## Appendix: Libraries Used

<table>
<thead>
<tr>
<th>Name</th>
<th>First page cited</th>
</tr>
</thead>
<tbody>
<tr>
<td>ape</td>
<td>289</td>
</tr>
<tr>
<td>aplpack</td>
<td>59</td>
</tr>
<tr>
<td>astsa</td>
<td>337</td>
</tr>
<tr>
<td>boot</td>
<td>58</td>
</tr>
<tr>
<td>BradleyTerry2</td>
<td>340</td>
</tr>
<tr>
<td>CCA</td>
<td>346</td>
</tr>
<tr>
<td>cluster</td>
<td>294</td>
</tr>
<tr>
<td>clValid</td>
<td>303</td>
</tr>
<tr>
<td>cubature</td>
<td>46</td>
</tr>
<tr>
<td>datasets</td>
<td>151</td>
</tr>
<tr>
<td>energy</td>
<td>163</td>
</tr>
<tr>
<td>fBasics</td>
<td>127</td>
</tr>
<tr>
<td>fMultivar</td>
<td>155</td>
</tr>
<tr>
<td>foreign</td>
<td>32</td>
</tr>
<tr>
<td>fpc</td>
<td>303</td>
</tr>
<tr>
<td>gap</td>
<td>283</td>
</tr>
<tr>
<td>gplots</td>
<td>307</td>
</tr>
<tr>
<td>graphics</td>
<td>65</td>
</tr>
<tr>
<td>HSAUR2</td>
<td>283</td>
</tr>
<tr>
<td>Hotelling</td>
<td>202</td>
</tr>
<tr>
<td>ICSNP</td>
<td>177</td>
</tr>
<tr>
<td>kernlab</td>
<td>274</td>
</tr>
<tr>
<td>kohonen</td>
<td>308</td>
</tr>
<tr>
<td>MASS</td>
<td>149</td>
</tr>
<tr>
<td>MCMCpack</td>
<td>25</td>
</tr>
<tr>
<td>nortest</td>
<td>127</td>
</tr>
<tr>
<td>MVA</td>
<td>67</td>
</tr>
<tr>
<td>mvShapiroTest</td>
<td>163</td>
</tr>
<tr>
<td>mvtnorm</td>
<td>40</td>
</tr>
<tr>
<td>mvnormtest</td>
<td>199</td>
</tr>
<tr>
<td>mvtree</td>
<td>279</td>
</tr>
<tr>
<td>psych</td>
<td>199</td>
</tr>
<tr>
<td>psychometric</td>
<td>160</td>
</tr>
<tr>
<td>rpart</td>
<td>311</td>
</tr>
<tr>
<td>robustbase</td>
<td>256</td>
</tr>
<tr>
<td>stats</td>
<td>295</td>
</tr>
<tr>
<td>TeachingDemos</td>
<td>48</td>
</tr>
<tr>
<td>TH.data</td>
<td>229</td>
</tr>
<tr>
<td>TSA</td>
<td>337</td>
</tr>
<tr>
<td>vars</td>
<td>325</td>
</tr>
</tbody>
</table>
Selected Solutions and Hints

2.1 Some solutions are

\[
\begin{align*}
&c(0 : 10 / 10, 2 : 10) \\
&c(seq(0, 1, .1), 2 : 10) \\
&c(seq(0, 1, .1), seq(2, 10))
\end{align*}
\]

2.3 (a)

\[
\text{mad} <- \text{function(x)} \\
{ \text{mean( abs( x - mean(x) ) )} }
\]

2.4 The R code

\[
> \text{pchisq}(1.2, 1) - \text{pchisq}(0.5, 1) \\
[1] 0.2061784
\]

provides the result.

2.5 All the test scores are very highly correlated with each other. This suggests any one test score is representative of the whole data from each country. See also Exercise 8.7.

2.10 (c) Here are examples of operations resulting in Inf and NaN. Notice R only produces an error message in the second of these examples but not the third.

\[
> 5 / 0 \\
[1] \text{Inf}
\]
> sqrt(-3)

[1] NaN
Warning message:
In sqrt(-3) : NaNs produced

> 0/0

[1] NaN

2.11 A simple solution is

```r
spacing(x) <- function(x) diff(sort(x))
```

Notice this code does not check for empty or a scalar x, situations where spacings are undefined. A somewhat more cryptic definition including the error checking is

```r
spacing <- function(x) if(length(x) > 1) diff(sort(x)) else NA
```

3.7 This R program draws and peels away five convex hulls.

```r
x <- housing$Apartment  # temporary copies of the data
y <- housing$House
plot(x, y, xlab = "Apartment", ylab = "House",
     pch = 19, col = 2, cex = 1.25)  # initial scatterplot
for (i in 1:5)  # number of onion layers
{
    ch <- chull(x, y)  # indices of convex hull
    chl <- c(ch, ch[1])  # loop back to the first point
    lines(x[chl], y[chl], type = "l",
          col = 3)  # draw the layer
    x <- x[ -ch]  # peel away the layer
    y <- y[ -ch]
}
```

4.3 If \( y \) is a vector of all 1s, then \( y'X \) is a vector containing the column sums in \( X \) and \( Xy \) (for \( y \) of a possibly different length) are the row sums in \( X \).
4.6 The matrix inverse is
\[
\begin{pmatrix}
1 & -1 \\
-1 & 2
\end{pmatrix}.
\]

The eigenvalues are \((3 \pm \sqrt{5})/2\) or approximately 0.382 and 2.618.

