x=Treat,y=WtGain))+
  geom_point()
```



Gambar 14.9 Penambahan Berat Badan versus Perawatan dalam kerangka data anoreksia.

Analisis varians atau analisis regresi linier akan sesuai untuk menguji ini:

$$H_0: \mu_{Cont} = \mu_{CBT} = \mu_{FT}$$

$$H_1: \text{Not } H_0$$

Berikut model regresi liniernya:

```
> anorexia.linreg <-lm(WtGain ~ Treat, data=anorexia)
> summary(anorexia.linreg)

Call:
lm(formula = WtGain ~ Treat, data = anorexia)

Residuals:
    Min      1Q  Median      3Q     Max
-12.565  -4.543  -1.007   3.846  17.893

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.007      1.398   2.151   0.0350 *
TreatCont     -3.457      2.033  -1.700   0.0936 .
TreatFT        4.258      2.300   1.852   0.0684 .
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.528 on 69 degrees of freedom
Multiple R-squared:  0.1358,   Adjusted R-squared:  0.1108
F-statistic: 5.422 on 2 and 69 DF,  p-value: 0.006499
```

Rasio-F dan nilai-p pada intinya memberi tahu Anda bahwa Anda dapat menolak hipotesis nol. Mari kita lihat koefisiennya. Intersep mewakili CBT. Ini adalah dasar yang Anda bandingkan dengan perawatan lainnya. Nilai-t dan probabilitas terkait (lebih besar dari 0,05) memberi tahu Anda bahwa tak satu pun dari level tersebut berbeda dari CBT. Rasio-F yang signifikan harus dihasilkan dari beberapa perbandingan lain.

Juga, periksa koefisien terhadap sarana pengobatan. Berikut cara cepat dan mudah untuk menemukan cara perawatan: Gunakan fungsi tapply() untuk menerapkan mean() dan temukan mean WtGain di level Treat:

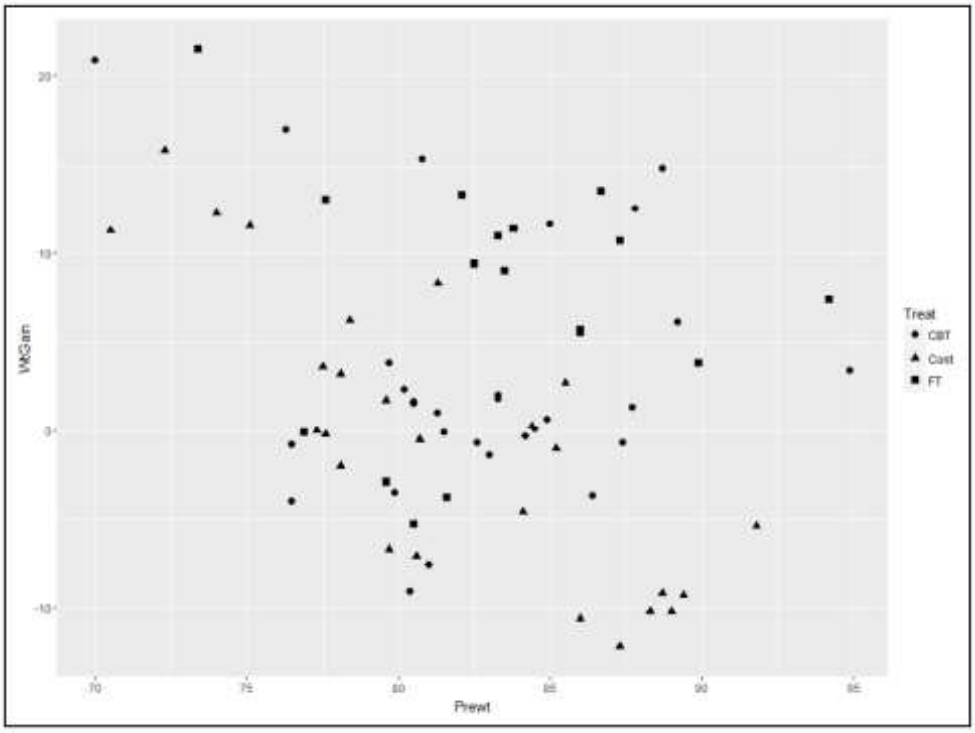
```
> with (anorexia, tapply(WtGain,Treat,mean))
      CBT      Cont        FT
 3.006897 -0.450000  7.264706
```

Intersep, ingat, adalah rata-rata untuk CBT. Tambahkan intersep ke koefisien berikutnya untuk menghitung mean untuk Cont, dan tambahkan intersep ke koefisien akhir untuk menghitung mean untuk FT.

Jika Anda lebih suka melihat rasio-F dan statistik terkait dalam tabel ANOVA, Anda dapat menerapkan fungsi anova() ke model:

```
> anova(anorexia.linreg)
Analysis of Variance Table

Response: WtGain
          Df Sum Sq Mean Sq F value   Pr(>F)
Treat      2  614.6 307.322  5.4223 0.006499 **
Residuals 69 3910.7  56.677
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anda bisa menggali lebih dalam. Misalkan penambahan berat badan tidak hanya bergantung pada jenis perawatan tetapi juga pada berat badan awal seseorang (yang disebut kovariat). Mempertimbangkan PreWt mungkin menghasilkan gambaran yang lebih akurat. Treat adalah variabel kategori, dan Prewt adalah variabel numerik. Gambar 14-10 menunjukkan plot berdasarkan dua variabel.



Gambar 14-.0 Berat Badan versus Perlakukan dan Prewt dalam kerangka data anoreksia.

Kode untuk plot ini adalah:

```
ggplot(anorexia, aes(x=Prewt,y=WtGain, shape = Treat)) +
  geom_point(size=2.5)
```

Pernyataan pertama memetakan Prewt ke sumbu x, WtGain ke sumbu y, dan Treat to shape. Dengan demikian, bentuk titik data mencerminkan kelompok perlakuannya. Pernyataan kedua menentukan bahwa poin muncul dalam plot. Argumen ukurannya memperbesar titik data dan membuatnya lebih mudah dilihat.

Untuk analisis kovarians, saya menggunakan fungsi lm() untuk membuat model berdasarkan Treat dan Prewt:

```
> anorexia.T.and.P <- lm(WtGain ~ Treat + Prewt, data=anorexia)
> summary(anorexia.T.and.P)

Call:
lm(formula = WtGain ~ Treat + Prewt, data = anorexia)
```

```
Call:
lm(formula = WtGain ~ Treat + Prewt, data = anorexia)

Residuals:
     Min       1Q   Median       3Q      Max
-14.1083  -4.2773  -0.5484   5.4838  15.2922

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.7711    13.3910   3.717 0.000410 ***
TreatCont    -4.0971     1.8935  -2.164 0.033999 *
TreatFT       4.5631     2.1333   2.139 0.036035 *
Prewt        -0.5655     0.1612  -3.509 0.000803 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.978 on 68 degrees of freedom
Multiple R-squared:  0.2683,   Adjusted R-squared:  0.236
F-statistic: 8.311 on 3 and 68 DF,  p-value: 8.725e-05
```

Perhatikan di baris terakhir bahwa derajat kebebasan telah berubah dari analisis pertama: Menambahkan Prewt mengambil derajat kebebasan dari df Residual dan menambahkannya ke df untuk Perlakukan. Perhatikan juga bahwa rasio-F lebih tinggi dan nilai-p jauh lebih rendah daripada analisis pertama.

Dan sekarang lihat koefisiennya. Berbeda dengan analisis asli, nilai-t dan probabilitas terkait (kurang dari 0,05) untuk Lanjutan dan FT menunjukkan bahwa masing-masing berbeda secara signifikan dari CBT. Jadi tampaknya menambahkan Prewt ke dalam analisis telah membantu mengungkap perbedaan pengobatan. Intinya: ANCOVA menunjukkan bahwa ketika mengevaluasi efek pengobatan anoreksia, penting juga untuk mengetahui berat badan sebelum perawatan seseorang.

Tetapi "tampaknya" tidak cukup untuk ahli statistik. Bisakah Anda benar-benar yakin bahwa ANCOVA menambah nilai? Untuk mengetahuinya, Anda harus membandingkan model regresi linier dengan model ANCOVA. Untuk membuat perbandingan, gunakan fungsi anova(), yang berfungsi ganda: Selain membuat tabel ANOVA untuk model (seperti yang Anda lihat sebelumnya), Anda dapat menggunakannya untuk membandingkan model. Begini caranya:

```
> anova(anorexia.linreg,anorexia.T.and.P)
Analysis of Variance Table

Model 1: WtGain ~ Treat
Model 2: WtGain ~ Treat + Prewt
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     69 3910.7
2     68 3311.3  1    599.48 12.311 0.0008034 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Apa arti angka-angka dalam tabel? RSS menunjukkan jumlah sisa kuadrat dari setiap model. Mereka berada di sebelah derajat kebebasan mereka di kolom Res.DF. Pada kolom Df, 1 adalah selisih antara kedua Res.Dfs. Pada kolom Sum of Sq, 599,48 adalah selisih antara kedua RSS. Rasio-F adalah hasil pembagian dua kuadrat rata-rata: Rata-rata kuadrat pembilangnya adalah 599,48 dibagi df (1), dan kuadrat rata-rata penyebutnya adalah 3311,3 dibagi dfnya (68). Rasio-F yang tinggi dan Pr(>F) yang rendah (probabilitas kesalahan Tipe 1) memberi tahu Anda bahwa menambahkan Prewt secara signifikan menurunkan jumlah kuadrat sisa. Dalam bahasa Inggris, itu berarti ada baiknya menambahkan Prewt ke dalam campuran. Ahli statistik akan mengatakan bahwa analisis ini secara statistik mengontrol efek kovariat (Prewt).

Dalam analisis kovarians, penting untuk menanyakan apakah hubungan antara variabel dependen dan variabel prediktor numerik adalah sama di seluruh level variabel kategori. Dalam contoh ini, itu sama dengan menanyakan apakah kemiringan garis regresi antara WtGain dan Prewt sama untuk skor di Cont seperti untuk skor di CBT dan untuk skor di FT. Jika kemiringannya sama, itu disebut homogenitas regresi. Jika tidak, Anda memiliki interaksi Prewt and Treat, dan Anda harus berhati-hati dalam menyatakan kesimpulan.

Menambahkan garis regresi ke plot pada Gambar 14.10 sangat membantu. Untuk melakukan ini, saya menambahkan baris ini ke kode yang menghasilkan Gambar 14.10:

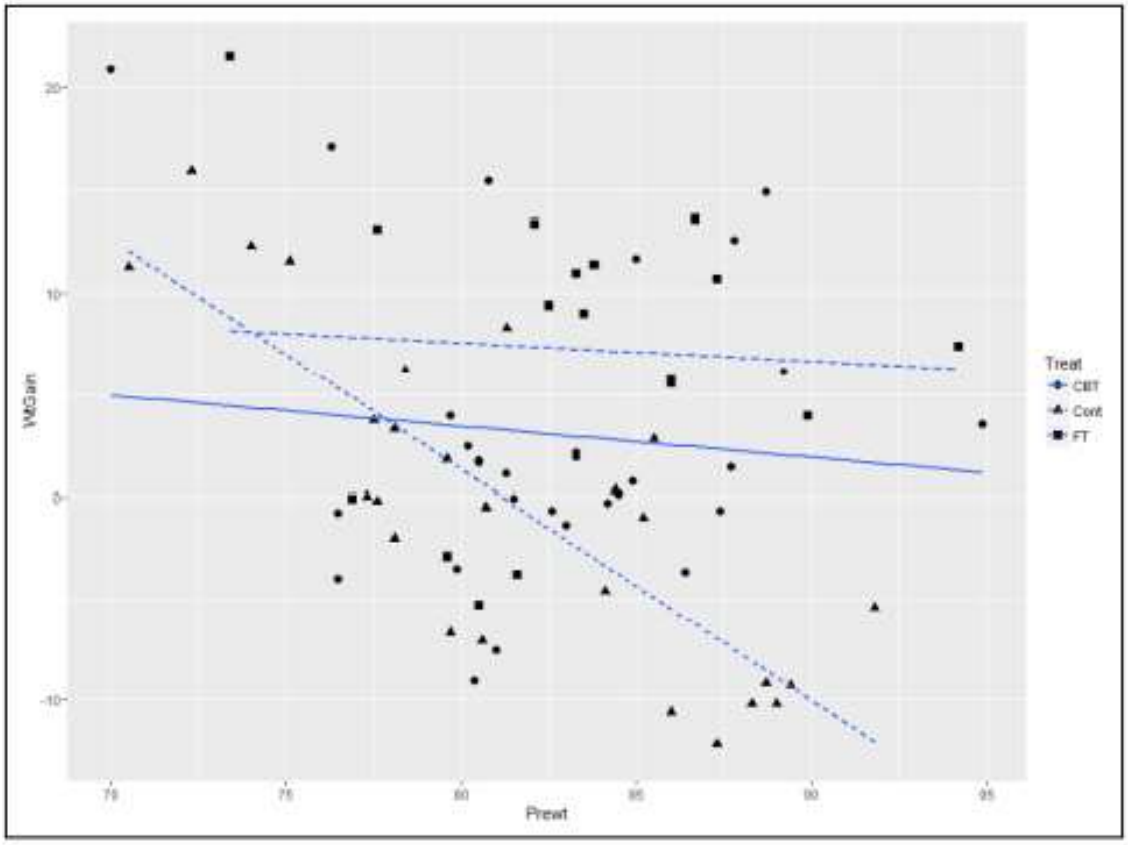
```
geom_smooth(method = lm,se = FALSE, aes(linetype=Treat))
```

Ini menginstruksikan ggplot untuk menambahkan baris terpisah yang "memperhalus" data dalam setiap kelompok perlakuan. Argumen metode menentukan lm (pemodelan linier) sehingga setiap baris adalah garis regresi. Argumen berikutnya, se=FALSE, mencegah plot interval kepercayaan di sekitar setiap baris. Terakhir, pemetaan estetis menunjukkan bahwa garis untuk setiap level Treat akan terlihat berbeda. Jadi kode lengkapnya adalah:

```
ggplot(anorexia, aes(x=Prewt,y=WtGain, shape = Treat)) +
  geom_point(size=2.5) +
  geom_smooth(method = lm,se = FALSE, aes(linetype=Treat))
```

dan hasilnya adalah Gambar 14.11.

Seperti yang Anda lihat, tiga garis regresi miring negatif tidak paralel. Garis untuk CBT sejajar dengan garis untuk FT, tetapi garis untuk Cont (kondisi kontrol) memiliki kemiringan negatif yang jauh lebih besar. Dengan asumsi bahwa pasien dalam kelompok kontrol tidak menerima pengobatan, ini terdengar cukup intuitif: Karena mereka tidak menerima pengobatan, banyak dari pasien anoreksia ini (yang lebih berat) terus menurunkan berat badan (daripada menambah berat badan), sehingga sangat negatif. kemiringan untuk garis itu.



Gambar 14.11 Penambahan Berat Badan versus Perlakuan dan Prewt dalam kerangka data anoreksia, dengan garis regresi untuk skor di setiap tingkat Perlakuan.

Rupanya, kami memiliki interaksi Treat X Prewt. Apakah analisis mendukung hal ini?
Untuk memasukkan interaksi dalam model, saya harus menambahkan Treat*Prewt ke rumus:

```
anorexia.w.interaction <- lm(WtGain ~ Treat + Prewt +
          Treat*Prewt, data=anorexia)
```

Apakah menambahkan interaksi membuat perbedaan?

```
> anova(anorexia.T.and.P,anorexia.w.interaction)
Analysis of Variance Table

Model 1: WtGain ~ Treat + Prewt
Model 2: WtGain ~ Treat + Prewt + Treat * Prewt
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     68 3311.3
2     66 2844.8  2    466.48 5.4112 0.006666 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

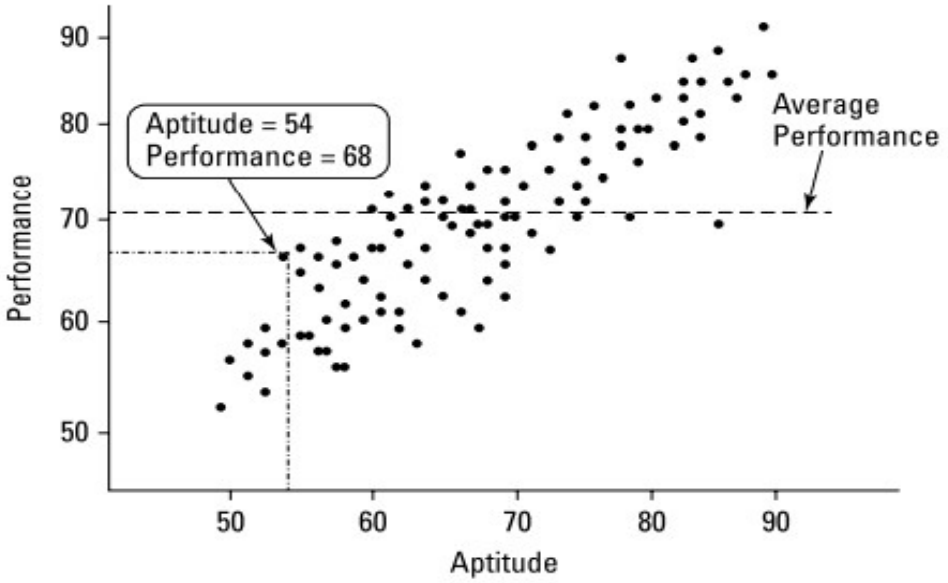
Itu pasti! Dalam kesimpulan Anda tentang penelitian ini, Anda harus menyertakan peringatan bahwa hubungan antara pra-berat dan penambahan berat badan berbeda untuk kontrol daripada untuk perawatan kognitif-perilaku dan untuk perawatan keluarga.

BAB 15
KORELASI: KEBANGKITAN DAN JATUHAN HUBUNGAN

Dalam Bab 14, saya memperkenalkan konsep regresi, alat untuk meringkas dan menguji hubungan antara (dan di antara) variabel. Dalam bab ini, saya memperkenalkan Anda pada naik turunnya korelasi, alat lain untuk melihat hubungan. Saya menggunakan contoh bakat dan kinerja karyawan dari Bab 14 dan menunjukkan cara berpikir tentang data dengan cara yang sedikit berbeda. Konsep baru terhubung dengan apa yang saya tunjukkan di Bab 14, dan Anda akan melihat bagaimana koneksi tersebut bekerja. Saya juga menunjukkan cara menguji hipotesis tentang hubungan dan cara menggunakan fungsi R untuk korelasi.

15.1 PLOT PENCAR

Sebuah plot pencar adalah cara grafis untuk menunjukkan hubungan antara dua variabel. Dalam Bab 14, saya menunjukkan kepada Anda plot pencar data untuk karyawan di FarMisht Consulting, Inc. Saya mereproduksi plot pencar tersebut di sini sebagai Gambar 15.1. Setiap poin mewakili skor satu karyawan pada ukuran Aptitude (pada sumbu x) dan pada ukuran Kinerja (pada sumbu y).



Gambar 15.1 Bakat dan Kinerja di FarMisht Consulting.

15.2 MEMAHAMI KORELASI

Dalam Bab 14, saya mengacu pada Bakat sebagai variabel bebas dan Kinerja sebagai variabel terikat. Tujuan dalam Bab 14 adalah menggunakan Aptitude untuk memprediksi Kinerja. Meskipun saya menggunakan skor pada satu variabel untuk memprediksi skor pada

variabel lainnya, saya tidak bermaksud bahwa skor pada satu variabel menyebabkan skor pada variabel lainnya. "Hubungan" tidak selalu berarti "kausalitas".

Korelasi adalah cara statistik untuk melihat suatu hubungan. Ketika dua hal yang berkorelasi, itu berarti bahwa mereka berbeda bersama-sama. Korelasi positif berarti bahwa skor tinggi di satu pihak dikaitkan dengan skor tinggi di sisi lain, dan skor rendah di satu sisi dikaitkan dengan skor rendah di sisi lain. Plot pencar pada Gambar 15.1 adalah contoh korelasi positif.

Korelasi negatif, di sisi lain, berarti bahwa skor tinggi pada hal pertama dikaitkan dengan skor rendah pada hal kedua. Korelasi negatif juga berarti bahwa skor rendah pada yang pertama dikaitkan dengan skor tinggi pada yang kedua. Contohnya adalah korelasi antara berat badan dan waktu yang dihabiskan untuk program penurunan berat badan. Jika program tersebut efektif, semakin tinggi jumlah waktu yang dihabiskan untuk program tersebut, semakin rendah berat badan. Juga, semakin rendah jumlah waktu yang dihabiskan untuk program, semakin tinggi berat badan. Tabel 15.1, pengulangan Tabel 14.2, menunjukkan data untuk 16 konsultan FarMisht.

Tabel 15.1 Skor Bakat dan Skor Kinerja untuk 16 Konsultan FarMisht

Konsultan	Bakat	Pertunjukan
1	45	56
2	81	74
3	65	56
4	87	81
5	68	75
6	91	84
7	77	68
8	61	52
9	55	57
10	66	82
11	82	73
12	93	90
13	76	67
14	83	79
15	61	70
16	74	66
Mean	72.81	70.63
Variance	181.63	126.65
Standar Deviasi	13.48	11.25

Sesuai dengan cara saya menggunakan Aptitude dan Performance di Bab 14, Aptitude adalah variabel x dan Performance adalah variabel y.
Rumus untuk menghitung korelasi antara keduanya adalah:

$$r = \frac{\left[\frac{1}{N-1}\right]\sum(x-\bar{x})(y-\bar{y})}{s_x s_y}$$

Istilah di sebelah kiri, r, disebut koefisien korelasi. Ini juga disebut koefisien korelasi momen produk Pearson, setelah penciptanya, Karl Pearson.

Dua suku penyebut di sebelah kanan adalah simpangan baku variabel x dan simpangan baku variabel y. Istilah dalam pembilang disebut kovarians. Cara lain untuk menulis rumus ini adalah:

$$r = \frac{\mathrm{cov}(x,y)}{s_x s_y}$$

Kovarians mewakili x dan y bervariasi bersama-sama. Membagi kovarians dengan produk dari dua standar deviasi memberlakukan beberapa batasan. Batas bawah koefisien korelasi adalah -1,00, dan batas atas adalah +1,00.

Koefisien korelasi -1,00 mewakili korelasi negatif sempurna (skor x rendah terkait dengan skor y tinggi, dan skor x tinggi terkait dengan skor y rendah). Korelasi +1,00 mewakili korelasi positif sempurna (skor x rendah terkait dengan skor y rendah dan skor x tinggi terkait dengan skor y tinggi). Korelasi sebesar 0,00 berarti kedua variabel tersebut tidak berhubungan.
Menerapkan rumus ke data pada Tabel 15.1,

$$r = \frac{\left[\frac{1}{N-1}\right]\sum(x-\bar{x})(y-\bar{y})}{s_x s_y}$$

$$= \frac{\left[\frac{1}{16-1}\right]\left[(45-72.81)(56-70.63)+\ldots+(74-72.81)(66-70.83)\right]}{(13.48)(11.25)} = .783$$

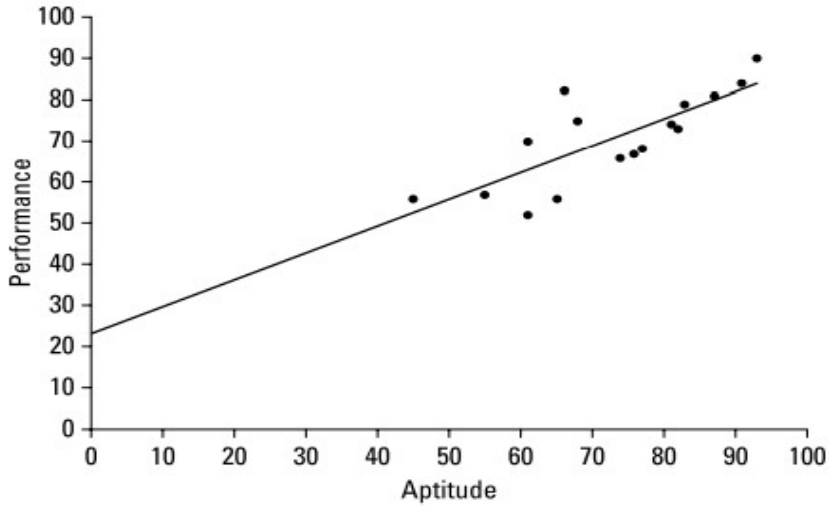
Apa sebenarnya arti angka ini? Saya akan memberitahu Anda.

15.3 KORELASI DAN REGRESI

Gambar 15.2 menunjukkan plot pencar dari hanya 16 karyawan pada Tabel 15-1 dengan garis yang “paling sesuai” dengan poin. Dimungkinkan untuk menggambar garis dalam jumlah tak terbatas melalui titik-titik ini. Yang mana yang terbaik?

Untuk menjadi yang terbaik, sebuah garis harus memenuhi standar tertentu: Jika Anda menggambar jarak dalam arah vertikal antara titik dan garis, dan Anda mengkuadratkan jarak tersebut, dan kemudian Anda menambahkan jarak kuadrat itu, garis adalah garis yang membuat jumlah jarak kuadrat itu sekecil mungkin. Garis ini disebut garis regresi.

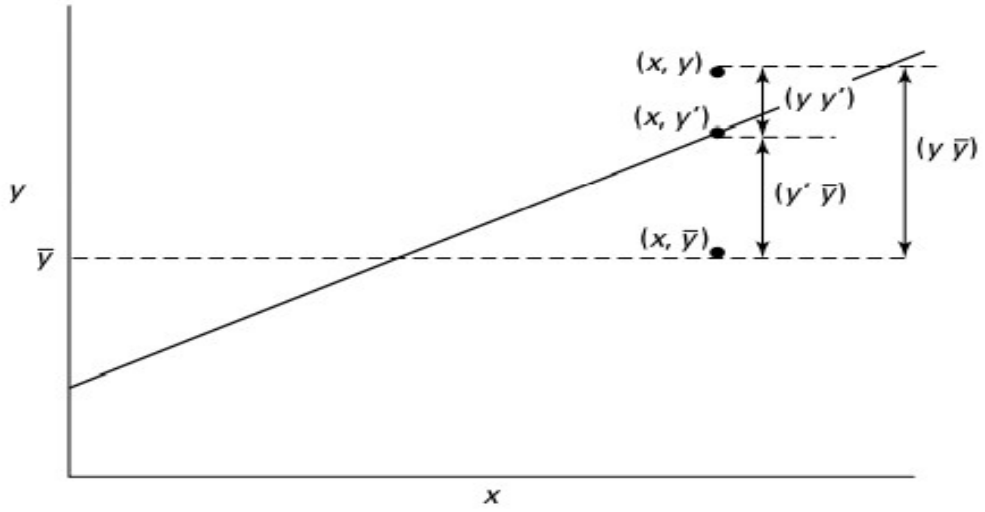
Tujuan garis regresi dalam hidup adalah untuk memungkinkan Anda membuat prediksi. Seperti yang saya sebutkan di Bab 14, tanpa garis regresi, nilai prediksi terbaik dari variabel y adalah rata-rata dari y. Garis regresi memperhitungkan variabel-x dan memberikan prediksi yang lebih tepat. Setiap titik pada garis regresi mewakili nilai prediksi untuk y. Dalam simbologi regresi, setiap nilai prediksi adalah y'.



Gambar 15.2 Scatter plot dari 16 konsultan FarMisht, termasuk garis regresi.

Mengapa saya memberitahu Anda semua ini? Karena korelasi erat kaitannya dengan regresi. Gambar 15.3 berfokus pada satu titik dalam plot pencar, dan pada jaraknya ke garis regresi dan rata-rata. (Ini adalah pengulangan dari Gambar 14.3).

Perhatikan tiga jarak yang ditunjukkan pada gambar. Jarak berlabel (y-y') adalah perbedaan antara titik dan prediksi garis regresi di mana titik seharusnya berada. (Dalam Bab 14, saya menyebutnya residual.) Jarak berlabel (y-$\bar{y}$) adalah perbedaan antara titik dan rata-rata dari y. Jarak berlabel (y'-$\bar{y}$) adalah keuntungan dalam kemampuan prediksi yang Anda dapatkan dari menggunakan garis regresi untuk memprediksi titik alih-alih menggunakan rata-rata untuk memprediksi titik.



Gambar 15.3: Satu titik dalam plot pencar dan jarak yang terkait

Gambar 15.3 menunjukkan bahwa ketiga jarak tersebut berhubungan seperti ini:

$$(y - y') + (y' - \bar{y}) = (y - \bar{y})$$

Seperti yang saya tunjukkan di Bab 14, Anda dapat mengkuadratkan semua residu dan menjumlahkannya, mengkuadratkan semua deviasi titik yang diprediksi dari mean dan menjumlahkannya, dan mengkuadratkan semua deviasi titik aktual dari mean dan menambahkannya juga. .

Ternyata jumlah kuadrat ini terkait dengan cara yang sama seperti penyimpangan yang baru saja saya tunjukkan kepada Anda:

$$SS_{Residual} + SS_{Regression} = SS_{Total}$$

Jika $SS_{Regression}$ besar dibandingkan dengan $SS_{Residual}$, hubungan antara variabel x dan variabel y adalah kuat. Artinya, sepanjang plot pencar, variabilitas di sekitar garis regresi kecil.

Sebaliknya, jika $SS_{Regresi}$ kecil dibandingkan dengan $SS_{Residual}$, hubungan antara variabel x dan variabel y lemah. Dalam hal ini, variabilitas di sekitar garis regresi besar di seluruh plot pencar.

Salah satu cara untuk menguji $SS_{Regression}$ terhadap $SS_{Residual}$ adalah dengan membagi masing-masing dengan derajat kebebasannya (1 untuk $SS_{Regression}$ dan N–2 untuk $SS_{Residual}$) untuk membentuk estimasi varians (juga dikenal sebagai mean-squares, atau MS), dan kemudian membagi satu dengan yang lain untuk menghitung F. Jika $MS_{Regression}$ secara signifikan lebih besar dari $MS_{Residual}$, Anda memiliki bukti bahwa hubungan xy kuat. (Lihat Bab 14 untuk detailnya.)

Inilah yang menentukan, sejauh menyangkut korelasi: Cara lain untuk menilai ukuran $SS_{Regression}$ adalah membandingkannya dengan SS_{Total}. Bagi yang pertama dengan yang kedua. Jika rasionya besar, ini menunjukkan bahwa hubungan xy kuat. Rasio ini memiliki nama. Ini

disebut koefisien determinasi. Simbolnya adalah r^2. Ambil akar kuadrat dari koefisien ini, dan Anda memiliki . . . koefisien korelasi!

$$r = r^2 = \pm\sqrt{\frac{SS_{Regression}}{SS_{Total}}}$$

Tanda plus-atau-minus (±) berarti bahwa r adalah akar kuadrat positif atau negatif, tergantung pada kemiringan garis regresi positif atau negatif. Jadi, jika Anda menghitung koefisien korelasi dan Anda ingin segera mengetahui apa artinya nilainya, kuadratkan saja. Jawabannya — koefisien determinasi — memungkinkan Anda mengetahui proporsi SS_{Total} yang terikat dalam hubungan antara variabel x dan variabel y. Jika proporsinya besar, koefisien korelasi menandakan hubungan yang kuat. Jika proporsinya kecil, koefisien korelasi menandakan hubungan yang lemah.

Dalam contoh Aptitude-Performance, koefisien korelasinya adalah 0,783. Koefisien determinasinya adalah:

$$r^2 = (.783)^2 = .613$$

Dalam sampel 16 konsultan ini, $SS_{Regression}$ adalah 61,3 persen dari SS_{Total}. Kedengarannya seperti proporsi yang besar, tapi apa yang besar? Apanya yang kecil? Pertanyaan-pertanyaan itu berteriak untuk tes hipotesis.

15.4 PENGUJIAN HIPOTESIS TENTANG KORELASI

Pada bagian ini, saya menunjukkan kepada Anda bagaimana menjawab pertanyaan-pertanyaan penting tentang korelasi. Seperti jenis pengujian hipotesis lainnya, idenya adalah menggunakan statistik sampel untuk membuat kesimpulan tentang parameter populasi. Di sini, statistik sampel adalah r, koefisien korelasi. Berdasarkan konvensi, parameter populasi adalah ρ (rho), padanan Yunani untuk r. (Ya, itu memang terlihat seperti huruf p, tapi itu benar-benar padanan bahasa Yunani dari r.) Dua jenis pertanyaan penting sehubungan dengan korelasi: (1) Apakah koefisien korelasi lebih besar dari 0? (2) Apakah dua koefisien korelasi berbeda satu sama lain?

Apakah koefisien korelasi lebih besar dari nol?

Kembali sekali lagi ke contoh Aptitude-Performance, Anda dapat menggunakan sampel r untuk menguji hipotesis tentang populasi — koefisien korelasi untuk semua konsultan di FarMisht Consulting.

Dengan asumsi bahwa Anda tahu sebelumnya (sebelum Anda mengumpulkan data sampel apa pun) bahwa korelasi apa pun antara Bakat dan Kinerja harus positif, hipotesisnya adalah:

$$H_0: \rho \leq 0$$

$$H_1: \rho > 0$$

Tetapkan α = 0,05.

Uji statistik yang sesuai adalah uji-t. Rumusnya adalah

$$t = \frac{r - \rho}{s_r}$$

Tes ini memiliki N-2 df.

Sebagai contoh, nilai dalam pembilang ditetapkan: r adalah 0,783 dan (dalam H0) adalah 0. Bagaimana dengan penyebutnya? Saya tidak akan membebani Anda dengan detailnya. Aku hanya akan memberitahumu itu

$$\sqrt{\frac{1-r^2}{N-2}}$$

Dengan sedikit aljabar, rumus untuk uji-t disederhanakan menjadi

$$t = \frac{r\sqrt{N-2}}{\sqrt{1-r^2}}$$

Sebagai contoh,

$$t = \frac{r\sqrt{N-2}}{\sqrt{1-r^2}} = \frac{.783\sqrt{16-2}}{\sqrt{1-.783^2}} = 4.707$$

Dengan df 14 dan α = .05 (one-tailed), nilai kritis t adalah 1,76. Karena nilai yang dihitung lebih besar dari nilai kritis, keputusannya adalah menolak H_0.

