Supplementary Notes

Data Analysis and Graphics Using R – An Example-Based Approach

John Maindonald and John Braun

These will be updated from time to time.

Chapter 2

Use of `xyplot()` to plot columns in parallel

Where column names on the left or right of the equation are separated by a plus sign (+), the plot is done in parallel for all the columns that are specified. This makes it possible, in the code for Figures 2.9A and 2.9B, to work directly from the `jobs`. The following code is for Figure 2.9A:

```
xyplot(BC+Alberta+Prairies+Ontario+Quebec+Atlantic ~ Date, data=jobs,
       ylab="Number of jobs", type="l")
```

The default is to overlay the plots on the one panel, as here. The approach that is demonstrated on p.433 (Subsection 14.3.4) might be used to add labels to the end of each line. Below, there is a note in connection with Subsection 14.3.4 that gives details.

The argument `outer=TRUE` gives separate panels, as in Figure 2.9B. In the following, data are plotted on a sliced logarithmic scale:

```
xyplot(BC+Alberta+Prairies+Ontario+Quebec+Atlantic ~ Date, data=jobs,
       type="l", layout=c(3,2),
       scales=list(y=list(relation="sliced", log=TRUE)), outer=TRUE)
```

Finally, the following modifies the code in Footnote 7 on page 56 to use the `+` operator and work directly from the `jobs` data frame. Additionally, a date object is plotted on the x-axis, and labels of the form `Jan95` are specified. The y-axis labels show both the numbers and values on a $\log_e$ scale:

```
ylabpos <- exp(pretty(log(unlist(jobs[,-7])), 100))
ylabels <- paste(round(ylabpos),"\n(", log(ylabpos), ")", sep="")
## Create a date object 'startofmonth'; use this instead of 'Date'
startofmonth <- seq(from=as.Date("1Jan1995", format="%d%b%Y"),
                    by="1 month", length=24)
atdates <- seq(from=as.Date("1Jan1995", format="%d%b%Y"),
               by="6 month", length=4)
datelabs <- format(atdates, "%b%y")
xyplot(BC+Alberta+Prairies+Ontario+Quebec+Atlantic ~ startofmonth,
       data=jobs, type="l", layout=c(3,2),
       scales=list(x=list(at=atdates, labels=datelabs),
                   y=list(relation="sliced", log=TRUE,
                   at=ylabpos, labels=ylabels)),
       ylab="Number of workers",
       outer=TRUE)
```

Chapter 4

Section 4..4: Analysis of variance

The statement on p.121 that "the analysis of variance table is obtained using the `anova()` function" is cryptic. As the code makes clear, the functions `anova()` and `aov()` are used in tandem. First, `aov()` is used to fit the analysis of variance model. The `anova()` function is then called with the resulting `aov` object as argument.

Here, `aov()` is used to fit a one-way classification model. The result of the model fit is to give the mean for each treatment level as the fitted value for that level. The function anova() is used with an `aov` or `lm` object as argument, to give an analysis of variance table.

One-way classification (or other `aov`) models can equivalently be fitted using `lm()`, as explained on page 242. The default output is then different.

Chapter 6

Subsection 6.8.1 – Errors in x

This subsection discusses just one of a variety of possible "errors in x" models, described in Carroll (2006) as the "classical" model. See Carroll (2006, pp. 49-52) for a summary of different types of models that have been proposed.

Multiple and orthogonal regression are discussed very briefly in the final paragraph on p. 209, again with reference to the classical model. Attenuation of regression coefficients is not the only possibility. As a simple example, suppose that one variable only is measured with error. Its coefficient is attenuated, while the coefficients of other variables may be reversed in sign, or show an effect when there is none. See Carroll (2006, pp. 52-55) for summary comment. Some further details follow.

Regression with a single covariate

Consider first regression with a single covariate. Under the classical model errors in explanatory variables, if they are sufficiently extreme, have two effects:

1. Estimates of the coefficient will be reduced, relative to the coefficient for the variable that is measured without error.

2. Very large samples may be required to show a statistically detectable coefficient.

The model is

$$y = \alpha + \beta x + \epsilon$$

We measure, not x, but $w = x + u$, where u is "measurement error".

Now assume that w is unbiased for x and that u is independent of x and ϵ.

