Preliminaries

> library(DAAG)

Exercise 1
The data set cities lists the populations (in thousands) of Canada’s largest cities over 1992 to 1996. There is a division between Ontario and the West (the so-called “have” regions) and other regions of the country (the “have-not” regions) that show less rapid growth. To identify the “have” cities we can specify

```r
data(cities) # DAAG package
cities$have <- factor((cities$REGION=="ON") | (cities$REGION=="WEST"))
```

Plot the 1996 population against the 1992 population, using different colors to distinguish the two categories of city, both using the raw data and taking logarithms of data values, thus:

```r
plot(POP1996 ~ POP1992, data=cities, col=as.integer(cities$have))
plot(log(POP1996) ~ log(POP1992), data=cities, col=as.integer(cities$have))
```

Which of these plots is preferable? Explain.

Now carry out the regressions

```r
cities.lm1 <- lm(POP1996 ~ have+POP1992, data=cities)
cities.lm2 <- lm(log(POP1996) ~ have+log(POP1992), data=cities)
```

and examine diagnostic plots. Which of these seems preferable? Interpret the results.

The required plots are given below.

![Figure 1: Red circles indicate the ‘have’ cities, and black circles indicate the ‘have-not’ cities. In the left panel, data are untransformed, while the right panel uses logarithmic scales.](image)

The second plot is preferable, since it spreads the plotted points out more evenly, while the first plot contains the large cluster of points in one corner. Population comparisons are
usually best made using ratios instead of differences; differences of logarithms correspond to logarithms of ratios, which is another reason for preferring the second plot.

We plot residuals against fitted values, first for the untransformed data and then for the transformed data.

```r
> par(mfrow = c(1, 2))
> cities.lm1 <- lm(POP1996 ~ have + POP1992, data = cities)
> cities.lm2 <- lm(log(POP1996) ~ have + log(POP1992), data = cities)
> plot(cities.lm1, which = 1)
> plot(cities.lm2, which = 1)
> par(mfrow = c(1, 1))
```

These plots indicate the need for transformation.

It is also a good idea to check plots of the residuals versus the predictors, as in

```r
plot(resid(cities.lm2) ~ log(cities$POP1992))
plot(resid(cities.lm2) ~ cities$have)
```

These plots (not shown) and plots of Cook’s distance and normal probability plots (also not shown) do not indicate any problems.

Here is the regression summary:

```r
> summary(cities.lm2)
```

**Call:**

```r
lm(formula = log(POP1996) ~ have + log(POP1992), data = cities)
```

**Residuals:**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.03478</td>
<td>-0.01698</td>
<td>-0.00332</td>
<td>0.01836</td>
<td>0.04821</td>
</tr>
</tbody>
</table>

**Coefficients:**

|                     | Estimate | Std. Error | t value | Pr(>|t|) |
|---------------------|----------|------------|---------|----------|
| (Intercept)         | -0.05565 | 0.03062    | -1.82   | 0.083    |
| haveTRUE            | 0.02254  | 0.01004    | 2.25    | 0.035    |
| log(POP1992)        | 1.01352  | 0.00523    | 193.92  | <2e-16   |

**Residual standard error:** 0.0239 on 22 degrees of freedom

**Multiple R-Squared:** 0.999, **Adjusted R-squared:** 0.999

**F-statistic:** 2.05e+04 on 2 and 22 DF, **p-value:** <2e-16

This suggests that the ‘have’ cities grew faster between 1992 and 1996 than the ‘have-not’ cities.
Exercise 2

In the data set cement (MASS package), examine the dependence of y (amount of heat produced) on x1, x2, x3 and x4 (which are proportions of four constituents). Begin by examining the scatterplot matrix. As the explanatory variables are proportions, do they require transformation, perhaps by taking log(x/(100 − x))? What alternative strategies might be useful for finding an equation for predicting heat?

