

Chapter 17

Stuff for Students

17.1 R/Splus and Arc

R is the free version of *Splus*. The website (<http://www.stat.umn.edu>) has useful links for *Arc* which is the software developed by Cook and Weisberg (1999a). The website (<http://www.stat.umn.edu>) also has a link to **Cran** which gives *R* support. As of June 2009, the author's personal computer has Version 2.4.1 (December 18, 2006) of *R*, *Splus*-2000 (see Mathsoft 1999ab) and Version 1.03 (August 2000) of *Arc*. Many of the text *R/Splus* functions and figures were made in the middle 1990's using *Splus* on a workstation.

Downloading the book's data.lsp files into Arc

Many homework problems use data files for *Arc* contained in the book's website (www.math.siu.edu/olive/regbk.htm). As an example, open *cbrain.lsp* file with *Notepad*. Then use the menu commands "File>Save As". A window appears. On the top "Save in" box change what is in the box to "Floppy(A:)" in order to save the file on a disk. Then in *Arc* activate the *cbrain.lsp* file with the menu commands "File > Load > 3 1/2 Floppy(A:) > cbrain.lsp."

Alternatively, open *cbrain.lsp* file with *Notepad*. Then use the menu commands "File>Save As". A window appears. On the top "Save in" box change what is in the box to "My Documents". Then go to *Arc* and use the menu commands "File>Load". A window appears. Change "Arc" to "My Documents" and open *cbrain.lsp*.

Downloading the book's R/Splus functions *regpack.txt* into *R* or *Splus*:

Many of the homework problems use *R/Splus* functions contained in the book's website (www.math.siu.edu/olive/regbk.htm) under the file name *regpack.txt*. Suppose that you download *regpack.txt* onto a disk. Enter *R* and wait for the cursor to appear. Then go to the *File* menu and drag down *Source R Code*. A window should appear. Navigate the *Look in* box until it says *3 1/2 Floppy(A:)*. In the *Files of type* box choose *All files(*.*)* and then select *regpack.txt*. The following line should appear in the main *R* window.

```
> source("A:/regpack.txt")
```

Type *ls()*. About 70 *R/Splus* functions from *regpack.txt* should appear.

When you finish your *R/Splus* session, enter the command *q()*. A window asking “*Save workspace image?*” will appear. Click on *No* if you do not want to save the programs in *R*. (If you do want to save the programs then click on *Yes*.)

If you use *Splus*, the command

```
> source("A:/regpack.txt")
```

will enter the functions into *Splus*. Creating a special workspace for the functions may be useful.

This section gives tips on using *R/Splus*, but is no replacement for books such as Becker, Chambers, and Wilks (1988), Chambers (1998), Crawley (2005, 2007), Fox (2002) or Venables and Ripley (2003). Also see Mathsoft (1999ab) and use the website (www.google.com) to search for useful websites. For example enter the search words *R documentation*.

The command *q()* gets you out of *R* or *Splus*.

Least squares regression is done with the function *lsfit*.

The commands *help(fn)* and *args(fn)* give information about the function *fn*, eg if *fn = lsfit*.

Type the following commands.

```
x <- matrix(rnorm(300),nrow=100,ncol=3)
y <- x%*%1:3 + rnorm(100)
out<- lsfit(x,y)
out$coef
ls.print(out)
```

The first line makes a 100 by 3 matrix x with $N(0,1)$ entries. The second line makes $y[i] = 0 + 1 * x[i, 1] + 2 * x[i, 2] + 3 * x[i, 2] + e$ where e is $N(0,1)$. The term $1:3$ creates the vector $(1, 2, 3)^T$ and the matrix multiplication operator is `%*%`. The function `lsfit` will automatically add the constant to the model. Typing “out” will give you a lot of irrelevant information, but `out$coef` and `out$resid` give the OLS coefficients and residuals respectively.

To make a residual plot, type the following commands.

```
fit <- y - out$resid
plot(fit,out$resid)
title("residual plot")
```

The first term in the plot command is always the horizontal axis while the second is on the vertical axis.