5.2 Let \(U\) have a uniform distribution between 0 and 1. Then
\[
\Pr[U \leq p] = p
\]
for any number \(p\), between zero and one. We also have
\[
\Pr[U \leq \Phi(z)] = \Phi(z)
\]
for any real number \(z\) and
\[
\Pr[\Phi^{-1}(U) \leq z] = \Phi(z).
\]

This shows the distribution of \(X = \Phi^{-1}(U)\) has a standard normal distribution.

5.8 Denote the estimate of the mean by \(\hat{\mu}\) and standard deviation by \(\hat{\sigma}\). Use (5.7) to show the endpoints of the \(k\) categories with equal expectations are
\[
\{-\infty, \quad \hat{\mu} + \hat{\sigma} \Phi^{-1}(1/k)\}
\]
\[
\{\hat{\mu} + \hat{\sigma} \Phi^{-1}(1/k), \quad \hat{\mu} + \hat{\sigma} \Phi^{-1}(2/k)\}
\]
\[
\vdots
\]
\[
\{\hat{\mu} + \hat{\sigma} \Phi^{-1}((k - 1)/(k), \quad +\infty\}.
\]

6.2 (a) Figure 6.2 was produced using the following R code:

```r
CORCON <- function(x, y, correl)
{
    nX <- length(x)
    nY <- length(y)
    Z <- matrix(rep(0, nX * nY), nX, nY)
    for (i in 1 : nX)
    {
        for (j in 1 : nY)
        {
            Z[i,j] <- dmvnorm(c(x[i], y[j]),
```


c(0, 0),
         matrix(c(1, correl, correl, 1),
                2, 2)
   }
   }
return(z)
}

library(mvtnorm)
del <- .05    # how fine the grid
lim <- 3.25    # std normals plotted on +/- lim
par(mfrow = c(2, 4), mar = c(5, 0, 5, 0))    # Four plots across
contour(corcon(seq(-lim, lim, del), seq(-lim, lim, del), -.5),
         xlab = "Corr = -.5",
         drawlabels = FALSE, axes = FALSE, frame = TRUE)
contour(corcon(seq(-lim, lim, del), seq(-lim, lim, del), 0),
         xlab = "Corr = 0",
         drawlabels = FALSE, axes = FALSE, frame = TRUE)
contour(corcon(seq(-lim, lim, del), seq(-lim, lim, del), .5),
         xlab = "Corr = .5",
         drawlabels = FALSE, axes = FALSE, frame = TRUE)
contour(corcon(seq(-lim, lim, del), seq(-lim, lim, del), .9),
         xlab = "Corr = .9",
         drawlabels = FALSE, axes = FALSE, frame = TRUE)

(b) Again, using the corcon function, defined above, Fig.6.3 was
drawn using

library(MASS,mvtnorm,graphics)
layout(t(matrix(c(1 : 2, rep(0, 2)), 2, 2)), widths = c(1, 1))
del < .025    # how fine the grid
lim <- 1.25    # std normals plotted on +/- lim
image(corcon(seq(-lim, lim, del), seq(-lim, lim, del), 0.8), axes = FALSE)
del <- .3      # how fine the grid
lim <- 2.7     # std normals plotted on +/- lim
persp(corcon(seq(-lim, lim, del), seq(-lim, lim, del), .8),
       axes = FALSE, xlab = "", ylab = "", box = FALSE,
       col = "lightblue", shade = .05)

6.3 This program generates bivariate normals and transforms these to the
shape of Fig.6.7:

quad <- function(n)
{
   quad <- NULL
   for (i in 1 : n)
   {

Some trial and error suggests rad should be about 1.83.

6.5 Here is a useful reparameterization and objective function:

```r
biv5r <- function(par) # all five parameter, reparameterized
{
  sig1 <- exp(par[3])
  sig2 <- exp(par[4])
  cov <- rho * sig1 * sig2
  biv5 <- sum(
    -dmvnorm(cancer, mean = c(par[1], par[2]),
      sigma = matrix(c(sig1 ^ 2, cov, cov, sig2 ^ 2), 2, 2),
      log = TRUE)
  )
  print(c(par[1:2], sig1, sig2, rho, biv5))
  biv5
}
```

The code

```r
.nlm(biv5r, c(45, 45, 7.25, 7.25, 2))
```
then estimates the five parameters without warnings.

The estimated value of \( \text{par}[5] \) is 2.053445, so the estimate of \( \rho \) is

\[
> 2.053445 / \sqrt{1 + 2.053445 ^ 2} \\
[1] 0.8990582
\]

7.5 Consider an election where citizens cast ballots for one of \( p \) different candidates. The data is the number of votes received by each candidate. Any vote for one candidate means fewer votes for all of the others.