First, obtain the scatterplot matrix for the untransformed cement data:

Since the explanatory variables are proportions, a transformation such as that suggested might be helpful, though the bigger issue is the fact that the sum of the explanatory variables is nearly constant. Thus, there will be severe multicollinearity as indicated by the variance inflation factors:

```r
> cement.lm <- lm(y ~ x1 + x2 + x3 + x4, data = cement)
> vif(cement.lm)

x1  x2  x3  x4
38.50 254.42 46.87 282.51
```

The scatterplot matrix indicated that x4 and x2 are highly correlated, so we may wish to include just one of these variables as in

```r
> cement.lm2 <- lm(y ~ x1 + x2 + x3, data = cement)
> vif(cement.lm2)

x1  x2  x3
3.251 1.064 3.142
```
The multicollinearity is less severe, and we can proceed. We consult the standard diagnostics using

\[
\text{par}(\text{mfrow} = c(1, 4))
\]
\[
\text{plot(cement.lm2)}
\]
\[
\text{par}(\text{mfrow} = c(1, 1))
\]

![Diagnostic plots for the model cement.lm2](image)

Figure 4: Diagnostic plots for the model cement.lm2

Nothing seems amiss on these plots. The three variable model seems satisfactory. Upon looking at the summary, one might argue in favour of removing the variable \(x_3\).

For the logit analysis, first define the logit function:

\[
\text{logit} \leftarrow \text{function}(x) \log(x/(100 - x))
\]

Now form the transformed data frame, and show the scatterplot matrix:

\[
\text{logitcement} \leftarrow \text{data.frame(logit(cement[, -5]), y = cement[, 5])}
\]
\[
\text{pairs(logitcement)}
\]

![Scatterplot matrix for the logits of the proportions.](image)

Figure 5: Scatterplot matrix for the logits of the proportions.
Chapter 6 Exercises

Notice that the relationship between \(x_2\) and \(x_4\) is now more nearly linear. This is helpful; it is advantageous for collinearities or multicollinearities to be explicit.

Now fit the full model, and plot the diagnostics:

\[
\text{logitcement.lm} <- \text{lm}(y \sim x_1 + x_2 + x_3 + x_4, \text{data = logitcement})
\]

\[
\text{par(mfrow = c(1, 4))}
\]

\[
\text{plot(logitcement.lm)}
\]

\[
\text{par(mfrow = c(1, 1))}
\]

Figure 6: Diagnostic plots for the model that works with logits.

This time, the multicollinearity problem is less extreme, though it is still notable. Some observations have now influential outliers. In this problem, we may be best off not transforming the predictors.

Exercise 3
The data frame \text{hills2000} in our \text{DAAG} package has data, based on information from the Scottish Running Resource web site, that updates the 1984 information in the data set \text{hills}. Fit a regression model, for men and women separately, based on the data in \text{hills2000}. Check whether it fits satisfactorily over the whole range of race times. Compare the equation that you obtain with that based on the \text{hills} data frame.

The \text{hills2000} data frame contains record times for more than just hill races, so the first thing we will do is subset the data frame so that we focus only on the 56 hill races contained therein.

\[
\text{data(hills2000)}
\]

\[
\text{hills2K} <- \text{hills2000[\text{hills2000}\text{type} == "hill", -seq(1, 6)]}
\]

We have also eliminated the first 6 columns which contain the record times for both sexes in terms of hours, minutes and seconds; these columns are extraneous, since we have the record times in seconds in the \text{time} and \text{timef} variables.

We begin with the same kind of transformed model that we tried in Section 6.3 for the \text{hills} data.

\[
\text{hills2K.loglm} <- \text{lm(log(time) \sim log(dist) + log(climb), data = hills2K)}
\]
The first of the diagnostic plots (residuals versus fitted values) reveals three outliers, identified as 12 Trig Trog, Chapelgill, and Caerketton. The two points which lie away from the main body of residuals and seem to be skewing them are for 12 Trig Trog and Caerketton; once these are removed, Chapelgill is no longer anomalous. A careful analyst will remove 12 Trig Trog first, then Caerketton.

> use <- !row.names(hills2K) %in% c("12 Trig Trog", "Caerketton")
> hills2Kr.loglm <- lm(log(time) ~ log(dist) + log(climb), data = hills2K[use,])
> plot(hills2Kr.loglm, panel = panel.smooth, which = 1)

There is clear evidence of curvature in the plot of residuals.