To put a graph in Word, hold down the *Ctrl* and *c* buttons simultaneously. Then select “paste” from the *Word* Edit menu.

To enter data, open a data set in *Notepad* or *Word*. You need to know the number of rows and the number of columns. Assume that each case is entered in a row. For example, assuming that the file *cyp.lsp* has been saved on your disk from the webpage for this book, open *cyp.lsp* in *Word*. It has 76 rows and 8 columns. In *R* or *Splus*, write the following command.

```
cyp <- matrix(scan(),nrow=76,ncol=8,byrow=T)
```

Then copy the data lines from *Word* and paste them in *R/Splus*. If a curser does not appear, hit *enter*. The command `dim(cyp)` will show if you have entered the data correctly.

Enter the following commands

```
cypy <- cyp[,2]
cypx<- cyp[,-c(1,2)]
lsfit(cypx,cypy)$coef
```

to produce the output below.

Intercept	X1	X2	X3	X4
205.40825985	0.94653718	0.17514405	0.23415181	0.75927197
X5	X6			
-0.05318671	-0.30944144			

To check that the data is entered correctly, fit LS in *Arc* with the response variable *height* and the predictors *sternal height*, *finger to ground*, *head length*, *nasal length*, *bigonal breadth*, and *cephalic index* (entered in that order). You should get the same coefficients given by *R* or *Splus*.

Making functions in R and Splus is easy.

For example, type the following commands.

```
mysquare <- function(x){  
# this function squares x  
r <- x^2  
r }
```

The second line in the function shows how to put comments into functions.

Modifying your function is easy.

Use the `fix` command.

```
fix(mysquare)
```

This will open an editor such as *Notepad* and allow you to make changes.

In *Splus*, the command `Edit(mysquare)` may also be used to modify the function *mysquare*.

To save data or a function in *R*, when you exit, click on *Yes* when the “*Save worksheet image?*” window appears. When you reenter *R*, type `ls()`. This will show you what is saved. You should rarely need to save anything for the material in the first thirteen chapters of this book. In *Splus*, data and functions are automatically saved. To remove unwanted items from the worksheet, eg *x*, type `rm(x)`,

`pairs(x)` makes a scatterplot matrix of the columns of *x*,

`hist(y)` makes a histogram of *y*,

`boxplot(y)` makes a boxplot of *y*,

`stem(y)` makes a stem and leaf plot of *y*,

`scan()`, `source()`, and `sink()` are useful on a *Unix* workstation.

To type a simple list, use `y <- c(1,2,3.5)`.

The commands `mean(y)`, `median(y)`, `var(y)` are self explanatory.

The following commands are useful for a scatterplot created by the command `plot(x,y)`.

```
lines(x,y), lines(lowess(x,y,f=.2))
```

```
identify(x,y)
```

```
abline(out$coef), abline(0,1)
```

The usual arithmetic operators are $2 + 4$, $3 - 7$, $8 * 4$, $8/4$, and

$2^{\{10\}}$.

The *i*th element of vector *y* is `y[i]` while the *ij* element of matrix *x* is `x[i, j]`. The second row of *x* is `x[2,]` while the 4th column of *x* is `x[, 4]`. The transpose of *x* is `t(x)`.

The command `apply(x,1,fn)` will compute the row means if `fn = mean`. The command `apply(x,2,fn)` will compute the column variances if `fn = var`. The commands `cbind` and `rbind` combine column vectors or row vectors with an existing matrix or vector of the appropriate dimension.

Downloading the book's R/Splus data sets *robdata.txt* into *R* or *Splus* is done in the same way for downloading *rpack.txt*. Use the command

```
> source("A:/robdata.txt")
```

For example the command

```
> lsfit(belx,bely)
```

will perform the least squares regression for the Belgian telephone data.

Transferring Data to and from *Arc* and *R* or *Splus*.

