Errata and Notes: "Statistical Computing with R", Chapman & Hall/CRC (2008)

Maria L. Rizzo

28 October 2017

This document contains errata, notes, and some programming notes for the first edition (2008) of "Statistical Computing with R", ISBN 978-1-58488-545-0 (Hardcover). For readers of an ebook version, the list has been updated with margin notes that give more details about the location than the page number.

Errata

p. 11 [1] For R versions >= 3.0.0, `mean` for data frames is defunct. Change the function `mean` to `colMeans`:

```
by(iris[,1:4], iris$Species, colMeans)
```

[1] Section 1.6, Data Frames

p. 30 Last line:[2] $x \geq 0, \nu = 1, 2, \ldots$.

[2] Section 2.3, Chisquare and t

p. 35 [3]In the first two displayed equations: $\lim_{n \to \infty}$.

[3] Section 2.5

p. 58 Item 4. [4]

$$Z_1 = \sqrt{-2 \log U} \cos(2\pi V),$$
$$Z_2 = \sqrt{-2 \log U} \sin(2\pi V)$$

[4] Section 3.4

p. 114 Example 4.11[5] In `stars`, the labels argument must be a vector of character strings: use `levels` of the factor for labels.

```
stars(x[4:8], draw.segments = TRUE,
  labels = levels(x$sp), nrow = 4,
  ylim = c(-2,10), key.loc = c(3,-1))
```

[5] Section 4.5.3, code for segment plot in Figure 4.10

p. 121 [6] In step 2: $\overline{g(X)} = \frac{1}{m} \sum_{i=1}^{m} g(X_i)$.

[6] Section 5.2.1, Example 5.2

p. 132 [7] In sentence above (5.9): $\hat{\theta}_c = g(X) + c(f(X) - \mu)$.

[7] Section 5.5

p. 133 [8] last displayed equation, rightmost paren. expression should be squared. (The numerical answers following are correct.)

[8] Example 5.7

p. 142 [9]Example 5.10.

The standard errors will be the vector `se/sqrt{m}\verb`, so the summary is:

```
> rbind(theta.hat, se/sqrt(m))
                 [,1]        [,2]      [,3]         [,4]        [,5]
theta.hat 0.524114007 0.531358351 0.5461507 0.5250698759 0.526049238
          0.002436559 0.004181264 0.0096613 0.0009658794 0.001427685
```

[9] Section 5.6

p. 143 In "Variance in Importance Sampling"[10], in the equation for $Var(\hat{\theta})$, ds should be dx in the integral.

[10] Section 5.6

p. 158 Table 6.1[11] The right two columns reporting $n\,\widehat{se}$ are not correct. To obtain results for $n\,\widehat{MSE}$ in Table 6.1, `set.seed(522)`. The corrected table is given below.

	Normal		$p = 0.95$		$p = 0.90$	
k	$n\,\widehat{MSE}$	$n\,\widehat{se}$	$n\,\widehat{MSE}$	$n\,\widehat{se}$	$n\,\widehat{MSE}$	$n\,\widehat{se}$
1	0.976	0.140	6.229	0.353	11.485	0.479
2	1.019	0.143	1.954	0.198	4.126	0.287
3	1.009	0.142	1.304	0.161	1.956	0.198
4	1.081	0.147	1.168	0.153	1.578	0.178
5	1.048	0.145	1.280	0.160	1.453	0.170
6	1.103	0.149	1.395	0.167	1.423	0.169
7	1.316	0.162	1.349	0.164	1.574	0.177
8	1.377	0.166	1.503	0.173	1.734	0.186
9	1.382	0.166	1.525	0.175	1.694	0.184
10	1.491	0.172	1.646	0.181	1.843	0.192

p. 187 Line 1.[12] Remove hat from the first 'se'.

p. 200 Example 7.10. Misplaced right paren.; correction: [13]

```
#normal
print(boot.obj$t0 + qnorm(alpha) * sd(boot.obj$t))
```

p. 204 [14]

$$\hat{a} = \frac{\sum_{i=1}^{n}(\overline{\theta_{(.)}} - \theta_{(i)})^3}{6\left(\sum_{i=1}^{n}(\overline{\theta_{(.)}} - \theta_{(i)})^2\right)^{3/2}}, \tag{7.11}$$

p. 216 [15] The upper limit on the sum is $\binom{N}{n}$ in (8.4) and in the last equation on page 216.

p. 246 First paragraph, in second sentence:[16] "n tends to infinity" should be "m tends to infinity".

p. 260 [17] Last paragraph: delete the second sentence "Then an observed sample is generated."

p. 263 Example 9.7. [18]

$$E[X_2|x_1] = \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1 - \mu_1)$$

p. 283 [19] Sturges's Rule: "For large k (large n) the distribution of Binomial$(n, 1/2)$ is ..."

p. 312 Example 10.15.[20] The mean vectors used to generate the samples in the code and in the plots in Figure 10.13 are

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad \mu_3 = \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

p. 315 Exercise 10.5. The skewness adjustment factor is given in (10.8).