Apakah dua koefisien korelasi berbeda?

FarKlempt Robotics memiliki cabang konsultan yang menilai bakat dan kinerja dengan alat pengukuran yang sama yang digunakan FarMisht Consulting. Dalam sampel 20 konsultan di FarKlempt Robotics, korelasi antara Aptitude dan Performance adalah 0,695. Apakah ini berbeda dengan korelasi (.783) di FarMisht Consulting? Jika Anda tidak memiliki cara untuk mengasumsikan bahwa satu korelasi harus lebih tinggi dari yang lain, hipotesisnya adalah:

$$H_0: \rho_{\text{FarMisht}} = \rho_{\text{FarKlempt}}$$

$$H_1: \rho_{\text{FarMisht}} \neq \rho_{\text{FarKlempt}}$$

Sekali lagi, α = 0,05.

Untuk alasan yang sangat teknis, Anda tidak dapat menyiapkan uji-t untuk yang satu ini. Faktanya, Anda bahkan tidak dapat bekerja dengan 0,783 dan 0,695, dua koefisien korelasi. Sebaliknya, yang Anda lakukan adalah mengubah setiap koefisien korelasi menjadi sesuatu yang lain dan kemudian bekerja dengan dua "sesuatu yang lain" dalam formula yang memberi Anda — percaya atau tidak — uji-z.

Transformasi tersebut disebut transformasi r ke z Fisher. Fisher adalah ahli statistik yang dikenang sebagai F dalam uji-F. Dia mengubah r menjadi z dengan melakukan ini:

$$z_r = \frac{1}{2}\left[\log_e(1+r) - \log_e(1-r)\right]$$

Jika Anda tahu apa artinya log, baiklah. Jika tidak, jangan khawatir tentang itu. (Saya menjelaskannya di Bab 16.) R menangani semua ini untuk Anda, seperti yang Anda lihat sebentar lagi.

Bagaimanapun, untuk contoh ini:

$$z_{.783} = \frac{1}{2}\left[\log_e(1+.783) - \log_e(1-.783)\right] = 1.0530$$

$$z_{.695} = \frac{1}{2}\left[\log_e(1+.695) - \log_e(1-.695)\right] = 0.8576$$

Setelah Anda mengubah r ke z, rumusnya adalah:

$$Z = \frac{z_1 - z_2}{\sigma_{z_1 - z_2}}$$

Penyebutnya ternyata lebih mudah dari yang Anda kira.

Itu adalah

$$\sigma_{z_1 - z_2} = \sqrt{\frac{1}{N_1 - 3} + \frac{1}{N_2 - 3}}$$

Untuk contoh ini,

$$\sigma_{z_1 - z_2} = \sqrt{\frac{1}{N_1 - 3} + \frac{1}{N_2 - 3}} = \sqrt{\frac{1}{16 - 3} + \frac{1}{20 - 3}} = .368$$

Seluruh rumusnya adalah:

$$Z = \frac{z_1 - z_2}{\sigma_{z_1 - z_2}} = \frac{1.0530 \quad 0.8576}{.368} = .531$$

Langkah selanjutnya adalah membandingkan nilai yang dihitung dengan distribusi normal standar. Untuk uji dua arah dengan α = 0,05, nilai kritis dalam distribusi normal standar adalah 1,96 pada ekor atas dan -1,96 pada ekor bawah. Nilai yang dihitung berada di antara keduanya, sehingga keputusannya adalah tidak menolak H_0.

15.5 KORELASI DALAM R

Di bagian ini, saya bekerja dengan contoh FarMisht. Bingkai data, FarMisht. frame, memegang titik data yang ditunjukkan pada Tabel 14-4. Inilah cara saya membuatnya:

[[-

```
Aptitude <- c(45, 81, 65, 87, 68, 91, 77, 61, 55, 66, 82, 93,
         76, 83, 61, 74)
Performance <- c(56, 74, 56, 81, 75, 84, 68, 52, 57, 82, 73, 90,
         67, 79, 70, 66)
Personality <- c(9, 15, 11, 15, 14, 19, 12, 10, 9, 14, 15, 14,
         16, 18, 15, 12)
FarMisht.frame <- data.frame(Aptitude, Performance, Personality)
```

Menghitung koefisien korelasi

Untuk mencari koefisien korelasi hubungan antara Aptitude dan Performance, saya menggunakan fungsi cor():

```
> with(FarMisht.frame, cor(Aptitude,Performance))
[1] 0.7827927
```

Koefisien korelasi momen produk Pearson yang dihitung cor() dalam contoh ini adalah default untuk argumen metodenya:

```
cor(Farmisht.frame, method = "pearson")
```

Dua kemungkinan nilai lain untuk metode adalah "spearman" dan "kendall", yang saya bahas di Lampiran B.

Menguji koefisien korelasi

Untuk menemukan koefisien korelasi, dan mengujinya pada saat yang sama, R memberikan cor. tes(). Berikut adalah uji satu sisi (ditentukan oleh alternatif = "lebih besar"):

```
> with(FarMisht.frame, cor.test(Aptitude,Performance,
         alternative = "greater"))

         Pearson's product-moment correlation

data:  Aptitude and Performance
t = 4.7068, df = 14, p-value = 0.0001684
alternative hypothesis: true correlation is greater than 0
95 percent confidence interval:
 0.5344414 1.0000000
sample estimates:
      cor
0.7827927
```

Seperti halnya cor(), Anda dapat menentukan "spearman" atau "kendall" sebagai metode untuk cor.test().

Menguji perbedaan antara dua koefisien korelasi

Pada bagian sebelumnya "Apakah dua koefisien korelasi berbeda?" Saya membandingkan koefisien korelasi Aptitude-Performance (0,695) untuk 20 konsultan di FarKlempt Robotics dengan korelasi (,783) untuk 16 konsultan di FarMisht Consulting. Perbandingan dimulai dengan transformasi r ke z Fisher untuk setiap koefisien. Statistik uji (Z) adalah selisih dari nilai-nilai yang ditransformasi dibagi dengan galat baku dari selisih tersebut.

Sebuah fungsi yang disebut r.test() melakukan semua pekerjaan jika Anda memberikan koefisien dan ukuran sampel. Fungsi ini hidup dalam paket psych, jadi pada tab Paket, klik Sisipkan. Kemudian pada kotak dialog Insert Packages, ketik psych. Saat psych muncul di tab Packages, pilih kotak centangnya.

Inilah fungsinya, dan argumennya:

```
r.test(r12=.783, n=16, r34=.695, n2=20)
```

Yang ini cukup khusus tentang bagaimana Anda menyatakan argumen. Argumen pertama adalah koefisien korelasi pertama. Yang kedua adalah ukuran sampelnya. Argumen ketiga adalah koefisien korelasi kedua, dan yang keempat adalah ukuran sampelnya. Label 12 untuk koefisien pertama dan label 34 untuk koefisien kedua menunjukkan bahwa kedua koefisien saling bebas.

Jika Anda menjalankan fungsi itu, inilah hasilnya:

```
Correlation tests
Call:r.test(n = 16, r12 = 0.783, r34 = 0.695, n2 = 20)
Test of difference between two independent correlations
 z value 0.53    with probability  0.6
```

Menghitung matriks korelasi

Selain menemukan koefisien korelasi tunggal, cor() dapat menemukan semua koefisien korelasi berpasangan untuk kerangka data, menghasilkan matriks korelasi:

```
> cor(FarMisht.frame)
             Aptitude Performance Personality
Aptitude    1.0000000   0.7827927   0.7499305
Performance 0.7827927   1.0000000   0.7709271
Personality 0.7499305   0.7709271   1.0000000
```

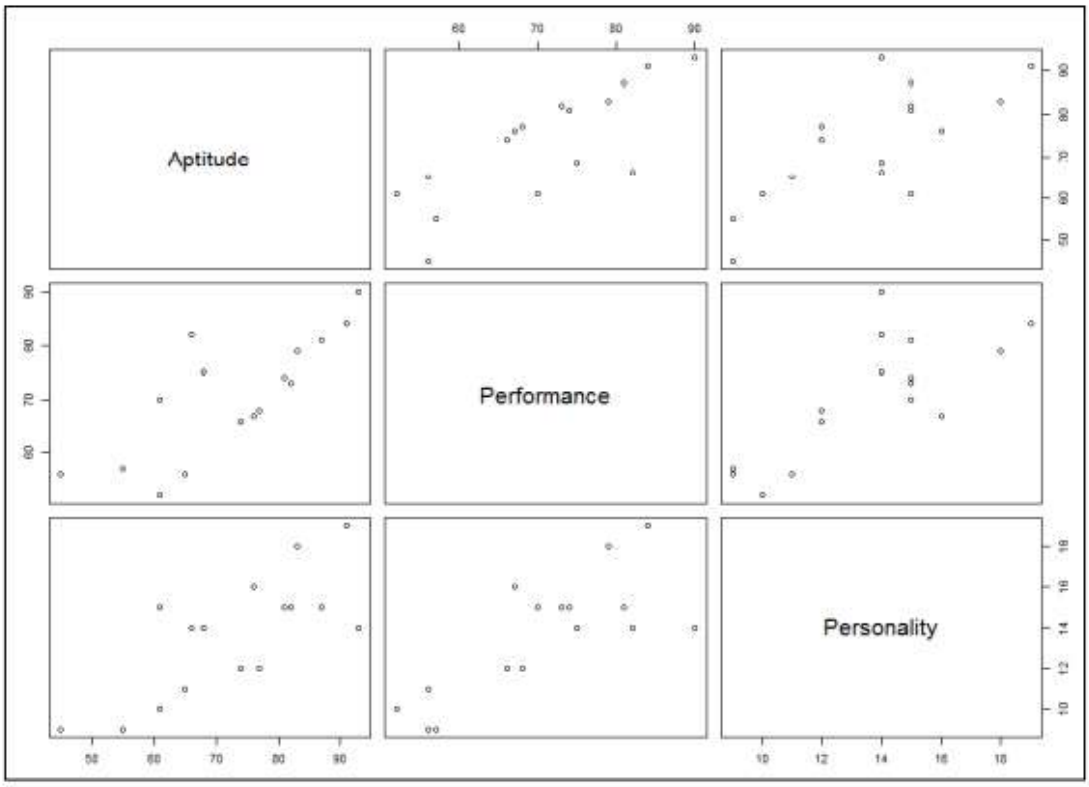
Memvisualisasikan matriks korelasi

Dalam Bab 3, saya menjelaskan beberapa cara untuk memvisualisasikan matriks seperti yang ada di bagian sebelumnya. Berikut cara melakukannya dengan grafik R dasar:

```
pairs(FarMisht.frame)
```

Fungsi ini menghasilkan Gambar 15.4.

Diagonal utama, tentu saja, memuat nama-nama variabel. Setiap sel off-diagonal adalah plot pencar dari pasangan variabel yang disebutkan dalam baris dan kolom. Misalnya, sel di sebelah kanan langsung Aptitude adalah plot sebar Aptitude (sumbu y) dan Kinerja (sumbu x). Sel tepat di bawah Aptitude adalah kebalikannya — ini adalah plot pencar Performance (sumbu y) dan Aptitude (sumbu x).



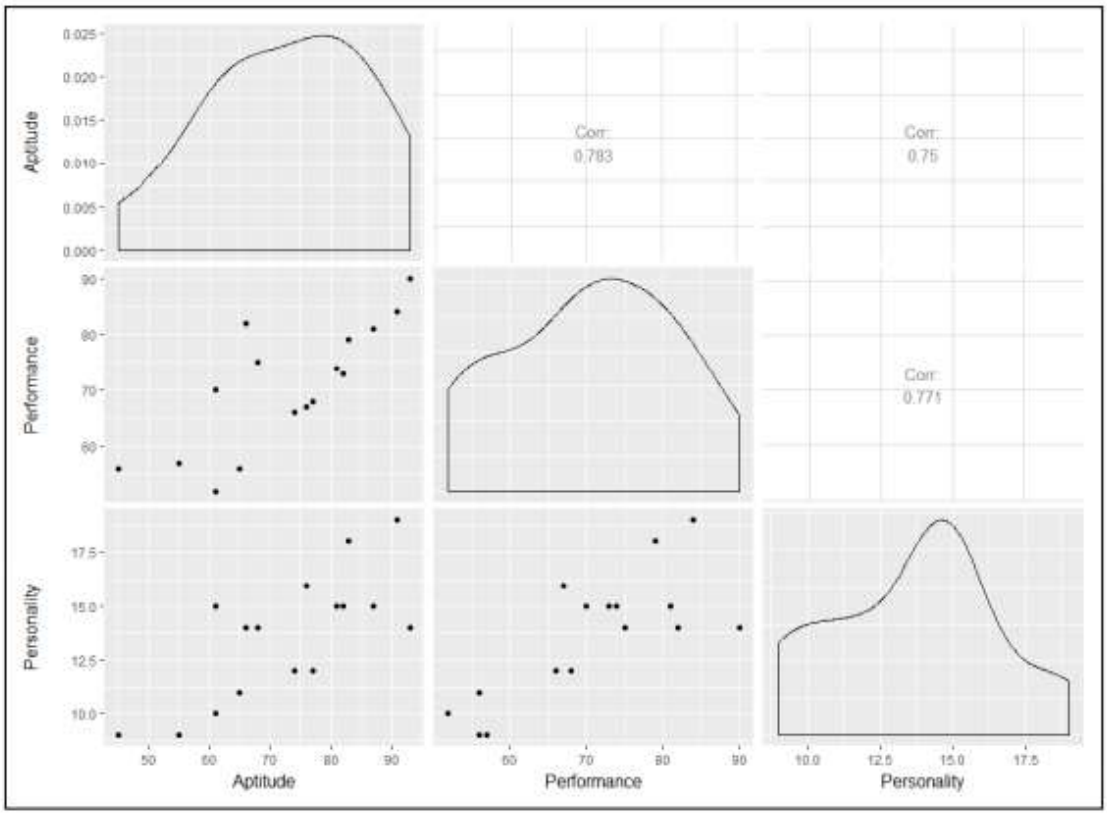
Gambar 15.4 Matriks korelasi untuk Bakat, Kinerja, dan Kepribadian, direnderdalam grafik R dasar.

Seperti yang juga saya sebutkan di Bab 3, sebuah paket bernama GGally (dibangun di atas ggplot2) menyediakan ggpairs(), yang menghasilkan sedikit lebih banyak. Temukan GGally pada tab Packages dan pilih kotak centangnya. Kemudian menggambar Gambar 15.5.

```
ggpairs(FarMisht.frame)
```

Diagonal utama menyediakan fungsi kepadatan untuk setiap variabel, sel-sel di luar diagonal atas menyajikan koefisien korelasi, dan sel-sel yang tersisa menunjukkan plot pencar berpasangan.

Tampilan yang lebih rumit dimungkinkan dengan paket corrgram. Pada tab Paket, klik Instal, dan di kotak dialog Instal, ketik corrgram dan klik Instal. (Bersabarlah. Paket ini menginstal banyak item.) Kemudian, pada tab Packages, temukan corgram dan centang kotaknya.

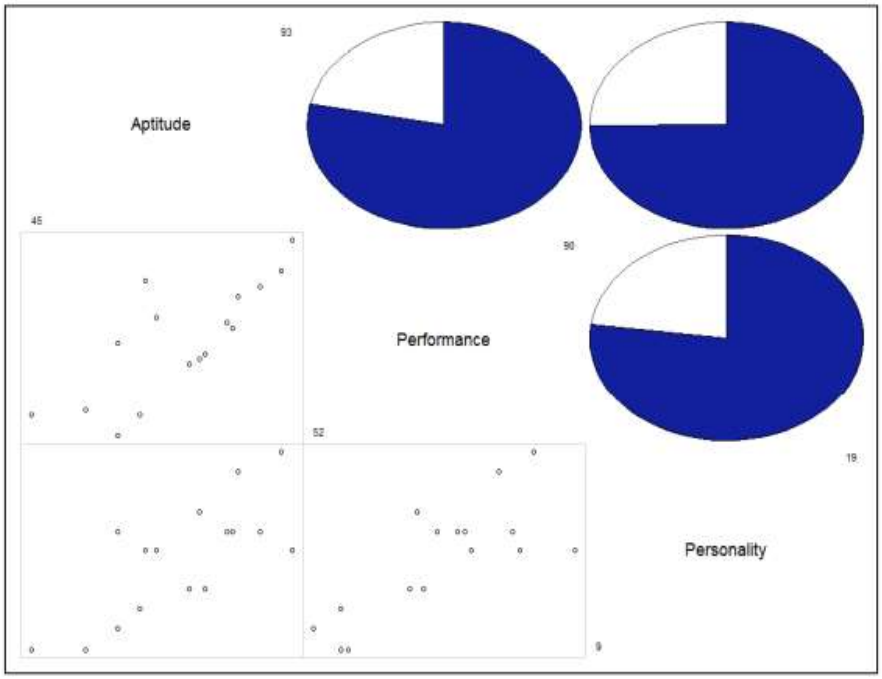


Gambar 15.5 Matriks korelasi untuk Bakat, Kinerja, dan Kepribadian, direnderer dalam GGally (paket berbasis ggplot2).

Fungsi corrgram() bekerja dengan bingkai data dan memungkinkan Anda memilih opsi untuk apa yang masuk ke diagonal utama (diag.panel) dari matriks yang dihasilkan, apa yang masuk ke sel di bagian atas matriks (panel atas) , dan apa yang masuk ke dalam sel di bagian bawah matriks (lower.panel). Untuk diagonal utama, saya memilih untuk menunjukkan nilai minimum dan maksimum untuk setiap variabel. Untuk setengah bagian atas, saya menentukan diagram lingkaran untuk menunjukkan nilai koefisien korelasi: Proporsi yang diisi mewakili nilai. Untuk bagian bawah, saya ingin plot pencar di setiap sel:

```
corrgram(FarMisht.frame, diag.panel=panel.minmax,
                         upper.panel = panel.pie,
                                lower.panel = panel.pts)
```

Hasilnya adalah Gambar 15.6.



Gambar 15.6 Matriks korelasi untuk Aptitude, Performance, dan Personality, direnderdalam paket corrgram.

15.6 KORELASI GANDA

Koefisien korelasi dalam matriks korelasi yang dijelaskan pada bagian sebelumnya digabungkan untuk menghasilkan koefisien korelasi berganda. Ini adalah angka yang merangkum hubungan antara variabel dependen — Kinerja, dalam contoh ini — dan dua variabel independen (*Aptitude* dan *Personality*).

Untuk menunjukkan kepada Anda bagaimana koefisien korelasi ini digabungkan, saya menyingkat *Performance* sebagai P, *Aptitude* sebagai A, dan *Personality* sebagai F (FarMisht Personality Inventory). Jadi rPA adalah koefisien korelasi untuk *Performance* dan *Aptitude* (.7827927), rPF adalah koefisien korelasi untuk *Performance* dan *Personality* (.7709271), dan rAF adalah koefisien korelasi untuk *Aptitude* dan *Personality* (.7499305).

Inilah rumus yang menyatukan semuanya:

$$R_{P.AF} = \sqrt{\frac{r_{PA}^2 + r_{PF}^2 - 2r_{PA}r_{PF}r_{AF}}{1 - r_{AF}^2}}$$

Huruf besar R di sebelah kiri menunjukkan bahwa ini adalah koefisien korelasi berganda, berlawanan dengan huruf kecil r, yang menunjukkan korelasi antara dua variabel. Subskrip P.AF berarti korelasi ganda antara Kinerja dan kombinasi Bakat dan Kepribadian.

Untuk contoh ini,

$$R_{P.AF} = \sqrt{\frac{(.7827927)^2 + (.7709271)^2 - 2(.7827927)(.7709271)(.7499305)}{1 - (.7499305)^2}} = .8306841$$

Jika Anda mengkuadratkan angka ini, Anda mendapatkan koefisien determinasi berganda. Di Bab 14, Anda bertemu dengan Multiple R-Squared, dan itulah yang terjadi. Untuk contoh ini, hasilnya adalah:

$$R^2_{P.AF} = (.830641)^2 = .6900361$$

Korelasi berganda dalam R

Cara termudah untuk menghitung koefisien korelasi berganda adalah dengan menggunakan lm() dan melanjutkan seperti dalam regresi berganda:

```
> FarMisht.multreg <- lm(Performance ~ Aptitude + Personality,
          data = FarMisht.frame)
> summary(FarMisht.multreg)

Call:
lm(formula = Performance ~ Aptitude + Personality, data =
          FarMisht.frame)

Residuals:
   Min     1Q Median     3Q    Max
-8.689 -2.834 -1.840  2.886 13.432

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.2825     9.6595   2.100   0.0558 .
Aptitude      0.3905     0.1949   2.003   0.0664 .
Personality   1.6079     0.8932   1.800   0.0951 .
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.73 on 13 degrees of freedom
Multiple R-squared:   0.69,   Adjusted R-squared:  0.6423
F-statistic: 14.47 on 2 and 13 DF,  p-value: 0.0004938
```

Di baris berikutnya hingga terakhir, Beberapa R-kuadrat ada di sana, menunggu Anda. Jika Anda harus bekerja dengan kuantitas itu untuk beberapa alasan, itu

```
> summary(FarMisht.multreg)$r.squared
[1] 0.6900361
```

Dan untuk menghitung R:

```
> Mult.R.sq <- summary(FarMisht.multreg)$r.squared
> Mult.R <- sqrt(Mult.R.sq)
> Mult.R
[1] 0.8306841
```

Menyesuaikan R-kuadrat

Dalam output lm(), Anda melihat Adjusted R-squared. Mengapa perlu "menyesuaikan" R-kuadrat?

Dalam regresi berganda, menambahkan variabel independen (seperti Kepribadian) terkadang membuat persamaan regresi menjadi kurang akurat. Koefisien determinasi berganda, R-kuadrat, tidak mencerminkan hal ini. Penyebutnya adalah SSTotal (untuk variabel terikat), dan itu tidak pernah berubah. Pembilang hanya bisa bertambah atau tetap sama. Jadi setiap penurunan akurasi tidak menghasilkan R-kuadrat yang lebih rendah.

Lihat Lainnya Pada Ganda Korelasi

Sekarang saya akan menggunakan R (perangkat lunak statistik) sebagai alat pengajaran untuk menunjukkan kepada Anda apa yang saya katakan sebelumnya tentang R (koefisien korelasi berganda): R (koefisien) adalah korelasi antara variabel dependen dan kombinasi dari dua variabel bebas.

Anda tidak akan pernah melakukan ini dalam praktik, tetapi begini: Saya membuat korelasi antara Kinerja dan kombinasi Bakat dan Kepribadian. Yang penting adalah menimbang variabel-variabel ini dengan koefisiennya (sebagaimana ditentukan oleh lm()):

```
> with(FarMisht.frame, cor(Performance, .390519*Aptitude +
1.607918*Personality))
[1] 0.8306841
```

Sekali lagi, Anda tidak akan pernah melakukan ini — Anda harus menjalankan lm() untuk menghitung koefisien, dan setelah Anda melakukannya, Anda sudah memiliki semua yang Anda butuhkan. Saya hanya berpikir ini mungkin membantu Anda memahami banyak R.

Mempertimbangkan derajat kebebasan untuk memperbaiki kekurangannya. Setiap kali Anda menambahkan variabel independen, Anda mengubah derajat kebebasan, dan itu membuat semua perbedaan. Asal tahu saja, berikut penyesuaiannya:

$$Adjusted\ R^2 = 1 - \left(1 - R^2\right)\left[\frac{(N-1)}{(N-k-1)}\right]$$

K pada penyebut adalah banyaknya variabel bebas.

Jika Anda pernah harus bekerja dengan kuantitas ini (dan saya tidak yakin mengapa Anda melakukannya), berikut ini cara mengambilnya:

```
> summary(FarMisht.multreg)$adj.r.squared
[1] 0.6423494
```

15.7 KORELASI PARSIAL

Kinerja dan Bakat dikaitkan dengan Kepribadian (dalam contoh). Asosiasi masing-masing dengan Kepribadian mungkin entah bagaimana menyembunyikan korelasi sebenarnya di antara mereka. Apa korelasinya jika Anda dapat menghapus asosiasi itu? Cara lain untuk

menanyakan ini: Apa yang akan menjadi korelasi Performance-Aptitude jika Anda dapat mempertahankan Personality konstan?

Salah satu cara untuk menjaga Kepribadian tetap konstan adalah dengan menemukan korelasi Kinerja-Aptitude untuk sampel konsultan yang memiliki satu skor Kepribadian — 17, misalnya. Dalam sampel seperti ini, korelasi setiap variabel dengan Kepribadian adalah 0. Namun, hal ini biasanya tidak mungkin dilakukan di dunia nyata. Cara lain adalah dengan menemukan korelasi parsial antara Kinerja dan Bakat. Ini adalah cara statistik untuk menghilangkan hubungan setiap variabel dengan Kepribadian dalam sampel Anda. Anda menggunakan koefisien korelasi dalam matriks korelasi untuk melakukan ini:

$$r_{PA.F} = \frac{r_{PA} - r_{PF} r_{AF}}{\sqrt{1 - r_{PF}^2}\sqrt{1 - r_{AF}^2}}$$

Sekali lagi, P adalah singkatan dari Performance, A untuk Aptitude, dan F untuk Personality. Subskrip PA.F berarti korelasi antara Performa dan Bakat dengan Kepribadian "terpisah". Untuk contoh ini,

$$r_{PA.F} = \frac{.7827927 - (.7709271)(.7499305)}{\sqrt{1 - (.7709271)^2}\sqrt{1 - (.7499305)^2}} = .4857198$$

15.8 KORELASI PARSIAL DALAM R

Paket bernama ppcor memiliki fungsi untuk menghitung korelasi parsial dan untuk menghitung korelasi semiparsial, yang akan saya bahas di bagian selanjutnya. Pada tab Paket, klik Instal. Di kotak dialog Instal Paket, ketik ppcor lalu klik Instal. Selanjutnya, temukan ppcor di kotak dialog Packages dan pilih kotak centangnya.

Fungsi pcor.test() menghitung korelasi antara Performa dan Bakat dengan Kepribadian secara parsial:

```
> with (FarMisht.frame, pcor.test(x=Performance, y=Aptitude,
          z=Personality))
   estimate    p.value statistic  n gp  Method
1 0.4857199 0.06642269    2.0035 16  1 pearson
```

Selain koefisien korelasi (ditampilkan di bawah perkiraan), ini menghitung uji-t korelasi dengan N–3 df (ditampilkan di bawah statistik) dan nilai-p terkait. Jika Anda lebih suka menghitung semua kemungkinan korelasi parsial (dan nilai-p dan statistik-t terkait) dalam bingkai data, gunakan pcor():

```
> pcor(FarMisht.frame)
$estimate
            Aptitude Performance Personality
Aptitude    1.0000000   0.4857199   0.3695112
Performance 0.4857199   1.0000000   0.4467067
Personality 0.3695112   0.4467067   1.0000000

$p.value
             Aptitude Performance Personality
Aptitude    0.00000000  0.06642269  0.17525219
Performance 0.06642269  0.00000000  0.09506226
Personality 0.17525219  0.09506226  0.00000000

$statistic
            Aptitude Performance Personality
Aptitude    0.000000    2.003500    1.433764
Performance 2.003500    0.000000    1.800222
Personality 1.433764    1.800222    0.000000
```

Setiap sel di bawah $estimate adalah korelasi parsial dari variabel baris sel dengan variabel kolom sel, dengan variabel ketiga dikeluarkan sebagian. Jika Anda memiliki lebih dari tiga variabel, setiap sel adalah korelasi parsial baris-kolom dengan yang lainnya dipisah sebagian.

15.9 KORELASI SEMIPARSIAL

Dimungkinkan untuk menghapus korelasi dengan Personality hanya dari Aptitude tanpa menghapusnya dari Performance. Ini disebut korelasi semiparsial. Rumus yang satu ini juga menggunakan koefisien korelasi dari matriks korelasi:

$$r_{P(A.F)} = \frac{r_{PA} - r_{PF} r_{AF}}{\sqrt{1 - r_{AF}^2}}$$

Subskrip P(A.F) berarti bahwa korelasi antara Kinerja dan Bakat dengan Kepribadian hanya sebagian dari Bakat.

Menerapkan rumus ini ke contoh,

$$r_{P(A.F)} = \frac{.7827927 - (.7709271)(.7499305)}{\sqrt{1 - (.7499305)^2}} = .3093663$$

Beberapa buku teks statistik menyebut korelasi semiparsial sebagai korelasi bagian.

Korelasi Semiparsial dalam R

Seperti yang saya sebutkan sebelumnya dalam bab ini, paket ppcor memiliki fungsi untuk menghitung korelasi semiparsial. Untuk menemukan korelasi semiparsial antara Kinerja dan Bakat dengan Kepribadian yang hanya sebagian dari Bakat, gunakan spcor.test():

```
> with (FarMisht.frame, spcor.test(x=Performance, y=Aptitude,
          z=Personality))
   estimate   p.value statistic  n gp  Method
1 0.3093664 0.2618492  1.172979 16  1 pearson
```

Seperti yang Anda lihat, outputnya mirip dengan output untuk pcor.test(). Sekali lagi, estimasi adalah koefisien korelasi dan statistik adalah uji-t dari koefisien korelasi dengan N–3 df. Untuk menemukan korelasi semiparsial untuk seluruh bingkai data, gunakan spcor():

```
> spcor(FarMisht.frame)
$estimate
             Aptitude Performance Personality
Aptitude    1.0000000   0.3213118   0.2299403
Performance 0.3093664   1.0000000   0.2779778
Personality 0.2353503   0.2955039   1.0000000

$p.value
             Aptitude Performance Personality
Aptitude    0.0000000   0.2429000   0.4096955
Performance 0.2618492   0.0000000   0.3157849
Personality 0.3984533   0.2849315   0.0000000

$statistic
             Aptitude Performance Personality
Aptitude    0.0000000    1.223378   0.8518883
Performance 1.1729794    0.000000   1.0433855
Personality 0.8730923    1.115260   0.0000000
```

Perhatikan bahwa, tidak seperti matriks dalam output untuk pcor(), dalam matriks ini angka-angka di atas diagonal tidak sama dengan angka-angka di bawah diagonal.

Cara termudah untuk menjelaskan adalah dengan sebuah contoh. Dalam matriks $estimate, nilai pada kolom pertama, baris kedua (0,3093364) adalah korelasi antara Performa (variabel baris) dan Aptitude (variabel kolom) dengan Personality yang dipisahkan dari Aptitude. Nilai di kolom kedua, baris pertama (0,3213118) adalah korelasi antara Aptitude (yang sekarang menjadi variabel baris) dan Performa (yang sekarang menjadi variabel kolom) dengan Personality di luar Performance.

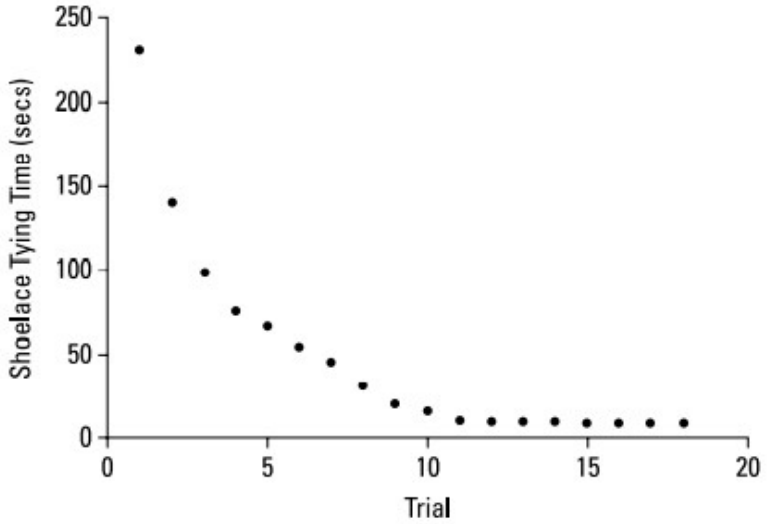
Apa yang terjadi ketika Anda memiliki lebih dari tiga variabel? Dalam hal ini, setiap nilai sel adalah korelasi baris-kolom dengan semua hal lain yang sebagian di luar variabel kolom.

BAB 16
REGRESI CURVILINEAR
KETIKA HUBUNGAN MENJADI RUMIT

Dalam Bab 14 dan 15, saya menjelaskan regresi linier dan korelasi — dua konsep yang bergantung pada garis lurus sebagai ringkasan yang paling cocok untuk plot sebar. Tapi garis tidak selalu yang paling cocok. Proses di berbagai bidang, dari biologi hingga bisnis, lebih sesuai dengan kurva daripada garis.

Misalnya, pikirkan saat Anda mempelajari suatu keterampilan — seperti mengikat tali sepatu. Saat pertama kali mencobanya, butuh waktu cukup lama bukan? Dan kemudian setiap kali Anda mencobanya lagi, semakin sedikit waktu yang Anda butuhkan untuk menyelesaikannya, bukan? Sampai akhirnya, Anda dapat mengikat tali sepatu Anda dengan sangat cepat tetapi Anda tidak bisa lebih cepat lagi — Anda sekarang melakukannya seefisien mungkin. Jika Anda memplot waktu pengikatan tali sepatu (dalam detik) pada sumbu y dan percobaan (kejadian ketika Anda mencoba mengikat sepatu Anda) pada sumbu x, grafiknya mungkin terlihat seperti Gambar 16.1. Garis lurus jelas bukan ringkasan terbaik dari plot seperti ini.

Bagaimana Anda menemukan kurva yang paling pas? (Cara lain untuk mengatakan ini: "Bagaimana Anda merumuskan model untuk data ini?") Saya akan dengan senang hati menunjukkan kepada Anda, tetapi pertama-tama saya harus memberi tahu Anda tentang logaritma, dan tentang bilangan penting yang disebut e. Mengapa? Karena konsep-konsep tersebut membentuk dasar dari tiga jenis regresi nonlinier.



Gambar 16.1 Plot hipotetis untuk mempelajari suatu keterampilan — seperti mengikat tali sepatu.