7.7 (a) This program calculates energy residuals:

```r
msqrt <- function(a) 
# finds matrix square root of positive definite matrix 
{ 
a.eig <- eigen(a) # eigenvalues and eigenvectors of x 
if (min(a.eig$values) < 0) # check for positive definite 
warning("Matrix not positive definite")

return(a.eig$vectors %*% diag(sqrt(a.eig$values)) %*% t(a.eig$vectors)) 
}

energy.resid <- function(dat, R=300) 
{ 
n <- dim(dat)[1] # observations 
p <- dim(dat)[2] # variables 
std <- dat - t(matrix(colMeans(dat), p, n)) 
s <- var(dat) 
std <- as.matrix(std) %*% solve(msqrt(s)) # standardized data 
rand <- matrix(rnorm(p * R), R, p) # independent normal data 
A <- rep(0, n) 
B <- rep(0, n) 
for (i in 1 : n) 
{ 
a <- 0 
b <- 0 
for (j in 1 : R) a <- a + sqrt(sum((std[i, ] - rand[j, ]) ^ 2)) 
A[i] <- a / R # ave dist between data and random 
for (j in 1 : n) b <- b + sqrt(sum((std[i, ] - std[j, ]) ^ 2)) 
B[i] <- b / (n - 1) # ave dist between data 
} 
cc <- 0 
for(i in 2 : R) for (j in 1 : (i - 1)) 
cc <- cc + sqrt(sum((rand[i, ] - rand[j, ]) ^ 2)) 
C <- 2 * cc /(R * (R - 1)) # ave dist between random values 
2 * A - B - C 
}```
7.9 This program computes the autocorrelation when columns of the data represent sequential years:

```r
autocov <- function(data) {
  # Autocorrelation of annual columns in data
  nyears <- dim(data)[2] - 1
  autocov <- NULL
  for (lag in 1 : nyears) {
    lagcor <- 0
    for (year in 1 : (nyears - lag + 1))
      lagcor <- lagcor + cor(data[, year], data[, year + lag])
    lagcor <- lagcor / (nyears - lag + 1)
    autocov <- c(autocov, lagcor)
  }
  autocov
}

ac <- autocov(CS)
```

8.7 The academic scores are highly correlated with each other. Further, these have been standardized. (See Exercise 2.5.) The loadings of the principal components analysis are almost the same for each test score. This suggests any one score is representative for the whole set of values for each country.

8.12 The biplot for the correlation matrix of the oil consumption data given in Table 8.4 appears here:

This figure separates low population, oil producing nations (Saudi Arabia and Canada) from high population, oil consuming nations (India and China).

9.1 The biplot of the three-dimensional residuals (given here) shows all dependent variables (cost of living, apartment rents, and house prices) remain highly correlated after correcting for the two explanatory variables (population and average state income). This first principal component explains over 90% of the total variability. Standout outliers include Hawaii (high rents and house prices) and Nevada (high rents but moderate house prices), both states with low populations.
9.4 The mining technology has changed over this time span, as has the number of workers employed in mining and the amount of coal produced. We can’t use this data alone as evidence of safer working conditions.
9.5 The R code

```r
jaw <- read.table(file = "Ramus.txt", header = TRUE, row.names = 1)
n <- dim(jaw)[1]
plot(x = NA, type = "n", xlim = c(8, 9.5), ylim = c(min(jaw), max(jaw)), xlab = "Age", ylab = "Ramus")
age <- c(8., 8.5, 9, 9.5)
longa <- NULL
longj <- NULL
for (i in 1:n)
{
  longa <- c(longa, age)
  ramus <- jaw[i,]
  longj <- c(longj, as.double(ramus))
  lines (age, ramus, col = "red")
}
lines(longa, longj, pch = 16, col = "blue", type = "p", cex = .8)
model <- lm(longj ~ longa)
lines ( age, fit, type = "l", col = "green", lwd = 3)
jres <- jaw - fit # residuals
sapply(jres, sd)
```

produces the spaghetti plot of observed and fitted values:

![Spaghetti plot of observed and fitted values](image-url)
This plot provides good evidence the variances are constant across ages.

10.16 The R code

```r
require(mvpart, datasets, graphics)
univ <- mvpart(mpg ~ cyl + disp + am + carb, data = mtcars)
```

produces a regression tree with four leaves:

```
cyl>=5  |  cyl< 5
      /
  cyl>=7  |  cyl< 7
       /
disp>=450  |  disp< 450
  /
10.4     n=2  15.9     n=12
```

```
cyl< 7
  /
disp< 101.6  |  disp>=101.6
   /
19.7     n=7  23.2     n=6
```

```
10.4     n=2  15.9     n=12  19.7     n=7  23.2     n=6  30.9     n=5
```

Error: 0.077  CV Error: 0.202  SE: 0.0498

11.6 (a) The code for clustering and principal components of the milk data