For example, suppose that the Belgium telephone data (Rousseeuw and Leroy 1987, p. 26) has the predictor *year* stored in *x* and the response *number of calls* stored in *y* in *R* or *Splus*. Combine the data into a matrix *z* and then use the *write.table* command to display the data set as shown below. The

```
sep=' '
```

separates the columns by two spaces.

```
> z <- cbind(x,y)
> write.table(data.frame(z),sep='  ')
```

row.names	z.1	y
1	50	0.44
2	51	0.47
3	52	0.47
4	53	0.59
5	54	0.66
6	55	0.73
7	56	0.81
8	57	0.88
9	58	1.06
10	59	1.2
11	60	1.35
12	61	1.49
13	62	1.61
14	63	2.12
15	64	11.9
16	65	12.4

17	66	14.2
18	67	15.9
19	68	18.2
20	69	21.2
21	70	4.3
22	71	2.4
23	72	2.7073
24	73	2.9

To enter a data set into *Arc*, use the following template *new.lsp*.

```
dataset=new
begin description
Artificial data.
Contributed by David Olive.
end description
begin variables
col 0 = x1
col 1 = x2
col 2 = x3
col 3 = y
end variables
begin data
```

Next open *new.lsp* in *Notepad*. (Or use the *vi* editor in Unix. Sophisticated editors like *Word* will often work, but they sometimes add things like page breaks that do not allow the statistics software to use the file.) Then copy the data lines from *R/Splus* and paste them below *new.lsp*. Then modify the file *new.lsp* and save it on a disk as the file *belg.lsp*. (Or save it in *mdata* where *mdata* is a data folder added within the *Arc data* folder.) The header of the new file *belg.lsp* is shown below.

```
dataset=belgium
begin description
Belgium telephone data from
Rousseeuw and Leroy (1987, p. 26)
end description
begin variables
```

```

col 0 = case
col 1 = x = year
col 2 = y = number of calls in tens of millions
end variables
begin data
1 50 0.44
. . .
. . .
. . .
24 73 2.9

```

The file above also shows the first and last lines of data. The header file needs a data set name, description, variable list and a *begin data* command. Often the description can be copied and pasted from source of the data, eg from the STATLIB website. Note that the first variable starts with *Col 0*.

To transfer a data set from Arc to R or Splus, select the item “Display data” from the dataset’s menu. Select the variables you want to save, and then push the button for “Save in R/Splus format.” You will be prompted to give a file name. If you select *bodfat*, then two files *bodfat.txt* and *bodfat.Rd* will be created. The file *bodfat.txt* can be read into either *R* or *Splus* using the *read.table* command. The file *bodfat.Rd* saves the documentation about the data set in a standard format for *R*.

As an example, the following command was used to enter the body fat data into *Splus*. (The *mdata* folder does not come with *Arc*. The folder needs to be created and filled with files from the book’s website. Then the file *bodfat.txt* can be stored in the *mdata* folder.)

```

bodfat <- read.table("C:\\ARC\\DATA\\MDATA\\BODFAT.TXT",header=T)
bodfat[,16] <- bodfat[,16]+1

```

The last column of the body fat data consists of the case numbers which start with 0 in *Arc*. The second line adds one to each case number.

As another example, use the menu commands “File>Load>Data>Arcg>forbes.lsp” to activate the forbes data set. From the *Forbes* menu, select *Display Data*. A window will appear. Double click on *Temp* and *Pressure*. Click on *Save Data in R/Splus Format* and save as *forbes.txt* in the folder *mdata*.

Enter *Splus* and type the following command.


```
forbes<-read.table("C:\\ARC\\DATA\\ARCG\\FORBES.TXT",header=T)
```

The command *forbes* will display the data set.

Getting information about a library in R

In *R*, a *library* is an add-on package of *R* code. The command *library()* lists all available libraries, and information about a specific library, such as *lqs* for robust estimators like *cov.mcd* or *ts* for time series estimation, can be found, eg, with the command *library(help=lqs)*.