16.1 APA ITU LOGARITMA?

Jelas dan sederhana, logaritma adalah eksponen — kekuatan di mana Anda menaikkan angka. Dalam persamaan

$$10^2 = 100$$

2 adalah eksponen. Apakah itu berarti bahwa 2 juga merupakan logaritma? Sehat . . . Ya. Dalam hal logaritma,

$$\log_{10} 100 = 2$$

Itu benar-benar hanya cara lain untuk mengatakan 102 100. Matematikawan membacanya sebagai "logaritma dari 100 ke basis 10 sama dengan 2." Artinya, jika Anda ingin menaikkan 10 ke beberapa kekuatan untuk mendapatkan 100, kekuatan itu adalah 2.

Bagaimana dengan 1.000? Seperti yang Anda ketahui

$$10^3 = 1000$$

Jadi

$$\log_{10} 1000 = 3$$

Bagaimana dengan 763? Uh. . . . Hmm. Itu seperti mencoba memecahkan

$$10^x = 763$$

Apa yang bisa menjadi jawaban itu? 10^2 berarti 10 × 10 dan itu memberi Anda 100. 10^3 berarti 10 × 10 × 10 dan itu 1.000. Tapi 763?

Di sinilah Anda harus berpikir di luar kotak dialog. Anda harus membayangkan eksponen yang bukan bilangan bulat. Saya tahu, saya tahu: Bagaimana Anda bisa mengalikan angka dengan dirinya sendiri sebagian kecil pada suatu waktu? Jika Anda bisa, entah bagaimana, angka dalam persamaan 763 itu harus berada di antara 2 (yang membuat Anda menjadi 100) dan 3 (yang membuat Anda menjadi 1.000).

Pada abad ke-16, matematikawan John Napier menunjukkan bagaimana melakukannya, dan logaritma lahir. Mengapa Napier repot-repot dengan ini? Salah satu alasannya adalah bahwa itu sangat membantu para astronom. Para astronom harus berurusan dengan angka-angka yang, yah, astronomi. Logaritma meredakan ketegangan komputasi dalam beberapa cara. Salah satu caranya adalah dengan mengganti bilangan kecil dengan bilangan besar: Logaritma dari 1.000.000 adalah 6, dan logaritma dari 100.000.000 adalah 8. Selain itu, bekerja dengan logaritma akan membuka serangkaian pintasan komputasi yang berguna. Sebelum kalkulator dan komputer muncul di tempat kejadian, ini adalah masalah yang sangat besar.

Kebetulan,

$$10^{2.882525} = 763$$

yang berarti bahwa

$$\log_{10} 763 = 2.882525$$

Anda dapat menggunakan fungsi log10() R untuk memeriksanya:

```
> log10(763)
[1] 2.882525
```

Jika Anda membalikkan prosesnya, Anda akan melihatnya

```
> 10^2.882525
[1] 763.0008
```

Jadi, 2.882525 sedikit meleset, tetapi Anda mengerti.

Sedikit sebelumnya, saya menyebutkan "pintasan komputasi" yang dihasilkan dari logaritma. Ini satu: Jika Anda ingin mengalikan dua angka, tambahkan logaritmanya, lalu temukan angka yang logaritmanya adalah jumlah. Bagian terakhir itu disebut "menemukan antilogaritma." Berikut adalah contoh singkatnya: Untuk mengalikan 100 dengan 1.000:

$$\log_{10}(100)+\log_{10}(1000)=$$
$$2+3=5$$
$$\text{antilog}_{10}(5)=10^5=100{,}000$$

Berikut ini cara pintas komputasi lainnya: Mengalikan logaritma bilangan x dengan bilangan b sama dengan menaikkan x ke pangkat b. Sepuluh, angka yang dipangkatkan ke eksponen, disebut basis. Karena ini juga merupakan basis dari sistem bilangan kami dan semua orang mengenalnya, logaritma dari basis 10 disebut logaritma umum. Dan, seperti yang baru saja Anda lihat, logaritma umum di R adalah log10.

Apakah itu berarti Anda dapat memiliki basis lain? Sangat. Setiap angka (kecuali 0 atau 1 atau angka negatif) dapat menjadi basis. Sebagai contoh,

$$7.8^2=60.84$$

Jadi

$$\log_{7.8}60.84=2$$

Dan Anda dapat menggunakan fungsi log() R untuk memeriksanya:

```
> log(60.84,7.8)
[1] 2
```

Dalam hal basis, satu nomor istimewa. . .

16.2 APA ITU E?

Yang membawa saya ke e, konstanta itu semua tentang pertumbuhan. Bayangkan jumlah yang sangat besar dari Rp 15.000 yang disimpan di rekening bank. Misalkan tingkat bunga adalah 2 persen setahun. (Ya, ini hanya sebuah contoh!) Jika itu adalah bunga sederhana, bank menambahkan Rp 300 setiap tahun, dan dalam 50 tahun Anda memiliki Rp 30.000.

Jika bunga majemuk, pada akhir 50 tahun Anda memiliki $(1 + 0{,}02)^{50}$ — yang hanya sedikit lebih dari Rp 40.200, dengan asumsi bahwa bank mengumpulkan bunga setahun sekali.

Tentu saja, jika bank melipatgandakan bunga dua kali setahun, setiap pembayaran adalah Rp 150, dan setelah 50 tahun bank telah melipatgandakannya 100 kali. Itu memberi Anda $(1 + 0{,}01)^{100}$, atau lebih dari Rp 40.500. Bagaimana dengan peracikan empat kali setahun? Setelah 50 tahun — 200 peracikan — Anda memiliki $(1 + 0.005)^{200}$, yang menghasilkan jangan-habiskan- jumlah all-in-one-place Rp 40.650 dan sedikit lebih banyak lagi.

Berfokus pada "sedikit lagi" dan "sedikit lagi," dan membawanya ke ekstrem, setelah 100.000 peracikan, Anda memiliki $2.718268. Setelah 100 juta, Anda memiliki $2.718282. Jika Anda bisa membuat bank menggandakan lebih banyak lagi dalam 50 tahun itu, jumlah uang Anda mendekati batas — jumlah yang sangat dekat dengannya, tetapi tidak pernah benar-benar tercapai. Batas tersebut adalah e.

Cara saya mengatur contohnya, aturan untuk menghitung jumlahnya adalah:

$$\left(1+\left(1/n\right)\right)^n$$

di mana n mewakili jumlah pembayaran. Dua sen adalah 1/50 dolar dan saya menentukan 50 tahun — 50 pembayaran. Kemudian saya menentukan dua pembayaran setahun (dan pembayaran setiap tahun harus ditambahkan hingga 2 persen) sehingga dalam 50 tahun Anda memiliki 100 pembayaran sebesar 1/100 dolar, dan seterusnya.

Untuk melihat konsep ini dalam tindakan,

```
x <- c(seq(1,10,1),50,100,200,500,1000,10000,100000000)
> y <- (1+(1/x))^x
> data.frame(x,y)
       x        y
1  1e+00 2.000000
2  2e+00 2.250000
3  3e+00 2.370370
4  4e+00 2.441406
5  5e+00 2.488320
6  6e+00 2.521626
7  7e+00 2.546500
8  8e+00 2.565785
9  9e+00 2.581175
10 1e+01 2.593742
11 5e+01 2.691588
12 1e+02 2.704814
13 2e+02 2.711517
14 5e+02 2.715569
15 1e+03 2.716924
16 1e+04 2.718146
17 1e+08 2.718282
```

Jadi e dikaitkan dengan pertumbuhan. Nilainya adalah 2.718282 . . . Tiga titik berarti Anda tidak pernah mendapatkan nilai yang tepat (seperti , konstanta yang memungkinkan Anda

menemukan luas lingkaran). Nomor e muncul di semua jenis tempat. Ada dalam rumus untuk distribusi normal (bersama dengan ; lihat Bab 8), dan ada dalam distribusi yang saya bahas di Bab 18 dan di Lampiran A). Banyak fenomena alam yang berkaitan dengan e.

Sangat penting bahwa para ilmuwan, matematikawan, dan analis bisnis menggunakannya sebagai dasar untuk logaritma. Logaritma ke basis e disebut logaritma natural. Dalam banyak buku teks, logaritma natural disingkat ln. Di R, itu log. Tabel 16.1 menyajikan beberapa perbandingan (dibulatkan hingga tiga tempat desimal) antara logaritma umum dan logaritma natural.

Tabel 16.1 Beberapa Logaritma Umum (Log10) dan Logaritma Natural (Log)

Jumlah	Log10	Log
e	0.434	1.000
10	1.000	2.303
50	1.699	3.912
100	2.000	4.605
453	2.656	6.116
1000	3.000	6.908

Satu hal lagi: Dalam banyak rumus dan persamaan, sering kali e harus dipangkatkan. Terkadang pangkat adalah ekspresi matematika yang cukup rumit. Karena superskrip biasanya dicetak dalam font kecil, dapat menjadi beban jika harus terus-menerus membacanya. Untuk meringankan kelelahan mata, matematikawan telah menemukan notasi khusus: exp. Setiap kali Anda melihat exp diikuti oleh sesuatu dalam tanda kurung, itu berarti menaikkan e ke pangkat apa pun yang ada di dalam tanda kurung. Sebagai contoh,

$$\exp(1.6) = e^{1.6} = 4.953032$$

Fungsi exp() R melakukan perhitungan itu untuk Anda:

```
> exp(1.6)
[1] 4.953032
```

Menerapkan fungsi exp() dengan logaritma natural seperti mencari antilog dengan logaritma umum. Berbicara tentang meningkatkan e, ketika eksekutif di Google, Inc., mengajukan IPO, mereka mengatakan ingin mengumpulkan $2.718.281.828, yang merupakan e kali satu miliar dolar dibulatkan ke dolar terdekat. Dan sekarang . . . kembali ke regresi lengkung.

16.3 REGRESI DAYA

Ahli biologi telah mempelajari hubungan timbal balik antara ukuran dan berat bagian tubuh. Salah satu hubungan yang menarik adalah hubungan antara berat badan dan berat otak. Salah satu cara untuk mempelajari ini adalah dengan menilai hubungan antar spesies

yang berbeda. Secara intuitif, sepertinya hewan yang lebih berat seharusnya memiliki otak yang lebih berat — tetapi apa sifat sebenarnya dari hubungan itu?

Dalam paket MASS, Anda akan menemukan kerangka data yang disebut Hewan yang berisi bobot tubuh (dalam kilogram) dan bobot otak (dalam gram) dari 28 spesies. (Untuk mengikuti, pada tab Paket klik Instal. Kemudian, di kotak dialog Instal Paket, ketik MASS. Ketika MASS muncul di tab Paket, pilih kotak centangnya).
Enam baris pertama Hewan adalah:

```
> head(Animals)
                  body brain
Mountain beaver    1.35   8.1
Cow              465.00 423.0
Grey wolf         36.33 119.5
Goat              27.66 115.0
Guinea pig         1.04   5.5
Dipliodocus    11700.00  50.0
```

Pernahkah Anda melihat dipliodokus? Tidak? Di luar museum sejarah alam, tidak ada orang lain juga. Selain dinosaurus di baris 6 ini, Hewan memiliki triceratops di baris 16 dan brachiosaurus di baris 26. Di sini, saya akan menunjukkan kepada Anda:

```
> Animals[c(6,16,26),]
             body brain
Dipliodocus 11700  50.0
Triceratops  9400  70.0
Brachiosaurus 87000 154.5
```

Untuk membatasi pekerjaan Anda pada spesies hidup, buat

```
> Animals.living <- Animals[-c(6,16,26),]
```

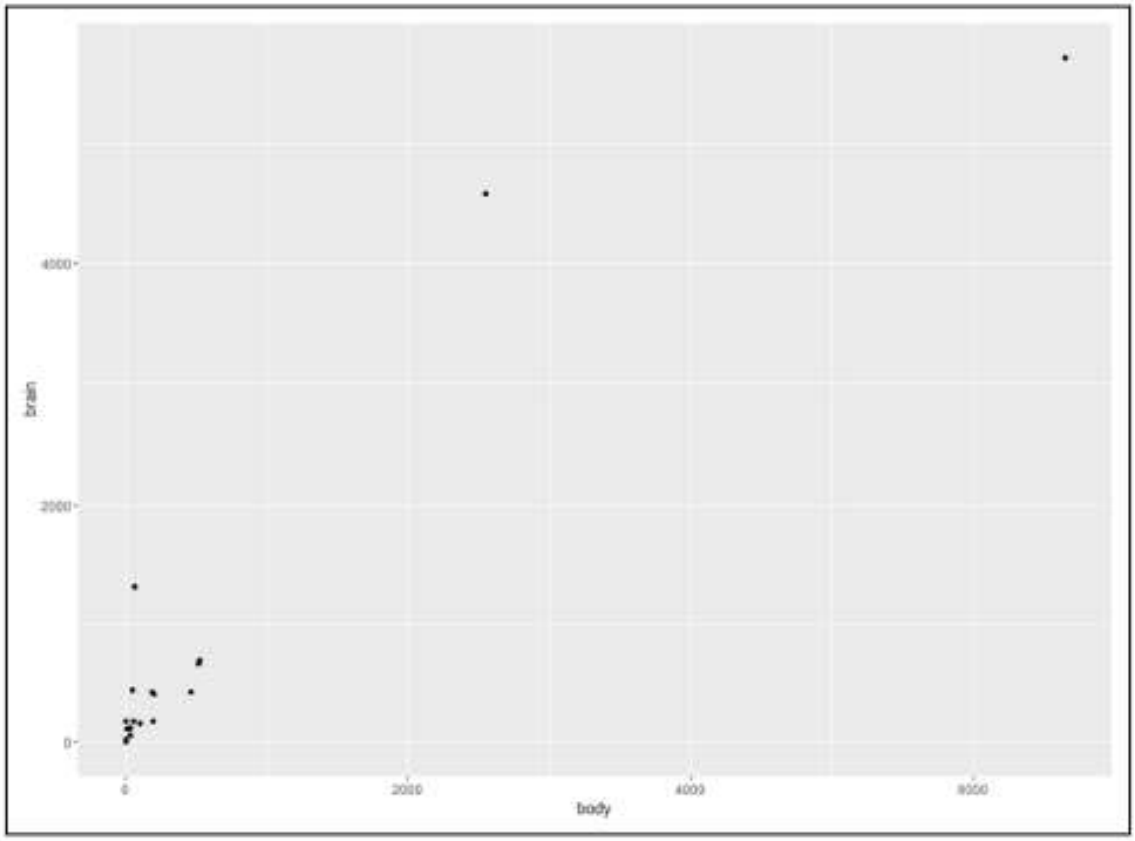
yang menyebabkan ketiga dinosaurus tersebut menghilang dari data frame sama pasti dengan menghilangnya mereka dari muka bumi.
Mari kita lihat poin datanya. Cuplikan kode ini

```
ggplot(Animals.living, aes(x=body, y=brain))+
  geom_point()
```

menghasilkan Gambar 16.2. Perhatikan bahwa idenya adalah menggunakan berat badan untuk memprediksi berat otak.

Tidak terlihat seperti hubungan linier, bukan? Sebenarnya tidak. Hubungan dalam bidang ini sering berbentuk

$$y' = ax^b$$



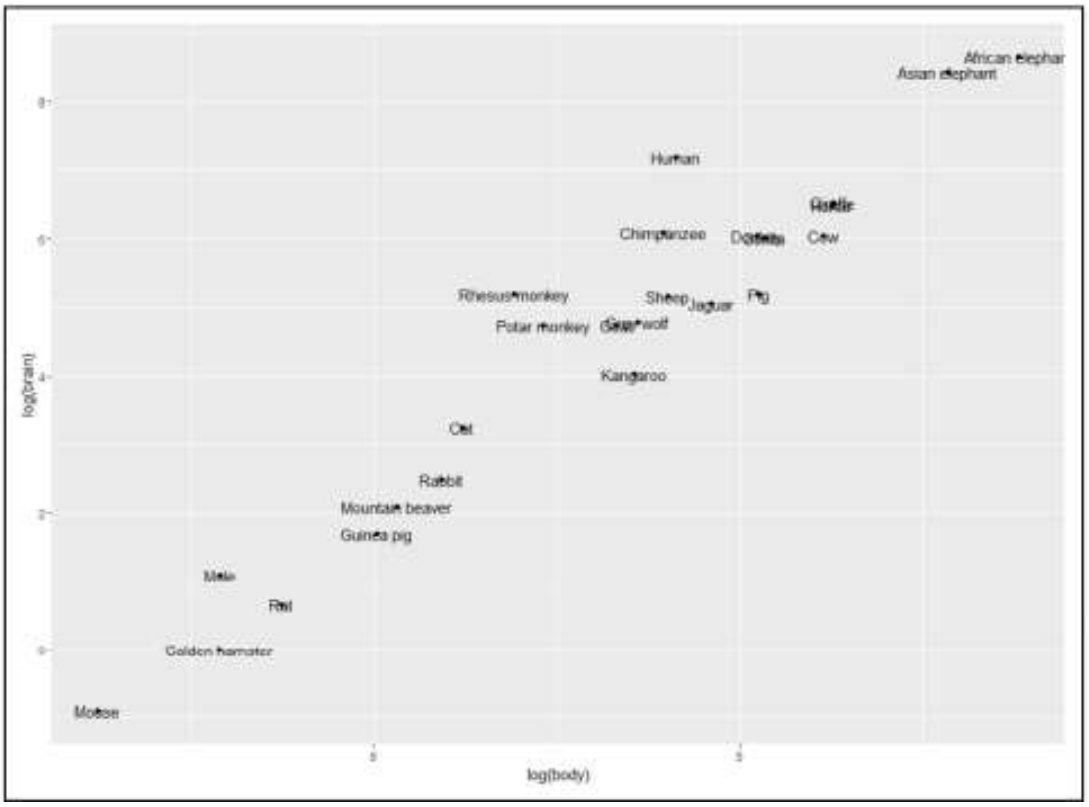
Gambar 16.2 Hubungan antara berat badan dan berat otak untuk 25 spesies hewan.

Karena variabel bebas (prediktor) x (berat badan, dalam hal ini) dipangkatkan, model jenis ini disebut regresi daya. R tidak memiliki fungsi khusus untuk membuat model regresi daya. Fungsi lm()-nya membuat model linier, seperti yang dijelaskan dalam Bab 14. Tetapi Anda dapat menggunakan lm() dalam situasi ini jika Anda dapat mengubah data sehingga hubungan antara berat badan yang diubah dan berat otak yang diubah adalah linier. Dan inilah mengapa saya memberi tahu Anda tentang logaritma.

Anda dapat "memlinearisasikan" scatterplot dengan bekerja dengan logaritma dari berat badan dan logaritma dari berat otak. Berikut beberapa kode untuk melakukan hal itu. Untuk ukuran yang baik, saya akan memasukkan nama hewan untuk setiap titik data:

```
ggplot(Animals.living, aes(x=log(body), y=log(brain)))+
  geom_point()+
  geom_text(aes(label=rownames(Animals.living)))
```

Gambar 16.3 menunjukkan hasilnya.



Gambar 16.3 Hubungan antara log berat badan dan log berat otak untuk 25 spesies hewan.

Saya heran dengan kedekatan keledai dan gorila, tapi mungkin konsep gorila saya berasal dari King Kong. Kejutan lainnya adalah kedekatan kuda dan jerapah.

Bagaimanapun, Anda dapat memasukkan garis regresi melalui sebar ini. Berikut kode plot dengan garis dan tanpa nama hewan:

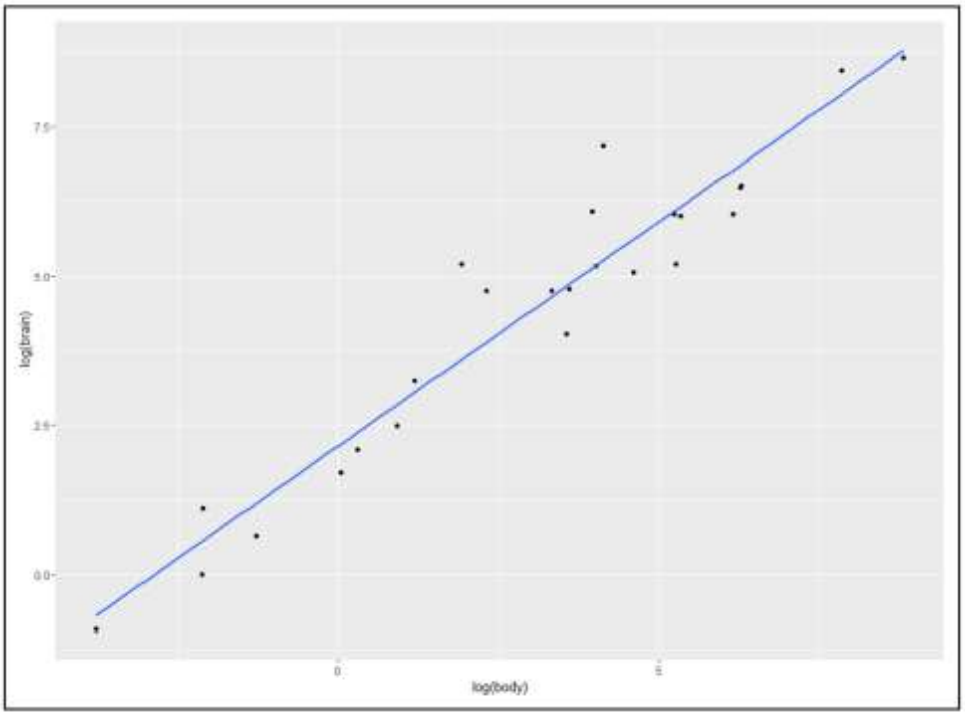
```
ggplot(Animals.living, aes(x=log(body), y=log(brain)))+
  geom_point()+
  geom_smooth(method = "lm",se=FALSE)
```

Argumen pertama dalam pernyataan terakhir (metode = "lm") cocok dengan garis regresi ke titik data. Argumen kedua (se=FALSE) mencegah ggplot memplot interval kepercayaan 95 persen di sekitar garis regresi. Baris kode ini menghasilkan Gambar 16.4.

Prosedur ini — bekerja dengan log masing-masing variabel dan kemudian memasang garis regresi — adalah persis apa yang harus dilakukan dalam kasus seperti ini. Berikut analisisnya:

```
powerfit <- lm(log(brain) ~ log(body), data = Animals.living)
```

Seperti biasa, lm() menunjukkan model linier, dan variabel dependen berada di sisi kiri tilde (~) dengan variabel prediktor di sisi kanan. Setelah menjalankan analisis,



Gambar 16.4 Hubungan antara log berat badan dan log berat otak untuk 25 spesies hewan, dengan garis regresi.

```
> summary(powerfit)

Call:
lm(formula = log(brain) ~ log(body), data = Animals.living)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9125 -0.4752 -0.1557  0.1940  1.9303

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.15041    0.20060   10.72 2.03e-10 ***
log(body)    0.75226    0.04572   16.45 3.24e-14 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7258 on 23 degrees of freedom
Multiple R-squared:  0.9217,  Adjusted R-squared:  0.9183
F-statistic: 270.7 on 1 and 23 DF,  p-value: 3.243e-14
```

Nilai F yang tinggi (270,7) dan nilai p yang sangat rendah menunjukkan bahwa model tersebut cocok. Koefisien memberitahu Anda bahwa dalam bentuk logaritma, persamaan regresi adalah:

$$\log(y') = \log(a + bx)$$

$$\log(\text{brainweight}') = \log(2.15041 + (.75226 \times \text{bodyweight}))$$

Untuk persamaan regresi daya, Anda harus mengambil antilog dari kedua sisi. Seperti yang saya sebutkan sebelumnya, ketika Anda bekerja dengan logaritma natural, itu sama dengan menerapkan fungsi exp():

$$\exp\left(\log\left(y'\right)\right) = \exp\left(\log\left(a + bx\right)\right)$$

$$y' = \exp\left(a\right)x^{b}$$

$$\text{brainweight}' = \exp(2.15041) \times \text{bodyweight}^{.75226}$$

$$\text{brainweight}' = 8.588397 \times \text{bodyweight}^{.75226}$$

Semua ini sesuai dengan apa yang saya katakan sebelumnya dalam bab ini:

- Menambahkan logaritma angka sama dengan mengalikan angka.
- Mengalikan logaritma x dengan b sama dengan menaikkan x ke pangkat b.

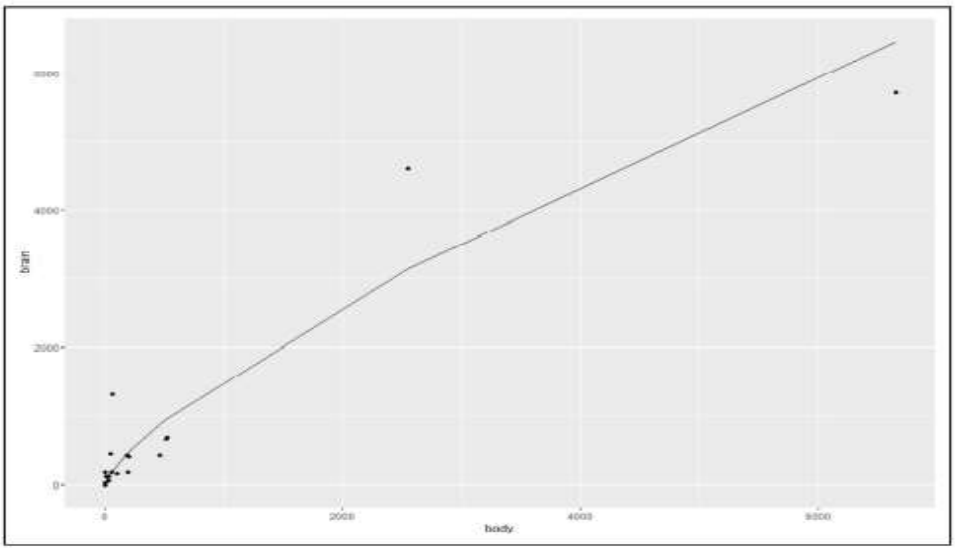
Berikut cara menggunakan R untuk menemukan exp dari intersep:

```
> a <- exp(powerfit$coefficients[1])
> a
(Intercept)
   8.588397
```

Anda dapat memplot persamaan regresi daya sebagai kurva di scatterplot asli:

```
ggplot(Animals.living, aes(x=body, y=brain))+
  geom_point()+
  geom_line(aes(y=exp(powerfit$fitted.values)))
```

Pernyataan terakhir itu adalah tujuan bisnis, tentu saja: powerfit$fitted.values berisi bobot otak yang diprediksi dalam bentuk logaritmik, dan menerapkan exp() ke nilai-nilai itu mengubah prediksi tersebut ke satuan ukuran asli. Anda memetakannya ke y untuk memposisikan kurva. Gambar 16.5 menunjukkan plot.



Gambar 16.5 Plot asli bobot otak dan bobot tubuh dari 25 spesies, dengan kurva regresi daya.

16.4 REGRESI EKSPONENSIAL

Seperti yang saya sebutkan sebelumnya, e menggambarkan proses di berbagai bidang. Beberapa dari proses tersebut, seperti bunga majemuk, melibatkan pertumbuhan. Lainnya melibatkan pembusukan. Berikut ini contohnya. Jika Anda pernah menuangkan segelas bir dan membiarkannya berdiri, Anda mungkin memperhatikan bahwa kepalanya semakin kecil (dengan kata lain "meluruh") seiring berjalannya waktu. Anda belum melakukannya? Oke. Silakan dan tuangkan yang tinggi dan dingin dan tonton selama enam menit. Aku akan menunggu.

Dan kami kembali. Apakah saya benar? Perhatikan bahwa saya tidak meminta Anda untuk mengukur tinggi kepala saat membusuk. Fisikawan Arnd Leike melakukannya untuk kami untuk tiga merek bir. Dia mengukur tinggi kepala setiap 15 detik dari 0 hingga 120 detik setelah menuangkan bir, kemudian setiap 30 detik dari 150 detik menjadi 240 detik, dan akhirnya, pada 300 detik dan 360 detik. (Dalam semangat sains yang sebenarnya, dia kemudian meminum birnya.) Berikut adalah interval sebagai vektor:

```
seconds.after.pour <- c(seq(0,120,15), seq(150,240,30),
  c(300,360))
```

dan berikut adalah tinggi kepala yang diukur (dalam sentimeter) untuk salah satu merek tersebut:

```
head.cm <- c(17, 16.1, 14.9, 14, 13.2, 12.5, 11.9, 11.2,
             10.7, 9.7, 8.9, 8.3, 7.5, 6.3, 5.2)
```

Saya menggabungkan vektor-vektor ini ke dalam bingkai data:

```
beer.head <- data.frame(seconds.after.pour,head.cm)
```

Mari kita lihat seperti apa plotnya. Cuplikan kode ini

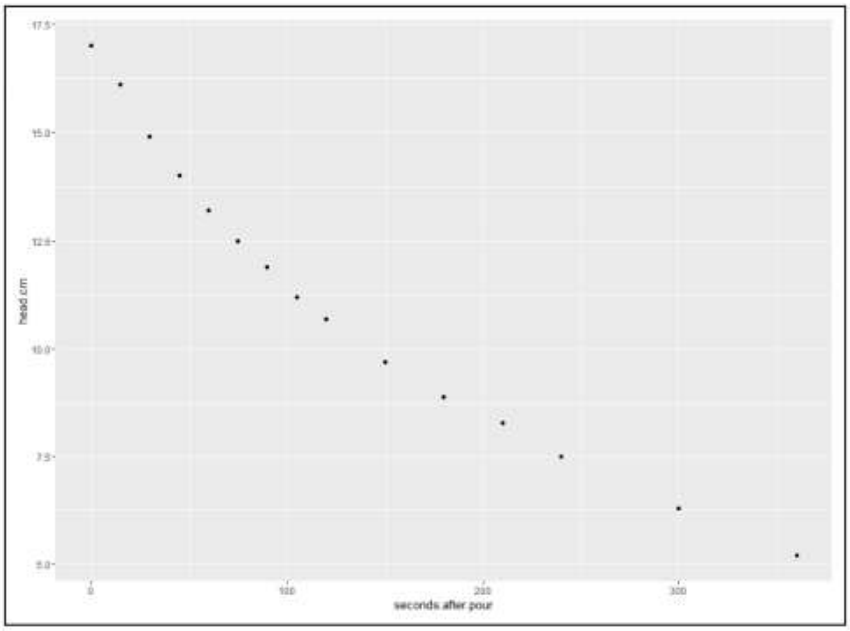
```
ggplot(beer.head, aes(x=seconds.after.pour,y=head.cm))+
  geom_point()
```

menghasilkan Gambar 16.6.

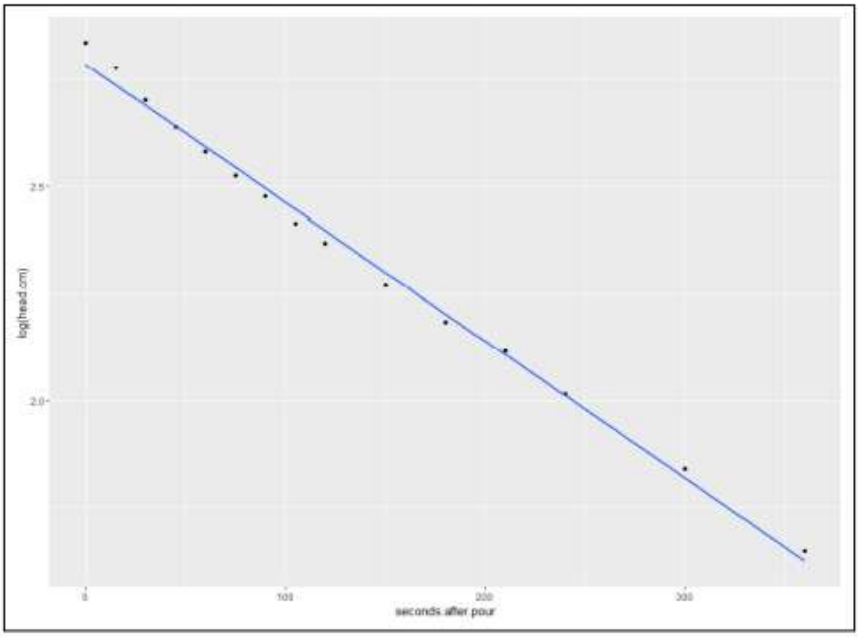
Yang ini menangis (dalam birnya?) Untuk model lengkung, bukan? Salah satu cara untuk linierisasi plot (sehingga Anda dapat menggunakan lm() untuk membuat model) adalah bekerja dengan log variabel y:

```
ggplot(beer.head, aes(x=
          seconds.after.pour,y=log(head.cm)))+
  geom_point()+
  geom_smooth(method="lm",se=FALSE)
```

Pernyataan terakhir menambahkan garis regresi (metode = “lm”) dan tidak menarik interval kepercayaan di sekitar garis (se = FALSE). Anda dapat melihat semua ini pada Gambar 16.7.



Gambar 16.6 Bagaimana bir setinggi kepala (head.cm) meluruh seiring waktu.



Gambar 16.7 Bagaimana log(head.cm) meluruh dari waktu ke waktu, termasuk garis regresi.