```r
library(robustbase)  # library with the milk dataset
milk2 <- milk[ -70 , ]  # omit outlier
colnames(milk2) <- c("dens", "fat", "prot", "casein", "Fdry", "Ldry", "drysub", "cheese")  # supply new names
# color schemes for K-means
color3 <- rainbow(3)[kmeans(scale(milk2), centers = 3)$cluster]
```
```r
pcm <- princomp(milk2, scores = TRUE)# principal components
plot(pcm$scores[,1], pcm$scores[,2], col = color3, pch = 16,
xlab = "First principal component", xaxt = "n",
ylab = "Second principal component",
main = "K-means clusters plotted by principal components")
```

produces the figure:

![K-means clusters plotted by principal components](image)

11.7 The distribution of $X$ is hypergeometric with probability mass function

$$
\Pr[X = x] = \binom{m}{x} \binom{N - m}{m' - x} / \binom{N}{m'}
$$

defined for

$$
\max(0, m + m' - N) \leq x \leq \min(m, m').
$$
In the hierarchical cluster

we see DC, Puerto Rico, Guam, and Hawaii cluster together, as do New York, New Jersey, and Delaware. Indiana, Illinois, Alaska, Vermont, and Utah each appear different from all other states.

(b) There are $K = 3$ clusters and the simulation probably alternates between two different pairs of these identified as closest.
References


REFERENCES


About the Author

Daniel Zelterman, Ph.D., is Professor in the Department of Biostatistics at Yale University. His research areas include computational statistics, models for discrete valued data, and the design of clinical trials in cancer studies. In his spare time he plays oboe and bassoon in amateur orchestral groups and has backpacked hundreds of miles of the Appalachian Trail.

Other Books by the Author

Models for Discrete Data, Oxford University Press, 1999
Advanced Log-Linear Models Using SAS, SAS Institute, 2002
Discrete Distributions: Application in the Health Sciences, J. Wiley, 2004
Applied Linear Models with SAS, Cambridge University Press, 2010
Index

> prompt, 18
K-means, 300, 309
+ prompt, 35
#, 39
2-tuple, 89
3-tuple, 89

ability tests, 226
abs(), 50
academic scores, 5, 50, 202, 363, 369
acf(), 323
activities of daily living, 252
adaptIntegrate(), 46
aesthetics, 124
Affymetrics, 303
age adjusted rates, 6, 164
AIC, 254, 331
albedo, 5
alphaindex(), 276
analysis of variance, see also ANOVA, 235
Anderson TW, xi, 375
anorexia, 87
ANOVA, 235, 238, 259, 295, 296
ansari.test(), 50, 139
Anscombe, FJ, xi, 375
answers to selected problems, 363
aov(), 237
ape library, 289, 290
APL, computer language, xi
aplpack library, 59, 81, 85
apples and oranges, 89
ARIMA model, 329
as.integer(), 259, 307
as.matrix(), 307
assignment <- in parenthesis, 21
astronomical unit, AU, 5
astronomy, 10
astrostatistics, 10
astsa library, 337
attach(), 31
autocorrelation, 180, 322, 324, 325
autocorrelation matrix, 190
autoregression, 322, 324
average distance, 308
axes=, 66, 67
back-to-back stem and leaf, 59
backward stepwise regression, 254
bagplot, 76, 79
banded correlation matrix, 205
bartlett.test(), 50, 139
baseball teams, 360
bats, 292
bct(), 123
beech trees, 286
beer bottles, 202
Behavior Risk Factor Surveillance System, see BRFSS, 244
bell curve, see also: normal distribution, 117, 118
big data, 217
binomial distribution, 117
biological homogeneity index, 308
biological stability index, 308
biplot, 213, 214, 216, 217, 223, 225, 229, 242, 243, 249, 251, 369
bird phylogeny, 292
INDEX

birth rate, 337
bivariate boxplot, 67, 68, 76, 85, 152
bivariate normal distribution, 151, 153, 195
black cherry trees, 228
body fat, 229
Bogat, GA, 199
bold font for vectors, 89
bootstrap, 301, 311
Box, GEP, 375
Box-Cox transformation, 123, 147
boxplot, 59, 86
Bradley-Terry model, 341, 360
BradleyTerry2 library, 340
breakdown, 204
breast cancer, 312
BRFSS, 238, 243, 342, 346
bubbleplot, 71
burger data, 12, 76, 82
bvbox(), 68
c(), 19, 31
Canadian health care workers, 288
cancer, 6, 164, 225, 316
candy, 195–201
canonical correlation, 342
canonical scores, 344
car.test.frame data, 256
cars, 231, 256, 286, 311
CART, 279, 287
Case-Schiller index, 178
cat(), 23
Cauchy distribution, 122, 350
causality, 231
cause and effect, 231
Caveat emptor, 17
cbind(), 31, 161, 169, 239, 259
CCA library, 346
ccf(), 323
CD4 counts, 58
CDC, see: Centers for Disease Control, vii
census, 2
Centers for Disease Control, vii, 163, 227, 244, 245, 248