Downloading a library into R

Many researchers have contributed a *library* of *R* code that can be downloaded for use. To see what is available, go to the website (<http://cran.us.r-project.org/>) and click on the Packages icon. Suppose you are interested the Weisberg (2002) dimension reduction library *dr*. Scroll down the screen and click on *dr*. Then click on the file corresponding to your type of computer, eg *dr 2.0.0.zip* for *Windows*. My unzipped files are stored in my directory

```
C:\unzipped.
```

The file

```
C:\unzipped\dr
```

contains a folder *dr* which is the *R library*. Cut this folder and paste it into the *R* library folder. (On my computer, I store the folder *rw1011* in the file

```
C:\R-Gui.
```

The folder

```
C:\R-Gui\rw1011\library
```

contains the library packages that came with *R*.) Open *R* and type the following command.

```
library(dr)
```

Next type *help(dr)* to make sure that the library is available for use.

17.2 Hints for Selected Problems

Chapter 1

1.1 $\beta^T \mathbf{x} = \mathbf{x}^T \beta$

Chapter 2

2.1 $F_o = 0.904$, p-value > 0.1 , fail to reject H_o , so the reduced model is good

2.2 a) 25.970

b) $F_o = 0.600$, p-value > 0.5 , fail to reject H_o , so the reduced model is good

2.3 a) (1.229, 3.345)

b) (1.0825, 3.4919)

2.4 c) $F_o = 265.96$, pvalue = 0.0, reject H_o , there is a MLR relationship between the response variable height and the predictors sternal height and finger to ground.

2.6 No, the relationship should be linear.

2.7 No, since 0 is in the CI. X could be a very useful predictor for Y , eg if $Y = X^2$.

2.11 a) $7 + \beta X_i$

b) $b = \sum (Y_i - 7)X_i / \sum X_i^2$

2.14 a) $b_3 = \sum X_{3i}(Y_i - 10 - 2X_{2i}) / \sum X_{3i}^2$. The second partial derivative = $\sum X_{3i}^2 > 0$.

2.21 d) The first assumption to check would be the constant variance assumption.

Chapter 3

3.1 The model uses constant, finger to ground and sternal height. (You can tell what the variable are by looking at which variables are deleted.)

3.2 Use L3. L1 and L2 have more predictors and higher C_p than L3 while L4 does not satisfy the $C_p \leq 2k$ screen.

3.3 a) L2.

b) Use L3 since L1 has too many predictors while L4 does not satisfy the $C_p \leq 2k$ screen.

3.4 Use a constant, A, B and C since this is the only model that satisfies the $C_p \leq 2k$ screen.

b) Use the model with a constant and B since it has the smallest C_p and the smallest k such that the $C_p \leq 2k$ screen is satisfied.

3.7 a) The plot looks roughly like the SW corner of a square.

b) No, the plot is nonlinear.

c) Want to spread small values of y , so make λ smaller. Hence use $y^{(0)} = \log(y)$.

3.8 Several of the marginal relationships are nonlinear, including $E(M|H)$.

3.9 This problem has the student reproduce Example 5.1. Hence $\log(Y)$ is the appropriate response transformation.

3.10 Plots b), c) and e) suggest that $\log(ht)$ is needed while plots d), f) and g) suggest that $\log(ht)$ is not needed. Plots c) and d) show that the residuals from both models are quite small compared to the fitted values. Plot d) suggests that the two models produce approximately the same fitted values. Hence if the goal is prediction, the expensive $\log(ht)$ measurement does not seem to be needed.

3.11 h) The submodel is ok, but the response and residual plots found in f) for the submodel do not look as good as those for the full model found in d). Since the submodel residuals do not look good, more terms are probably needed in the model.

3.12 b) Forward selection gives constant, $(\text{size})^{1/3}$, age, sex, breadth and cause.

c) Backward elimination gives constant, age, cause, cephalic, headht, length and sex.

d) Forward selection is better because it has fewer terms and a smaller C_p .

e) The variables are highly correlated. Hence backward elimination quickly eliminates the single best predictor $(\text{size})^{1/3}$ and can not get a good model that only has a few terms.

f) Although the model in c) could be used, a better model uses constant, age, sex and $(\text{size})^{1/3}$.

j) The FF and RR plots are good and so are the response and residual plots if you ignore the good leverage points corresponding to the 5 babies.

