See also documents, including Maindonald (2008), that are listed under **Contributed Documentation** on the CRAN sites. For careful detailed accounts of the R language, see Chambers (2007), Gentleman (2008).

Books and papers that set out principles of good graphics include Cleveland (1993, 1994), Tufte (1997), Wainer (1997), and Wilkinson and Task Force on Statistical Inference (1999). See also the imaginative uses of R's graphical abilities that are demonstrated in Murrell (2005). Maindonald (1992) comments very briefly on graphical design.

References for further reading

Chambers, J. M. 2007. *Software for Data Analysis: Programming with* R.

Cleveland, W. S. 1993. *Visualizing Data.*

Cleveland, W. S. 1994. *The Elements of Graphing Data*, revised edn.

Dalgaard, P. 2008. *Introductory Statistics with* R.

Fox, J. 2002. *An* R *and* S-PLUS *Companion to Applied Regression.*

Gentleman, R. 2008. R *Programming for Bioinformatics*.

Maindonald, J. H. 1992. Statistical design, analysis and presentation issues. *New Zealand Journal of Agricultural Research* 35: 121–41.

Maindonald, J. H. 2008. *Using* R *for Data Analysis and Graphics.* Available as a pdf file at `http://www.maths.anu.edu.au/~johnm/r/usingR.pdf`

Murrell, P. 2005. R *Graphics*. `http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html`

R Development Core Team. 2009a. *An Introduction to* R.

Tufte, E. R. 1997. *Visual Explanations.*

Venables, W. N. and Ripley, B. D. 2002. *Modern Applied Statistics with* S, 4th edn.

Wainer, H. 1997. *Visual Revelations.*

Wilkinson, L. and Task Force on Statistical Inference. 1999. Statistical methods in psychology journals: guidelines and explanation. *American Psychologist* 54: 594–604.

See the references at the end of the book for fuller bibliographic details.

1.9 Exercises

1. The following table gives the size of the floor area (ha) and the price ($A000), for 15 houses sold in the Canberra (Australia) suburb of Aranda in 1999.

```
  area sale.price
1  694 192.0
2  905 215.0
3  802 215.0
4 1366 274.0
5  716 112.7
6  963 185.0
7  821 212.0
8  714 220.0
9 1018 276.0
```

```
10   887 260.0
11   790 221.5
12   696 255.0
13   771 260.0
14 1006 293.0
15 1191 375.0
```

Type these data into a data frame with column names `area` and `sale.price`.

(a) Plot `sale.price` versus `area`.
(b) Use the `hist()` command to plot a histogram of the sale prices.
(c) Repeat (a) and (b) after taking logarithms of sale prices.
(d) The two histograms emphasize different parts of the range of sale prices. Describe the differences.

2. The `orings` data frame gives data on the damage that had occurred in US space shuttle launches prior to the disastrous Challenger launch of 28 January 1986. The observations in rows 1, 2, 4, 11, 13, and 18 were included in the pre-launch charts used in deciding whether to proceed with the launch, while remaining rows were omitted.

   Create a new data frame by extracting these rows from `orings`, and plot `total` incidents against `temperature` for this new data frame. Obtain a similar plot for the full data set.

3. For the data frame `possum` (*DAAG* package)

   (a) Use the function `str()` to get information on each of the columns.
   (b) Using the function `complete.cases()`, determine the rows in which one or more values is missing. Print those rows. In which columns do the missing values appear?

4. For the data frame `ais` (*DAAG* package)

   (a) Use the function `str()` to get information on each of the columns. Determine whether any of the columns hold missing values.
   (b) Make a table that shows the numbers of males and females for each different sport. In which sports is there a large imbalance (e.g., by a factor of more than 2:1) in the numbers of the two sexes?

5. Create a table that gives, for each species represented in the data frame `rainforest`, the number of values of `branch` that are `NA`s, and the total number of cases.
   [*Hint:* Use either `!is.na()` or `complete.cases()` to identify `NA`s.]