[11] Section 6.2.2

[12] Section 7.1.1, Example 7.2, paragraph following the code

[13] Section 7.4.3, Example 7.10, above R note 7.3

[14] Section 7.5

[15] Equation (8.4)

[16] Section 9.1.2

[17] Section 9.2.4, Example 9.6, paragraph before the code

[18] Section 9.3

[19] Section 10.1.1

[20] Section10.3.3

Notes

p. 56 [21]Remark on Example 3.7. Although 6 is an upper bound, it is not the least upper bound. The generator is more efficient if $c = 1.5$, the maximum value of $f(x)/g(x)$ for $0 \le x \le 1$.

[21] Section 3.3

p. 57 Code above para. 2: `#See Ch. 2` [22]

[22] Section 3.3, Example 3.7

p. 71–72 Example 3.16, summary statistics. [23] The `rmvn.eigen` generator takes the covariance matrix `Sigma` as an argument, so in general one may want to display `cov(X)` for comparison with `Sigma` rather than the sample correlation matrix `cor(X)`. (Here our `Sigma` was a correlation matrix.)

[23] Section 3.6.1

p. 178–179 Examples 6.14–6.15.[24] Although mathematically it is not an error, it is unnecessary to subtract the sample means in the expression `tests` of Example 6.14 because the sample means are subtracted in the function `count5test`. Mathematically, if $Z_i = X_i - \overline{X}$, then $\overline{Z} = 0$. The same is true in the expression for `alphahat` in Example 6.15. The `count5test` can be applied without centering the data first, as in Example 6.16.

[24] Section 6.4

p. 225–228 Examples 8.4-8.6.[25] The `knnFinder` package with `nn` function for finding nearest neighbors has been withdrawn from CRAN. These examples have been revised using the `ann` function in the `yaImpute` package.

[25] Section 8.3, Nearest Neighbor Tests

p. 323-324 Example 11.3.[26] In `system.time` the timings are hardware dependent; however, the vectorized version should be faster on all platforms.

[26] Section 11.1, Evaluating Functions

pp. 338–339 Example 11.11.[27] The first code snippet will produce a graph similar to Figure 11.3 but with x-axis ranging from about 2 to 8. To produce Figure 11.3 as shown, replace 8 with 15 in `seq(2, 8, .001)`.

[27] Section 11.5

p. 341 Example 11.12.[28] The histograms will of course vary slightly from Figure 11.4 because the data is generated at random. According to my notes, `set.seed(333)` before the first line of code should produce samples matching the histograms as shown on page 341. See the note below concerning `par(ask=TRUE)` to wait for user input before displaying each graph.

[28] Section 11.5

p. 342–343 Example 11.13.[29] `set.seed(333)` was set prior to run.

[29] Section 11.6

Programming Notes

1. `curve` is convenient for plotting a function. It can replace `lines` in some examples; e.g. in Example 3.2 to add the density curve to the histogram, instead of `lines` we can use:

```
curve(3*x^2, add=TRUE)
```

2. Displaying a sequence of graphs: `par(ask=TRUE)` has the effect that the user is asked for input before each new figure is drawn. Follow it with `par(ask=FALSE)` to restore to default behavior.

3. `sapply` can be used instead of `apply` in some examples, which eliminates the need for MARGIN and the need for the argument to have a dimension attribute. See e.g. Example 5.4 on page 123. The lines:

```
dim(x) <- length(x)
p <- apply(x, MARGIN=1, function(x, z) {mean(z < x)}, z=z)
```

   can be replaced with either version below:

```
p <- sapply(x, FUN=function(x,z) mean(z<x), z=z)
p <- sapply(x, function(x) mean(z<x))
```

4. A more elegant approach to the comparison of generators in Example 3.19 is to wrap the repeated statements in a function that takes the name of the generator (e.g. rmvn.eigen) as an argument. An example of a function that has a functional argument is `boot`; see Example 7.10 for a typical example.

Acknowledgements Thanks to several readers who sent comments, corrections, and suggestions.