8.3. See Example 8.6.

9.3. See Example 9.2.

10.2 a) $ESP = 1.11108$, $\exp(ESP) = 3.0376$ and $\hat{\rho} = \exp(ESP)/(1 + \exp(ESP)) = 3.0376/(1 + 3.0376) = 0.7523$.

10.3 $G^2(O|F) = 62.7188 - 13.5325 = 49.1863$, $df = 3$, $p\text{-value} = 0.00$, reject H_0 , there is a LR relationship between ape and the predictors lower jaw, upper jaw and face length.

10.4 $G^2(R|F) = 17.1855 - 13.5325 = 3.653$, $df = 1$, $0.05 < p\text{-value} < 0.1$, fail to reject H_0 , the reduced model is good.

10.5 a) B4

b) EE plot

c) B3 is best. B1 has too many predictors with large Wald p -values, B2 still has too many predictors (want $\leq 300/10 = 30$ predictors) while B4 has too small of a p -value for the change in deviance test.

10.10 b) Use the log rule: $(\max \text{ age})/(\min \text{ age}) = 1400 > 10$.

e) The slice means track the logistic curve very well if 8 slices are used.

i) The EE plot is linear.

j) The slice means track the logistic curve very well if 8 slices are used.

n) The slice form -0.5 to 0.5 is bad since the symbol density is not approximately constant from the top to the bottom of the slice.

10.11 c) Should have 200 cases, $df = 178$ and deviance = 112.168.

d) The ESS plot with 12 slices suggests that the full model is good.

h) The submodel I_1 that uses a constant, AGE, CAN, SYS, TYP and FLOC and the submodel I_2 that is the same as I_1 but also uses FRACE seem to be competitors. If the factor FRACE is not used, then the EY plot follows 3 lines, one for each race. The Wald p -values suggest that FRACE is not needed, but the EE plot suggests that FRACE is needed. I think that the EE plot is generally more trustworthy, so use model I_2 .

10.12 b) The ESS plot (eg with 4 slices) is bad, so the LR model is bad.

d) Now the ESS plot (eg with 12 slices) is good in that slice smooth and the logistic curve are close where there is data (also the LR model is good at classifying 0's and 1's).

f) The MLE does not exist since there is perfect classification (and the logistic curve can get close to but never equal a discontinuous step function). Hence Wald p-values tend to have little meaning; however, the change in deviance test tends to correctly suggest that there is an LR relationship when there is perfect classification.

For this problem, $G^2(O|F) = 62.7188 - 0.00419862 = 62.7146$, $df = 1$, $p\text{-value} = 0.00$, so reject H_0 and conclude that there is an LR relationship between ape and the predictor x_3 .

10.14 The ESS plot should look ok, but the function uses a default number of slices rather than allowing the user to select the number of slices using a “slider bar” (a useful feature of *Arc*).

10.15 a)

Number in Model	Rsquare	C(p)	Variables in model						
6	0.2316	7.0947	X3	X4	X6	X7	X9	X10	

c) The slice means follow the logistic curve fairly well with 8 slices.

e) The EE plot is linear.

f) The slice means follow the logistic curve fairly well with 8 slices.

11.1a $ESP = 0.2812465$ and $\hat{\mu} = \exp(ESP) = 1.3248$.

11.2 $G^2(O|F) = 187.490 - 138.685 = 48.805$, $df = 2$, $p\text{-value} = 0.00$, reject H_0 , there is a LLR relationship between possums and the predictors habitat and stags.

11.5 a) A good submodel uses a constant, Bar, Habitat and Stags as predictors.

d) The EY and EE plots are good as are the Wald p-values. Also $AIC(\text{full}) = 141.506$ while $AIC(\text{sub}) = 139.644$.