Seperti pada bagian sebelumnya, membuat plot ini menunjukkan cara untuk melakukan analisis. Persamaan umum untuk model yang dihasilkan adalah:

$$y' = ae^{bx}$$

Karena variabel prediktor muncul dalam eksponen (dimana e dinaikkan), ini disebut regresi eksponensial. Dan inilah cara melakukan analisis:

```
expfit <- lm(log(head.cm) ~ seconds.after.pour,
             data = beer.head)
```

Sekali lagi, lm() menunjukkan model linier, dan variabel dependen berada di sisi kiri tilde (~), dengan variabel prediktor di sisi kanan. Setelah menjalankan analisis,

```
> summary(expfit)

Call:
lm(formula = log(head.cm) ~ seconds.after.pour, data = beer.
          head)

Residuals:
      Min        1Q    Median        3Q       Max
-0.031082 -0.019012 -0.001316  0.017338  0.047806

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.785e+00  1.110e-02  250.99  < 2e-16 ***
seconds.after.pour -3.223e-03  6.616e-05  -48.72  4.2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02652 on 13 degrees of freedom
Multiple R-squared:  0.9946,  Adjusted R-squared:  0.9941
F-statistic:  2373 on 1 and 13 DF,  p-value: 4.197e-16
```

Nilai F dan p menunjukkan bahwa model ini sangat cocok secara fenomenal. R-kuadrat adalah salah satu yang tertinggi yang pernah Anda lihat. Faktanya, Arnd melakukan semua ini untuk menunjukkan kepada murid-muridnya bagaimana proses eksponensial bekerja. [Jika Anda ingin melihat datanya untuk dua merek lainnya, lihat Leike, A. (2002), "Demonstration of the exponential decay law using beer froth," European Journal of Physics, 23(1), 21–26].
Berdasarkan koefisiennya, persamaan regresi dalam bentuk logaritma adalah:

$$\log(y') = a + bx$$

$$\log(\text{head.cm}') = 2.785 + \left((-.003223) \times \text{seconds.after.pour}\right)$$

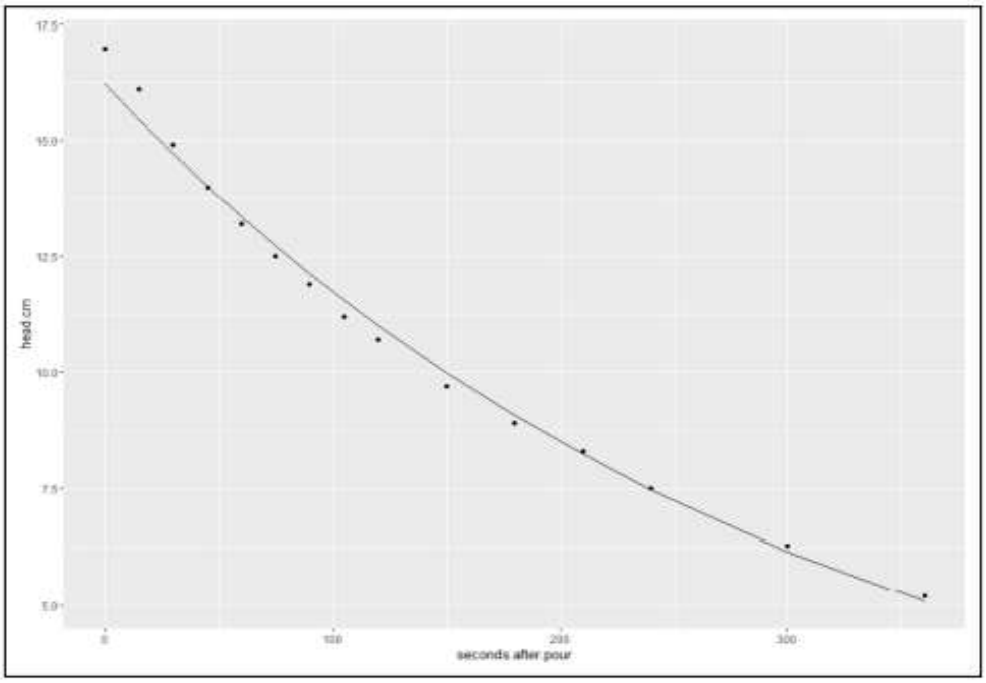
Untuk persamaan regresi eksponensial, Anda harus mengambil eksponensial dari kedua sisi — dengan kata lain, Anda menerapkan fungsi exp() :

$$\exp\left(\log\left(y'\right)\right)=\exp\left(a+bx\right)$$
$$y'=\exp\left(a\right)e^{bx}$$
$$\text{head.cm}'=\exp(2.785)\times e^{-.003223\text{seconds.after.pour}}$$
$$\text{head.cm}'=16.20642\times e^{-.003223\text{seconds.after.pour}}$$

Analog dengan apa yang Anda lakukan di bagian sebelumnya, Anda dapat memplot persamaan regresi eksponensial sebagai kurva di scatterplot asli:

```
ggplot(beer.head, aes(x= seconds.after.pour,y=head.cm))+
  geom_point()+
  geom_line(aes(y=exp(expfit$fitted.values)))
```

Dalam pernyataan terakhir, expfit$fitted.values berisi prediksi ketinggian bir dalam bentuk logaritmik, dan menerapkan exp() ke nilai tersebut mengubah prediksi tersebut ke satuan ukuran asli. Memetakan mereka ke posisi y kurva. Gambar 16.8 menunjukkan plot.



Gambar 16-8: Peluruhan head.cm dari waktu ke waktu, dengan kurva regresi eksponensial.

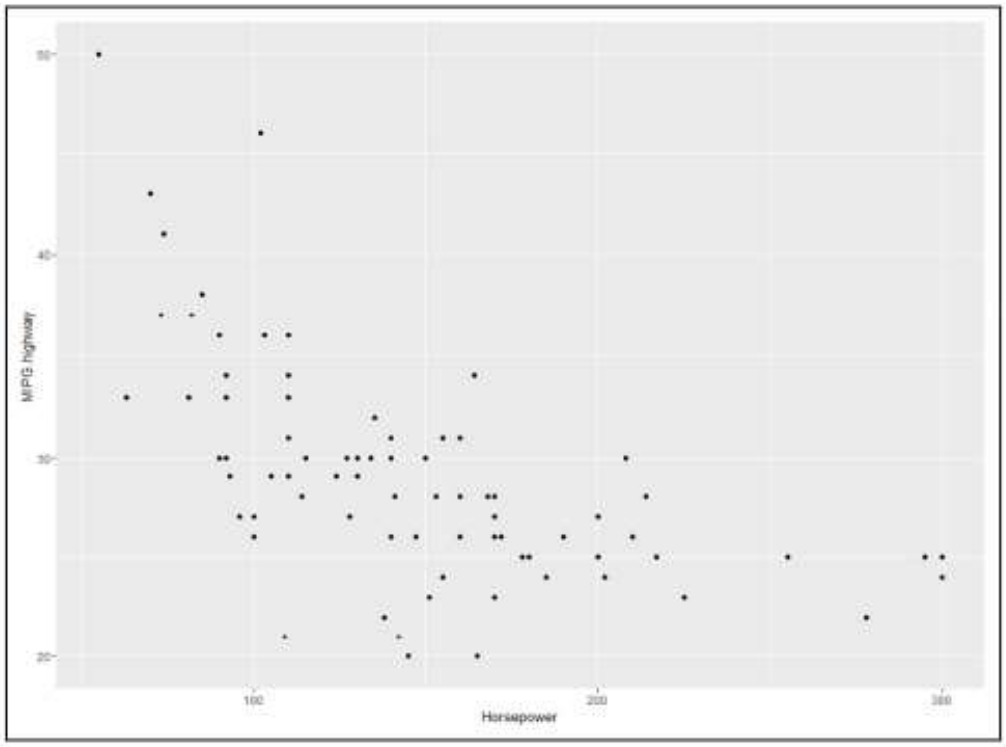
16.5 REGRESI LOGARITMA

Dalam dua bagian sebelumnya, saya menjelaskan bagaimana analisis regresi daya bekerja dengan log variabel x dan log variabel y, dan bagaimana eksponensial analisis regresi bekerja dengan log hanya variabel y. Seperti yang Anda bayangkan, satu lagi kemungkinan analitik tersedia untuk Anda — bekerja hanya dengan log dari variabel-x. Persamaan modelnya terlihat seperti ini:

$$y'=a+b\log\left(x\right)$$

Karena logaritma diterapkan pada variabel prediktor, ini disebut regresi logaritmik. Berikut adalah contoh yang menggunakan data frame Cars93 dalam paket MASS. (Pastikan Anda menginstal paket MASS. Pada tab Packages, cari kotak centang MASS dan jika tidak dipilih, klik.) Kerangka data ini, yang ditampilkan secara menonjol di Bab 3, menyimpan data tentang sejumlah variabel untuk 93 mobil dalam model tahun 1993. Di sini, saya fokus pada hubungan antara Horsepower (variabel x) dan MPG.highway (variabel y).
Ini adalah kode untuk membuat scatterplot pada Gambar 16.9:

```
ggplot(Cars93, aes(x=Horsepower,y=MPG.highway))+
  geom_point()
```



Gambar 16.9 MPG.highway dan Horsepower dalam kerangka data Cars93.

Untuk contoh ini, linearisasikan plot dengan mengambil log Horsepower. Dalam plot, sertakan garis regresi, dan inilah cara menggambarnya:

```
ggplot(Cars93, aes(x=log(Horsepower),y=MPG.highway))+
  geom_point()+
  geom_smooth(method="lm",se=FALSE)
```

Gambar 16.10 menunjukkan hasilnya.
Dengan log(Horsepower) sebagai variabel x, analisisnya adalah:

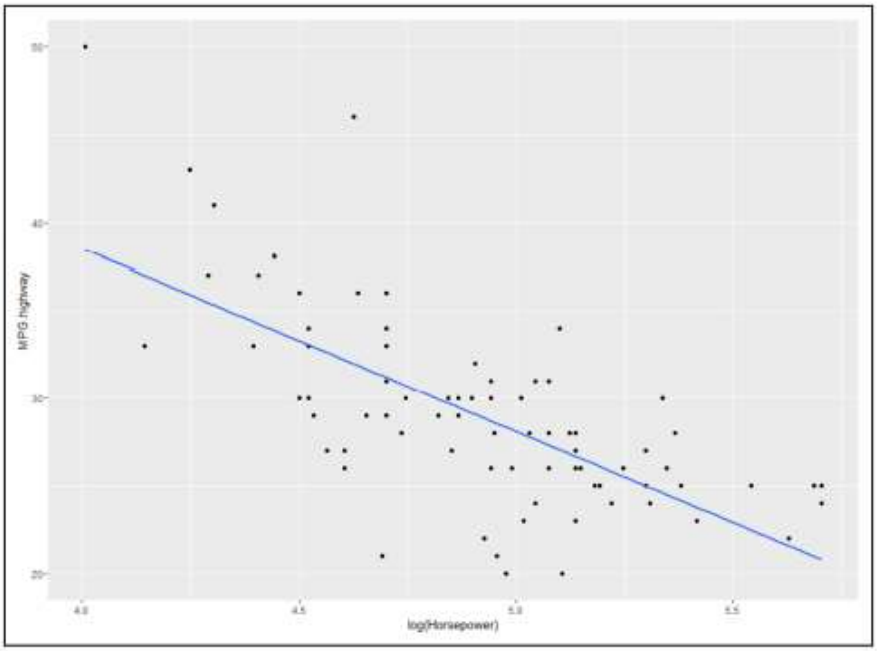
```
logfit <- lm(MPG.highway ~ log(Horsepower), data=Cars93)
```

Setelah melakukan analisis itu, summary() memberikan detailnya:

```
> summary(logfit)

Call:
lm(formula = MPG.highway ~ log(Horsepower), data = Cars93)

Residuals:
     Min       1Q   Median       3Q      Max
-10.3109  -2.2066  -0.0707   2.0031  14.0002

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       80.003      5.520  14.493  < 2e-16 ***
log(Horsepower)  -10.379      1.122  -9.248 9.55e-15 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.849 on 91 degrees of freedom
Multiple R-squared:  0.4845,  Adjusted R-squared:  0.4788
F-statistic: 85.53 on 1 and 91 DF,  p-value: 9.548e-15
```

Nilai F yang tinggi dan nilai p yang sangat rendah menunjukkan kecocokan yang sangat baik. Dari koefisien, persamaan regresinya adalah:

$$\text{MPG.highway}' = 80.03 - 10.379\log(\text{Horsepower})$$

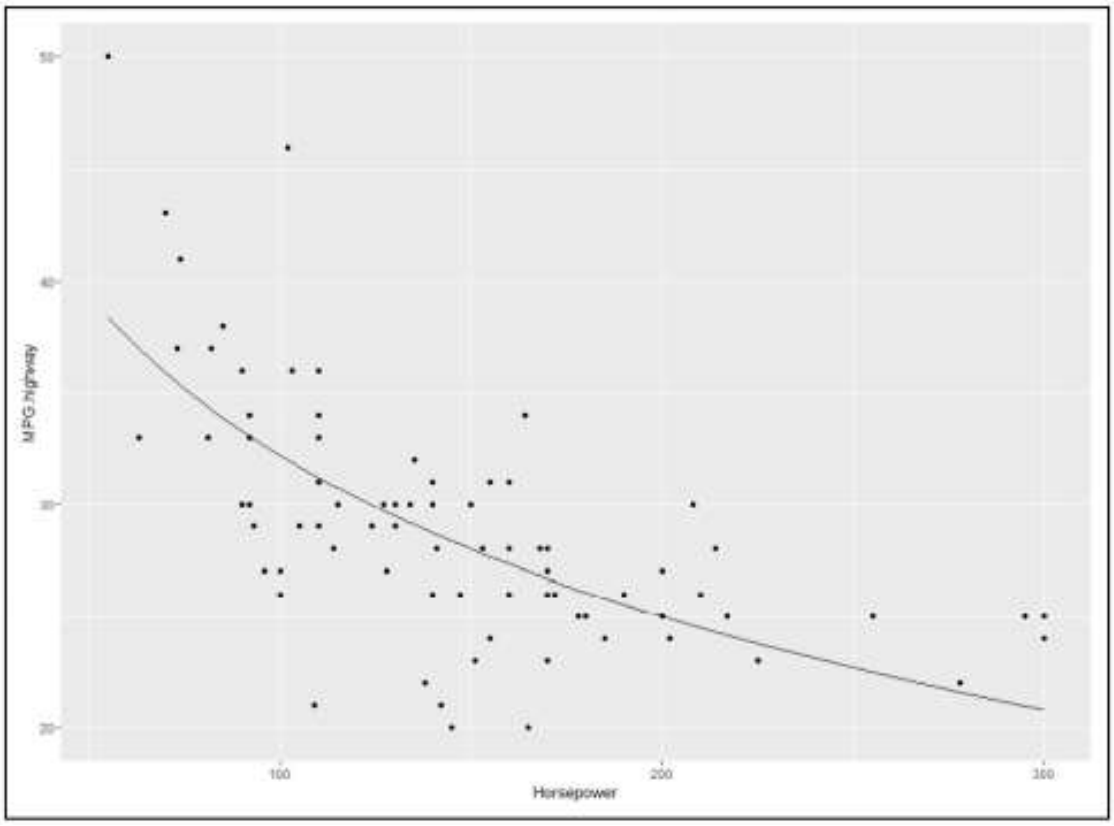


Gambar 16.10 MPG.highway dan Log(Horsepower) di Cars93, bersama dengan garis regresi.

Seperti pada bagian sebelumnya, saya memplot kurva regresi di plot asli:

```
ggplot(Cars93, aes(x=Horsepower,y=MPG.highway))+
  geom_point()+
  geom_line(aes(y=logfit$fitted.values))
```

Gambar 16.11 menunjukkan plot dengan kurva regresi.



Gambar 16.11 MPG.highway dan Horsepower, dengan kurva regresi logaritmik.

16.6 REGRESI POLINOMIAL: KEKUATAN LEBIH TINGGI

Dalam semua jenis regresi yang saya jelaskan sebelumnya dalam bab ini, modelnya adalah garis atau kurva yang tidak berubah arah. Namun, dimungkinkan untuk membuat model yang menggabungkan perubahan arah. Ini adalah provinsi regresi polinomial. Saya menyentuh perubahan arah di Bab 12, dalam konteks analisis tren. Untuk memodelkan satu perubahan arah, persamaan regresi harus memiliki suku x yang dipangkatkan ke dua:

$$y' = a + b_1 x + b_2 x^2$$

Untuk memodelkan dua perubahan arah, persamaan regresi harus memiliki suku x yang dipangkatkan ketiga:

$$y' = a + b_1 x + b_2 x^2 + b_3 x^3$$

Dan seterusnya.

Saya menggambarkan regresi polinomial dengan kerangka data lain dari paket MASS. (Pada tab Paket, temukan MASS. Jika kotak centangnya tidak dipilih, klik.) Kerangka data ini disebut Boston. Ini menyimpan data tentang nilai perumahan di pinggiran kota Boston. Di antara 14 variabelnya adalah rm (jumlah kamar di hunian) dan medv (nilai median hunian). Saya fokus pada dua variabel dalam contoh ini, dengan rm sebagai variabel prediktor.
Mulailah dengan membuat scatterplot dan garis regresi:

```
ggplot(Boston, aes(x=rm,y=medv))+
  geom_point()+
  geom_smooth(method=lm, se=FALSE)
```

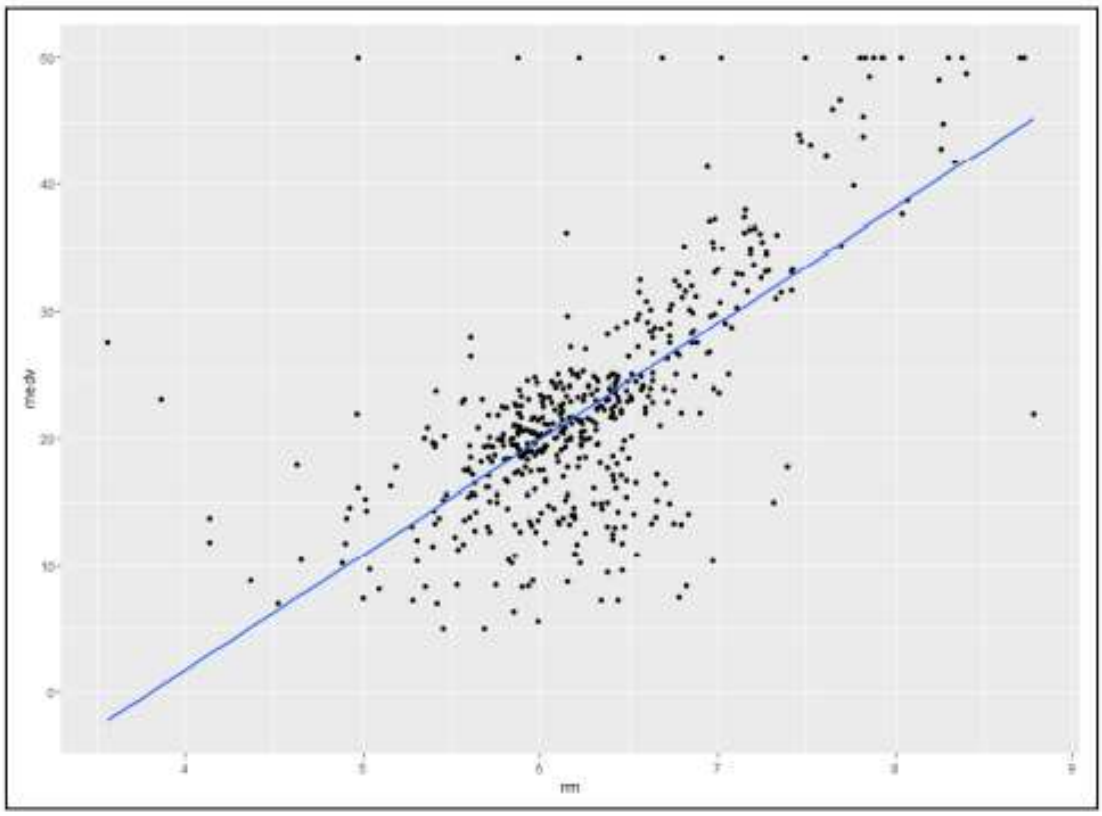
Gambar 16.12 menunjukkan apa yang dihasilkan kode ini. Model regresi linier adalah:

```
linfit <- lm(medv ~ rm, data=Boston)
```

```
> summary(linfit)

Call:
lm(formula = medv ~ rm, data = Boston)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-23.346  -2.547   0.090   2.986  39.433

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,  Adjusted R-squared:  0.4825
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

Nilai F dan p menunjukkan bahwa ini cocok. R-kuadrat memberitahu Anda bahwa sekitar 48 persen dari SSTotal untuk medv terikat dalam hubungan antara rm dan medv. (Lihat Bab 15 jika kalimat terakhir itu terdengar asing).



Gambar 16.12: Scatterplot nilai median (medv) vs kamar (rm) dalam kerangka data Boston, dengan garis regresi.

Koefisien memberitahu Anda bahwa model linier adalah:

$$\text{medv}' = -34.671 + 9.102\text{rm}$$

Tapi mungkin model dengan perubahan arah memberikan kecocokan yang lebih baik. Untuk mengatur ini di R, buat variabel baru rm2 — yang hanya rm kuadrat:

```
rm2 <- Boston$rm^2
```

Sekarang perlakukan ini sebagai analisis regresi berganda dengan dua variabel prediktor: rm dan rm2:

```
polyfit2 <-lm(medv ~ rm + rm2, data=Boston)
```

Anda tidak bisa melanjutkan dan menggunakan rm^2 sebagai variabel prediktor kedua: lm() tidak akan bekerja dengannya dalam bentuk itu.
Setelah Anda menjalankan analisis, berikut adalah detailnya:

```
> summary(polyfit2)
```

```
Call:
lm(formula = medv ~ rm + rm2, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-35.769  -2.752   0.619   3.003  35.464

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  66.0588    12.1040   5.458 7.59e-08 ***
rm          -22.6433     3.7542  -6.031 3.15e-09 ***
rm2           2.4701     0.2905   8.502  < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.193 on 503 degrees of freedom
Multiple R-squared:  0.5484,  Adjusted R-squared:  0.5466
F-statistic: 305.4 on 2 and 503 DF,  p-value: < 2.2e-16
```

Sepertinya lebih cocok daripada model linier. F-statistik di sini lebih tinggi, dan kali ini R-kuadrat memberitahu Anda bahwa hampir 55 persen dari SSTotal untuk medv adalah karena hubungan antara medv dan kombinasi rm dan rm^2. Peningkatan F dan R-kuadrat dikenakan biaya — model kedua memiliki 1 df lebih sedikit (503 versus 504).
Koefisien menunjukkan bahwa persamaan regresi polinomial adalah:

$$\text{medv}' = 66.0588 - 22.6433\text{rm} + 2.4701\text{rm}^2$$

Apakah sepadan dengan upaya untuk menambahkan rm^2 ke model? Untuk mengetahuinya, saya menggunakan anova() untuk membandingkan model linier dengan model polinomial:

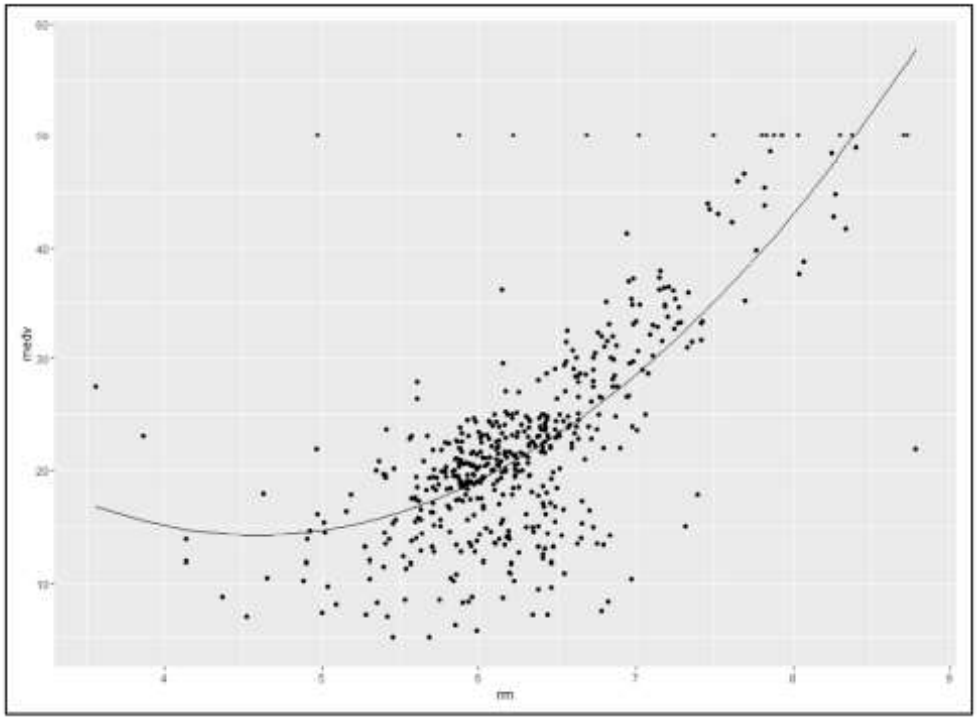
```
> anova(linfit,polyfit2)
Analysis of Variance Table

Model 1: medv ~ rm
Model 2: medv ~ rm + rm2
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    504 22062
2    503 19290  1    2772.3 72.291 < 2.2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Rasio F yang tinggi (72,291) dan Pr(>F) yang sangat rendah menunjukkan bahwa menambahkan rm^2 adalah ide yang bagus.

Berikut kode untuk scatterplot, bersama dengan kurva untuk model polinomial:

```
ggplot(Boston, aes(x=rm,y=medv))+
  geom_point()+
  geom_line(aes(y=polyfit2$fitted.values))
```

Nilai prediksi untuk model polinomial berada di polyfit2$fitted.values, yang Anda gunakan dalam pernyataan terakhir untuk memposisikan kurva regresi pada Gambar 16.13.



Gambar 16.13 Scatterplot nilai median (medv) versus kamar (rm) dalam kerangka data Boston, dengan kurva regresi polinomial.

Kurva pada gambar menunjukkan sedikit tren penurunan nilai hunian saat kamar meningkat dari kurang dari empat menjadi sekitar 4,5, dan kemudian kurva tren naik lebih tajam.

16.7 MODEL MANA YANG HARUS ANDA GUNAKAN?

Saya menyajikan berbagai model regresi dalam bab ini. Memutuskan yang tepat untuk data Anda tidak selalu mudah. Satu jawaban dangkal mungkin mencoba masing-masing dan melihat mana yang menghasilkan F dan R-kuadrat tertinggi.

Kata operasi dalam kalimat terakhir adalah "dangkal." Pilihan model harus bergantung pada pengetahuan Anda tentang domain dari mana data berasal dan proses dalam domain itu. Jenis regresi mana yang memungkinkan Anda merumuskan teori tentang apa yang mungkin terjadi dalam data? Misalnya, dalam contoh Boston, model polinomial menunjukkan bahwa nilai tempat tinggal sedikit menurun seiring dengan meningkatnya jumlah kamar di

ujung bawah, dan kemudian nilainya terus meningkat seiring dengan bertambahnya jumlah kamar. Model linier tidak dapat membedakan tren seperti itu. Mengapa tren itu bisa terjadi? Bisakah Anda membuat teori? Apakah teorinya masuk akal?

Saya akan meninggalkan Anda dengan latihan. Ingat contoh mengikat tali sepatu di awal bab ini? Yang saya berikan hanyalah Gambar 16-1, tapi ini nomornya:

```
trials <-seq(1,18,1)
time.sec <- c(230, 140, 98, 75, 66, 54, 45, 31, 20, 15,
          10, 9, 9, 9, 8, 8, 8, 8)
```

Model apa yang bisa Anda buat? Dan bagaimana hal itu membantu Anda menjelaskan data?

BAGIAN 4
BEKERJA DENGAN PROBABILITAS
BAB 17
MEMPERKENALKAN PROBABILITAS

Probabilitas adalah dasar dari pengujian hipotesis dan statistik inferensial, jadi saya menggunakan konsep ini di seluruh buku ini. (Sepertinya waktu yang tepat untuk memperkenalkannya!) Sebagian besar waktu saya mewakili probabilitas sebagai proporsi area di bawah bagian dari distribusi. Sebagai contoh, probabilitas kesalahan Tipe I (juga dikenal sebagai) adalah area di bagian ekor dari distribusi normal standar, atau di bagian ekor dari distribusi t.

Saatnya untuk memeriksa probabilitas secara lebih rinci, termasuk variabel acak, permutasi, dan kombinasi. Saya menunjukkan kepada Anda beberapa dasar dan aplikasi probabilitas, dan kemudian saya fokus pada beberapa distribusi probabilitas spesifik dan juga memberi tahu Anda tentang beberapa fungsi R terkait probabilitas.

17.1 APA ITU PROBABILITAS?

Sebagian besar dari kita memiliki gagasan intuitif tentang probabilitas. Lempar koin yang adil, dan Anda memiliki peluang 50-50 untuk muncul. Lempar dadu yang adil (salah satu dari sepasang dadu) dan Anda memiliki peluang 1-dalam-6 bahwa dadu itu muncul dengan 2.

Jika Anda ingin lebih formal dalam definisi Anda, kemungkinan besar Anda akan mengatakan sesuatu tentang semua hal yang mungkin terjadi, dan proporsi hal-hal yang Anda pedulikan. Dua hal dapat terjadi ketika Anda melempar koin, dan jika Anda hanya peduli pada salah satu dari mereka (kepala), kemungkinan peristiwa itu terjadi adalah satu dari dua. Enam hal dapat terjadi ketika Anda melempar dadu, dan jika Anda hanya peduli pada salah satunya (2), peluang terjadinya peristiwa itu adalah satu dari enam.

Eksperimen, percobaan, peristiwa, dan ruang sampel

Ahli statistik dan orang lain yang bekerja dengan probabilitas merujuk pada proses seperti melempar koin atau melempar dadu sebagai eksperimen. Setiap kali Anda menjalani prosesnya, itu adalah cobaan. Ini mungkin tidak sesuai dengan definisi pribadi Anda tentang eksperimen (atau percobaan, dalam hal ini), tetapi bagi ahli statistik, eksperimen adalah proses apa pun yang menghasilkan salah satu dari setidaknya dua hasil berbeda (seperti kepala atau ekor).

Inilah bagian lain dari definisi eksperimen: Anda tidak dapat memprediksi hasilnya dengan pasti. Setiap hasil yang berbeda disebut hasil dasar. Letakkan sekelompok hasil dasar bersama-sama dan Anda memiliki sebuah acara. Misalnya, dengan sebuah dadu, hasil dasar 2, 4, dan 6 membentuk kejadian “bilangan genap”. Satukan semua kemungkinan hasil dasar dan Anda mendapatkan ruang sampel untuk diri Anda sendiri. Bilangan 1, 2, 3, 4, 5, dan 6

merupakan ruang sampel sebuah dadu. Kepala dan ekor membentuk ruang sampel untuk koin.

Ruang sampel dan probabilitas

Bagaimana peristiwa, hasil, dan ruang sampel berperan dalam probabilitas? Jika setiap hasil dasar dalam ruang sampel memiliki peluang yang sama, peluang suatu kejadian adalah

$$\text{pr}(\text{Event}) = \frac{\text{Number of Elementary Outcomes in the Event}}{\text{Number of Elementary Outcomes in the Sample Space}}$$

Jadi peluang munculnya sebuah dadu dan muncul bilangan genap adalah:

$$\text{pr}(\text{Even Number}) = \frac{\text{Number of Even-Numbered Elementary Outcomes}}{\text{Number of Possible Outcomes of a Die}} = \frac{3}{6} = .5$$

Jika hasil dasar tidak sama kemungkinannya, Anda menemukan probabilitas suatu peristiwa dengan cara yang berbeda. Pertama, Anda harus memiliki beberapa cara untuk menetapkan probabilitas untuk masing-masing. Kemudian Anda menjumlahkan peluang hasil dasar yang membentuk kejadian tersebut.

Beberapa hal yang perlu diingat tentang probabilitas hasil:

- Setiap probabilitas harus antara 0 dan 1.
- Semua peluang hasil elementer dalam ruang sampel harus berjumlah 1,00.

Bagaimana Anda menetapkan probabilitas itu? Kadang-kadang Anda memiliki informasi awal — seperti mengetahui bahwa sebuah koin bias menuju kepala yang muncul 60 persen dari waktu. Kadang-kadang Anda hanya perlu memikirkan situasinya untuk mengetahui probabilitas suatu hasil.

Berikut adalah contoh singkat dari "memikirkan situasi." Misalkan sebuah dadu bias sehingga probabilitas suatu hasil sebanding dengan label numerik dari hasil tersebut: A 6 muncul enam kali lebih sering dari 1, a 5 muncul lima kali lebih sering dari 1, dan seterusnya. Berapa probabilitas dari setiap hasil? Semua peluang harus dijumlahkan 1,00, dan semua angka pada dadu dijumlahkan dengan 21 (1 + 2 + 3 + 4 + 5 + 6 = 21), jadi peluangnya adalah: pr (1) = 1/21, pr (2) 2/21, , pr (6) = 6/21.

17.2 PERISTIWA MAJEMUK

Beberapa aturan untuk menangani peristiwa majemuk membantu Anda "memikirkan." Suatu peristiwa gabungan terdiri dari lebih dari satu peristiwa. Dimungkinkan untuk menggabungkan acara dengan penyatuan atau persimpangan (atau keduanya).

Persatuan dan persimpangan

Pada pelemparan sebuah dadu yang adil, berapa peluang terambilnya angka 1 atau 4? Matematikawan memiliki simbol untuk atau. Ini disebut serikat, dan terlihat seperti ini: ∪. Dengan menggunakan simbol ini, peluang munculnya a 1 atau a 4 adalah pr(1 ∪ 4). Dalam mendekati probabilitas semacam ini, akan sangat membantu untuk melacak hasil dasar. Satu hasil dasar ada di setiap peristiwa, jadi peristiwa "1 atau 4" memiliki dua hasil dasar. Dengan ruang sampel enam hasil, peluangnya adalah 2/6, atau 1/3. Cara lain untuk menghitung ini adalah:

$$pr(1 \cup 4) = pr(1) + pr(4) = \frac{1}{6} + \frac{1}{6} = \frac{2}{6} = \frac{1}{3}$$

Inilah yang sedikit lebih terlibat: Berapa probabilitas mendapatkan angka antara 1 dan 3 atau angka antara 2 dan 4?

Hanya menambahkan hasil dasar di setiap acara tidak akan menyelesaikannya kali ini. Tiga hasil dalam acara "antara 1 dan 3" dan tiga di acara "antara 2 dan 4." Probabilitas tidak mungkin 3 3 dibagi dengan enam hasil di ruang sampel, karena itu 1,00, tidak menyisakan apa pun untuk pr(5) dan pr(6). Untuk alasan yang sama, Anda tidak bisa hanya menambahkan probabilitas. Tantangan muncul dalam tumpang tindih dua peristiwa. Hasil dasar di "antara 1 dan 3" adalah 1, 2, dan 3. Hasil dasar dalam "antara 2 dan 4" adalah 2, 3, dan 4. Dua hasil tumpang tindih: 2 dan 3. Agar tidak dihitung dua kali, triknya adalah menguranginya dari total.