CEO compensation, 358
cex.lab=, 67
cex=, 61, 66–68, 70, 73, 76, 125, 161
characteristic polynomial, 107
chart junk, 65
Chernoff’s faces, 81, 82
chi-squared distribution, ix, 50, 56, 122, 182, 187
Chiroptera, 292
chull(), 69, 70
CIr(), 160
citations in academic journals, 340
cities, 72
Clz(), 160
clear(), 308
climate change data, 317
clipboard, 32
cluster library, 294, 310
clustering, 8, 287
clustering diagnostics, 301
cluttered desk, 47
clValid library, 303, 308
col=, 66, 68, 70, 73, 76, 83, 84, 125, 161
colMeans(), 176
colnames(), 31
colors, see also col= and rainbow(), 66, 84
comments in programs, 39
commutative operation, 96
compensation of non-profit CEO’s, 358
complete.cases(), 36, 228, 311
complex conjugate, 107
complex numbers, 107
complexity parameter, 281
computer-aided regression trees, 279
conditional distribution, 156, 176
conditional probability, ix
conditional variance, 156
conditioning plot, 76
confidence interval, 2, 134
connectivity, 308
contour plot, 269
convex hull, 69, 76, 79, 85, 274
coplot, 76, 80, 172
copy and paste, 32
cor(), 34, 41, 159
cor.test(), 160, 170
correlation, 42, 152, 159, 170, 342
correlation matrix, 34, 50
correlogram, 323–325
cotton workers, 202
cov(), 159
cov.rob(), 204
cov.trob(), 204
covariance, 154, 158, 159
covariance of ability tests, 226
Cox, DR, 375
CP, 281
cpgram(), 335
Crabs, 279, 281
cross product in 3 dimensions, 97
cross validation, 275, 276, 280, 285
cross(), 285
cross-correlation, 323, 327
crossprod(), 97
crow, flying, 94
cubature library, 46
cumulative normal distribution, 118, 120
cumulative periodogram, 335, 336
curvature of the spine, 286
cut and paste, 32

Daily Beast, vii
daisy(), 294
Dalgaard, P, 190
data frame, 29
data to ink ratio, 65
data.frame, 64, 159, 239
data.frame(), 169
datasets library, 151, 231
dBase, 32
dchisq(), chi-squared density, 56
de Moivre, A, 117
decomposed.ts object, 319
degrees of freedom, df, ix
deleting variables, 19
dendrogram, 287, 289, 290, 305
derivative, 145
determinant, 102, 106
Detroit housing prices, 180, 191, 205
dev.off(), 57
df, 132
df, fractional, 134
diag(), 98, 169
diagnostics for clustering, 301
diagonal matrix, 96, 103
diagonalizable matrix, 108
diana(), 308
diff(), 53, 364
disability, 252
discriminant function, 272
dist(), 290, 293
distance, 94
DNA microarray, 303
dnorm(), 119
documenting programs, 39
domain of attraction, 350
dot-dash-plot, 66
Dunn Index, 303, 310, 311
duplicated(), 36, 276
daily Beast, vii
daisy(), 294
Dalgaard, P, 190
data frame, 29
data to ink ratio, 65
data.frame, 64, 159, 239
data.frame(), 169
datasets library, 151, 231
dBase, 32
dchisq(), chi-squared density, 56
de Moivre, A, 117
decomposed.ts object, 319
degrees of freedom, df, ix
deleting variables, 19
dendrogram, 287, 289, 290, 305
derivative, 145
determinant, 102, 106
Detroit housing prices, 180, 191, 205
dev.off(), 57
df, 132
df, fractional, 134
diag(), 98, 169
diagnostics for clustering, 301
diagonal matrix, 96, 103
diagonalizable matrix, 108
diana(), 308
diff(), 53, 364
disability, 252
discriminant function, 272
dist(), 290, 293
distance, 94
DNA microarray, 303
dnorm(), 119
documenting programs, 39
domain of attraction, 350
dot-dash-plot, 66
Dunn Index, 303, 310, 311
duplicated(), 36, 276
effective df, 264
Egyptian skulls, 283
eigen(), 209
eigenvalue, 106
eigenvector, 106, 115, 209
El Niño, 317
everse, 43, 44
energy library, 163
energy statistic, 203
energy test, 163, 201
eqscplot(), 74
equality of variances, 50, 139
error checking, 42
estimated correlation, 159
estimated covariance distribution, 187
estimated mean, 159
estimated variance, 159
Euclid, 93
Euclidean length, 94, 201, 203, 290, 293
Euler constant (0.5772...), 351
Euler L, 351
Excel, 32
exchangeable correlation matrix, 190
exchangeable model, 171, 187
exoplanet, 356
explained sum of squares, 296
exponential distribution, 352, 353
extrapolation, 233
extremes, 198
Eye, vA, 199

F distribution, 177
F, see also FALSE, 22
F-ratio, 235
faces, 48, 81, 82, 86
factanal(), 221
factor analysis, 10, 84, 218
factor loadings, 219
factor variable type, 262
FALSE, 22, 25, 43
fanny(), 308
fast Fourier transform, 334
Fayette, AR, 248
fBasics library, 127, 128
fg=, 72
Fisher linear discriminant function, 272
Fisher RA, 82
Fisher Z-transformation, 160
Fisher’s iris data, see iris dataset, 82, 283
fMultivar library, 155