11.6 k) The deleted point is certainly influential. Without this case, there does not seem to be a LLR relationship between the predictors and the

response.

m) The weighted residual plot suggests that something is wrong with the model since the plotted points scatter about a line with positive slope rather than a line with 0 slope. The deviance residual plot does not suggest that anything is wrong with the model.

11.7 a) Since this is simulated LLR data, the EY plot should look ok, but the function uses a default lowess smoothing parameter rather than allowing the user to select smoothing parameter using a “slider bar” (a useful feature of *Arc*).

b) The data should the identity line in the weighted forward response plots. In about 1 in 20 plots there will be a very large count that looks like an outlier. The weighted residual plot based on the MLE usually looks better than the plot based on the minimum chi-square estimator (the MLE plot tend to have less of a “left opening megaphone shape”).

Chapter 14

14.1 a) $X_2 \sim N(100, 6)$.

b)

$$\begin{pmatrix} X_1 \\ X_3 \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 49 \\ 17 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ -1 & 4 \end{pmatrix} \right).$$

c) $X_1 \perp\!\!\!\perp X_4$ and $X_3 \perp\!\!\!\perp X_4$.

d)

$$\rho(X_1, X_2) = \frac{\text{Cov}(X_1, X_3)}{\sqrt{\text{VAR}(X_1)\text{VAR}(X_3)}} = \frac{-1}{\sqrt{3}\sqrt{4}} = -0.2887.$$

14.2 a) $Y|X \sim N(49, 16)$ since $Y \perp\!\!\!\perp X$. (Or use $E(Y|X) = \mu_Y + \Sigma_{12}\Sigma_{22}^{-1}(X - \mu_x) = 49 + 0(1/25)(X - 100) = 49$ and $\text{VAR}(Y|X) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = 16 - 0(1/25)0 = 16$.)

b) $E(Y|X) = \mu_Y + \Sigma_{12}\Sigma_{22}^{-1}(X - \mu_x) = 49 + 10(1/25)(X - 100) = 9 + 0.4X$.

c) $\text{VAR}(Y|X) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = 16 - 10(1/25)10 = 16 - 4 = 12$.

14.4 The proof is identical to that given in Example 10.2. (In addition, it is fairly simple to show that $M_1 = M_2 \equiv M$. That is, M depends on Σ but not on c or g .)

14.6 a) Sort each column, then find the median of each column. Then $\text{MED}(\mathbf{W}) = (1430, 180, 120)^T$.

b) The sample mean of $(X_1, X_2, X_3)^T$ is found by finding the sample mean of each column. Hence $\bar{\mathbf{x}} = (1232.8571, 168.00, 112.00)^T$.

14.11 $\Sigma\mathbf{B} = E[E(\mathbf{X}|\mathbf{B}^T\mathbf{X})\mathbf{X}^T\mathbf{B}] = E(\mathbf{M}_B\mathbf{B}^T\mathbf{X}\mathbf{X}^T\mathbf{B}) = \mathbf{M}_B\mathbf{B}^T\Sigma\mathbf{B}$. Hence $\mathbf{M}_B = \Sigma\mathbf{B}(\mathbf{B}^T\Sigma\mathbf{B})^{-1}$.

14.13 The 4 plots should look nearly identical with the five cases 61–65 appearing as outliers.

Chapter 15

15.1

a) $\hat{e}_i = Y_i - T(Y)$.

b) $\hat{e}_i = Y_i - \mathbf{x}_i^T\hat{\boldsymbol{\beta}}$.

c)

$$\hat{e}_i = \frac{Y_i}{\hat{\beta}_1 \exp[\hat{\beta}_2(x_i - \bar{x})]}.$$

d) $\hat{e}_i = \sqrt{w_i}(Y_i - \mathbf{x}_i^T\hat{\boldsymbol{\beta}})$.