Beberapa hal akan membuat hidup lebih mudah saat saya melanjutkan. Saya menyingkat "antara 1 dan 3" sebagai A dan "antara 2 dan 4" sebagai B. Juga, saya menggunakan simbol matematika untuk "tumpang tindih." Simbolnya adalah ⋂ dan itu disebut persimpangan. Dengan menggunakan simbol-simbol tersebut, peluang “antara 1 dan 3” atau “antara 2 dan 4” adalah:

$$pr(A \cup B) = \frac{\text{Number of Outcomes in A} + \text{Number of Outcomes in B} - \text{Number of Outcomes in } (A \cap B)}{\text{Number of Outcomes in the Sample Space}}$$

$$pr(A \cup B) = \frac{3+3-2}{6} = \frac{4}{6} = \frac{2}{3}$$

Anda juga dapat bekerja dengan probabilitas:

$$pr(A \cup B) = \frac{3}{6} + \frac{3}{6} - \frac{2}{6} = \frac{4}{6} = \frac{2}{3}$$

Rumus umumnya adalah:

$$pr(A \cup B) = pr(A) + pr(B) - pr(A \cap B)$$

Mengapa tidak apa-apa untuk hanya menambahkan probabilitas bersama-sama dalam contoh sebelumnya? Karena pr(1 ∩ 4) adalah nol: Tidak mungkin mendapatkan angka 1 dan 4 dalam pelemparan sebuah dadu yang sama. Setiap kali pr(A ∩ B) = 0, A dan B dikatakan saling lepas.

Persimpangan lagi

Bayangkan melempar koin dan melempar dadu secara bersamaan. Kedua eksperimen ini independen, karena hasil yang satu tidak mempengaruhi hasil yang lain. Berapa probabilitas mendapatkan kepala dan 4? Anda menggunakan simbol persimpangan dan menulis ini sebagai pr(heads 4):

$$\text{pr}(\text{Heads} \cap 4) = \frac{\text{Number of Elementary Outcomes in Heads } \cap \text{ 4}}{\text{Number of Elementary Outcomes in the Sample Space}}$$

Mulailah dengan ruang sampel. Tabel 17.1 mencantumkan semua hasil dasar.

Tabel 17.1 Hasil Dasar di Ruang Sampel untuk Melempar Koin dan Melempar Sebuah Die

Heads, 1	Tails, 1
Heads, 2	Tails, 2
Heads, 3	Tails, 3
Heads, 4	Tails, 4
Heads, 5	Tails, 5
Heads, 6	Tails, 6

Seperti yang ditunjukkan tabel, 12 hasil yang mungkin. Berapa banyak hasil dalam acara "kepala dan 4"? Hanya satu. Jadi

$$\text{pr}(\text{Heads} \cap 4) = \frac{\text{Number of Elementary Outcomes in Heads} \cap 4}{\text{Number of Elementary Outcomes in the Sample Space}} = \frac{1}{12}$$

Anda juga dapat bekerja dengan probabilitas:

$$\text{pr}(\text{Heads} \cap 4) = \text{pr}(\text{Heads}) \times \text{pr}(4) = \frac{1}{12}$$

Secara umum, jika A dan B saling bebas,

$$pr(A \cap B) = pr(A) \times pr(B)$$

17.3 Probabilitas Bersyarat

Dalam beberapa keadaan, Anda mempersempit ruang sampel. Sebagai contoh, misalkan saya melempar sebuah dadu dan saya memberi tahu Anda bahwa hasilnya lebih besar dari 2. Berapa probabilitas bahwa itu adalah 5? Biasanya peluang munculnya angka 5 adalah 1/6. Namun dalam kasus ini, ruang sampelnya bukan 1, 2, 3, 4, 5, dan 6. Jika diketahui hasilnya lebih besar dari 2, ruang sampelnya menjadi 3, 4, 5, dan 6. Probabilitas dari 5 sekarang 1/4.

Ini adalah contoh probabilitas bersyarat. Ini "bersyarat" karena saya telah memberikan "syarat" — lemparan menghasilkan angka yang lebih besar dari 2. Notasi untuk ini adalah (pr 5|Lebih dari 2).

Garis vertikal (|) adalah singkatan untuk kata "diberikan," dan Anda membaca notasi itu sebagai "probabilitas 5 diberikan lebih besar dari 2."

Bekerja dengan probabilitas

Secara umum, jika Anda memiliki dua peristiwa A dan B,

$$pr(A|B) = \frac{pr(A \cap B)}{pr(B)}$$

selama pr(B) tidak nol.

Untuk perpotongan pada pembilang di sebelah kanan, ini bukan kasus di mana Anda hanya mengalikan peluang. Faktanya, jika Anda bisa melakukan itu, Anda tidak akan memiliki probabilitas bersyarat, karena itu berarti A dan B independen. Jika mereka independen, satu

peristiwa tidak dapat bergantung pada yang lain. Anda harus memikirkan kemungkinan persimpangan. Dalam sebuah dadu, berapa banyak hasil pada kejadian "5 Lebih besar dari 2"? Hanya satu, jadi pr(5 Lebih besar dari 2) adalah 1/6, dan

$$\text{pr}(5 \mid \text{Greater than } 2) = \frac{\text{pr}(5 \cap \text{Greater than } 2)}{\text{pr}(\text{Greater than } 2)} = \frac{1/6}{4/6} = \frac{1}{4}$$

Dasar pengujian hipotesis

Semua pengujian hipotesis yang saya bahas dalam bab-bab sebelumnya melibatkan probabilitas bersyarat. Saat Anda menghitung statistik sampel, menghitung uji statistik, dan kemudian membandingkan statistik uji dengan nilai kritis, Anda mencari probabilitas bersyarat. Secara khusus, Anda mencoba menemukan statistik uji yang diperoleh pr (atau nilai yang lebih ekstrem|H_0 benar).

Jika probabilitas bersyarat itu rendah (kurang dari 0,05 dalam semua contoh yang saya tunjukkan dalam bab pengujian hipotesis), Anda menolak H_0.

17.4 RUANG SAMPEL BESAR

Ketika berhadapan dengan probabilitas, penting untuk memahami ruang sampel. Dalam contoh yang telah saya tunjukkan sejauh ini dalam bab ini, ruang sampelnya kecil. Dengan koin atau dadu, mudah untuk membuat daftar semua hasil dasar. Dunia, tentu saja, tidak sesederhana itu. Faktanya, bahkan masalah probabilitas yang ada di buku teks statistik tidak sesederhana itu. Sebagian besar waktu, ruang sampel berukuran besar dan tidak nyaman untuk membuat daftar setiap hasil dasar.

Ambil contoh, melempar dadu dua kali. Berapa banyak hasil elementer dalam ruang sampel yang terdiri dari kedua lemparan? Anda dapat duduk dan membuat daftarnya, tetapi lebih baik untuk memikirkannya: Enam kemungkinan untuk lemparan pertama, dan masing-masing dari enam itu dapat dipasangkan dengan enam kemungkinan pada lemparan kedua. Jadi ruang sampel memiliki 6 x 6 = 36 kemungkinan hasil dasar. Ini mirip dengan ruang sampel koin-dan-mati pada Tabel 17-1, di mana ruang sampel terdiri dari 2 x 6 = 12 hasil dasar. Dengan 12 hasil, mudah untuk membuat daftar semuanya dalam sebuah tabel. Dengan 36 hasil itu mulai mendapatkan, yah, tidak pasti. (Maaf).

Peristiwa sering membutuhkan pemikiran juga. Berapa peluang pelemparan sebuah dadu dua kali dan berjumlah 5? Anda harus menghitung banyaknya cara kedua lemparan tersebut berjumlah 5, dan kemudian membaginya dengan jumlah hasil dasar dalam ruang sampel (36). Anda menjumlahkan 5 dengan mendapatkan salah satu dari pasangan lemparan ini: 1 dan 4, 2 dan 3, 3 dan 2, atau 4 dan 1. Itu total empat cara, dan mereka tidak tumpang tindih (permisi — berpotongan), jadi

$$\text{pr}(5) = \frac{\text{Number of Ways of Rolling a 5}}{\text{Number of Possible Outcomes of Two Tosses}} = \frac{4}{36} = .11$$

Mendaftar semua hasil dasar untuk ruang sampel sering kali merupakan mimpi buruk. Untungnya, pintasan tersedia, seperti yang saya tunjukkan di subbagian yang akan datang.

Karena setiap pintasan dengan cepat membantu Anda menghitung sejumlah item, nama lain untuk pintasan adalah aturan penghitungan.

Percaya atau tidak, saya baru saja menyelipkan satu aturan penghitungan melewati Anda. Beberapa paragraf yang lalu, saya katakan bahwa dalam dua kali pelemparan sebuah dadu Anda memiliki ruang sampel 6 x 6 = 36 kemungkinan hasil. Ini adalah aturan produk: Jika hasil N_1 mungkin pada percobaan pertama dari suatu percobaan, dan hasil N_2 mungkin pada percobaan kedua, jumlah hasil yang mungkin adalah N_1N_2. Setiap hasil yang mungkin pada percobaan pertama dapat diasosiasikan dengan semua hasil yang mungkin pada percobaan kedua. Bagaimana dengan tiga percobaan? Itu $N_1N_2N_3$.

Sekarang untuk beberapa aturan penghitungan lagi.

Permutasi

Misalkan Anda harus mengatur lima objek ke dalam urutan. Berapa banyak cara Anda dapat melakukannya? Untuk posisi pertama dalam urutan, Anda memiliki lima pilihan. Setelah Anda membuat pilihan itu, Anda memiliki empat pilihan untuk posisi kedua. Kemudian Anda memiliki tiga pilihan untuk yang ketiga, dua untuk yang keempat, dan satu untuk yang kelima. Banyaknya cara adalah (5) (4) (3) (2) (1) = 120.

Secara umum banyaknya barisan dari N objek adalah N(N–1)(N–2). . .(2)(1). Perhitungan semacam ini cukup sering terjadi di dunia probabilitas, dan memiliki notasinya sendiri: N! Anda tidak membaca ini dengan berteriak "N" dengan suara keras. Sebaliknya, ini adalah "N faktorial." Menurut definisi, 1! 1, dan 0! 1.

Sekarang untuk hal-hal yang baik. Jika Anda harus memesan 26 huruf alfabet, jumlah urutan yang mungkin adalah 26!, jumlah yang sangat besar. Tetapi misalkan tugasnya adalah membuat urutan lima huruf sehingga tidak ada huruf yang berulang dalam urutan tersebut. Berapa banyak cara Anda dapat melakukannya? Anda memiliki 26 pilihan untuk huruf pertama, 25 untuk yang kedua, 24 untuk yang ketiga, 23 untuk yang keempat, 22 untuk yang kelima, dan hanya itu. Jadi itu (26)(25) (24)(23)(22). Inilah bagaimana produk itu terkait dengan 26!:

$$\frac{26!}{21!}$$

Setiap barisan disebut permutasi. Secara umum, jika Anda mengambil permutasi dari N hal r pada satu waktu, notasinya adalah NPr (P adalah singkatan dari permutasi). Rumusnya adalah:

$$_N P_r = \frac{N!}{(N-r)!}$$

Hanya untuk kelengkapan, inilah kerutan lain. Misalkan saya mengizinkan pengulangan dalam urutan 5 ini. Artinya, aabbc adalah urutan yang diizinkan. Dalam hal ini, jumlah barisan adalah 26 x 26 x 26 x 26 x 26, atau seperti yang dikatakan ahli matematika, "26 dipangkatkan ke lima." Atau seperti yang akan ditulis oleh ahli matematika, "26^5."

Kombinasi

Dalam contoh sebelumnya, urutan ini berbeda satu sama lain: abcde, adbce, dbcae, dan terus dan terus. Bahkan, Anda bisa mendapatkan 5! = 120 dari barisan yang berbeda ini hanya untuk huruf a, b, c, d, dan e.

Misalkan saya menambahkan batasan bahwa salah satu dari urutan ini tidak berbeda dari yang lain, dan yang saya khawatirkan adalah memiliki set lima huruf yang tidak berulang tanpa urutan tertentu. Setiap himpunan disebut kombinasi. Untuk contoh ini, jumlah kombinasi adalah jumlah permutasi dibagi 5!:

$$\frac{26!}{5!(21!)}$$

Secara umum, notasi untuk kombinasi N hal yang diambil r pada suatu waktu adalah NCr (C singkatan dari kombinasi). Rumusnya adalah:

$$_NC_r = \frac{N!}{r!(N-r)!}$$

Saya menyentuh topik ini di Lampiran B. Dalam konteks uji statistik yang disebut uji jumlah peringkat Wilcoxon, saya menggunakan sebagai contoh jumlah kombinasi dari delapan hal yang diambil empat sekaligus:

$$_8C_4 = \frac{8!}{4!4!} = 70$$

Sekarang untuk kelengkapan itu kerut lagi. Misalkan saya mengizinkan pengulangan dalam urutan ini. Berapa banyak urutan yang akan saya miliki? Ternyata setara dengan N+r-1 hal yang diambil N-1 sekaligus, atau $_{N+r+1}C_{N-1}$. Untuk contoh ini, itu akan menjadi $_{30}C_{25}$.

17.5 FUNGSI R UNTUK ATURAN PENGHITUNGAN

R menyediakan faktorial() untuk menemukan faktorial suatu bilangan:

```
> factorial(6)
[1] 720
```

Anda juga dapat menggunakan fungsi ini untuk mencari faktorial dari setiap bilangan dalam sebuah vektor:

```
> xx <- c(2,3,4,5,6)
> factorial(xx)
[1]   2   6  24 120 720
```

Untuk kombinasi, R memberikan beberapa kemungkinan. Fungsi select() menghitung NCr — jumlah kombinasi dari N hal yang diambil r pada suatu waktu. Jadi, untuk 8 hal yang diambil 4 sekaligus (lihat contoh dari Lampiran B), yaitu

```
> choose(8,4)
[1] 70
```

Untuk membuat daftar semua kombinasi, gunakan combn(). Saya ilustrasikan dengan 4C2. Saya memiliki vektor yang berisi nama empat saudara Marx

```
Marx.Bros <- c("Groucho","Chico","Harpo","Zeppo")
```

dan saya ingin membuat daftar semua kemungkinan kombinasi dari mereka yang diambil dua sekaligus:

```
> combn(Marx.Bros,2)
     [,1]      [,2]      [,3]      [,4]    [,5]    [,6]
[1,] "Groucho" "Groucho" "Groucho" "Chico" "Chico" "Harpo"
[2,] "Chico"   "Harpo"   "Zeppo"   "Harpo" "Zeppo" "Zeppo"
```

Matriks ini memberi tahu saya bahwa enam kombinasi seperti itu dimungkinkan, dan dua baris di setiap kolom menunjukkan dua nama di setiap kombinasi. Dalam pandangan saya, fungsi terbaik untuk menangani kombinasi dan permutasi ada di paket gtools. Pada tab Paket, temukan gtools dan pilih kotak centangnya.
Berikut adalah fungsi kombinasi() dan permutasi() dari gtools di tempat kerja:

```
> combinations(4,2,v=Marx.Bros)
     [,1]      [,2]
[1,] "Chico"   "Groucho"
[2,] "Chico"   "Harpo"
[3,] "Chico"   "Zeppo"
[4,] "Groucho" "Harpo"
[5,] "Groucho" "Zeppo"
[6,] "Harpo"   "Zeppo"

> permutations(4,2,v=Marx.Bros)
      [,1]      [,2]
 [1,] "Chico"   "Groucho"
 [2,] "Chico"   "Harpo"
 [3,] "Chico"   "Zeppo"
 [4,] "Groucho" "Chico"
 [5,] "Groucho" "Harpo"
 [6,] "Groucho" "Zeppo"
 [7,] "Harpo"   "Chico"
 [8,] "Harpo"   "Groucho"
 [9,] "Harpo"   "Zeppo"
[10,] "Zeppo"   "Chico"
[11,] "Zeppo"   "Groucho"
[12,] "Zeppo"   "Harpo"
```

Untuk setiap fungsi, argumen pertama adalah N, yang kedua adalah r, dan yang ketiga adalah vektor yang berisi item. Tanpa vektor, inilah yang terjadi:

```
> combinations(4,2)
     [,1] [,2]
[1,]    1    2
[2,]    1    3
[3,]    1    4
[4,]    2    3
[5,]    2    4
[6,]    3    4
```

Jika semua yang ingin Anda lakukan adalah memecahkan jumlah kombinasi:

```
> nrow(combinations(4,2))
[1] 6
```

Tentu saja, Anda dapat melakukan hal yang sama untuk permutasi.

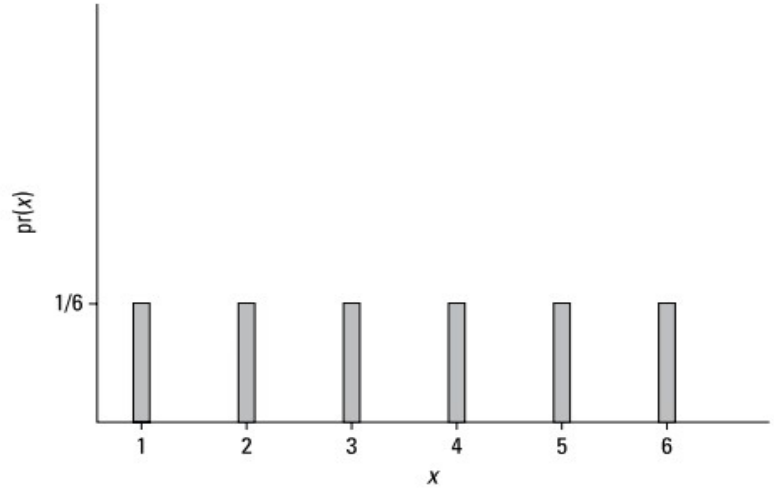
17.6 VARIABEL ACAK: DISKRIT DAN KONTINU

Mari saya kembali ke lemparan dadu yang adil, di mana enam hasil dasar dimungkinkan. Jika saya menggunakan x untuk merujuk ke hasil lemparan, x dapat berupa bilangan bulat dari 1 hingga 6. Karena x dapat mengambil sekumpulan nilai, itu adalah variabel. Karena nilai x yang mungkin sesuai dengan hasil dasar dari sebuah eksperimen (artinya Anda tidak dapat memprediksi nilainya dengan kepastian mutlak), x disebut variabel acak.

Variabel acak datang dalam dua varietas. Salah satu varietas bersifat diskrit, di mana die-tossing adalah contoh yang baik. Sebuah variabel acak diskrit hanya dapat mengambil apa yang matematika suka sebut jumlah nilai yang dapat dihitung — seperti angka 1 sampai 6. Nilai antara seluruh angka 1 sampai 6 (seperti 1,25 dan 3,1416) tidak mungkin untuk variabel acak yang sesuai dengan hasil lemparan dadu. Jenis variabel acak lainnya adalah kontinu. Variabel acak kontinu dapat mengambil jumlah nilai yang tak terbatas. Suhu adalah contohnya. Tergantung pada ketepatan termometer, suhu seperti 34,516 derajat dimungkinkan.

Distribusi Probabilitas dan Fungsi Kepadatan

Kembali lagi ke die-tossing. Setiap nilai variabel acak x (1–6, ingat) memiliki probabilitas. Jika dadu itu adil, setiap peluang adalah 1/6. Pasangkan setiap nilai dari variabel acak diskrit seperti x dengan probabilitasnya, dan Anda memiliki distribusi probabilitas. Distribusi probabilitas cukup mudah untuk direpresentasikan dalam grafik. Gambar 17.1 menunjukkan distribusi probabilitas untuk x.



Gambar 17.1 Distribusi probabilitas untuk x, variabel acak berdasarkan pelemparan sebuah dadu yang adil.

Sebuah variabel acak memiliki mean, varians, dan standar deviasi. Menghitung parameter ini cukup mudah. Dalam dunia variabel acak, mean disebut nilai harapan, dan nilai harapan variabel acak x disingkat E(x). Inilah cara Anda menghitungnya:

$$E(x)=\sum x\left(pr(x)\right)$$

Untuk distribusi probabilitas pada Gambar 17-1, yaitu:

$$E(x)=\sum x\left(pr(x)\right)=(1)\left(\frac{1}{6}\right)+(2)\left(\frac{1}{6}\right)+(3)\left(\frac{1}{6}\right)+(4)\left(\frac{1}{6}\right)+(5)\left(\frac{1}{6}\right)+(6)\left(\frac{1}{6}\right)=3.5$$

Varians variabel acak sering disingkat V(x), dan rumusnya adalah:

$$V(x)=\sum x^2\left(pr(x)\right)-\left[E(x)\right]^2$$

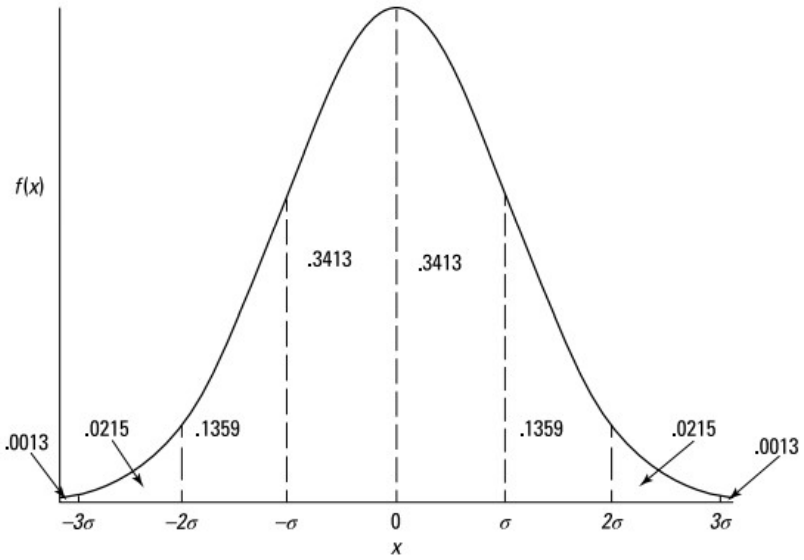
Bekerja dengan distribusi probabilitas pada Gambar 17.1 sekali lagi,

$$V(x)=\left(1^2\right)\left(\frac{1}{6}\right)+\left(2^2\right)\left(\frac{1}{6}\right)+\left(3^2\right)\left(\frac{1}{6}\right)+\left(4^2\right)\left(\frac{1}{6}\right)+\left(5^2\right)\left(\frac{1}{6}\right)+\left(6^2\right)\left(\frac{1}{6}\right)-\left[3.5\right]^2=2.917$$

Standar deviasi adalah akar kuadrat dari varians, yang dalam hal ini adalah 1,708.

Untuk variabel acak kontinu, segalanya menjadi sedikit lebih rumit. Anda tidak dapat memasangkan nilai dengan probabilitas, karena Anda tidak dapat benar-benar menentukan nilai. Sebaliknya, Anda mengasosiasikan variabel acak kontinu dengan aturan matematika (persamaan) yang menghasilkan kepadatan probabilitas, dan distribusinya disebut fungsi kepadatan probabilitas. Untuk menghitung mean dan varians dari variabel acak kontinu, Anda memerlukan kalkulus.

Dalam Bab 8, saya menunjukkan kepada Anda fungsi kepadatan probabilitas — distribusi normal standar. Saya mereproduksinya di sini sebagai Gambar 17.2.



Gambar 17.2 Distribusi normal standar: fungsi kepadatan probabilitas.

Pada gambar, f(x) mewakili kepadatan probabilitas. Karena kepadatan probabilitas dapat melibatkan beberapa konsep matematika kelas berat, saya tidak akan membahasnya. Seperti yang saya sebutkan di Bab 8, pikirkan kepadatan probabilitas sebagai sesuatu yang mengubah area di bawah kurva menjadi probabilitas.

Meskipun Anda tidak dapat berbicara tentang probabilitas nilai tertentu dari variabel acak kontinu, Anda dapat bekerja dengan probabilitas interval. Untuk menemukan probabilitas bahwa variabel acak mengambil nilai dalam suatu interval, Anda menemukan proporsi luas total di bawah kurva yang ada di dalam interval itu. Gambar 17.2 menunjukkan konsep ini. Probabilitas bahwa x berada di antara 0 dan 1σ adalah 0,3413.

Untuk sisa bab ini, saya hanya berurusan dengan variabel acak diskrit. Yang spesifik ada di depan.

17.7 DISTRIBUSI BINOMIAL

Bayangkan sebuah eksperimen yang memiliki lima karakteristik berikut:

- Percobaan terdiri dari N percobaan identik.
  Sebuah percobaan bisa menjadi lemparan dadu atau lemparan koin.
- Setiap percobaan menghasilkan salah satu dari dua hasil dasar.
  Merupakan standar untuk menyebut satu hasil sebagai sukses dan yang lainnya gagal. Untuk pelemparan diet, sukses mungkin merupakan lemparan yang muncul 3, dalam hal ini kegagalan adalah hasil lainnya.
- Probabilitas keberhasilan tetap sama dari percobaan ke percobaan.
  Sekali lagi, cukup standar untuk menggunakan p untuk mewakili probabilitas keberhasilan dan menggunakan 1–p (atau q) untuk mewakili probabilitas kegagalan.
- Uji coba bersifat independen.

- Variabel acak diskrit x adalah jumlah keberhasilan dalam N percobaan.

Jenis percobaan ini disebut percobaan binomial. Distribusi probabilitas untuk x mengikuti aturan ini:

$$pr(x) = \frac{N!}{x!(n-x)!} p^x (1-p)^{N-x}$$

Di paling kanan, px(1–p)N–x adalah probabilitas dari satu kombinasi x sukses dalam N percobaan. Suku di sebelah kiri langsungnya adalah NCx, banyaknya kemungkinan kombinasi x sukses dalam N percobaan. Ini disebut distribusi binomial. Anda menggunakannya untuk menemukan probabilitas seperti probabilitas Anda akan mendapatkan empat angka 3 dalam sepuluh kali pelemparan dadu:

$$pr(4) = \frac{10!}{4!(6!)} \left(\frac{1}{6}\right)^4 \left(\frac{5}{6}\right)^6 = .054$$

Distribusi binomial negatif berhubungan erat. Dalam distribusi ini, variabel acaknya adalah jumlah percobaan sebelum keberhasilan ke-x. Misalnya, Anda menggunakan binomial negatif untuk menemukan probabilitas lima kali lemparan yang menghasilkan apa pun kecuali 3 sebelum keempat kalinya Anda melempar 3.

Agar ini terjadi, dalam delapan lemparan sebelum 3 keempat, Anda harus mendapatkan lima non-3 dan tiga keberhasilan (lemparan ketika 3 muncul). Kemudian lemparan berikutnya menghasilkan a 3. Peluang kombinasi empat berhasil dan lima gagal adalah $p^4(1–p)^5$. Banyaknya cara Anda dapat memiliki kombinasi lima kegagalan dan keberhasilan empat banding satu adalah ${}_{5+4-1}C_{4-1}$. Jadi peluangnya adalah:

$$pr(5 \text{ failures before the 4th success}) = \frac{(5+4-1)!}{(4-1)!(5!)} \left(\frac{1}{6}\right)^4 \left(\frac{5}{6}\right)^5 = .017$$

Secara umum, distribusi binomial negatif (kadang-kadang disebut distribusi Pascal) adalah

$$pr(f \text{ failures before the } x\text{th success}) = \frac{(f+x-1)!}{(x-1)!(f!)} p^x (1-p)^f$$

17.8 BINOMIAL DAN BINOMIAL NEGATIF DALAM R

R menyediakan fungsi binom untuk distribusi binomial, dan fungsi nbinom untuk distribusi binomial negatif. Untuk kedua distribusi, saya bekerja dengan lemparan dadu sehingga p (probabilitas sukses) = 1/6.

Distribusi binomial

Seperti halnya untuk distribusi built-in lainnya, R menyediakan fungsi-fungsi ini untuk distribusi binomial: dbinom() (fungsi kepadatan), pbinom() (fungsi distribusi kumulatif), qbinom() (kuantil), dan rbinom() (acak generasi nomor).

Untuk menunjukkan kepada Anda distribusi binomial, saya menggunakan dbinon() untuk memplot fungsi kepadatan untuk jumlah keberhasilan dalam sepuluh pelemparan dadu yang adil. Saya mulai dengan membuat vektor untuk jumlah keberhasilan:

```
successes <- seq(0,10)
```

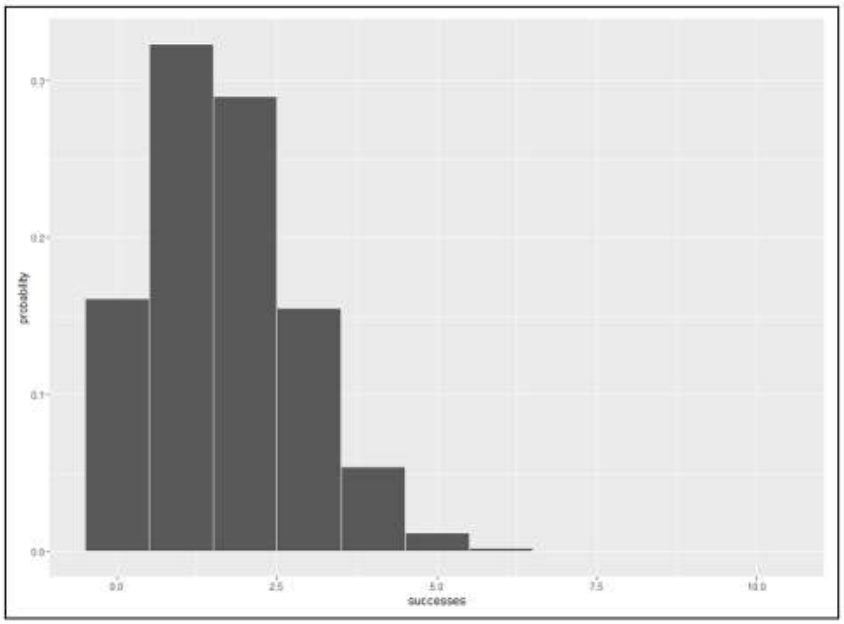
dan kemudian vektor untuk probabilitas terkait:

```
probability <- dbinom(successes,10,1/6)
```

Argumen pertama, tentu saja, adalah vektor keberhasilan, yang kedua adalah jumlah percobaan, dan yang ketiga (1/6) adalah probabilitas sukses dengan dadu bersisi enam yang adil. Untuk memplot fungsi kepadatan ini:

```
ggplot(NULL,aes(x=successes,y=probability))+
  geom_bar(stat="identity",width=1,color="white")
```

Argumen NULL di ggplot() menunjukkan bahwa saya belum membuat bingkai data — saya hanya menggunakan keberhasilan dan vektor probabilitas. Dalam geom_bar(), argumen stat= “identity” menunjukkan bahwa nilai dalam vektor probabilitas mengatur ketinggian batang, lebar = 1 melebarkan batang sedikit dari lebar default, dan warna = “putih” menambah kejelasan dengan meletakkan perbatasan putih di sekitar setiap batang. Kode menciptakan Gambar 17.3.



Gambar 17.3 Distribusi binomial dari jumlah keberhasilan dalam sepuluh pelemparan sebuah dadu yang adil.

Selanjutnya, saya menggunakan pbinom() untuk menunjukkan kepada Anda distribusi kumulatif untuk jumlah keberhasilan dalam sepuluh lemparan dadu yang adil:

```
cumulative <-pbinom(successes,10,1/6)
```

Dan inilah kode untuk plotnya:

```
ggplot(NULL,aes(x=successes,y=cumulative))+
  geom_step()
```

Pernyataan kedua menghasilkan fungsi bertahap yang Anda lihat pada Gambar 17.4:

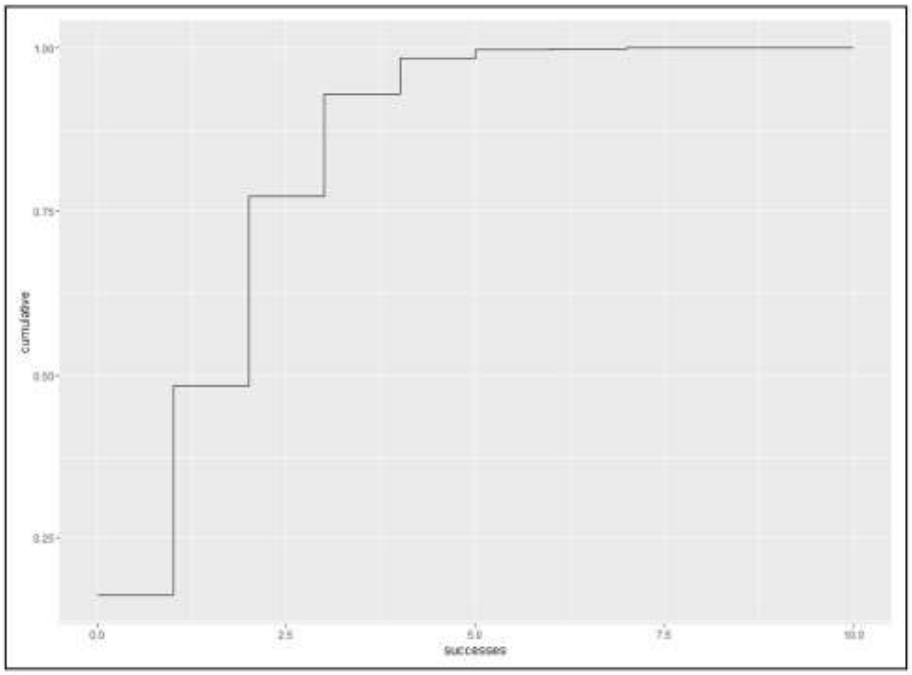
> *Setiap langkah mewakili probabilitas mendapatkan x atau lebih sedikit keberhasilan dalam sepuluh kali lemparan.*

Fungsi qbinom() menghitung informasi kuantil. Untuk setiap kuantil kelima dari 10 sampai 95 dalam distribusi binomial dengan N = 10 dan p = 1/6:

```
> qbinom(seq(.10,.95,.05),10,1/6)
 [1] 0 0 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 4
```

Untuk mengambil sampel 5 bilangan acak dari distribusi binomial ini

```
> rbinom(5, 10, 1/6)
[1] 4 3 3 0 2
```



Gambar 17.4 Distribusi kumulatif dari jumlah keberhasilan dalam sepuluh pelemparan dadu yang adil.