for(), 43
foreign library, 32
forests, 281
format(), 23
formula in liner models, 236
Fourier transformation, 334
fourth moment, see kurtosis, 128
fpc library, 303
fractional df, 134
frequency spectrum, 334
full rank matrix, 103
functions, 37
galaxies, 10
gamma distribution, 350
gap library, 283
garbage, 47
Gauss, CF, 117
Gaussian distribution, see also:
  normal distribution, 117
gene expression, 310
genralized inverse, 109
genralized linear model, see also
  glm, 237
getwd(), 47
ggplot2(), 85
Gini statistic, 353
glaucoma, 285
glm(), 237
glmrob(), 256
global environment data, 317
glyph, 80
gml(), 237
Golub, GH, 102
Gosset WS, 132
Gower metric, 294
gplots library, 307
gradient, 46, 145
graphical windows, 27
graphics, 27, 55
graphics library, 65
Gumbel distribution, 350
Gumbel, EJ, 350
Guttenberg, J, 84
hamburger data, 12, 76, 82, 223
haplotype, 286
hclust(), 290
header=, 29
health expenditures, 337
heatmap, 245, 305, 307
heavy tail, see also kurtosis, 128
Heineken beer bottles, 202
INDEX

help(), 23, 24, 48
hemangiomata, 219
hematocrit, 337
Hessian, 46, 169, 171, 184
hierarchical clustering, 287
hill races, 149
hist(), 26, 51
histogram, 26, 58, 126, 135
HIV, 58
HLA, 283
home prices, 178
Hotelling T2 statistic, 177
Hotelling, H, 177, 344
HSAUR2 library, 283, 356, 358
Hubble Space Telescope, 12
Huntington, WV, 248
hypergeometric distribution, 373
IAD, 252
ICSNP library, 177
identifiable model, 342
identity matrix, 96, 103, 104, 106, 187
if(), 43, 44
image(), 155, 170
immune system, 283
inches=, 72
independence, ix, 153, 157
independence of irrelevant alternatives, 265, 285
Inf, 52, 364
inner product, 90, 92
installed.packages(), 48
installing R, 17
installing packages, 48
integrate(), 46, 51
interactive program, 18
inverse Fourier transform, 334
inverse of a matrix, 104
iris dataset, 82, 83, 283, 288, 290
iteration, 44
Izenman, A, x

Jaccard Index, 302, 310
jackknife, 256, 308
January temperatures in U.S. cities, 72
Jarque-Bera test, 128, 147, 199, 327
jitter(), 62, 259
jittering, 301
joint likelihood function, 141
Journal Citation Reports, 340
journal citations, 340
judges dataset, 226
Jupiter, 356
Kedem, B, x
Kendall \( \tau \), 170
kernlab library, 274
kohonen library, 308
Kolmogorov-Smirnov test, 130, 147
Krause A, 14
Krige, DG, 72
kriging, 73, 74, 274
ksvm(), 274
Kuiper Belt, 5, 214
kurtosis, 128, 198
kyphosis, 286

lag, 180, 322
Lang, S, 102, 377
largest oil spills, 357
LASERI, 177, 178, 204
latitude, 10
lattice plot, 85
lda(), 266
LDA, see linear discriminant analysis, 265
least squares, 69, 74, 233
length of a vector, 94
length(), 21
library(), 48
light tail, see also kurtosis, 128
likelihood, 141
likelihood ratio test, 182
line width, see also lwd=, 70
linear combination of multivariate normal variates, 175
linear discrimination analysis, 265, 266, 272
INDEX

153, 156, 184
linear regression, 153, 156, 184

114, 115
linear space, 114, 115

93
linearly dependent vectors, 93

56, 67, 70, 161
lines(), 56, 67, 70, 161

329, 333
Ljung–Box test, 329, 333

48
loading packages into R: see installing packages, 48

208
loadings in principal components, 208

219
loadings, in factor analysis, 219

38
local variables, 38

76, 78, 86, 241, 274, 321, 336
loess, 76, 78, 86, 241, 274, 321, 336

350
log-gamma distribution, 350

142
log-likelihood, 142

30, 142
logarithm, 30, 142

262, 286
logistic regression, 262, 286

341
logit, 341

128
long tail, see also kurtosis, 128

10
longitude, 10

43
loop in programming, 43

19, 47
ls(), 19, 47

310
lung cancer, 310

67, 70, 72, 76, 161
lwd=, 67, 70, 72, 76, 161

276
machine learning, 276

14
macro, 14

12
magnitude, in astronomy, 12

195, 201, 203, 204, 240
Mahalanobis distance, 195, 201, 203, 204, 240

286
mammography, 286

293
Manhattan norm, 293

238
MANOVA, 238

270
MAP, 270

199
mardia(), 199

198
Mardia, KV, 198

153
marginal analysis, 153

154, 156, 176
marginal distribution, 154, 156, 176

199
Margolin, BH, 199

70
masking, 70

149
MASS library, 82, 149

73
match(), 22, 68, 73

94
matrix, 94

102, 106
matrix determinant, 102, 106

109
matrix generalized inverse, 109

104
matrix inverse, 104

100
matrix multiplication in R, 100

103
matrix of full rank, 103

96
matrix operation in R, 96

98
matrix outer product in R, 98

103
matrix rank, 103

76–79, 260
matrix scatterplot, 76–79, 260