15.2

a) Since Y is a (random) scalar and $E(\mathbf{w}) = \mathbf{0}$, $\Sigma_{\mathbf{x},Y} = E[(\mathbf{x} - E(\mathbf{x}))(Y - E(Y))^T] = E[\mathbf{w}(Y - E(Y))] = E(\mathbf{w}Y) - E(\mathbf{w})E(Y) = E(\mathbf{w}Y)$.

b) Using the definition of z and \mathbf{r} , note that $Y = m(z) + e$ and $\mathbf{w} = \mathbf{r} + (\Sigma_{\mathbf{x}}\boldsymbol{\beta})\boldsymbol{\beta}^T\mathbf{w}$. Hence $E(\mathbf{w}Y) = E[(\mathbf{r} + (\Sigma_{\mathbf{x}}\boldsymbol{\beta})\boldsymbol{\beta}^T\mathbf{w})(m(z) + e)] = E[(\mathbf{r} + (\Sigma_{\mathbf{x}}\boldsymbol{\beta})\boldsymbol{\beta}^T\mathbf{w})m(z)] + E[\mathbf{r} + (\Sigma_{\mathbf{x}}\boldsymbol{\beta})\boldsymbol{\beta}^T\mathbf{w}]E(e)$ since e is independent of \mathbf{x} . Since $E(e) = 0$, the latter term drops out. Since $m(z)$ and $\boldsymbol{\beta}^T\mathbf{w}m(z)$ are (random) scalars, $E(\mathbf{w}Y) = E[m(z)\mathbf{r}] + E[\boldsymbol{\beta}^T\mathbf{w}m(z)]\Sigma_{\mathbf{x}}\boldsymbol{\beta}$.

c) Using result b), $\Sigma_{\mathbf{x}}^{-1}\Sigma_{\mathbf{x},Y} = \Sigma_{\mathbf{x}}^{-1}E[m(z)\mathbf{r}] + \Sigma_{\mathbf{x}}^{-1}E[\boldsymbol{\beta}^T\mathbf{w}m(z)]\Sigma_{\mathbf{x}}\boldsymbol{\beta} = E[\boldsymbol{\beta}^T\mathbf{w}m(z)]\Sigma_{\mathbf{x}}^{-1}\Sigma_{\mathbf{x}}\boldsymbol{\beta} + \Sigma_{\mathbf{x}}^{-1}E[m(z)\mathbf{r}] = E[\boldsymbol{\beta}^T\mathbf{w}m(z)]\boldsymbol{\beta} + \Sigma_{\mathbf{x}}^{-1}E[m(z)\mathbf{r}]$ and the result follows.

d) $E(\mathbf{w}z) = E[(\mathbf{x} - E(\mathbf{x}))\mathbf{x}^T\boldsymbol{\beta}] = E[(\mathbf{x} - E(\mathbf{x}))(\mathbf{x}^T - E(\mathbf{x}^T) + E(\mathbf{x}^T))\boldsymbol{\beta}] = E[(\mathbf{x} - E(\mathbf{x}))(\mathbf{x}^T - E(\mathbf{x}^T))]\boldsymbol{\beta} + E[\mathbf{x} - E(\mathbf{x})]E(\mathbf{x}^T)\boldsymbol{\beta} = \Sigma_{\mathbf{x}}\boldsymbol{\beta}$.

e) If $m(z) = z$, then $c(\mathbf{x}) = E(\boldsymbol{\beta}^T\mathbf{w}z) = \boldsymbol{\beta}^TE(\mathbf{w}z) = \boldsymbol{\beta}^T\Sigma_{\mathbf{x}}\boldsymbol{\beta} = 1$ by result d).

f) Since z is a (random) scalar, $E(z\mathbf{r}) = E(\mathbf{r}z) = E[(\mathbf{w} - (\Sigma\mathbf{x}\boldsymbol{\beta})\boldsymbol{\beta}^T\mathbf{w})z] = E(\mathbf{w}z) - (\Sigma\mathbf{x}\boldsymbol{\beta})\boldsymbol{\beta}^T E(\mathbf{w}z)$. Using result d), $E(\mathbf{r}z) = \Sigma\mathbf{x}\boldsymbol{\beta} - \Sigma\mathbf{x}\boldsymbol{\beta}\boldsymbol{\beta}^T\Sigma\mathbf{x}\boldsymbol{\beta} = \Sigma\mathbf{x}\boldsymbol{\beta} - \Sigma\mathbf{x}\boldsymbol{\beta} = \mathbf{0}$.