Distribusi binomial negatif

Untuk fungsi binomial negatif, dnbinom() menyediakan fungsi kepadatan, pnbinom() memberi Anda fungsi distribusi kumulatif, qnbinom() memberikan informasi kuantum, dan

rnbinom() menghasilkan angka acak. Contoh yang saya tunjukkan sebelumnya melibatkan jumlah kegagalan sebelum keberhasilan keempat dari lemparan mati. Kasus itu adalah probabilitas 5 kegagalan sebelum lemparan keempat, dan saya menggunakan dnbinom() untuk menghitung probabilitas itu:

```
> dnbinom(5,4,1/6)
[1] 0.01736508
```

Argumen pertama untuk dnbinom() adalah jumlah kegagalan, yang kedua adalah jumlah keberhasilan, dan yang ketiga adalah probabilitas keberhasilan. Jika saya ingin mengetahui kemungkinan lima kegagalan atau lebih sedikit sebelum keberhasilan keempat:

```
> pnbinom(5,4,1/6)
[1] 0.04802149
```

yang sama dengan

```
> sum(dnbinom(seq(0,5),4,1/6))
[1] 0.04802149
```

Untuk setiap kuantil kelima dari tanggal 10 hingga 95 dari jumlah kegagalan sebelum empat keberhasilan (dengan p = 1/6):

```
> qnbinom(seq(.10,.95,.05),4,1/6)
 [1]  8  9 11 12 13 14 16 17 18 20 21 22 24 26 28 31 35 41
```

Dan untuk sampel lima angka acak dari binomial negatif dengan 4 keberhasilan dan p = 1/6:

```
> rnbinom(5, 4, 1/6)
[1] 10  5  4 23  7
```

17.9 PENGUJIAN HIPOTESIS DENGAN DISTRIBUSI BINOMIAL

Uji hipotesis terkadang melibatkan distribusi binomial. Biasanya, Anda memiliki beberapa gagasan tentang kemungkinan keberhasilan, dan Anda memasukkan gagasan itu ke dalam hipotesis nol. Kemudian Anda melakukan N percobaan dan mencatat jumlah keberhasilan. Terakhir, Anda menghitung probabilitas mendapatkan banyak keberhasilan atau jumlah yang lebih ekstrem jika H0 Anda benar. Jika probabilitasnya rendah, tolak H0.

Saat Anda menguji dengan cara ini, Anda menggunakan statistik sampel untuk membuat kesimpulan tentang parameter populasi. Di sini, parameter itu adalah probabilitas keberhasilan dalam populasi percobaan. Dengan konvensi, huruf Yunani mewakili parameter. Ahli statistik menggunakan π (pi), padanan bahasa Yunani dari p, untuk menyatakan probabilitas keberhasilan dalam populasi.

Melanjutkan contoh pelemparan dadu, misalkan Anda memiliki dadu dan Anda ingin menguji apakah itu adil atau tidak. Anda menduga bahwa jika tidak, itu bias ke 3. Tentukan

lemparan yang menghasilkan 3 sebagai sukses. Anda melemparkannya sepuluh kali. Lima lemparan adalah keberhasilan. Menyebutkan semua ini ke dalam istilah pengujian hipotesis:

$$H_0: \pi \leq 1/6$$

$$H_1: \pi > 1/6$$

Seperti yang biasa saya lakukan, saya menetapkan α = 0,05.

Untuk menguji hipotesis ini, Anda harus mencari peluang untuk mendapatkan setidaknya empat keberhasilan dalam sepuluh kali lemparan dengan p = 1/6. Peluang tersebut adalah pr (5) + pr (6) + pr (7) + pr (8) + pr (9) + pr (10). Jika totalnya kurang dari 0,05, tolak H_0. Sekali waktu, itu akan menjadi banyak perhitungan. Dengan R, tidak begitu banyak. Fungsi binom.test() melakukan semua pekerjaan:

```
binom.test(5,10,1/6, alternative="greater")
```

Argumen pertama adalah jumlah keberhasilan, yang kedua adalah jumlah lemparan, yang ketiga adalah , dan yang keempat adalah hipotesis alternatif. Menjalankan fungsi ini menghasilkan

```
> binom.test(5,10,1/6, alternative="greater")

        Exact binomial test

data:  5 and 10
number of successes = 5, number of trials = 10,
p-value = 0.01546
alternative hypothesis: true probability of success is greater
        than 0.1666667
95 percent confidence interval:
 0.2224411 1.0000000
sample estimates:
probability of success
                   0.5
```

Nilai p (0,01546) jauh lebih kecil dari 0,05, dan itu memberitahu saya untuk menolak hipotesis nol. Juga, perhatikan informasi tambahan tentang interval kepercayaan dan perkiraan probabilitas keberhasilan (jumlah keberhasilan yang diperoleh dibagi dengan jumlah percobaan).

Jika Anda telah mengikuti diskusi tentang distribusi binomial, Anda tahu bahwa dua cara lain untuk menghitung nilai-p adalah:

```
> sum(dbinom(seq(5,10),10,1/6))
[1] 0.01546197
```

Dan

```
> 1-pbinom(4,10,1/6)
[1] 0.01546197
```

Dengan cara apa pun Anda memotongnya, keputusannya adalah menolak hipotesis nol.

17.10 LEBIH LANJUT TENTANG PENGUJIAN HIPOTESIS: R VERSUS TRADISI

Ketika N $\pi \geq 5$ (Jumlah percobaan × probabilitas keberhasilan yang dihipotesiskan) dan N(1 π) ≥ 5 (jumlah percobaan × probabilitas kegagalan yang dihipotesiskan) keduanya lebih besar dari 5, distribusi binomial mendekati distribusi normal standar. Dalam kasus tersebut, buku teks statistik biasanya meminta Anda untuk menggunakan statistik distribusi normal untuk menjawab pertanyaan tentang distribusi binomial. Demi tradisi, mari kita lanjutkan dan bandingkan dengan binom.test().

Statistik tersebut melibatkan z-score, yang berarti Anda harus mengetahui mean dan standar deviasi binomial. Untungnya, mereka mudah dihitung. Jika N adalah banyaknya percobaan dan adalah peluang sukses, rata-ratanya adalah:

$$\mu = N\pi$$

variansnya adalah:

$$\sigma^2 = N\pi(1-\pi)$$

dan simpangan bakunya adalah:

$$\sigma = \sqrt{N\pi(1-\pi)}$$

Saat Anda menguji hipotesis, Anda membuat kesimpulan tentang dan Anda harus mulai dengan perkiraan. Anda menjalankan N percobaan dan mendapatkan x keberhasilan. Perkiraannya adalah:

$$P = \frac{x}{N}$$

Untuk membuat z-score, Anda memerlukan satu informasi lagi — kesalahan standar P. Ini terdengar lebih sulit daripada itu, karena kesalahan standar ini hanya

$$\sigma_P = \sqrt{\frac{\pi(1-\pi)}{N}}$$

Sekarang Anda siap untuk uji hipotesis.

Berikut ini contohnya. CEO FarKlempt Robotics, Inc., percaya bahwa 50 persen robot FarKlempt dibeli untuk digunakan di rumah. Sampel dari 1.000 pelanggan FarKlempt menunjukkan bahwa 550 dari mereka menggunakan robot mereka di rumah. Apakah ini berbeda secara signifikan dari apa yang diyakini CEO? Hipotesis:

$H_0: \pi = .50$

$H_1: \pi \neq .50$

Saya menetapkan $\alpha = .05$

N π = 500, dan N (1- π) = 1500, sehingga pendekatan normal adalah tepat.

Pertama, hitung P:

$$P = \frac{x}{N} = \frac{550}{1000} = .55$$

Sekarang buat skor-z

$$z = \frac{P - \pi}{\sqrt{\frac{\pi(1-\pi)}{N}}} = \frac{.55 - .50}{\sqrt{\frac{(.50)(1-.50)}{1000}}} = \frac{.05}{\sqrt{\frac{.25}{1000}}} = 3.162$$

Dengan 0,05, apakah 3,162 merupakan nilai-z yang cukup besar untuk menolak H_0?

```
> pnorm(3.162,lower.tail = FALSE)*2
[1] 0.001566896
```

Ini jauh lebih kecil dari 0,05, jadi keputusannya adalah menolak H_0.

Dengan sedikit pemikiran, Anda dapat melihat mengapa ahli statistik merekomendasikan prosedur ini di masa lalu. Untuk menghitung probabilitas yang tepat, Anda harus menghitung probabilitas setidaknya 550 keberhasilan dalam 1.000 percobaan. Itu akan menjadi pr (550) + pr 551 +… + pr (1000) , jadi perkiraan berdasarkan distribusi terkenal sangat diterima — terutama di buku teks statistik.

Tapi sekarang

```
> binom.test(550,1000,.5,alternative="two.sided")

        Exact binomial test

data:  550 and 1000
number of successes = 550, number of trials = 1000,
p-value = 0.001731
alternative hypothesis: true probability of success is not
        equal to 0.5
95 percent confidence interval:
 0.5185565 0.5811483
sample estimates:
probability of success
                  0.55
```

Voila! Fungsi binom.test() menghitung probabilitas yang tepat dalam sekejap mata. Seperti yang Anda lihat, probabilitas pasti (0,001731) sedikit berbeda dari nilai-p yang diperkirakan secara normal tetapi kesimpulannya (tolak H_0) adalah sama.

BAB 18
MEMPERKENALKAN PEMODELAN

Model adalah sesuatu yang Anda ketahui dan dapat digunakan untuk membantu Anda memahami sesuatu yang hanya sedikit Anda ketahui. Seorang model seharusnya meniru, dalam beberapa hal, hal yang dimodelkannya. Bola dunia, misalnya, adalah model dari bumi. Peta jalan adalah model lingkungan. Cetak biru adalah model sebuah bangunan.

Peneliti menggunakan model untuk membantu mereka memahami proses dan fenomena alam. Analis bisnis menggunakan model untuk membantu mereka memahami proses bisnis. Model yang digunakan orang-orang ini mungkin termasuk konsep dari matematika dan statistik — konsep yang sangat terkenal sehingga mereka dapat menjelaskan hal yang tidak diketahui. Idenya adalah untuk membuat model yang terdiri dari konsep yang Anda pahami, menempatkan model melalui langkahnya, dan melihat apakah hasilnya terlihat seperti hasil dunia nyata. Dalam bab ini, saya membahas pemodelan. Tujuan saya adalah untuk menunjukkan bagaimana Anda dapat memanfaatkan R untuk membantu Anda memahami proses di dunia Anda.

18.1 PERMODELAN DISTRIBUSI

Dalam satu pendekatan pemodelan, Anda mengumpulkan data dan mengelompokkannya ke dalam distribusi. Selanjutnya, Anda mencoba mencari tahu proses yang menghasilkan distribusi semacam itu. Nyatakan kembali proses itu dalam istilah statistik sehingga dapat menghasilkan distribusi, dan kemudian lihat seberapa baik distribusi yang dihasilkan cocok dengan yang asli. “Proses yang Anda temukan dan nyatakan kembali dalam istilah statistik” adalah modelnya.

Jika distribusi yang Anda hasilkan cocok dengan data sebenarnya, apakah ini berarti model Anda "benar"? Apakah itu berarti proses yang Anda duga adalah proses yang menghasilkan data? Sayangnya tidak ada. Logikanya tidak bekerja seperti itu. Anda dapat menunjukkan bahwa model itu salah, tetapi Anda tidak dapat membuktikan bahwa itu benar.

Terjun ke dalam distribusi Poisson

Di bagian ini, saya memandu Anda melalui contoh pemodelan dengan distribusi Poisson. Saya membahas distribusi ini di Lampiran A, di mana saya memberi tahu Anda bahwa distribusi ini tampaknya mencirikan serangkaian proses di dunia nyata. Dengan "mencirikan suatu proses," maksud saya bahwa distribusi data dunia nyata sangat mirip dengan distribusi Poisson. Ketika ini terjadi, ada kemungkinan bahwa jenis proses yang menghasilkan distribusi Poisson juga bertanggung jawab untuk menghasilkan data.

Apa itu proses? Mulailah dengan variabel acak x yang melacak jumlah kemunculan peristiwa tertentu dalam suatu interval. Dalam Lampiran A, "interval" adalah sampel dari 1.000 sambungan universal, dan kejadian spesifiknya adalah "sambungan rusak". Distribusi

racun juga sesuai untuk peristiwa yang terjadi dalam interval waktu tertentu, dan peristiwa tersebut dapat berupa “kedatangan di pintu tol”.

Selanjutnya, saya menguraikan kondisi untuk proses Poisson dan menggunakan sambungan yang rusak dan kedatangan pintu tol untuk menggambarkan:

- Banyaknya kejadian dalam dua interval yang tidak tumpang tindih adalah bebas.
  Jumlah sambungan yang rusak dalam satu sampel tidak tergantung pada jumlah sambungan yang rusak di sampel lainnya. Jumlah kedatangan di gardu tol selama satu jam tidak tergantung pada jumlah kedatangan pada jam lainnya.
- Probabilitas terjadinya peristiwa sebanding dengan ukuran interval.
  Peluang Anda akan menemukan sambungan yang rusak lebih besar dalam sampel 10.000 daripada sampel 1.000. Peluang sampai di pintu tol lebih besar selama satu jam daripada setengah jam.
- Probabilitas lebih dari satu kejadian dalam interval kecil adalah 0 atau mendekati 0.

Dalam sampel 1.000 sambungan universal, Anda memiliki kemungkinan yang sangat rendah untuk menemukan dua sambungan yang rusak tepat di sebelah satu sama lain. Setiap saat, dua kendaraan tidak tiba di pintu tol secara bersamaan.

Seperti yang saya tunjukkan di Lampiran A, rumus untuk distribusi Poisson adalah:

$$pr(x) = \frac{\mu^x e^{-\mu}}{x!}$$

Dalam persamaan ini, mewakili jumlah rata-rata kemunculan peristiwa dalam interval yang Anda lihat, dan e adalah konstanta 2.781828 (diikuti oleh tak terhingga lebih banyak tempat desimal).

Pemodelan dengan distribusi Poisson

Saatnya menggunakan Poisson dalam sebuah model. Di FarBlonJet Corporation, desainer web melacak jumlah klik per jam di halaman beranda intranet. Mereka memantau halaman selama 200 jam berturut-turut dan mengelompokkan data, seperti yang tercantum dalam Tabel 18.1.

Tabel 18.1 Hits Per Jam pada Home Page Intranet FarBlonJet

Hit per Jam	Jam Pengamatan	Hit/Jam X Jam Pengamatan
0	10	0
1	30	30
2	44	88
3	44	132
4	36	144
5	18	90
6	10	60
7	8	56

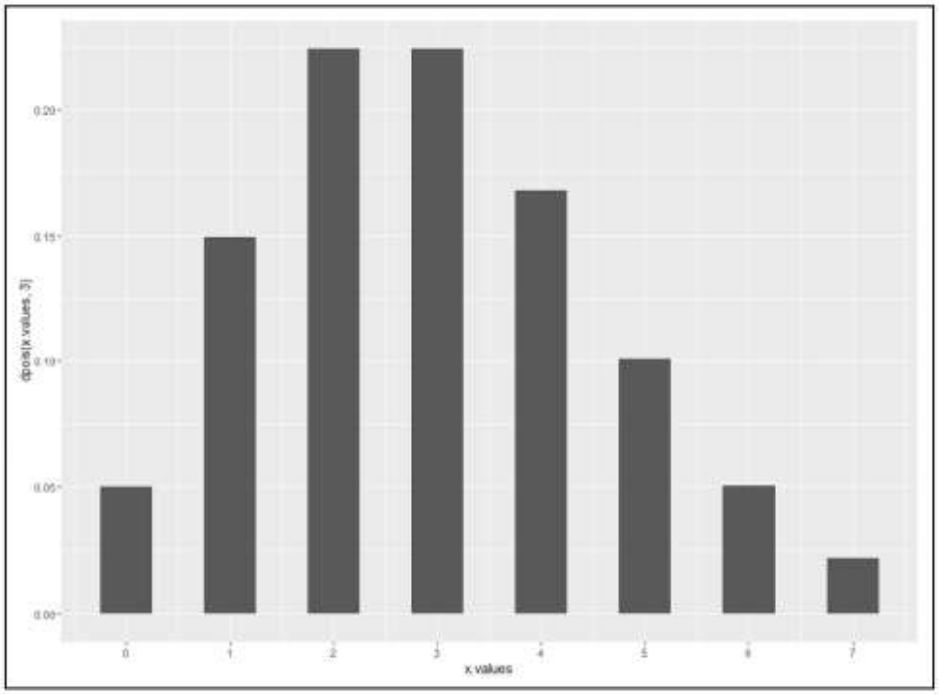
Total	200	600

Kolom pertama menunjukkan variabel Hits per Hour. Kolom kedua, Jam yang Diamati, menunjukkan jumlah jam di mana setiap nilai klik per jam terjadi. Dalam 200 jam yang diamati, 10 jam berlalu tanpa hit, 30 jam memiliki satu hit, 44 memiliki dua hit, dan seterusnya. Data ini mengarahkan desainer web untuk menggunakan distribusi Poisson untuk memodelkan hit per jam. Berikut cara lain untuk mengatakan ini: Mereka percaya bahwa proses Poisson menghasilkan jumlah klik per jam di halaman web. Mengalikan kolom pertama dengan kolom kedua menghasilkan kolom ketiga. Menjumlahkan kolom ketiga menunjukkan bahwa dalam 200 jam pengamatan, halaman intranet menerima 600 klik. Jadi rata-rata jumlah hit per jam adalah 3,00.

Menerapkan distribusi Poisson untuk contoh ini,

$$pr(x) = \frac{\mu^x e^{-\mu}}{x!} = \frac{3^x e^{-3}}{x!}$$

Gambar 18.1 menunjukkan fungsi densitas untuk distribusi Poisson dengan μ = 3.



Gambar 18.1: Distribusi Poisson dengan 3.

Label sumbu pada gambar mengisyaratkan cara membuatnya. Mulailah dengan vektor nilai untuk sumbu x.

```
x.values <- seq(0,7)
```

Kemudian, kerjakan fungsi kerapatan untuk distribusi Poisson (lihat Lampiran A):

```
dpois(x.values,3)
```

Itulah fungsi yang digunakan untuk pemetaan estetika y di ggplot():

```
ggplot(NULL,aes(x=x.values,y=dpois(x.values,3)))+
  geom_bar(stat="identity",width=.5)+
  scale_x_continuous(breaks=seq(0,7))
```

Pernyataan kedua memplot bar. Argumen pertamanya (stat="identity") menetapkan bahwa tinggi setiap batang adalah nilai fungsi densitas yang sesuai yang dipetakan ke y. Lebar yang ditunjukkan (.5) dalam argumen kedua mempersempit bilah sedikit dari nilai default (.9). Pernyataan ketiga menempatkan 0–7 pada sumbu x. Tujuan dari model adalah untuk memprediksi. Untuk model ini, Anda ingin menggunakan distribusi Poisson untuk memprediksi distribusi hit per jam. Untuk melakukannya, kalikan setiap probabilitas Poisson dengan 200 — jumlah total jam:

```
Predicted <- dpois(x.values,3)*200
```

Berikut prediksinya:

```
> Predicted
[1]  9.957414 29.872241 44.808362 44.808362 33.606271
           20.163763 10.081881  4.320806
```

Untuk bekerja dengan nilai yang diamati (Kolom 2 pada Tabel 18-1), buat vektor:

```
Observed <- c(10,30,44,44,36,18,10,8)
```

Anda ingin menggunakan ggplot untuk menunjukkan seberapa dekat jam yang diprediksi dengan jam yang diamati, jadi buat bingkai data. Ini melibatkan tiga vektor lagi:

```
Category <-c(rep("Observed",8),rep("Predicted",8))
Hits.Hr <- c(x.values,x.values)
Hours <- c(Observed,Predicted)
```

Dan sekarang Anda dapat membuat

```
FBJ.frame <-data.frame(Category,Hits.Hr,Hours)
```

yang terlihat seperti ini

```
> FBJ.frame
    Category Hits.Hr     Hours
1   Observed       0 10.000000
2   Observed       1 30.000000
3   Observed       2 44.000000
```

```
4   Observed       3 44.000000
5   Observed       4 36.000000
6   Observed       5 18.000000
7   Observed       6 10.000000
8   Observed       7  8.000000
9  Predicted       0  9.957414
10 Predicted       1 29.872241
11 Predicted       2 44.808362
12 Predicted       3 44.808362
13 Predicted       4 33.606271
14 Predicted       5 20.163763
15 Predicted       6 10.081881
16 Predicted       7  4.320806
```

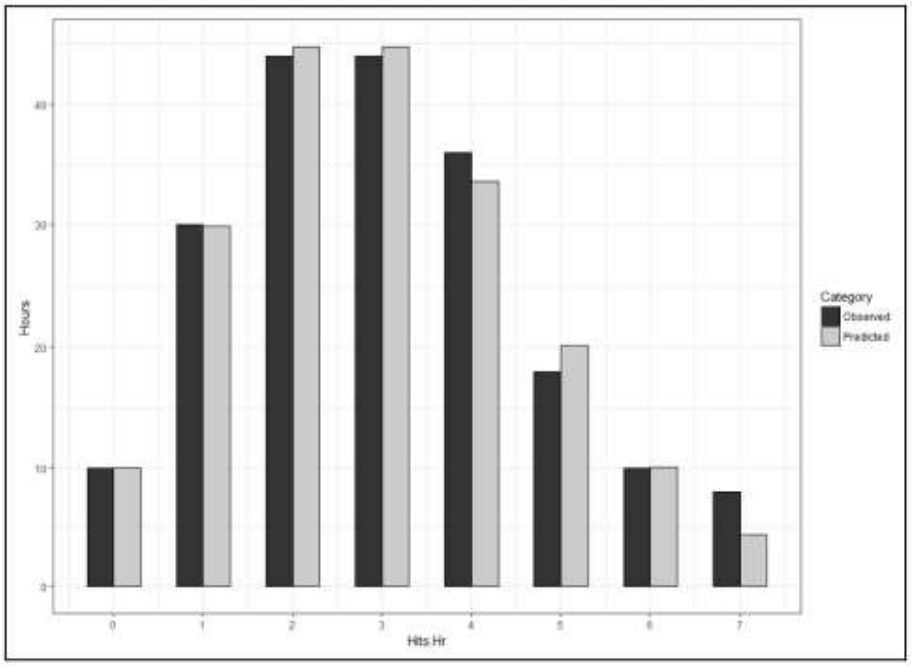
Untuk merencanakan semuanya:

```
ggplot(FBJ.frame,aes(x=Hits.Hr,y=Hours,fill=Category))+
  geom_bar(stat="identity", position="dodge", color="black",
           width=.6)+
  scale_x_continuous(breaks=x.values)+
  scale_fill_grey()+
  theme_bw()
```

Pernyataan pertama menggunakan bingkai data, dengan pemetaan estetika yang ditunjukkan ke x, y, dan isi. Pernyataan kedua memplot bar. Argumen position= "dodge" menempatkan dua kategori bar berdampingan, dan color = "black" menggambar batas hitam di sekitar bar (yang tentu saja tidak akan muncul di bar yang diisi hitam). Seperti sebelumnya, pernyataan ketiga menempatkan nilai dalam vektor x.values pada sumbu x. Pernyataan keempat mengubah warna isian bilah menjadi warna yang muncul di halaman yang Anda baca, dan pernyataan terakhir menghapus latar belakang abu-abu default. (Itu membuat palang lebih mudah dilihat.) Gambar 18.2 menunjukkan plot. Yang diamati dan yang diprediksi terlihat cukup dekat, bukan?

Menguji kecocokan model

Yah, "tampak cukup dekat" tidak cukup untuk seorang ahli statistik. Sebuah tes statistik adalah suatu keharusan. Seperti halnya semua uji statistik, uji ini dimulai dengan hipotesis nol dan hipotesis alternatif. Di sini mereka:



Gambar 18.2 Home page intranet FarBlonJet hits per jam, diamati dan diprediksi Poisson (μ =3).

H_0: Distribusi hit yang diamati per jam mengikuti distribusi Poisson.
H_1: Bukan H_0

Uji statistik yang sesuai melibatkan perluasan distribusi binomial. Ini disebut distribusi multinomial - "multi" karena mencakup lebih banyak kategori daripada hanya "sukses" dan "gagal." Ini adalah distribusi yang sulit untuk dikerjakan.

Untungnya, ahli statistik perintis Karl Pearson (penemu koefisien korelasi) memperhatikan bahwa 2 ("chi-kuadrat"), distribusi yang saya tunjukkan di Bab 10, mendekati multinomial. Awalnya ditujukan untuk uji hipotesis satu sampel tentang varians, 2 telah menjadi jauh lebih dikenal untuk aplikasi seperti yang akan saya tunjukkan kepada Anda.

Ide besar Pearson adalah ini: Jika Anda ingin mengetahui seberapa baik distribusi yang dihipotesiskan (seperti Poisson) cocok dengan sampel (seperti jam yang diamati), gunakan distribusi untuk menghasilkan sampel yang dihipotesiskan (jam prediksi Anda, misalnya), dan bekerja dengan rumus ini:

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Predicted})^2}{\text{Predicted}}$$

Biasanya, rumus ditulis dengan Expected daripada Predicted, dan keduanya Observed dan Expected disingkat. Bentuk umum dari rumus ini adalah:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Untuk contoh ini,

$$\chi^2 = \sum \frac{(O-E)^2}{E} = \frac{(10-9.9574)^2}{9.9574} + \frac{(30-29.8722)^2}{29.8722} + \ldots + \frac{(8-4.3208)^2}{4.3208}$$

itu totalnya sampai apa? Anda dapat menggunakan R sebagai kalkulator untuk mengetahuinya — saya sudah menyebut vektor nilai prediksi Predicted, dan saya tidak ingin mengubah namanya menjadi Expected:

```
> chi.squared <- sum(((Observed-Predicted)^2)/Predicted)
> chi.squared
[1] 3.566111
```

Oke. Sekarang apa? Apakah 3,566111 tinggi, atau rendah?

Untuk mengetahuinya, Anda mengevaluasi chi.squared terhadap distribusi 2. Tujuannya adalah untuk mencari probabilitas mendapatkan nilai minimal setinggi nilai yang dihitung, 3,566111. Caranya adalah dengan mengetahui berapa derajat kebebasan (df) yang Anda miliki. Untuk aplikasi yang cocok seperti ini:

$$df = k - m - 1$$

dimana k = jumlah kategori dan m = jumlah parameter yang diestimasi dari data. Jumlah kategorinya adalah 8 (0 Hits per Hour hingga 7 Hits per Hour). Banyaknya parameter? Saya menggunakan jam yang diamati untuk memperkirakan parameter , jadi m dalam contoh ini adalah 1. Itu berarti df = 8–1–1 = 6.

Untuk mencari peluang mendapatkan nilai chi.squared (3.566111) atau lebih, saya menggunakan pchisq() dengan 6 derajat kebebasan:

```
> pchisq(chi.squared,6,lower.tail = FALSE)
[1] 0.7351542
```

Argumen ketiga, lower.tail = FALSE, menunjukkan bahwa saya ingin area di sebelah kanan 3,56111 dalam distribusi (karena saya mencari probabilitas nilai yang ekstrim atau lebih tinggi). Jika α = .05, probabilitas yang dikembalikan (.7351542) memberi tahu saya untuk tidak menolak H0 — artinya Anda tidak dapat menolak hipotesis bahwa data yang diamati berasal dari distribusi Poisson. Ini adalah salah satu waktu yang jarang terjadi ketika bermanfaat untuk tidak menolak H_0 — jika Anda ingin membuat kasus bahwa proses Poisson menghasilkan data. Nilai 2 yang rendah menunjukkan kecocokan yang erat antara data dan prediksi Poisson. Jika probabilitasnya hanya sedikit lebih besar dari 0,05, tidak menolak H0 akan terlihat mencurigakan. Probabilitas tinggi, bagaimanapun, membuatnya masuk akal untuk tidak menolak H0 — dan untuk berpikir bahwa proses Poisson mungkin menjelaskan data.

Sepatah kata tentang chisq.test()

R menyediakan fungsi chisq.test(), yang menurut namanya menunjukkan bahwa Anda dapat menggunakannya sebagai ganti perhitungan yang saya tunjukkan di bagian sebelumnya. Bisa sih, tapi harus hati-hati.

Fungsi ini dapat memakan waktu hingga delapan argumen, tetapi saya hanya membahas tiga:

```
chisq.test(Observed,p=dpois(x.values,3),rescale.p=TRUE)
```

Argumen pertama adalah vektor data — nilai yang diamati. Yang kedua adalah vektor probabilitas yang diprediksi Poisson. Saya harus memasukkan p= karena itu bukan argumen kedua dalam daftar argumen yang dibutuhkan fungsi. Untuk alasan yang sama, saya menyertakan rescale.p= dalam argumen ketiga, yang memberi tahu fungsi untuk "menskalakan ulang" vektor probabilitas. Mengapa itu perlu? Satu persyaratan untuk fungsi ini adalah bahwa probabilitas harus berjumlah 1,00, dan probabilitas ini tidak:

```
> sum(dpois(x.values,3))
[1] 0.9880955
```

"Penskalaan ulang" mengubah nilai sehingga jumlahnya mencapai 1,00.
Saat Anda menjalankan fungsi itu, ini terjadi:

```
> chisq.test(Observed,p=dpois(x.values,3),rescale.p=TRUE)

        Chi-squared test for given probabilities

data:  Observed
X-squared = 3.4953, df = 7, p-value = 0.8357

Warning message:
In chisq.test(Observed, p = dpois(x.values, 3), rescale.p =
          TRUE) :
  Chi-squared approximation may be incorrect
```

Mari kita periksa outputnya. Pada baris sebelum pesan peringatan, perhatikan penggunaan X2 daripada 2. Ini karena nilai yang dihitung mendekati 2, dan bentuk serta tampilan X mendekati bentuk dan tampilan . Nilai X-kuadrat cukup dekat dengan nilai yang saya hitung sebelumnya, tetapi tidak aktif karena probabilitas yang diskalakan ulang.

Tapi masalah lain mengintai. Perhatikan bahwa df sama dengan 7 daripada nilai yang benar, 6, dan dengan demikian pengujian terhadap anggota keluarga x2 yang salah. Mengapa perbedaan itu? Karena chisq.test() tidak tahu bagaimana Anda sampai pada probabilitas. Tidak ada ide bahwa Anda harus menggunakan data untuk memperkirakan satu parameter (μ), dan dengan demikian kehilangan derajat kebebasan. Jadi selain pesan peringatan tentang pendekatan chi-kuadrat, Anda juga harus menyadari bahwa derajat kebebasan tidak benar untuk jenis contoh ini.

Kapan Anda akan menggunakan chisq.test()? Berikut ini contoh singkatnya: Anda melempar koin 100 kali dan muncul Kepala 65 kali. Hipotesis nol adalah bahwa koin itu adil. Keputusan Anda?

```
> chisq.test(c(65,35), p=c(.5,.5))

        Chi-squared test for given probabilities

data:  c(65, 35)
X-squared = 9, df = 1, p-value = 0.0027
```

Nilai p yang rendah memberitahu Anda untuk menolak hipotesis nol. Dalam Bab 20 saya tunjukkan aplikasi lain dari chisq.test().

Permainan Olah raga dengan model

Baseball adalah permainan yang menghasilkan statistik dalam jumlah besar — dan banyak orang mempelajari statistik ini dengan cermat. Society for American Baseball Research (SABR) telah muncul dari upaya sekelompok penggemar statistik (fantasi?) yang berdedikasi yang menyelidiki sudut dan celah statistik dari Great American Pastime. Mereka menyebut pekerjaan mereka sabermetrics. (Saya membuat "fantastician." Mereka menyebut diri mereka "sabermetricians.")

Alasan saya menyebutkan ini adalah bahwa sabermetrics menyediakan contoh pemodelan yang bagus. Ini didasarkan pada gagasan yang jelas bahwa selama pertandingan, tujuan tim bisbol adalah mencetak angka dan mencegah lawannya mencetak angka. Semakin baik tim melakukan kedua tugas, semakin banyak permainan yang dimenangkannya. Bill James, yang memberi sabermetrik namanya dan merupakan eksponen utamanya, menemukan hubungan yang rapi antara jumlah lari skor tim, jumlah lari yang diizinkan tim, dan persentase kemenangannya. Dia menyebutnya persentase Pythagoras:

$$\text{Pythagorean Percentage} = \frac{(\text{Runs Scored})^2}{(\text{Runs Scored})^2 + (\text{Runs Allowed})^2}$$

Kuadrat dalam ekspresi mengingatkan James pada teorema Pythagoras, oleh karena itu dinamakan “persentase Pythagoras.” Anggap saja sebagai model untuk memprediksi permainan yang dimenangkan. (Ini adalah formula asli James, dan saya menggunakannya sepanjang waktu. Selama bertahun-tahun, para ahli pedang telah menemukan bahwa 1,83 lebih akurat daripada 2.) Hitung persentase ini dan kalikan dengan jumlah permainan yang dimainkan tim. Kemudian bandingkan jawaban dengan kemenangan tim. Seberapa baik model memprediksi jumlah pertandingan yang dimenangkan setiap tim selama musim 2016?

Untuk mengetahuinya, saya menemukan semua data yang relevan (jumlah pertandingan yang dimenangkan dan kalah, skor lari, dan lari yang diperbolehkan) untuk setiap tim National League (NL) pada tahun 2016. (Terima kasih, www.baseball-reference.com.) Saya menempatkan data ke dalam bingkai data yang disebut NL2016.

```
> NL2016
   Team Won Lost Runs.scored Runs.allowed
1   ARI  69   93         752          890
2   ATL  68   93         649          779
3   CHC 103   58         808          556
4   CIN  68   94         716          854
5   COL  75   87         845          860
6   LAD  91   71         725          638
7   MIA  79   82         655          682
8   MIL  73   89         671          733
9   NYM  87   75         671          617
10  PHI  71   91         610          796
11  PIT  78   83         729          758
12  SDP  68   94         686          770
13  SFG  87   75         715          631
14  STL  86   76         779          712
15  WSN  95   67         763          612
```

Singkatan tiga huruf di kolom Tim menurut abjad mengurutkan tim NL dari ARI (Arizona Diamondbacks) ke WSN (Washington Nationals). (Saya sangat merasa bahwa angka yang jauh lebih tinggi di sebelah kanan NYM akan membuat dunia menjadi tempat yang lebih baik, tapi itu hanya saya).