Then, conditional on x, instead of

$$\hat{\beta} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

we have

$$\beta^* = \frac{\sum (w - \bar{w})(y - \bar{y})}{\sum (w - \bar{w})^2}$$

Then

$$\mathrm{E}[\sum (\mathrm{w} - \bar{\mathrm{w}})(\mathrm{y} - \bar{\mathrm{y}})] = \sum (\mathrm{x} - \bar{\mathrm{x}})(\mathrm{y} - \bar{\mathrm{y}})$$

This happens because y is independent of u.

$$\begin{aligned} \mathrm{E}[\sum(\mathrm{w}-\bar{\mathrm{w}})^2] &= \mathrm{E}[\sum(\mathrm{x}-\bar{\mathrm{x}}+\mathrm{u}-\bar{\mathrm{u}})^2] \\ &= \sum(x-\bar{x})^2+\sum(u-\bar{u})^2 \\ &= (n-1)\sigma_x^2+(n-1)\sigma_u^2 \end{aligned}$$

Then β^* is a consistent estimate of $\lambda\beta$, where

$$\lambda = \frac{\sigma_x^2}{\sigma_x^2+\sigma_u^2}$$

One covariate measured with error; others without error

The coefficient of the variable that is measured with error is attenuated, as in the single variable case. The coefficients of other variables may be reversed in sign, or show an effect when there is none. See Carroll (2006, pp. 52-55) for summary comment.

Suppose that

$$y = \beta_x\mathbf{x}+\beta_z z+\epsilon$$

If w is unbiased for x and the measurement error u is independent of x and z, then least squares regression yields a consistent estimate of $\lambda\beta_x$

$$\lambda = \frac{\sigma_{x|z}^2}{\sigma_{x|z}^2+\sigma_u^2}$$

The σ_x^2 that appears in the single covariate case is replaced by $\sigma_{x|z}^2$.

A new feature is the bias in the least squares estimate of β_z. The naive least squares estimator estimates

$$\beta_z+\beta_x(1-\lambda)\gamma_{x|z} \tag{1}$$

where $\gamma_{x|z}$ is the coefficient of z in the least squares regression of x on z. The least squares estimate may be non-zero value even though $\beta_z = 0$. Where $\beta_z \neq 0$, the least squares estimate may, depending on the relative values of β_z, β_x and λ be reversed in sign from β_z.

Where there are multiple explanatory variables that are measured without error, equation 1 can be applied to each of them in turn.

One covariate measured without error – a simulation exercise

Here is an exercise that illustrates how measurement error in one variable may lead to a spurious effect for a variable that is measured without error. The code simulates data in which the regression relationship $y = 2.5 + 1.5x$ is the same in each of two groups, which we call `CTL` and `TRT`. The plot identifies the two fitted lines. These are, to within statistical error, identical:

```
‘errorsINxGP‘ <-
  function(n=1000, xshift=1.25, a=2.5, b=1.5,
           gpSDx=2, yerr=0.5, sdRatio=1.5){
      ## xshift is the difference in mean value of x, between ctl and TRT
    xraw <- rnorm(2*n, 12.5, gpSDx)
    df <- data.frame(x=rnorm(2*n, 12.5, gpSDx) + rep(c(0,xshift), c(n,n)),
                     trt = rep(c("CTL", "TRT"), c(n,n)))
    df$y <- a + b*df$x + rnorm(2*n, sd=yerr)
      ## b is the slope of the regression line of y on the "true" x
    ## Now add random error to each of the $x$-values, thus:
      ## gpSDx is the conditional SD of x, given the treatment groups
    df$xWITHerr <- df$x + rnorm(2*n, sd=sdRatio*gpSDx)
```

```
    ## sdRatio is the ratio of the measurement error SD to SDx
  b <- lm(y ~ trt+x, data=df)$coef
  B <- lm(y ~ trt+xWITHerr, data=df)$coef
  atten <- 1/(1+sdRatio^2)  # Theoretical attenuation
  cat("\n", "Attenuation, simulated model =", round(atten,3))
  cat("\n", "Slope Estimates - On x:", round(b[3],3), "  On xWITHerr:",
      round(B[3],3), "\n")
  cat("\n", "Estimated group effect - Reg on x", round(b[2],3),
      "  Reg on xWITHerr", round(B[2],3), "\n")
  plt <- xyplot(y ~ x+xWITHerr, data=df, groups=trt,
                panel=function(x,y,...){
                  panel.superpose(x,y,type="p",...)
                  subs <- list(...)$subscripts
                  fac <- list(...)$groups[subs]
                  gps.lm <- lm(y~fac+x)
                  hat <- fitted(gps.lm)
                  panel.superpose(x,y=hat,type="r",...)
                },
                auto.key=list(columns=2)
                )
  print(plt)
  invisible(df)
}
```