6. Create a data frame called `Manitoba.lakes` that contains the lake's `elevation` (in meters above sea level) and `area` (in square kilometers) as listed below. Assign the names of the lakes using the `row.names()` function.

```
               elevation  area
Winnipeg             217 24387
Winnipegosis         254  5374
Manitoba             248  4624
SouthernIndian       254  2247
Cedar                253  1353
Island               227  1223
Gods                 178  1151
Cross                207   755
Playgreen            217   657
```

(a) Use the following code to plot `log2(area)` versus `elevation`, adding labeling information (there is an extreme value of `area` that makes a logarithmic scale pretty much essential):

```
attach(Manitoba.lakes)
plot(log2(area) ~ elevation, pch=16, xlim=c(170,280))
  # NB: Doubling the area increases log2(area) by 1.0
text(log2(area) ~ elevation,
     labels=row.names(Manitoba.lakes), pos=4)
text(log2(area) ~ elevation, labels=area, pos=2)
title("Manitoba's Largest Lakes")
detach(Manitoba.lakes)
```

Devise captions that explain the labeling on the points and on the y-axis. It will be necessary to explain how distances on the scale relate to changes in area.

(b) Repeat the plot and associated labeling, now plotting `area` versus `elevation`, but specifying `log="y"` in order to obtain a logarithmic y-scale. [*Note:* The `log="y"` setting carries across to the subsequent `text()` commands. See Subsection 2.1.5 for an example.]

7. Look up the help page for the R function `dotchart()`. Use this function to display the areas of the Manitoba lakes (a) on a linear scale, and (b) on a logarithmic scale. Add, in each case, suitable labeling information.

8. Using the `sum()` function, obtain a lower bound for the area of Manitoba covered by water.

9. The second argument of the `rep()` function can be modified to give different patterns. For example, to get four 2s, then three 3s, then two 5s, enter

```
rep(c(2,3,5), c(4,3,2))
```

(a) What is the output from the following command?

```
rep(c(2,3,5), 4:2)
```

(b) Obtain a vector of four 4s, four 3s, and four 2s.

(c) The argument `length.out` can be used to create a vector whose length is `length.out`. Use this argument to create a vector of length 50 that repeats, as many times as necessary, the sequence: `3 1 1 5 7`

(d) The argument `each` can be used to form a vector in which each element in the first argument is replaced by the specified number of repeats of itself. Use this to create a vector in which each of `3 1 1 5 7` is replaced by four repeats of itself. Show, also, how this can be done without use of the argument `each`.

10. The ^ symbol denotes exponentiation. Consider the following:

```
1000*((1+0.075)^5 - 1)  # Interest on $1000, compounded
                        # annually at 7.5% p.a. for five years
```

(a) Evaluate the above expression.

(b) Modify the expression to determine the amount of interest paid if the rate is 3.5% p.a.

(c) Explain the result obtained when the exponent `5` is changed to `seq(1, 10)`.

11. Run the following code:

```
gender <- factor(c(rep("female", 91), rep("male", 92)))
table(gender)
gender <- factor(gender, levels=c("male", "female"))
table(gender)
```

```
gender <- factor(gender, levels=c("Male", "female"))
                    # Note the mistake: "Male" should be "male"
table(gender)
table(gender, exclude=NULL)
rm(gender)          # Remove gender
```

Explain the output from the successive uses of `table()`.

12. Write a function that calculates the proportion of values in a vector `x` that exceed some value `cutoff`.

    (a) Use the sequence of numbers 1, 2, . . . , 100 to check that this function gives the result that is expected.
    (b) Obtain the vector `ex01.36` from the *Devore6* (or *Devore7*) package. These data give the times required for individuals to escape from an oil platform during a drill. Use `dotplot()` to show the distribution of times. Calculate the proportion of escape times that exceed 7 minutes.

13. The following plots four different transformations of the `Animals` data from the *MASS* package. What different aspects of the data do these different graphs emphasize? Consider the effect on low values of the variables, as contrasted with the effect on high values.

```
par(mfrow=c(2,2))     # 2 by 2 layout on the page
library(MASS)         # Animals is in the MASS package
plot(brain ~ body, data=Animals)
plot(sqrt(brain) ~ sqrt(body), data=Animals)
plot(I(brain^0.1) ~ I(body^0.1), data=Animals)
  # I() forces its argument to be treated "as is"
plot(log(brain) ~ log(body), data=Animals)
par(mfrow=c(1,1))     # Restore to 1 figure per page
```

14. Use the function `abbreviate()` to obtain six-character abbreviations for the row names in the data frame `cottonworkers` (*DAAG* package). Plot `survey1886` against `census1886`, and plot `avwage*survey1886` against `avwage*census1886`, in each case using the six-letter abbreviations to label the points. How should each of these graphs be interpreted? [*Hint:* Be sure to specify `I(avwage*survey1886)` and `I(avwage*census1886)` when plotting the second of these graphs.]