Langkah selanjutnya adalah mencari persentase Pythagoras untuk setiap tim:

```
pythag <- with(NL2016,
          Runs.scored^2/(Runs.scored^2 + Runs.allowed^2))
```

Saya menggunakan with(), untuk menghindari keharusan mengetik ekspresi seperti NL2016$Runs.scored^2.

Kemudian, saya menemukan prediksi jumlah kemenangan:

```
Predicted.wins <- with(NL2016, pythag*(Won + Lost))
```

Ungkapan Menang + Kalah, tentu saja, memberikan jumlah permainan yang dimainkan setiap tim. Bukankah mereka semua memainkan jumlah permainan yang sama? Tidak. Terkadang sebuah pertandingan diguyur hujan dan kemudian tidak dijadwalkan ulang jika hasilnya tidak mempengaruhi klasemen akhir.

Yang tersisa hanyalah menemukan 2 dan mengujinya terhadap distribusi khi-kuadrat:

```
> chi.squared <- with(NL2016,
                sum((Won-Predicted.wins)^2/Predicted.wins))
> chi.squared
[1] 3.402195
```

Saya tidak menggunakan data Won di Kolom 2 untuk memperkirakan parameter apa pun, seperti mean atau varians, dan kemudian menerapkan parameter tersebut untuk menghitung prediksi kemenangan. Sebaliknya, prediksi datang dari data lain — Runs Scored dan Runs Allowed. Untuk alasan ini, df = k–m–1= 15–0–1 = 14. Pengujiannya adalah:

```
> pchisq(chi.squared,14,lower.tail=FALSE)
[1] 0.9981182
```

Seperti pada contoh sebelumnya, lower.tail=FALSE menunjukkan bahwa saya menginginkan area di sebelah kanan 3.40215 dalam distribusi (karena saya mencari probabilitas nilai yang ekstrim atau lebih tinggi).

Nilai p yang sangat tinggi memberi tahu Anda bahwa dengan 14 derajat kebebasan, Anda memiliki peluang besar untuk menemukan nilai 2 setidaknya setinggi X2 yang Anda hitung dari nilai-nilai yang diamati ini dan nilai-nilai yang diprediksi ini. Cara lain untuk mengatakan ini: Nilai X2 yang dihitung sangat rendah, artinya kemenangan yang diprediksi mendekati kemenangan yang sebenarnya. Intinya: Model sangat cocok dengan data.

Jika Anda seorang penggemar bisbol (seperti saya), sangat menyenangkan untuk mencocokkan Won dengan Predicted.wins untuk setiap tim. Ini memberi Anda gambaran tentang tim mana yang berkinerja lebih baik dan mana yang berkinerja buruk mengingat berapa banyak lari yang mereka cetak dan berapa banyak yang diizinkan. Dua ekspresi ini

```
NL2016["Predicted"]<-round(Predicted.wins)
NL2016["W-P"] <- NL2016["Won"]-NL2016["Predicted"]
```

buat kolom untuk Predicted dan kolom untuk W-P (Won-Predicted), masing-masing, dalam bingkai data. Ini adalah kolom keenam dan ketujuh.

Ekspresi ini

```
NL2016 <-NL2016[,c(1,2,6,7,3,4,5)]
```

menempatkan kolom keenam dan ketujuh di sebelah Won, untuk memudahkan perbandingan. (Jangan lupa koma pertama dalam ekspresi kurung di sebelah kanan).

Bingkai data sekarang

```
> NL2016
   Team Won Predicted W-P Lost Runs.scored Runs.allowed
1   ARI  69        67   2   93         752          890
2   ATL  68        66   2   93         649          779
3   CHC 103       109  -6   58         808          556
4   CIN  68        67   1   94         716          854
5   COL  75        80  -5   87         845          860
6   LAD  91        91   0   71         725          638
7   MIA  79        77   2   82         655          682
8   MIL  73        74  -1   89         671          733
9   NYM  87        88  -1   75         671          617
10  PHI  71        60  11   91         610          796
11  PIT  78        77   1   83         729          758
12  SDP  68        72  -4   94         686          770
13  SFG  87        91  -4   75         715          631
14  STL  86        88  -2   76         779          712
15  WSN  95        99  -4   67         763          612
```

Kolom W-P menunjukkan bahwa PHI (Philadelphia Phillies) mengungguli prediksi mereka dengan 11 pertandingan — dan itu adalah kinerja berlebih terbesar di National League pada tahun 2016. Siapa yang berkinerja buruk terbesar? Yang cukup menarik, itu adalah CHC (Chicago Cubs — enam pertandingan lebih buruk dari prediksi mereka). Namun, jika Anda mengikuti postseason 2016, Anda tahu mereka lebih dari sekadar menebus ini. . . .

18.2 DISKUSI SIMULASI

Pendekatan lain untuk pemodelan adalah untuk mensimulasikan proses. Idenya adalah untuk mendefinisikan sebanyak mungkin tentang apa yang dilakukan suatu proses dan kemudian entah bagaimana menggunakan angka untuk mewakili proses itu dan melaksanakannya. Ini adalah cara yang bagus untuk mengetahui apa yang dilakukan suatu proses jika metode analisis lain sangat kompleks.

Mengambil kesempatan: Metode Monte Carlo

Banyak proses mengandung unsur keacakan. Anda tidak bisa memprediksi hasilnya dengan pasti. Untuk mensimulasikan jenis proses ini, Anda harus memiliki beberapa cara untuk mensimulasikan keacakan. Metode simulasi yang menggabungkan keacakan disebut simulasi Monte Carlo. Nama tersebut berasal dari kota di Monaco yang daya tarik utamanya adalah kasino judi.

Dalam beberapa bagian berikutnya, saya menunjukkan beberapa contoh. Contoh-contoh ini tidak begitu rumit sehingga Anda tidak dapat menganalisisnya. Saya menggunakannya hanya untuk alasan itu: Anda dapat memeriksa hasil dengan analisis.

Memuat dadu

Dalam Bab 17, saya berbicara tentang dadu (satu anggota dari sepasang dadu) yang bias muncul sesuai dengan angka di wajahnya: A 6 adalah enam kali lebih mungkin dari 1, 5 adalah lima kali lebih mungkin, dan seterusnya. Pada sembarang pelemparan, peluang terambilnya bilangan n adalah n/21.

Misalkan Anda memiliki sepasang dadu yang dimuat dengan cara ini. Seperti apa hasil dari 2.000 pelemparan dadu ini? Berapa rata-rata dari 2.000 kali lemparan itu? Apa yang akan menjadi varians dan standar deviasi? Anda dapat menggunakan R untuk menyiapkan simulasi Monte Carlo dan menjawab pertanyaan-pertanyaan ini.

Saya mulai dengan menulis fungsi R untuk menghitung probabilitas dari setiap kemungkinan hasil. Sebelum saya mengembangkan fungsinya, saya akan menelusuri alasannya untuk Anda. Untuk setiap hasil (2-12), saya harus memiliki semua cara untuk menghasilkan hasil. Misalnya, untuk melempar 4, saya dapat memiliki 1 pada dadu pertama dan 3 pada dadu kedua, 2 pada dadu pertama dan 2 pada dadu kedua, atau 3 pada dadu pertama dan 1 pada dadu kedua. Probabilitas (saya menyebutnya dimuat.pr) dari 4, maka, adalah:

$$loaded.pr(4) = \left(\frac{1}{21} \times \frac{3}{21}\right) + \left(\frac{2}{21} \times \frac{2}{21}\right) + \left(\frac{3}{21} \times \frac{1}{21}\right) = \frac{(1\times 3) + (2\times 2) + (3\times 1)}{21^2} = .02267574$$

Daripada menghitung semua kemungkinan untuk setiap hasil dan kemudian menghitung probabilitas, saya membuat fungsi yang disebut dimuat.pr() untuk melakukan pekerjaan. Saya ingin ini berfungsi seperti ini:

```
> loaded.pr(4)
[1] 0.02267574
```

Pertama, saya mengatur fungsi:

```
loaded.pr <-function(x){
```

Selanjutnya, saya ingin menghentikan semuanya dan mencetak peringatan jika x kurang dari 2 atau lebih besar dari 12:

```
if(x <2 | x >12) warning("x must be between 2 and 12,
          inclusive")
```

Kemudian saya menetapkan variabel bernama first yang melacak nilai dadu pertama, tergantung pada nilai x. Jika x kurang dari 7, saya set dulu ke 1. Jika x adalah 7 atau lebih, saya set dulu ke 6 (nilai maksimum lemparan dadu):

```
if(x < 7) first=1
    else first=6
```

Variabel kedua (nilai dadu kedua), tentu saja, adalah x-pertama:

```
second = x-first
```

Saya ingin melacak jumlah pembilangnya (seperti dalam persamaan yang baru saja saya tunjukkan), jadi saya memulai nilainya dari nol:

```
sum = 0
```

Sekarang tibalah akhir bisnis: loop for yang melakukan penghitungan dengan memberikan nilai pertama (lemparan dadu pertama) dan kedua (lemparan dadu kedua):

```
for(first in first:second){
    second = x-first
    sum = sum + (first*second)
  }
```

Karena pernyataan if sebelumnya, jika x kurang dari 7, kenaikan pertama dari 1 ke x-1 dengan setiap iterasi dari perulangan for (dan penurunan kedua). Jika x adalah 7 atau lebih besar, penurunan pertama dari 6 menjadi x–6 pada setiap iterasi (dan kedua meningkat).
Akhirnya, ketika loop selesai, fungsi mengembalikan jumlah dibagi 21^2:

```
}
  return(sum/21^2)
}
```

Di sini semuanya bersama-sama:

```
loaded.pr <- function(x){
  if(x < 2 | x > 12) warning("x must be between 2 and 12,
           inclusive")
  if(x < 7) first=1
    else first=6
  second = x-first
  sum = 0
  for(first in first:second){
    second = x-first
    sum=sum + (first*second)
  }
  return(sum/21^2)
}
```

Untuk mengatur distribusi probabilitas, saya membuat vektor untuk hasil

```
outcome <- seq(2,12)
```

dan gunakan for loop untuk membuat vektor pr.outcome untuk menampung probabilitas yang sesuai:

```
pr.outcome <- NULL
for(x in outcome){pr.outcome <- c(pr.outcome,loaded.pr(x))}
```

Dalam setiap iterasi dari loop, pernyataan kurung kurawal di sebelah kanan menambahkan probabilitas yang dihitung ke vektor. Berikut adalah probabilitas yang dibulatkan ke tiga tempat desimal agar terlihat bagus di halaman:

```
> round(pr.outcome,3)
 [1] 0.002 0.009 0.023 0.045 0.079 0.127 0.159 0.172 0.166
           0.136 0.082
```

Dan sekarang saya siap untuk mengambil sampel secara acak 2.000 kali dari distribusi probabilitas diskrit ini — setara dengan 2.000 lemparan sepasang dadu yang dimuat. Fungsi pengacakan di R benar-benar "pseudorandom." Mereka mulai dari nomor "benih" dan bekerja dari sana. Jika Anda mengatur benih, Anda dapat menentukan jalannya pengacakan; jika Anda tidak mengaturnya, pengacakan akan berjalan dengan sendirinya setiap kali Anda menjalankannya.

Jadi saya mulai dengan menetapkan benih:

```
set.seed(123)
```

Ini tidak perlu, tetapi jika Anda ingin mereproduksi hasil saya, mulailah dengan fungsi itu dan nomor benih itu. Jika tidak, hasil Anda tidak akan terlihat persis seperti hasil saya (yang belum tentu merupakan hal yang buruk).

Untuk pengambilan sampel acak, saya menggunakan fungsi sample() dan menetapkan hasilnya ke hasil:

```
results <- sample(outcome,size = 2000,replace = TRUE,
          prob=pr.outcome)
```

Argumen pertama, tentu saja, adalah himpunan nilai untuk variabel (kemungkinan lemparan dadu), yang kedua adalah jumlah sampel, yang ketiga menentukan pengambilan sampel dengan penggantian, dan yang keempat adalah vektor probabilitas yang baru saja saya hitung. Untuk mereproduksi hasil yang tepat, ingatlah untuk menyetel benih itu sebelum setiap kali Anda menggunakan sample().

Berikut ini sekilas tentang distribusi hasil:

```
> table(results)
results
  2   3   4   5   6   7   8   9  10  11  12
  3  28  39  79 154 246 335 356 311 284 165
```

Baris pertama adalah hasil yang mungkin, dan yang kedua adalah frekuensi hasil. Jadi, 39 dari 2.000 lemparan menghasilkan 4, dan 165 di antaranya menghasilkan 12. Saya meninggalkannya sebagai latihan bagi Anda untuk membuat grafik hasil ini.

Bagaimana dengan statistik lemparan simulasi ini?

```
> mean(results)
[1] 8.6925
> var(results)
[1] 4.423155
> sd(results)
[1] 2.10313
```

Bagaimana nilai-nilai ini cocok dengan parameter variabel acak? Inilah yang saya maksud sebelumnya dengan "memeriksa terhadap analisis." Dalam Bab 17, saya menunjukkan bagaimana menghitung nilai yang diharapkan (mean), varians, dan deviasi standar untuk variabel acak diskrit.

Nilai yang diharapkan adalah:

$$E(x) = \sum x(pr(x))$$

Saya dapat menghitungnya dengan cukup mudah di R:

```
> E.outcome = sum(outcome*pr.outcome)
> E.outcome
[1] 8.666667
```

Variansnya adalah:

$$V(x) = \sum x^2(pr(x)) - [E(x)]^2$$

Di R, itu

```
> Var.outcome <- sum(outcome^2*pr.outcome)-E.outcome^2
> Var.outcome
[1] 4.444444
```

Standar deviasinya tentu saja

```
> sd.outcome <- sqrt(Var.outcome)
> sd.outcome
[1] 2.108185
```

Tabel 18.2 menunjukkan bahwa hasil simulasi sangat cocok dengan parameter variabel acak. Anda dapat mencoba mengulangi simulasi dengan lebih banyak lemparan yang disimulasikan — 10.000, mungkin. Apakah lemparan yang meningkat akan menarik statistik simulasi lebih dekat ke parameter distribusi?

Tabel 18.2 Statistik dari Simulasi Pelemparan Dadu Bermuatan dan Parameter Distribusi Diskrit

	Statistik Simulasi	Parameter Distribusi
Mean	8.6925	8.666667
Variance	4.423155	4.444444
Standar Deviasi	2.10313	2.108185

Mensimulasikan teorema limit pusat

Ini mungkin mengejutkan Anda, tetapi ahli statistik sering menggunakan simulasi untuk menentukan beberapa statistik mereka. Mereka melakukan ini ketika analisis matematis menjadi sangat sulit. Misalnya, beberapa uji statistik bergantung pada populasi yang terdistribusi normal. Jika populasinya tidak normal, apa yang terjadi pada tes itu? Apakah mereka masih melakukan apa yang seharusnya mereka lakukan? Untuk menjawab pertanyaan itu, ahli statistik dapat membuat populasi angka yang tidak terdistribusi secara normal, mensimulasikan eksperimen dengan mereka, dan menerapkan uji statistik pada hasil simulasi.

Pada bagian ini, saya menggunakan simulasi untuk menguji item statistik penting: teorema limit pusat. Dalam Bab 9, saya memperkenalkan teorema ini sehubungan dengan distribusi sampling mean. Sebenarnya, saya mensimulasikan pengambilan sampel dari suatu populasi dengan hanya tiga kemungkinan nilai untuk menunjukkan kepada Anda bahwa bahkan dengan ukuran sampel yang kecil, distribusi pengambilan sampel mulai terlihat terdistribusi secara normal.

Di sini, saya membuat populasi yang terdistribusi normal dan menggambar 10.000 sampel yang masing-masing terdiri dari 25 skor. Saya menghitung rata-rata setiap sampel dan kemudian mengatur distribusi 10.000 cara tersebut. Idenya adalah untuk melihat bagaimana statistik distribusi itu cocok dengan prediksi teorema limit pusat.

Populasi untuk contoh ini memiliki parameter populasi skor pada tes IQ, distribusi yang saya gunakan sebagai contoh dalam beberapa bab. Ini adalah distribusi normal dengan 100 dan 15. Menurut teorema limit pusat, rata-rata distribusi rata-rata (distribusi sampling rata-rata) harus 100, dan standar deviasi (kesalahan standar rata-rata) harus 3 — simpangan baku populasi (15) dibagi dengan akar kuadrat dari ukuran sampel (5). Teorema limit pusat juga memprediksikan bahwa distribusi sampling dari mean terdistribusi normal.

Fungsi rnorm() melakukan pengambilan sampel. Untuk satu sampel dari 25 angka dari populasi yang terdistribusi normal dengan rata-rata μ = 100 dan standar deviasi σ = 15, fungsinya adalah:

```
rnorm(25,100,15)
```

dan jika saya ingin sampelnya berarti, itu

```
mean(rnorm(25,100,15))
```

Saya akan meletakkan fungsi itu di dalam for loop yang berulang 10.000 kali dan menambahkan setiap mean sampel yang baru dihitung ke vektor yang disebut sampling.distribution, yang saya inisialisasi:

```
sampling.distribution <- NULL
```

Perulangan untuk adalah

```
for(sample.count in 1:10000){
  set.seed(sample.count)
  sample.mean <- mean(rnorm(25,100,15))
  sampling.distribution <- c(sampling.distribution,sample.mean)
}
```

Sekali lagi, pernyataan set.seed() diperlukan hanya jika Anda ingin mereproduksi hasil saya. Bagaimana dengan statistik distribusi sampling?

```
> mean(sampling.distribution)
[1] 100.029
> sd(sampling.distribution)
[1] 3.005007
```

Cukup dekat dengan nilai prediksi! Pastikan untuk mereset sampling.distribution ke NULL sebelum setiap kali Anda menjalankan for loop.

Seperti apa distribusi samplingnya? Agar semuanya terlihat bersih, saya membulatkan sampel berarti di sampling.distribution dan kemudian membuat tabel:

```
table(round(sampling.distribution))
```

Saya akan menunjukkan kepada Anda tabelnya, tetapi angka-angkanya menjadi acak-acakan di halaman. Sebagai gantinya, saya akan melanjutkan dan menggunakan ggplot() untuk membuat grafik distribusi sampling.
Pertama, saya membuat bingkai data

```
sampling.frame <- data.frame(table(round(sampling.
           distribution)))
```

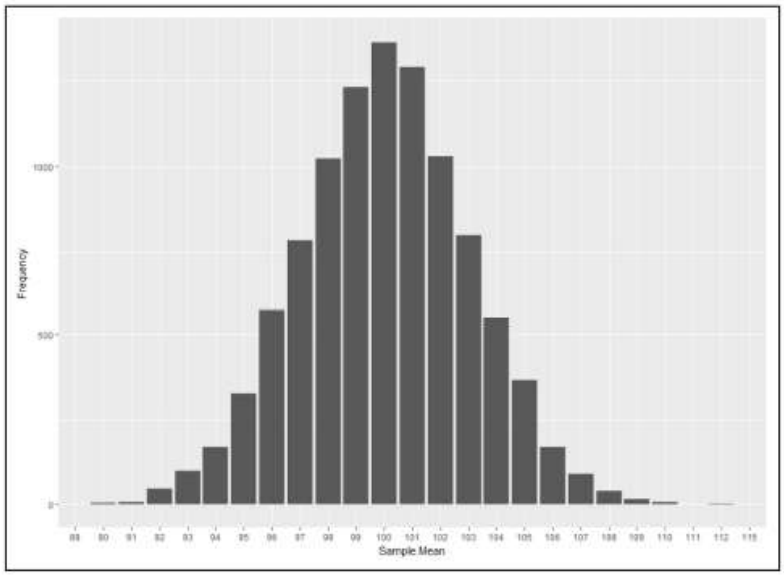
dan tentukan nama kolomnya:

```
colnames(sampling.frame) <- c("Sample.Mean","Frequency")
```

Sekarang untuk plotnya:

```
ggplot(sampling.frame,aes(x=Sample.Mean,y=Frequency))+
  geom_bar(stat="identity")
```

Hasilnya ditunjukkan pada Gambar 18.3, sebuah plot yang mendekati bentuk dan simetri dari distribusi normal.



Gambar 18.3 Distribusi sampel rata-rata (N = 25) berdasarkan 10.000 sampel dari distribusi normal dengan μ = 100 dan σ = 15.

BAGIAN 5
BAGIAN DARI PULUHAN
BAB 19
SEPULUH TIPS UNTUK EXCEL EMIGRÉS

Excel, program spreadsheet yang paling banyak digunakan, memiliki serangkaian alat analisis statistik yang mengesankan. Meskipun beberapa orang mencirikan Excel sebagai perangkat lunak analisis Rodney Dangerfield ("jangan tidak dihormati!"), Banyak orang menggunakan alat analisis Excel. (Dan percayalah, tidak ada yang lebih bahagia tentang itu daripada saya!)

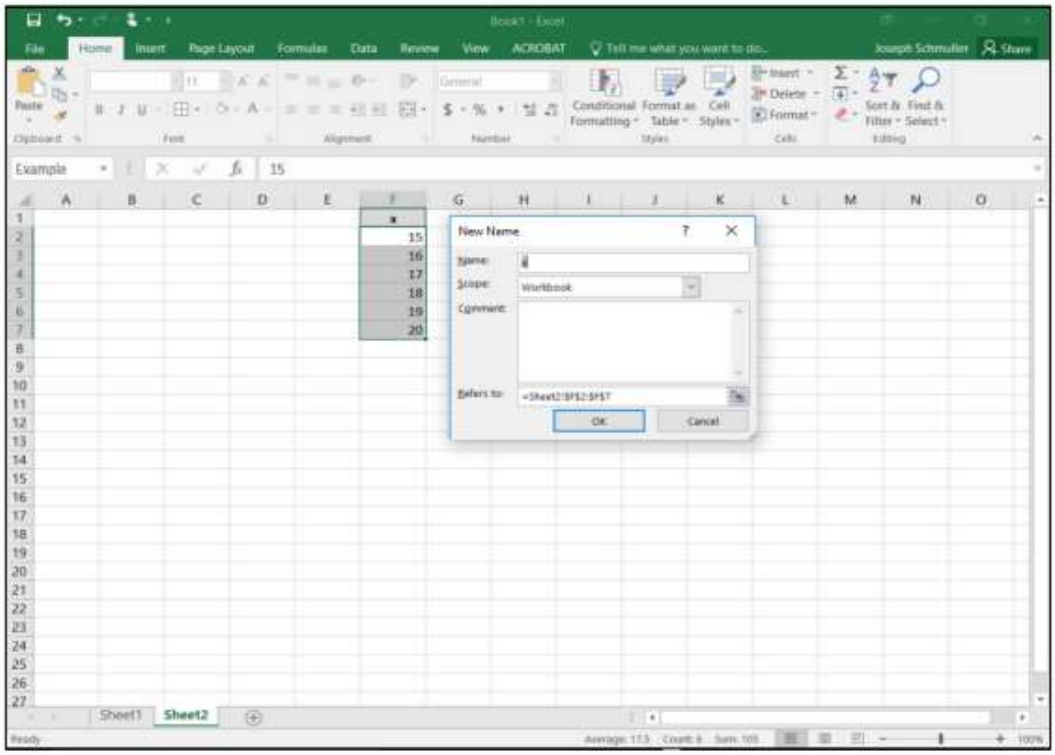
Jika Anda salah satu dari orang-orang itu dan Anda memerlukan sedikit bantuan untuk bertransisi ke R, bab ini cocok untuk Anda. Saya menunjukkan persamaan dan perbedaan yang mungkin membantu Anda membuat lompatan.

19.1 MENDEFINISIKAN VEKTOR DI R SEPERTI MEMBERI NAMA RENTANG DI EXCEL

Berikut adalah vektor varietas taman sehari-hari standar di R:

```
x <- c(15,16,17,18,19,20)
```

Jika Anda terbiasa menamai array di Excel, Anda sudah melakukan sesuatu seperti ini.



Gambar 19.1 Rentang di Excel, yang akan diberi nama x.

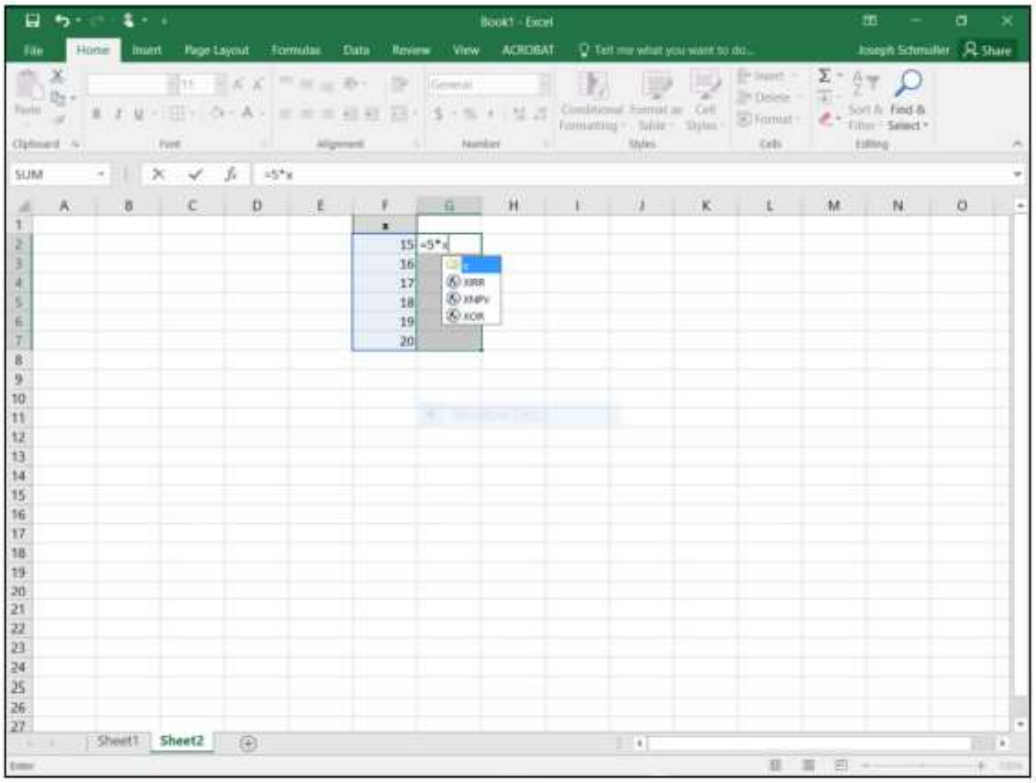
Gambar 19.1 menunjukkan spreadsheet dengan angka-angka ini dalam sel F2 sampai F7 dan dipimpin oleh x di F1. Gambar juga menunjukkan kotak dialog Nama Baru yang terbuka saat saya menyorot rentang tersebut, klik kanan, dan pilih Tentukan Nama dari menu yang muncul. Mengklik OK mendefinisikan x sebagai nama rentang itu, sama seperti pernyataan R membuat vektor x. Apa? Anda tidak memberi nama rentang di Excel?

19.2 BEROPERASI PADA VEKTOR SEPERTI BEROPERASI PADA RENTANG BERNAMA

Saya dapat mengalikan vektor x dengan konstanta:

```
> 5*x
[1]  75  80  85  90  95 100
```

Kembali ke spreadsheet dengan rentang bernama x. Saya memilih rentang sel dengan panjang yang sama dengan x — katakanlah, G2 hingga G —, dan ketik = 5*x di G2. Gambar 19.2 menunjukkan hal ini.



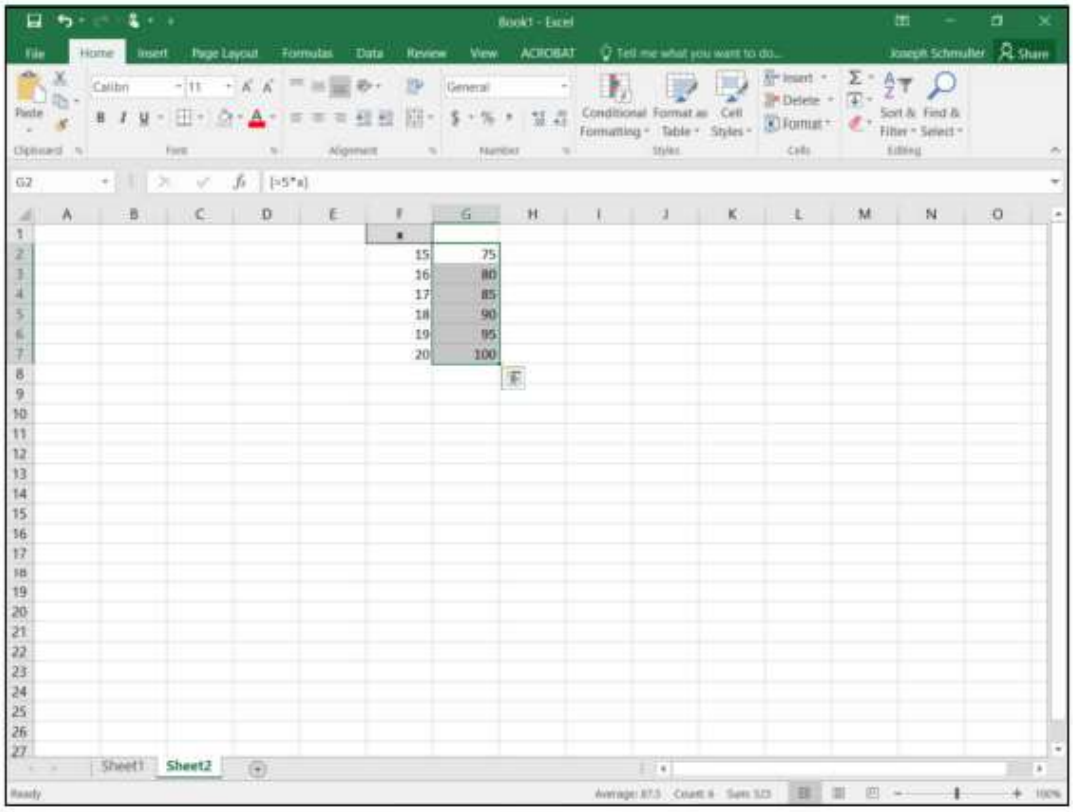
Gambar 19.2 Mengalikan rentang bernama x dengan 5.

Menekan kombinasi tombol Ctrl+Shift+Enter menempatkan hasilnya di G2 hingga G7, seperti yang ditunjukkan Gambar 19-3. Kombinasi tombol tersebut untuk fungsi larik di Excel — fungsi yang mengembalikan jawaban dalam larik sel, bukan dalam satu sel. Tentu saja, cara lain untuk melakukan perkalian adalah dengan mengetikkan =5*x ke G2, tekan Enter, lalu isi otomatis ke G7.

Kesamaannya berlimpah. Di R,

```
> sum(x)
[1] 105
```

menjumlahkan angka dalam x, seperti halnya =SUM(x) yang diketik ke dalam sel yang dipilih.



Gambar 19.3 Hasil perkalian kembali dalam bentuk larik.

Untuk menjumlahkan kuadrat dari angka-angka di x:

```
> sum(x^2)
[1] 1855
```

Di spreadsheet, pilih sel dan ketik =SUMSQ(x). Jika saya memiliki vektor lain y

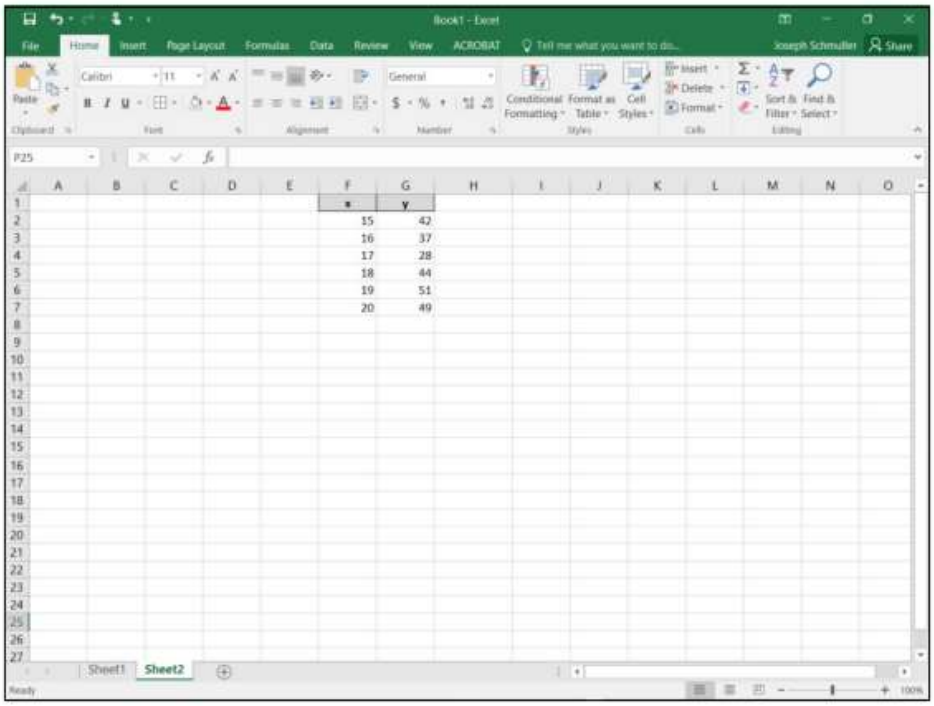
```
y <- c(42,37,28,44,51,49)
```

Kemudian

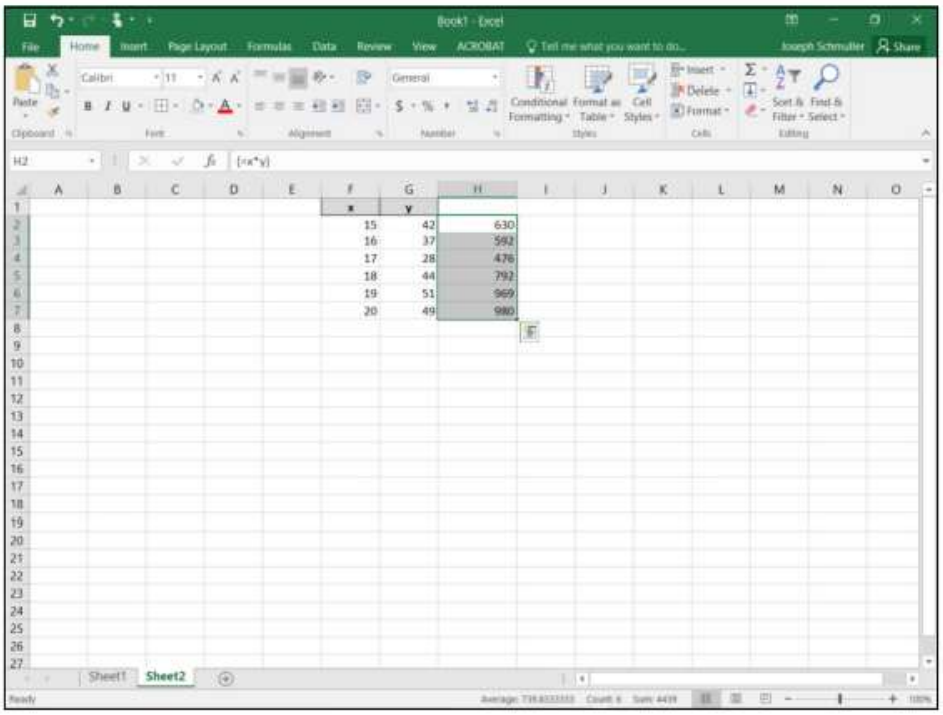
```
> x*y
[1] 630 592 476 792 969 980
```

Pada spreadsheet, saya dapat memiliki rentang bernama lain yang disebut y dalam sel G2 hingga G7, seperti pada Gambar 19.4. Memilih rentang seperti H2 hingga H7, mengetik =x*y,

dan menekan Ctrl+Shift+Enter menempatkan jawaban dalam larik yang dipilih, seperti yang ditunjukkan Gambar 19.5.