The left panel shows the regression lines when x is measured without error. The right panel shows the fitted regression lines when random error of the same order of magnitude as the within groups variation in x is added to x, giving the column of values $xWITHerr$. The default argument `n=1000` gives results that are close to the theoretical lines. Whereas the lines that appear in the left panel can be expected to overlap, those in the right panel are distinct. The "measurement" error in x leads to a spurious treatment (or group) effect.

(a) Run the function for several different values of `xshift` in the interval (0,1.5), and plot the estimate of the treatment effect (identified in the output as `trtTRT`) against `xshift`.

(b) Run the function for several different values of `sdRatio` in the interval (0,1.5) and plot the estimate of the treatment effect against `xshift`.

(c) Run the function with `b = -1.5`. How does the estimate of the treatment effect change, as compared with `b = 1.5`? Explain the change.

Chapter 8

Fitted and Predicted values for GLMs

Page 246 introduces the notation

$$f\left(\mathrm{E}[y]\right) = \alpha + \beta x$$

where $f()$ is the *link* function. In the fitted model, $\alpha + \beta x$ is the linear predictor, while $\mathrm{E}[y]$ is the expected value of the response.

Thus, for a logistic regression model

$$f(\pi) = \log(\frac{\pi}{1-\pi})$$

where π is the expected proportion.

For GLMs, there are two "types" of predicted values. The function `fitted()` delivers fitted values on the scale of the response. The text does not mention, however, the use of `predict()` to get fitted values on the scale of the linear predictor.

The following uses the example on the first three lines of page 250, where we had

```
anes.logit <- glm(nomove ~ conc, family=binomial(link="logit"),
                  data=anesthetic)
```

Scale used	Logistic regression	Code Used in R
response (`type="response"`)	$\pi = \mathrm{E}[p]$	`predict(anes.logit, type="response")` or `fitted(anes.logit)`
linear predictor (`type="link"`)	$\log(\frac{\pi}{1-\pi})$	`predict(anes.logit, type="link")` (for `predict()`, `type="link"` is the default)

Here are the two different types of predicted values, for the object `anes.logit` that was created above:

```
> fitted(anes.logit)
    1     2     3     4     5     6     7     8     9    10    11    12    13
0.289 0.553 0.790 0.790 0.553 0.999 0.920 0.118 0.920 0.790 0.118 0.920 0.999
   14    15    16    17    18    19    20    21    22    23    24    25    26
0.790 0.920 0.790 0.790 0.118 0.118 0.553 0.118 0.118 0.289 0.118 0.289 0.553
   27    28    29    30
0.289 0.553 0.289 0.553
> predict(anes.logit)
     1      2      3      4      5      6      7      8      9     10     11
-0.902  0.211  1.325  1.325  0.211  7.448  2.438 -2.015  2.438  1.325 -2.015
    12     13     14     15     16     17     18     19     20     21     22
 2.438  7.448  1.325  2.438  1.325  1.325 -2.015 -2.015  0.211 -2.015 -2.015
    23     24     25     26     27     28     29     30
-0.902 -2.015 -0.902  0.211 -0.902  0.211 -0.902  0.211
```

Chapter 10

The model fit

```
orthsame.lmer <-
  lmer(logdist ~ Sex + I(age - 11) + (I(age - 11) | Subject),
       data=Orthodont, method="ML", subset=keep)
```

may not converge. If you encounter this problem, install version 0.9975-11 of *lme4* or later, and use `lmer2()` in place of `lmer()`. See `help(lmer2)` for current limits on the use of `lmer2` objects in subsequent calculations. In due course, the function `lmer2()` will morph into an updated version of `lmer()`.