111, 201
matrix square root, 111, 201

106
matrix trace, 106

91, 94
matrix transpose, 91, 94

32, 98
matrix(), 32, 98

108
matrix, positive definite, 108

103
matrix, singular, 103

270
maximum a posteriori (MAP) estimator, 270

141
maximum likelihood estimates, 141

262
maximum likelihood estimation, 262

293
maximum norm, 293

118
mean, 118

49
mean absolute deviation, 49

33, 50, 133, 159
mean(), 33, 50, 133, 159

252, 337
Medicare, 252, 337

35
merge(), 35

303, 310
microarray, 303, 310

232, 233, 286
miles per gallon, 232, 233, 286

298, 300, 373
milk data, 298, 300, 373

32
Minitab, 32

293
Minkowski norm, 293

201, 337
missing data, 201, 337

201
MissMech package, 201

141
mle’s, 141

280
mode, 280

235
model sum of squares, 235

50, 139
mood.test(), 50, 139

109
Moore-Penrose matrix inverse, 109

72, 377
Mosteller F, 72, 377

322, 329, 335, 336
moving average, 322, 329, 335, 336

329, 330
moving window, 329, 330

198, 199
multivariate kurtosis, 198, 199

175, 269
multivariate normal distribution, 175, 269

164, 198
multivariate outlier, 164, 198

198, 199
multivariate skewness, 198, 199

204
multivariate t distribution, 204
mutual independence, 93
MVA library, 67
mvnormtest library, 199
mvpart library, 256
mvShapiroTest library, 163
mvtree library, 279

n-tuple, 89, 91
NA, 36, 39, 52
names(), 259
NaN, 39, 42, 52, 364
national academic scores, 5, 50, 202, 363, 369
National Cancer Institute, 316
National Vital Statistics System, 227
nearest neighbor, 308
negative subscript, 21, 30
nested models, 182
neural network, 308
New York Choral Society, 85
New York City birth rates, 318
Nichols, E, x
nlm(), 46, 51, 143, 144, 183, 192
non-identifiable model, 342
non-profit CEO compensation, 358
nonparametric methods, 70, 122, 170
normal distribution, 117, 157, 350
normal distribution histogram, 51
normal distribution mean, 118
normal distribution quantile function, 120, 124
normal distribution variance, 118
normal distribution, bivariate, 151
normalized spacings, 352
nortest library, 127
NULL, 52
numerical analysis, 105
numerical integration, 46
numerical operations, 46, 105

offset, 342
oil companies, 313
oil consumption, 229
oil spills, 357
Older Americans Act, 311
Olson M, 14
one-way ANOVA, 259
onion peeling, 70
oranges, 89
order statistics, 130, 349
organizational tips, 47
orthogonal vectors, 93, 108, 210
outer product in R, 98
outlier, 70, 76, 164

p value, ix
pairs(), 76–79, 150, 195, 217, 226, 244, 260, 300
palette, see also rainbow(), 67
pam(), 308
panel.smooth, 76, 86, 241
panel.violin, 86
panel=, 76
par(), 61
parallel coordinate plot, 82, 198, 260, 261
parcoord(), 82, 83, 198, 260
parenthesis and assignment, 21
patient outcomes, 284
pch=, 66, 68, 70, 73, 125
pchisq(), 50, 197
pdf(), 57
Pearl, J, 231, 377
Pearson, K, 126
Pearson chi-squared, 126
Pearson correlation coefficient, 159, 160
Pearson, K, 126
peeling an onion, 70
periodogram, 323, 335, 336
persp(), 142, 155, 170
perspective plot, 155, 269
pie chart, 85
PISA, 5
platelet count, 337
plot data, 55
plot(), 55, 66, 67, 70, 73, 161
plot() characters: pch=, 66
plot() options, 66
Pluto, former planet, 5
pmin(), 193
pnorm(), 119
Poisson distribution, 147, 148
population, 2
pos=, 73
positive definite matrix, 108
post-operative patient outcomes, 284
posterior probability, 270
pound sign #, 39
power, 45, 51
power transformations, 123
prediction strength, 303
principal component loadings, 208
principal components, 238, 242, 373
print(), 23
printcp(), 281
prior probability, 269
probability, ix
probability journals, 340
programming, 37
promoter gene, 284
prostate cancer, 285
pruning, 281
psych library, 199, 228
psychometric library, 160
pt(), 264
public health education, 286
pulmonary function, 202
Pythagoras, 94
q(), 19
qnorm(), 197
QQ plot, 124, 130, 135, 196, 199, 200, 240
qqline(), 125
qqplot(), 125
quadratic form, 106, 174
quadrature, 46
quantile function for normal, 120
quartile, 59, 125, 156, 196
quitting R, 19
r2z(), 160
rainbow(), 67, 79
Rand Index, 303, 310
random forests, 281
random numbers, 24
rank of a matrix, 103
rbind(), 32, 83
read.table(), 28, 32
read.table() from Web, 258
reading scores, 5
reading tab characters, 32
reciprocal, 104
recursive partitioning, 279
reference category, 262, 342
regression, 69, 74, 152
regression tree, 287, 312