A possibility is to try the addition of the interaction term \( \log(\text{dist}) \times \log(\text{climb}) \). This does not remove the curvature in the plot of residuals versus fitted values.

Additional Note: A model that uses spline curves to transform the explanatory variables does work well:
> library(splines)
> hills2K.bs <- lm(log(time) ~ bs(dist, 4) + bs(climb, 4), data = hills2K[use, + ])
> par(mfrow = c(1, 3))
> plot(hills2K.bs, panel = panel.smooth, which = 1)
> termplot(hills2K.bs, partial.resid = TRUE)

![Figure 9: Residuals vs fitted values, and termplots, for hills2K.bs](image)

The plot of residuals versus fitted values shows no evidence either of trend or of heterogeneity of variance.

(A further model that may be tried has time on the left-hand side. The plot of residuals against fitted values then shows clear evidence of curvature.)

Examination of the termplots that show the contribution of the individual terms shows why the logarithmic transformation does not quite work, and why it will be hard to find a parametric model that gives a satisfactory account of the data, over the whole range of times. This model is hardly affected if 12 Trig Trog is re-included. It is still not possible to accommodate Caerketton.

Turning to the women’s race times, we start where we left off with the men’s race times:

> par(mfrow = c(1, 3))
> hillsfr.bs <- lm(log(timef) ~ bs(dist, 4) + bs(climb, 4), data = hills2K[use, + ])
> plot(hillsfr.bs, which = 1)
> termplot(hillsfr.bs)
> par(mfrow = c(1, 1))

![Figure 10: Residuals vs fitted values, and termplots, for hillsfr.bs](image)

Again this is a good fit to the data, with the termplots showing very similar patterns of response. Notice the upward curvature at high fitted values, relative to the curves for log(dist) and log(climb).

The range of times is much greater for the hills2000 than for the hills data; compare:

> data(hills, package = "DAAG")
> range(hills$time)
We noticed above that the model that is linear in \( \log(\text{dist}) \) and \( \log(\text{climb}) \) does not work well for large values of \text{time}. A satisfactory way to compare parametric models may be to restrict data in \text{hills2K} to the same range of times as for \text{hills}, i.e., \text{time} \leq 3.4 \text{ hours}.

\begin{verbatim}
> range(hills2000$time)
[1] 0.2086 15.4389
\end{verbatim}

Even this is not a sufficient restriction of the range of times in the \text{hills2K} data. The curvature in the plot of residuals is a sign of impending problems. It is nevertheless interesting to compare the coefficients:

\begin{verbatim}
> summary(hills.loglm)$coef
Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.8816 0.28263 -13.734 1.009e-14
log(dist) 0.9092 0.06500 13.989 6.163e-15
log(climb) 0.2601 0.04839 5.375 7.330e-06

> summary(hills3.loglm)$coef
Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.9670 0.16722 -23.72 3.252e-29
log(dist) 0.7383 0.03175 23.26 8.252e-29
log(climb) 0.3181 0.02696 11.80 3.392e-16
\end{verbatim}

The coefficients are broadly comparable.

Additional Note: The following is interesting. We use the spline model, derived from the \text{hills2K} data, to determine predicted values for the \text{data}, compare these with predicted values from the spline model that is fitted to the \text{hills} data.

\begin{verbatim}
> hills2K.bs <- lm(log(time) ~ bs(dist, 4) + bs(climb, 4), data = hills2K[use, ])
> hills.bs <- lm(log(time) ~ bs(dist, 4) + bs(climb, 4), data = hills[-18, ])
> fits <- predict(hills.bs)
> fits2 <- predict(hills2K.bs, newdata = hills[-18, ])
> plot(fits, fits2, xlab = "Fitted values, from hills.bs", ylab = "Fitted values, hills2K.bs model")
> mtext(side = 3, line = 1, "All fitted values are for the hills data")
> abline(0, 1)
\end{verbatim}

The warnings arise because some values of \text{climb} for the \text{hills} data lie outside of the range of this variable for the \text{hills2K} data.
Chapter 6 Exercises

Exercise 4

For each covariate, compare the (NSW) treatment group in nsw74psid1 with the (PSID) control group. Use overlaid density plots to compare the continuous variables, and two-way tables to compare the binary (0/1) variables. Where are the greatest differences?