Chapter 12

Section 12.4: Further Reading

Principal components

Computationally, principal components analysis uses the a principal components form of matrix decomposition, applied to a variance-covariance or correlation matrix. The variance-covariance matrix has the form $X'X$, where X is derived from the observations by variables matrix by subtracting off means in each

column. For calculations with the correlation matrix, values in each column must, additionally, be scaled so that the squared values sum to 1.0.

Equivalently, it is possible to work directly with a version of X, and apply the singular value decomposition. This leads to a different form of description of the analysis. As an example, see Booth et al (2002), where the singular value decomposition is applied to a matrix that gives, for each of a number of age categories, mortalities for each of the years 1907-1999 or (for some of the analyses) 1968-1999. In the "Lee-Carter" methodology that is described and critically evaluated in that paper, scores on what is really the first principal component are used, after an adjustment that better reproduces the total number of deaths in any year, as a basis for mortality projections into the future.

Functional data analysis is an extension of such an approach.

Discriminant methods – Support Vector Machines

Support vector machines have recently come into prominence. The *e1071* package in R handles the fitting of Support Vector Machines. Meyer et al (2003) use a number of different data sets to compare Support Vector Machines with other approaches.

Chapter 14

Subsection 14.3.4

Addition of axis labels to lattice plots

Here is an alternative to the paragraph that starts at line 4 on page 433, now using the `+` operator to plot the first six columns of the data frame `jobs` in parallel:

```
atdates <- seq(from=95, by=0.5, length=4)
datelabs <- format(seq(from=as.Date("1Jan1995", format="%d%b%Y"),
                       by="6 month", length=4), "%b%y")
xyplot(BC+Alberta+Prairies+Ontario+Quebec+Atlantic ~ Date, data=jobs,
       xlab="Date", ylab="Number of Jobs",
       scales=list(x=list(at=atdates, labels=datelabs)))
```

Add labels thus:

```
library(grid)
trellis.focus("panel", row=1, column=1, clip.off=TRUE)
xpos <- unit(rep(max(jobs$Date),6),"native") + unit(1, "strwidth","a")
ypos <- bounce(unlist(jobs[23,1:6]),
               as.numeric(convertUnit(unit(1, "strheight","A"), "native")))
grid.text(label=names(jobs)[1:6], x=xpos, y=ypos,
          default.units="native", just="left")
trellis.unfocus()
```

Subsection 14.12

Version 0.17-2 of *lattice* introduced the function `simpleTheme()`. Among the arguments that are accepted are `col`, `col.points`, `col.line`, `cex`, `pch`, `font`, `lty` and `lwd`. They affect the trellis parameters in the manner that might be expected. The list that is returned by `simpleTheme()` can be supplied as argument to texttttrellis.par.set(), or given as an argument to `par.settings` in a lattice function call. In the former case it affects (until changed) all graphs that are subsequently sent to the current device. In the latter case changes affect only the graph that is created by that function call. In either case, the setting is made both for the graph and any key (e.g., created using the argument `auto.key`) which uses the global settings for lattice parameters.

Here is an example of its use:

```
library(DAAG)
aisBS <- subset(ais,
    sport %in% c("B_Ball", "Swim") )
aisBS$sport <- factor(aisBS$sport)
xyplot(hg ~ rcc | sex, type=c("p","r"), groups=sport, data=aisBS,
       auto.key=list(points=TRUE, lines=TRUE, columns=2),
       par.settings=simpleTheme(pch = c(3,16), col=c("black","gray"),
       lwd=1.5))
```

References

Booth, H., Maindonald, J., and Smith, L. 2002. Applying Lee-Carter under conditions of variable mortality decline. Population Studies, 56: 325-336.

Carroll, R., Ruppert, D. and Stefanski, L. A. 2006. *Measurement Error in Nonlinear Models.* 2^{nd} edition, Chapman and Hall.

Meyer, D., Leisch, F., and Hornik, K. 2003. The Support Vector Machine under test. Neurocomputing 55: 169-186.

Smith, J. A. and Todd, P.E. 2005. Does Matching overcome LaLonde's critique of nonexperimental estimators? *Journal of Econometrics* 125: 305-353.

John Maindonald
December 7, 2007