removing, deleting variables from R, 19
rep(), 20, 61
require(), 48
residual sum of squares, 235
residuals, 207, 233
return(), 37
right ascension, 10
Ripley BD, 14, 378
rm(), 47
rmvnorm(), 274
rnorm(), 24, 51, 133
robust methods, 49, 122, 204, 256, 310
robustbase library, 256, 283, 309
round(), 218
rounding error, 105, 264
row.names(), 28, 29, 32, 68, 169
rpart library, 311
Rstudio, 14, 39
rug fringe, 65, 236, 327
S and S-PLUS, computer languages, 14
S, computer language, 32
S, sample variance matrix, 159
sample, 2
sample mean, 132
sample size estimation, 45
sample variance, 132
sample(), 39
SAPA (Synthetic Aperture Personality Assessment), 228
sapply(), 34, 159, 215
Sarkar D, 377
SAS, 13, 27, 32, 355
scalar, 89, 91, 104, 364
scale(), 293, 327
scaling, 290
scan(), 32
scatterplot, 63, 76–79
schizophrenia, 283, 358
school demographic, 309
Schwager, SJ, x, 199
scope of variables, 38
Scottish hills, 149
scree plot, 210, 219, 229
sd(), 133
second derivative, 164
SEER, 316
semi-major axis, 5
semiTools library, 199
senior citizens, 311
setwd(), 47
sex differences in mathematics exams, 135
Shapiro SS, 130, 378
Shapiro-Wilk test, 147, 163, 199, 240
shapiro.test(), 197
short tail, see also kurtosis, 128
simulation, 24, 131
simulation of K-means, 309
sine wave, 274
singular matrix, 103
size bias, 356
skewness, 122, 128, 198
skulls, 283
SMART, 244
smoothed periodogram, 335
solution of linear equations, 101
solve(), 101, 105, 167, 169
som(), 308
sort(), 26, 32, 364
space spanned by a vector, 114
 spacings, 38, 42, 352
spaghetti plot, 63, 64, 372
spam, 284
span option in spec.pgram, 335
Spearman ρ, 170
spectral decomposition, 333, 334
spectrum, 334
spherical covariance matrix, 187
spinal surgery, 286
sqrt(), 37
square root, 37
standard normal, 117
standardized extreme values, 350
star plot, 78
Stata, 32
stationarity, 329
stationarity tests, 333
stationary time series, 322, 337, 338
statistical graphics, 55
statistical significance, 45, 51
stats library, 295
stem and leaf plot, 58
stem(), 58
stepwise regression, 254
stl(), 321, 336
stopping R, 19
Student’s t distribution, 122, 132
subscript, negative, 30, 53
sum of squares, 235
sum of vectors, 92
sums of normally distributed variables, 122
supertree, 292
support vector machine, 273
support vectors, 273, 274
supremum norm, 293
surf.ls, 74
SVM, 273
Swiss cantons, 309
swiss dataset, 151, 152, 155, 157–161, 309
Switzerland, 151
symbol, 72
symmetric matrix, 94, 174
t distribution, 132
t test, ix
t(), transpose of a matrix in R, 32, 99
T, see also TRUE, 22
t.test(), 133
tab characters, 32
tail of the distribution, see also kurtosis, 128, 349
take out the garbage, 47
TeachingDemos package, 48
test for normality, 135
text(), 57, 68, 72, 73
TH.data library, 229
third moment, see also skewness, 128
tilde \sim, 237
tile vertically, 27
time series, 180, 189, 315
total sum of squares, 235
trace, 102, 106
transformed data, 150
transpose, 32, 89, 91, 99, 173
transpose of matrix, 91, 94, 174
transpose of vector, 173
trees, 228
trellis plot, 85
TRUE, 22, 25, 43
ts object, 319
ts() function, 319
TSA library, 337
Tufte ER, 65, 84, 378
Tukey JW, 72, 84, 334, 377
type=, 161

uncorrelated normal variates, 157
unequal variance t-test, 134
unexplained sum of squares, 296

uniform distribution, 350
US cities, 72
US judges dataset, 226
validation of clustering, 301
Van Loan, CF, 102
var(), 34, 133, 159, 176, 204
var.test(), 50, 139
variance, 118
variance matrix, 34, 187
variogram, 323
vars library, 325, 327
vasoconstriction, 283
vector, 89, 91
vector cross product in 3 dimensions, 97
vector elements, 21
vector inner product, 90, 92, 97
vector subtraction, 92
vector sum, 92
vector transpose, 91
vector valued arguments, 26
Venables WN, 14, 378
violin plot, 61, 86
visual magnitude, 10
vocalists, 85

Walmart, 248
weather map, 73
wedding cake, 70
Weibull distribution, 350
Welch t-test, 134
whiskers, 59
white blood count, 337
white light, 334
white noise, 335
Wilk MB, 130, 378
Wilks SS, 182
window width of moving average, 322
wine cultivars, 258–261
Wisconsin breast cancer study, 312
Wishart distribution, 187
with(), 72
within-cluster variability, 297
working directory, see also getwd() and setwd(), 29, 47, 57
workspace, 19

xlab=, 67, 68, 161
xlim=, 66

ylab=, 67, 68, 161
ylim=, 66

Yu, C, x

z2r(), 160

zero vector, 92, 93, 114