15. The data frame `socsupport` (*DAAG*) has data from a survey on social and other kinds of support, for a group of university students. It includes Beck Depression Inventory (`BDI`) scores. The following are two alternative plots of `BDI` against `age`:

```
plot(BDI ~ age, data=socsupport)
plot(BDI ~ unclass(age), data=socsupport)
```

For examination of cases where the score seems very high, which plot is more useful? Explain. Why is it necessary to be cautious in making anything of the plots for students in the three oldest age categories (`25-30`, `31-40`, `40+`)?

16. Functions that can be useful for labeling points on graphs are `abbreviate()` (create abbreviated names), and `paste()` (create composite labels). A composite label might, for the data from `socsupport`, give information about `gender`, `country`, and row number. Try the following:

```
gender1 <- with(socsupport, abbreviate(gender, 1))
table(gender1)        # Examine the result
```

```
country3 <-  with(socsupport, abbreviate(country, 3))
table(country3)       # Examine the result
```

Now use the following to create a label that can be used with `text()` or with `identify()`:

```
num <- with(socsupport, seq(along=gender))   # Generate row numbers
lab <- paste(gender1, country3, num, sep=":")
```

Use `identify()` to place labels on all the points that the boxplots have identified as "outliers".

17. Given a vector `x`, the following demonstrates alternative ways to create a vector of numbers from 1 through n, where n is the length of the vector:

```
x <-  c(8, 54, 534, 1630, 6611)
seq(1, length(x))
seq(along=x)
```

Now set `x <- NULL` and repeat each of the calculations `seq(1, length(x))` and `seq(along=x)`. Which version of the calculation should be used in order to return a vector of length 0 in the event that the supplied argument is `NULL`.

18. The `Rabbit` data frame in the *MASS* library contains blood pressure change measurements on five rabbits (labeled as `R1, R2,...,R5`) under various control and treatment conditions. Read the help file for more information. Use the `unstack()` function (three times) to convert `Rabbit` to the following form:

```
   Treatment   Dose    R1    R2    R3    R4   R5
1    Control   6.25  0.50  1.00  0.75  1.25  1.5
2    Control  12.50  4.50  1.25  3.00  1.50  1.5
. . . .
6    Control 200.00 32.00 29.00 24.00 33.00 18.0
7        MDL   6.25  1.25  1.40  0.75  2.60  2.4
8        MDL  12.50  0.75  1.70  2.30  1.20  2.5
. . . .
12       MDL 200.00 37.00 28.00 25.00 22.00 19.0
```

19. The data frame `vlt` (*DAAG*) consists of observations taken on a video lottery terminal during a two-day period. Eight different objects can appear in each of three windows. Here, they are coded from 0 through 7. Different combinations of the objects give prizes (although with small probability). The first four rows are:

```
> head(vlt, 4)        # first few rows of vlt
  window1 window2 window3 prize night
1       2       0       0     0     1
2       0       5       1     0     1
3       0       0       0     0     1
4       2       0       0     0     1
>   # . . .
```

Use `stack()` to convert the first three columns of this data set to a case-by-variable format, then creating a tabular summary of the results, broken down by window.

```
vltcv <- stack(vlt[, 1:3])
head(vltcv)           # first few rows of vltcv
table(vltcv$values, vltcv$ind)
  # More cryptically, table(vltcv) gives the same result.
```

Does any window stand out as different?

20.* The help page for `iris` (type `help(iris)`) gives code that converts the data in `iris3` (*datasets* package) to case-by-variable format, with column names "Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", and "Species". Look up the help pages for the functions that are used, and make sure that you understand them. Then add annotation to this code that explains each step in the computation.

21.* The following uses the `for()` looping function to plot graphs that compare the relative population growth (here, by the use of a logarithmic scale) for the Australian states and territories.

```
oldpar <- par(mfrow=c(2,4))
for (i in 2:9){
plot(austpop[, 1], log(austpop[, i]), xlab="Year",
     ylab=names(austpop)[i], pch=16, ylim=c(0,10))}
par(oldpar)
```

Find a way to do this without looping. [*Hint:* Use the function `sapply()`, with `austpop[,2:9]` as the first argument.]