Gambar 19.4 Sebuah spreadsheet dengan dua array bernama, x dan y.



Gambar 19.5 Hasil perkalian dua array bernama.

Terkadang Fungsi Statistik Bekerja dengan Cara yang Sama. . .

Untuk mencari korelasi antara vektor x dan y di R:

```
> cor(x,y)
[1] 0.5900947
```

Untuk rentang bernama x dan y di spreadsheet, pilih sel dan masukkan

```
=CORREL(x,y)
```

Jawabannya muncul di sel yang dipilih.

. . . Dan Terkadang Mereka Tidak

Jika x dan y mewakili data dari dua kelompok, uji-t tepat untuk menguji perbedaan antara rata-rata. (Lihat Bab 11.)

Jika saya melakukan tes itu di R:

```
> t.test(x,y,alternative="two.sided",var.equal=FALSE)

        Welch Two Sample t-test

data:  x and y
t = -6.9071, df = 5.492, p-value = 0.000663
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -33.15068 -15.51598
sample estimates:
mean of x mean of y
 17.50000  41.83333
```

Argumen ketiga untuk t.test menentukan pengujian dua arah, dan argumen keempat menunjukkan bahwa kedua varians tidak sama. (Nilai untuk dua argumen terakhir adalah kondisi default, jadi tidak perlu menyatakannya.) Seperti yang Anda lihat, fungsi R's t.test() memberi Anda laporan lengkap.

Tidak demikian di Excel. Pilih sel dan masukkan

```
=T.TEST(x,y,2,3)
```

Argumen ketiga, 2, berarti ini adalah tes dua sisi. Argumen keempat, 3, menentukan varians yang tidak sama. Tekan Enter dan yang Anda dapatkan hanyalah nilai-p.

19.3 KONTRAS: EXCEL DAN R BEKERJA DENGAN FORMAT DATA BERBEDA

Di seluruh buku, saya membedakan antara format lebar

```
> wide.format
   x  y
1 15 42
2 16 37
3 17 28
4 18 44
5 19 51
6 20 49
```

dan format panjang

```
> long.format
   Group Score
1      x    15
2      x    16
3      x    17
4      x    18
5      x    19
6      x    20
7      y    42
8      y    37
9      y    28
10     y    44
11     y    51
12     y    49
```

Excel bekerja dengan format lebar.

Jika Anda bekerja dengan Excel 2011 untuk Mac (atau versi Mac yang lebih lama), Anda mungkin telah menginstal StatPlus:mac LE, add-in pihak ketiga yang menyediakan banyak alat analisis statistik untuk Excel versi Mac. StatPlus bekerja dengan data format panjang.

R, sebagian besar, menggunakan format panjang. Misalnya, fungsi t.test() yang baru saja saya tunjukkan kepada Anda juga dapat berfungsi seperti ini:

```
> t.test(Score ~ Group, alternative="two.sided", var.
  equal=FALSE, data=long.format)

        Welch Two Sample t-test

data:  Score by Group
t = -6.9071, df = 5.492, p-value = 0.000663
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -33.15068 -15.51598
sample estimates:
mean in group x mean in group y
       17.50000        41.83333
```

Perhatikan bahwa outputnya sama kecuali untuk data: Skor berdasarkan Grup daripada data: x dan y seperti pada contoh sebelumnya. Baris berikutnya-ke-terakhir juga sedikit berbeda.

19.4 FUNGSI DISTRIBUSI (AGAK) MIRIP

Baik Excel dan R memiliki fungsi bawaan yang bekerja dengan keluarga distribusi (seperti normal dan binomial). Karena R dikhususkan untuk pekerjaan statistik, ia memiliki fungsi untuk lebih banyak keluarga distribusi daripada Excel. Saya akan menunjukkan kepada Anda bagaimana keduanya bekerja dengan keluarga normal, dan Anda akan melihat kesamaannya.

Dalam distribusi normal dengan mean = 100 dan standar deviasi = 15, jika saya ingin mencari kepadatan yang terkait dengan 110 di Excel, itu =NORM.DIST(110,100,15,FALSE). Argumen keempat, FALSE, menunjukkan fungsi kepadatan. Di R, Anda akan menggunakan

```
> dnorm(110,100,15)
[1] 0.02129653
```

Untuk probabilitas kumulatif 110 dalam distribusi itu =NORM.DIST(110,100,15,TRUE) . Di sini, TRUE menunjukkan fungsi distribusi kumulatif. Versi R adalah

```
> pnorm(110,100,15)
[1] 0.7475075
```

Untuk mencari skor pada persentil ke-25 di Excel, saya menggunakan fungsi NORM.INV: =NORM.INV(0.25,100,15). Dan di R:

```
> qnorm(.25,100,15)
[1] 89.88265
```

Satu perbedaan: R memiliki fungsi untuk menghasilkan bilangan acak dari distribusi ini:

```
> rnorm(5,100,15)
[1] 85.06302 84.40067 99.73030 98.01737 61.75986
```

Untuk melakukan ini di Excel, Anda harus menggunakan alat Pembuatan Angka Acak di add-in yang disebut Alat Analisis Data.

19.5 BINGKAI DATA ADALAH (SESUATU) SEPERTI RENTANG BERNAMA MULTIKOLOM

Untuk bagian ini dan selanjutnya, saya menggunakan spreadsheet yang menyimpan larik multikolom yang sesuai dengan kerangka data NL2016 di Bab 20. Berikut kerangka datanya:

```
> NL2016
   Team Won Lost Runs.scored Runs.allowed
1   ARI  69   93          752          890
2   ATL  68   93          649          779
3   CHC 103   58          808          556
4   CIN  68   94          716          854
5   COL  75   87          845          860
6   LAD  91   71          725          638
7   MIA  79   82          655          682
8   MIL  73   89          671          733
9   NYM  87   75          671          617
10  PHI  71   91          610          796
11  PIT  78   83          729          758
12  SDP  68   94          686          770
13  SFG  87   75          715          631
14  STL  86   76          779          712
15  WSN  95   67          763          612
```

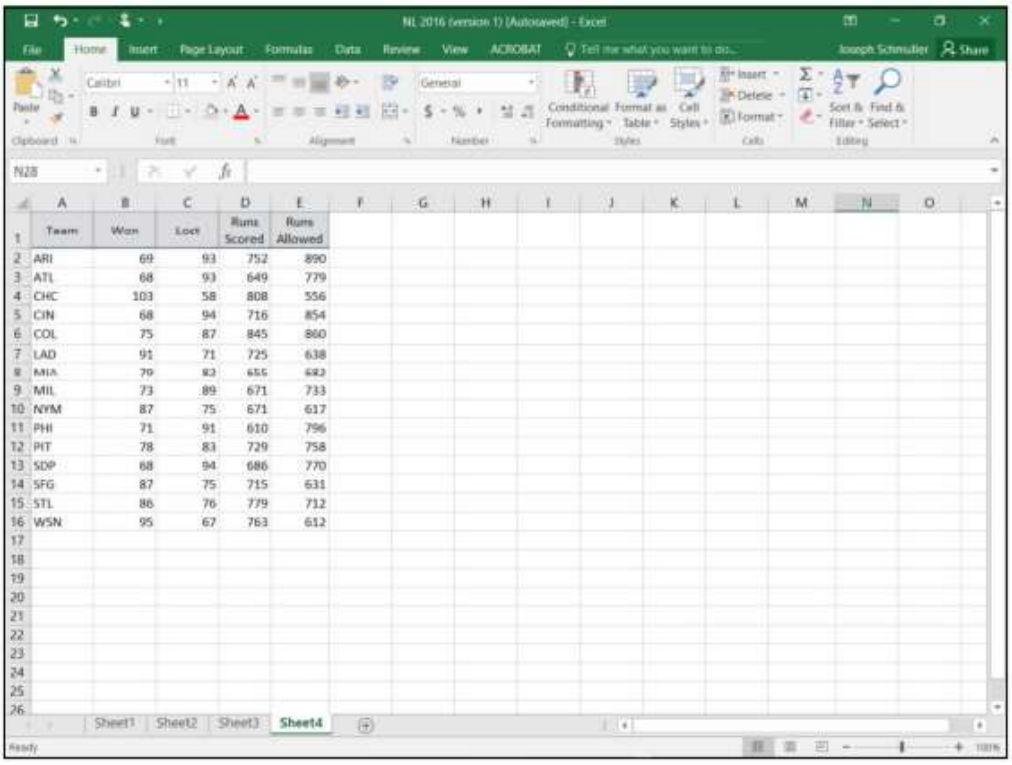
Gambar 19.6 menunjukkan spreadsheet. Saya telah mendefinisikan NL_2016 sebagai nama untuk seluruh tabel (sel A2 hingga E16). Di R, saya dapat menemukan rata-rata Runs.skor dengan cara ini:

```
> mean(NL2016[,4])
[1] 718.2667
```

Runs.scored ada di kolom 4, dan koma di dalam kurung siku menentukan semua baris di kolom itu. Di spreadsheet, saya memilih sel dan masuk

```
=AVERAGE(INDEX(NL_2016,,4))
```

Dua koma dalam tanda kurung menentukan semua baris di kolom 4.
Saya tahu saya tahu. Anda dapat melakukannya dengan beberapa cara lain di R dan Excel. Saya hanya mencoba menunjukkan kesamaan.



	A	B	C	D	E
1	Team	Won	Lost	Runs Scored	Runs Allowed
2	ARI	69	93	752	890
3	ATL	68	93	649	779
4	CHC	103	58	808	556
5	CIN	68	94	716	854
6	COL	75	87	845	860
7	LAD	91	71	725	638
8	MIA	79	83	655	682
9	MIL	73	89	671	733
10	NYM	87	75	671	617
11	PHI	71	91	610	796
12	PIT	78	83	729	758
13	SDP	68	94	686	770
14	SFG	87	75	715	631
15	STL	86	76	779	712
16	WSN	95	67	783	612

Gambar 19.6 Bingkai data NL2016 dalam spreadsheet. Nama Excel-nya adalah NL_2016.

"Beberapa cara lain" membuat segalanya berantakan. Misalnya, Excel tidak memiliki analogi dengan

```
> mean(NL2016$Runs.scored)
[1] 718.2667
```

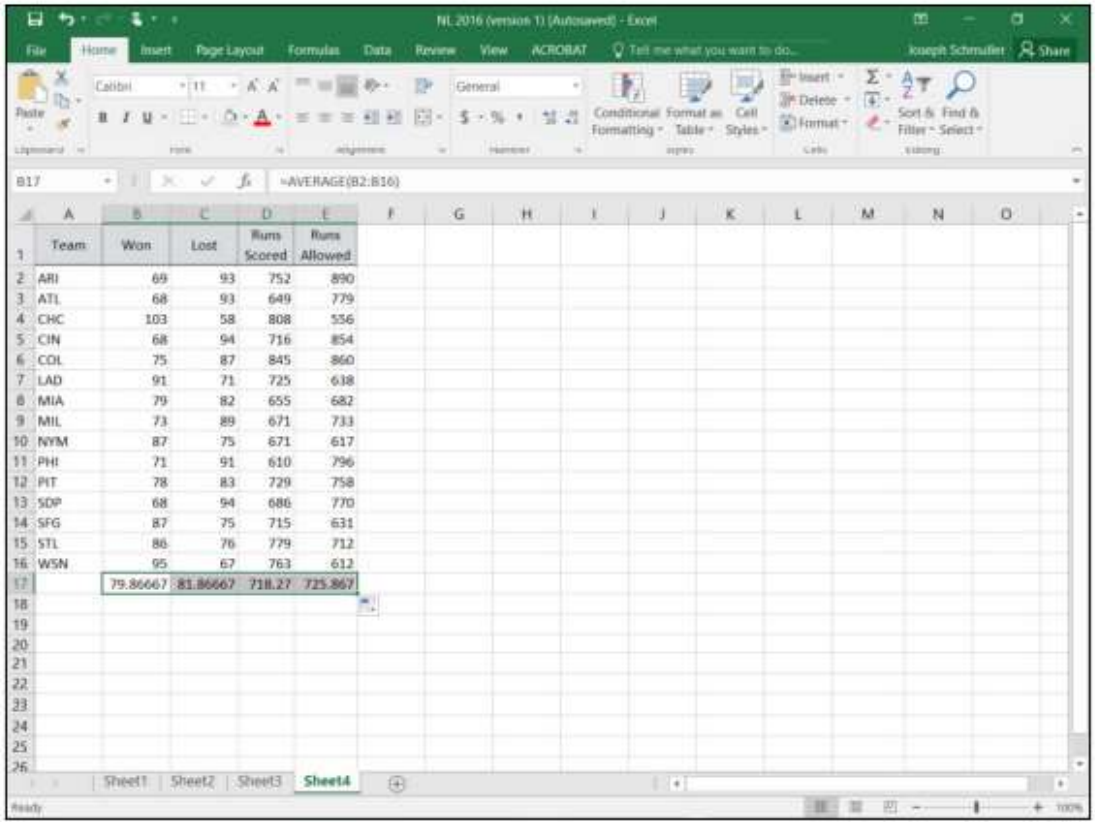
Fungsi sapply() Seperti Menyeret

Untuk menemukan semua kolom berarti di NL_2016 di Excel, saya dapat memilih sel B17 di bagian bawah kolom kedua dan masuk

=AVARAGE(B2:B16)

lalu seret melalui kolom ketiga, keempat, dan kelima. Gambar 19-7 menunjukkan hasil dragging. Untuk menghitung kolom tersebut berarti dalam R:

```
> sapply(NL2016[,2:5],mean)
         Won         Lost  Runs.scored Runs.allowed
    79.86667     81.86667    718.26667    725.86667
```



Gambar 19.7 Rata-rata melintasi kolom dengan menyeret dari kolom pertama.

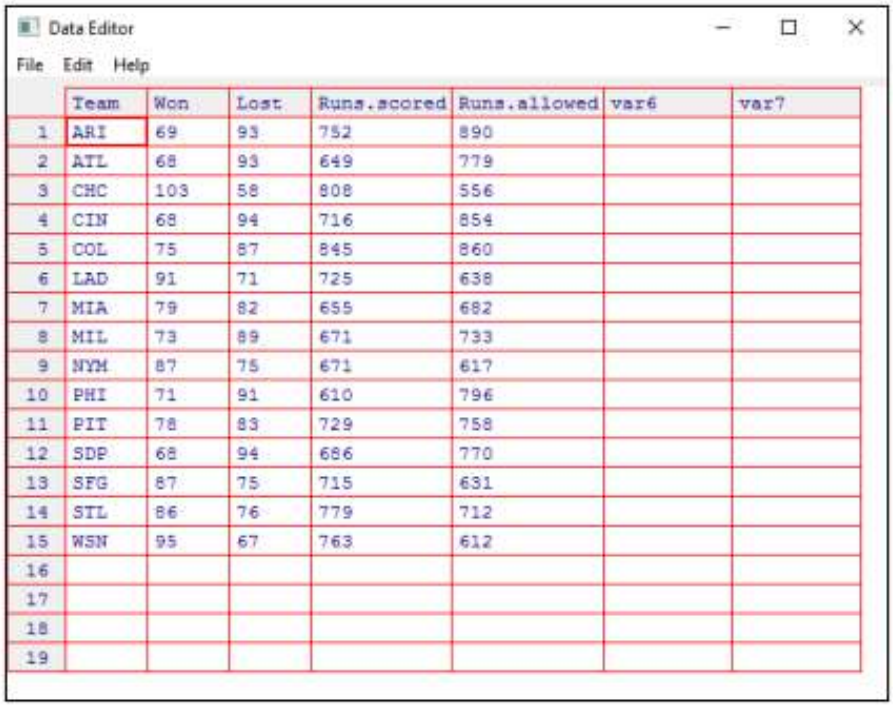
Menggunakan edit() Adalah (Hampir) Seperti Mengedit Spreadsheet

Dalam Bab 2, saya menyebutkan bahwa edit() membuka tampilan seperti spreadsheet ("spreadsheetsque"?) dari bingkai data. Ini sepertinya tempat yang bagus untuk membahasnya lagi, terutama jika Anda terbiasa dengan spreadsheet dan Anda merasa sulit untuk membuat perubahan pada bingkai data di R. Untuk mengubah bingkai data, saya menetapkannya dengan nama lain dan membukanya dengan edit():

```
> NL2016.changed <- edit(NL2016)
```

Ini akan membuka jendela Data Editor pada Gambar 19.8.

Sekarang saya bisa membuat perubahan. Misalnya, untuk memanjakan angan-angan saya, saya mengubah NYM's Won dari 87 menjadi 107, dan Lost dari 75 menjadi 55. Untuk melakukan itu, saya mengklik dua kali sel yang sesuai, membuat perubahan, dan memilih File Close dari menu utama .



	Team	Won	Lost	Runs.scored	Runs.allowed	var6	var7
1	ARI	69	93	752	890		
2	ATL	68	93	649	779		
3	CHC	103	58	808	556		
4	CIN	68	94	716	854		
5	COL	75	87	845	860		
6	LAD	91	71	725	638		
7	MIA	79	82	655	682		
8	MIL	73	89	671	733		
9	NYM	87	75	671	617		
10	PHI	71	91	610	796		
11	PIT	78	83	729	758		
12	SDP	68	94	686	770		
13	SFG	87	75	715	631		
14	STL	86	76	779	712		
15	WSN	95	67	763	612		
16							
17							
18							
19							

Gambar 19.8 Jendela Editor Data R.

Ketika saya membuka baris kesembilan dari bingkai data yang baru dinamai, saya melihat data yang jauh lebih enak, meskipun sayangnya tidak realistis:

```
> NL2016.changed[9,]
  Team Won Lost Runs.scored Runs.allowed
9  NYM 107   55         671          617
```

19.6 GUNAKAN CLIPBOARD UNTUK MENGIMPOR TABEL DARI EXCEL KE R

Jadi, Anda ingin menggunakan R untuk menganalisis data Anda, tetapi data Anda sebagian besar berada di lembar bentang. Apa yang kamu kerjakan?

Dalam Bab 2, saya menjelaskan paket xlsx. Paket ini menyediakan read.xlsx(), yang memungkinkan Anda membaca spreadsheet ke dalam R. Untuk menggunakan fungsi ini, Anda harus mengetahui direktori mana spreadsheet berada dan halaman spreadsheet mana yang ingin Anda impor. Tapi inilah cara termudah untuk mengimpor tabel Excel ke dalam bingkai data R: Salin tabel (ke clipboard) lalu gunakan

```
read.table("clipboard", header = TRUE)
```

Argumen kedua menetapkan bahwa baris pertama tabel berisi header kolom. Agar teknik ini berfungsi, Anda tidak boleh memiliki spasi pada nama di header kolom Anda. Misalkan saya ingin membawa tabel pada Gambar 19.4 ke R. Saya memilih sel F1 sampai G7 dan tekan Ctrl+C untuk menyalin sel yang dipilih ke clipboard.

Kemudian di R:

```
> clip.frame <-read.table("clipboard", header = TRUE)
> clip.frame
   x  y
1 15 42
2 16 37
3 17 28
4 18 44
5 19 51
6 20 49
```

dan Anda memiliki bingkai data sendiri.

Bagaimana Anda bisa yakin itu adalah bingkai data? Fungsi is.data.frame() mengembalikan TRUE jika argumennya adalah bingkai data; SALAH, jika tidak:

```
> is.data.frame(clip.frame)
[1] TRUE
```

BAB 20
SEPULUH SUMBER DAYA R ONLINE BERHARGA

Salah satu alasan peningkatan pesat R adalah komunitas R yang mendukung. Tampaknya begitu seseorang menjadi mahir dalam R, mereka segera ingin berbagi pengetahuan dengan orang lain dan web adalah tempat untuk melakukannya. Bab ini mengarahkan Anda ke beberapa sumber daya berbasis web yang bermanfaat yang telah dibuat oleh komunitas R.

20.1 SITUS WEB UNTUK PENGGUNA R

Saat Anda bekerja dengan R, Anda mungkin mengalami satu atau dua situasi yang memerlukan bantuan ahli. Situs web di bagian ini dapat memberikan bantuan yang Anda butuhkan.

R-blogger

Saat saya menulis ini, situs web R-bloggers terdiri dari upaya 750 blogger R. Pada saat Anda mengunjungi www.r-bloggers.com/, jumlah ini pasti akan lebih besar. Statistik Ph.D. kandidat Tal Galili menjalankan pertunjukan. Seperti yang dia katakan, tujuannya adalah untuk memberdayakan blogger R untuk memberdayakan pengguna R. Selain blog, Anda akan menemukan tautan ke kursus, konferensi, dan peluang kerja.

Jaringan Aplikasi Microsoft R

Sekali waktu, situs hebat bernama Inside-R menyediakan berbagai sumber daya untuk pengguna R. Baru-baru ini, Microsoft mengakuisisi perusahaan induk Inside-R, Revolution Analytics. Salah satu hasil dari akuisisi ini adalah Microsoft R Application Network, (MRAN) di mana Anda akan menemukan semua blog dan tautan yang dulu berada di Inside-R.Untuk mengunjungi MRAN, arahkan browser Anda ke https://mran.microsoft.com/. Hasil lain dari akuisisi tersebut adalah Microsoft R Open, yang disebut Microsoft sebagai distribusi R yang "ditingkatkan". Anda dapat mengunduh Microsoft R Open dari situs web MRAN.

Cepat-R

Profesor Universitas Wesleyan Rob Kabacoff membuat situs web ini untuk memperkenalkan Anda pada R dan penerapannya pada konsep statistik, baik pengantar maupun lanjutan. Anda akan menemukan konten yang ditulis dengan sangat baik (dan grafik yang rapi!) di www.statmethods.net/.

Pembelajaran Online RStudio

Orang-orang hebat di balik RStudio telah membuat halaman pembelajaran online yang menautkan ke tutorial dan contoh untuk membantu Anda menguasai R dan alat terkait — dan Anda juga dapat mempelajari dasar-dasar ilmu data. URL-nya adalah www.rstudio.com/online-learning/.

Stack Overflow

Tidak terbatas pada R, Stack Overflow adalah komunitas pemrogram dengan jutaan anggota yang berdedikasi untuk saling membantu. Anda dapat mencari basis Q&A mereka untuk bantuan atas suatu masalah, atau Anda dapat mengajukan pertanyaan. Namun, untuk mengajukan pertanyaan, Anda harus menjadi anggota (gratis) dan masuk. Situs ini juga menyediakan tautan ke pekerjaan, dokumentasi, dan banyak lagi. Tidak mengherankan, situs webnya ada di http://stackoverflow.com/.

20.2 BUKU DAN DOKUMENTASI ONLINE

Web memiliki banyak buku dan dokumen yang akan membantu Anda mendapatkan informasi terbaru tentang R. Salah satu cara untuk menautkannya adalah dengan mengeklik tombol Beranda pada tab Bantuan di RStudio.

Berikut adalah beberapa sumber lagi.

R manual

Jika Anda ingin langsung ke sumbernya, kunjungi halaman manual R di https://cran.rproject.org/manuals.html. Di situlah Anda akan menemukan tautan ke Definisi Bahasa R dan dokumentasi lainnya.

Dokumentasi R

Untuk tautan ke dokumentasi R lainnya, coba https://www.r-project.org/other-docs.html.

Dokumentasi

Tunggu. Bukankah saya baru saja menggunakan judul ini? Ya, baik. . . Liga Sepak Bola Kanada pernah memiliki tim bernama Rough Riders dan yang lain bernama Roughriders. Itu sesuatu seperti itu. Halaman RDocumentation di www.rdocumentation.org/ agak berbeda dengan halaman web di bagian sebelumnya. Yang ini tidak menautkan ke manual dan dokumen lainnya. Sebagai gantinya, situs web ini memungkinkan Anda untuk mencari paket dan fungsi R yang sesuai dengan kebutuhan Anda. Berapa banyak paket yang tersedia? Lebih dari 12.000!

ANDA CANanalytics

Gagasan Roopham Upadhyay, situs web YOU CANanalytics menyediakan sejumlah blog dan studi kasus yang bermanfaat, dan bisa masuk ke bagian utama pertama. Kenapa ada di yang ini? Karena halaman ini http://ucanalytics.com/blogs/learn-r-12-books-and-online-resources/ memungkinkan Anda mengunduh buku R klasik dalam format PDF. Beberapa judul ada di level pengantar, ada yang lanjutan, dan semuanya gratis!

Buku dalam format PDF adalah dokumen yang sangat panjang. Jika Anda membacanya di tablet, lebih mudah untuk mengubah file PDF menjadi e-book. Untuk melakukannya, unggah dokumen PDF Anda ke e-reader seperti Google Playbooks, dan voila — file PDF Anda menjadi e-book.

Jurnal R

Saya menyimpan yang ini untuk yang terakhir, karena ini pada tingkat lanjutan. Seperti publikasi akademis, The R Journal direferensikan — para ahli di bidangnya memutuskan apakah artikel yang dikirimkan layak untuk dipublikasikan. Lihatlah artikel di https://journal.r-project.org/ dan Anda akan melihat apa yang tersedia untuk Anda saat Anda menjadi salah satu pakar tersebut.

DAFTAR PUSTAKA

Advantages and Disadvantages. Management Information System 3(2): 029-033.

Andrie de Vries dan Joris Meys, 2012, R for Dummies, John Willey & Sons

Assaad, H. I., L. Zhou, R. J. Carroll, and G. Wu. 2014. Rapid Publication-Ready MS-Word Tables for One- way ANOVA. SpringerPlus 3(474):1-8.

Balachanthur, B., E. V. R. College, and T. Nadu. 2014. Open Source Softwares for Application Design and Development. International Journal of Advanced Multidisciplinary Research 1(1):73-85.

Benestad, R.E., A. Mezghani, and K.M. Parding. 2015. Esd - The Emperical - Statistical Downscaling tool and its visualisation capabilities. Met Report. Norwegian Meteorologi Institute. Oslo. Norwegian Meteorologi Institute . 55-58.

Brenning, A and D. Bangs. 2015. Introduction to Terrain Analysis with RSAGA : Landslide Susceptibility Modeling. Cran.r-project. pp.1–9.

Brenning, A. 2008. Statistical Geocomputing Combining R and Saga: T He E Xample Of L Andslide Susceptibility Analysis With Generalized Additive Models, Hamburger Beiträge zur Physischen Geographie und 24 Landschaftsökologie – Heft 19. pp. 23–32.

Chamber, J. 2008. Software for Data Analysis. Springer statistical and computing. New Yorl : Springer-Verlag New York. 1-10.

Conrad, O., B. Bechtel, M. Bock, H. Dietrich, E. Fischer, L. Gerlitz, and J. Böhner. 2015. System for Automated Geoscientific Analyses (SAGA) v . 2 .1 . 4, 1991–2007. Geosci.Model Dev., 8, 1991-2007, 2015.

Delipetrev, B., A. Jonoski, and D. P. Solomatine, 2014. Development of a web application for water resources based on open source software. Computers and Geosciences 62:35–42.

Durkovic, J., V. Vukovi, and L. Rakovi. 2008. Open Source Approach in Software Development.

Encinas, A.H., A. Q. Dios, L.H. Encina, and V.G Martinez. 2013. Statistical Analysis From Time Series Related to Climate Data. International Journal of Applied Physics and Mathematics 3(3): 203-207.

Garrett Grolemund, 2014, Hands-On Programming with R, O'Reilly Media, Inc.,

Gartina, D. 2009. Penggunaan Software Open Software dalam Mendukung Kegiatan Penelitian dan Administrasi perkantoran. Jurnal Informatika Pertanian 18 (1): 45-62.

Gasch, C. K., T. Hengl, B. Gräler, H. Meyer, T. S Magney and D. J Brown. 2015. Temperature and Electrical Conductivity in 3D + T : The Cook Agronomy Farm Data Set. Spatial Statistics 14: 70–90.

Gilleland, E. and R.W Katz. 2011. New software to analyze how extremes change over time. Eos 92(2):13–14.

Gilleland, E., M. Ribatet, and A. G. Stephenson. 2013. A Software Review for Extreme Value Analysis. Extremers 16: 103–119.

Girvetz, E. H., C. Zganjar, G. T. Raber, E.P. Maurer, P. Kareiva, and J. J. Lawler. 2009. Applied Climate-Change Analysis: The Climate Wizard Tool. PloS ONE 4(12): 1-19

Hermawati, F.A. 2013. Data Mining. Penerbit Andi.

http://www.milbo.org/rpart-plot/prp.pdf

https://cran.r-project.org/web/packages/caret/caret.pdf

https://cran.r-project.org/web/packages/rpart.plot/rpart.plot.pdf

https://cran.r-project.org/web/packages/rpart/rpart.pdf

https://cran.r-project.org/web/packages/tree/tree.pdf

https://rpubs.com/minma/cart_with_rpart

Jeelani, M. I., A. H. Mir, S. Maqbool, N. Nazir, N. Ashraf and A. Ahmad. 2012. Multiple Regression Analysis of Horticultural Data: An Illustration using R-Software. International Journal of Modern Social Sciences 1(1): 12–28.

Jeelani, M. I., N. Nazir, S. A. Mir, F. Jeelani, and N. A Dar. 2014. Application of Simple Random Sampling in Agriculture using R-software. Indian Journal of Science and Technology 7 (5): 706–709.

Kemp, R. 2009. Current developments in Open Source Software. Computer Law and Security Review 25(6): 569–582.

McCreight, James. 2012. Hands-on R for Climate Data Analysis. NASA Summer Short Course for Earth System Modeling and Supercomputing

McLeod, I. A., H. Yu, and E. Mahdi. 2015. Time Series Analisis with R in Handbook Statistics 30. Elsevier 30. 661-712.

Mendiburu, F. and R. Simon. 2015. Agricolae-Ten years of an Open source Statistical tool for experiments in Breeding, agriculture and biology. PeerJPreprint.0–17.

Miguez, F. E. 2010. BioCro: an R package for Crop Simulation and Statistics. Iowa State University.

Molanejad, M., Soltani, and A.R.S Abadi. 2014. Changes in Precipitation extremes in climate variability over northwest Iran. International Journal of Agricultural Policy and Research 2(10): 334-345.

Nina Zumel dan John Mount, 2014, Practical Data Science with R, Manning Publications

Norman Matloff, 2009, The Art of R Programming.

Peternelli, L. A. 2011. Program R: applications in plant breeding. Crop Breeding and Applied Biotechnology SI. pp. 91–92.

Prana Ugiana Gio dan Dasapta Erwin Irawan, 2016, Belajar Statistika dengan R.

Prasetyo, Eko. 2014. Data Mining, Mengolah Data Menjadi Informasi Menggunakan Matlab. Penerbit Andi.

Raymond, E. 2000. The Cathedral and the Bazaar; Musing on Linux and opensource by an accidental Revolutionary. O'Reilly Media. Sebastopol. 34-38.

Sohrabi, M.M, J.H. Ryu, and Abatzoglou, J. 2012. Climate Extreme and its Linkage to Regional Drought over Idaho, USA. Nat Hazard (65): 653-681. DOI 10.1007/s11060-012-0384-1.

Sohrabi, M.M., S. Marofi and B. Ababaei. 2013. Investigtation of temperature and precipitation Indecis by using RclimDex and R Software in Senmnan Province. Conference paper.

Sreekanth, P. D., S. K.Soam, and Kumar, K. V. 2013. Spatial decision support system for managing agricultural experimental farms. Current Science 105 (11): 1588-1592.

Steiniger, S. and E. Bocher. 2008. An Overview on Current Free and Open Source Desktop GIS Developments. Internation Journal Geographical Information Science. pp.1–24.

Steiniger, S. and G. J Hay. 2009. Ecological Informatics Free and open source geographic information tools for landscape ecology. Ecological Informatics 4(4):183–195.

Verzani, JA. 2002. Simple R - Using R for Introductory Statistics.

W. John Braum dan Duncan J. Murdoch, 2007, A First Course in Statistical Programming with R, Cambridge University Press.