g) Since z and \mathbf{r} are linear combinations of \mathbf{x} , the joint distribution of z and \mathbf{r} is multivariate normal. Since $E(\mathbf{r}) = \mathbf{0}$, z and \mathbf{r} are uncorrelated and thus independent. Hence $m(z)$ and \mathbf{r} are independent and $\mathbf{u}(\mathbf{x}) = \Sigma_{\mathbf{x}}^{-1}E[m(z)\mathbf{r}] = \Sigma_{\mathbf{x}}^{-1}E[m(z)]E(\mathbf{r}) = \mathbf{0}$.

15.4 The submodel I that uses a constant and A, C, E, F, H looks best since it is the minimum $C_p(I)$ model and I has the smallest value of k such that $C_p(I) \leq 2k$.

15.6 a) No strong nonlinearities for MVN data but there should be some nonlinearities present for the non-EC data.

b) The plot should look like a cubic function.

c) The plot should use 0% trimming and resemble the plot in b), but may not be as smooth.

d) The plot should be linear and for many students some of the trimmed views should be better than the OLS view.

e) The EY plot should look like a cubic with trimming greater than 0%.

f) The plot should be linear.

15.7 b) and c) It is possible that none of the trimmed views look much like the $\text{sinc}(\text{ESP}) = \sin(\text{ESP})/\text{ESP}$ function.

d) Now at least one of the trimmed views should be good.

e) More lms trimmed views should be good than the views from the other 2 methods, but since simulated data is used, one of the plots from b) or c) could be as good or even better than the plot in d).

17.3 Tables

Tabled values are $F(k, d, 0.95)$ where $P(F < F(k, d, 0.95)) = 0.95$.

00 stands for ∞ . Entries produced with the `qf(.95, k, d)` command in *R*. The numerator degrees of freedom are k while the denominator degrees of freedom are d .

k	1	2	3	4	5	6	7	8	9	00
d										
1	161	200	216	225	230	234	237	239	241	254
2	18.5	19.0	19.2	19.3	19.3	19.3	19.4	19.4	19.4	19.5
3	10.1	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.53
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.63
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.37
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	3.67
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.23
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	2.93
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	2.71
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.54
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.41
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.30
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.21
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.13
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.07
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.01
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	1.96
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	1.92
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	1.88
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	1.84
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	1.71
30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	1.62
00	3.84	3.00	2.61	2.37	2.21	2.10	2.01	1.94	1.88	1.00

Tabled values are $t_{d,\delta}$ where $P(t < t_{d,\delta}) = \delta$ where t has a t distribution with d degrees of freedom. If $d > 30$ use the $N(0, 1)$ cutoffs given in the second to last line with $d = Z = \infty$.

delta	0.95	0.975	0.995
d			
1	6.314	12.706	63.657
2	2.920	4.303	9.925
3	2.353	3.182	5.841
4	2.132	2.776	4.604
5	2.015	2.571	4.032
6	1.943	2.447	3.707
7	1.895	2.365	3.499
8	1.860	2.306	3.355
9	1.833	2.262	3.250
10	1.812	2.228	3.169
11	1.796	2.201	3.106
12	1.782	2.179	3.055
13	1.771	2.160	3.012
14	1.761	2.145	2.977
15	1.753	2.131	2.947
16	1.746	2.120	2.921
17	1.740	2.110	2.898
18	1.734	2.101	2.878
19	1.729	2.093	2.861
20	1.725	2.086	2.845
21	1.721	2.080	2.831
22	1.717	2.074	2.819
23	1.714	2.069	2.807
24	1.711	2.064	2.797
25	1.708	2.060	2.787
26	1.706	2.056	2.779
27	1.703	2.052	2.771
28	1.701	2.048	2.763
29	1.699	2.045	2.756
30	1.697	2.042	2.750
Z	1.645	1.960	2.576
CI	90%	95%	99%