To obtain the overlaid densities, we will find the following function, which has some similarities to overlap.density() in the DAAG package, helpful. Its principal argument x is a list consisting of the control and treatment vectors (named as such). (The function overlap.density() might alternatively be used to similar effect.)

```r
> overlay.density <- function(xx, xlab = "", main = "") {
+     names(xx) <- c("control", "treatment")
+     control.density <- density(xx$control)
+     treatment.density <- density(xx$treatment)
+     yrange <- range(control.density$y, treatment.density$y)
+     xrange <- range(control.density$x, treatment.density$x)
+     plot(control.density, xlab = xlab, ylab = "density (black=control, red=treatment)",
+          col = 1, main = main, xlim = xrange, ylim = yrange)
+     lines(treatment.density, col = 2)
+ }
```

This function can be applied to the age, re74, re75, re78, and educ variables as follows:

```r
> par(mfrow = c(2, 3))
> data(nsw74psid1)
> attach(nsw74psid1)
> overlay.density(split(age, trt), xlab = "age", main = "Age Density")
> overlay.density(split(re74, trt), xlab = "earnings", main = "Real 1974 Earnings")
> overlay.density(split(re75, trt), xlab = "earnings", main = "Real 1975 Earnings")
> overlay.density(split(re78, trt), xlab = "earnings", main = "Real 1978 Earnings")
> overlay.density(split(educ, trt), xlab = "earnings", main = "Education")
> par(mfrow = c(1, 1))
> detach(nsw74psid1)
```

![Overlaid density plots](image)

Figure 11: Overlaid density plots

The resulting plots are similar to those of Figure 6.12 (page 158) of the book. Note that the real earnings in 1974 and 1975 contrast sharply between the control and treatment groups.
Comparing the binary covariates, we see very different black/non-black proportions in the control and treatment groups. Similar discrepancies hold for married/non-married and nodegree/degree proportions.

```
> attach(nsw74psid1)
> table(trt, black)

          black
       trt 0  1
           0 1866 624
           1   29 156

> table(trt, hisp)

          hisp
       trt 0  1
           0 2409  81
           1  174  11

> table(trt, marr)

          marr
       trt 0  1
           0  333 2157
           1  150   35

> table(trt, nodeg)

          nodeg
       trt 0  1
           0 1730  760
           1  54  131

> detach(nsw74psid1)
```

**Exercise 5**
Repeat the analysis in Section 6.5, using the data set `nsw74psid1`, but now working with `log(re78)` and `log(re75)`. What difference does the use of the logarithms of the income variables have for the interpretation of the results? [Hint: For each observation, determine predicted values. Then \( \exp(\text{predicted values}) \) gives predicted incomes in 1978. Take \( \exp(\text{estimated treatment effect}) \) to get an estimate of the factor by which a predicted income for the control group must be multiplied to get a predicted income for the experimental group, if covariate values are the same.]

```
> nsw.lm <- lm(log(re78 + 1) ~ trt + (age + educ + log(re74 + 1) +
+ log(re75 + 1)) + (black + hisp + nodeg), data = nsw74psid1)
> summary(nsw.lm)

Call:
  lm(formula = log(re78 + 1) ~ trt + (age + educ + log(re74 + 1) +
      log(re75 + 1)) + (black + hisp + nodeg), data = nsw74psid1)

Residuals:
Chapter 6 Exercises

Min | 1Q | Median | 3Q | Max
---|---|---|---|---
-10.527 | 0.112 | 0.497 | 0.819 | 8.058

Coefficients:

|         | Estimate | Std. Error | t value | Pr(>|t|) |
|---------|----------|------------|---------|----------|
| (Intercept) | 2.55595 | 0.45126 | 5.66 | 1.6e-08 |
| trt | 1.80438 | 0.24827 | 7.27 | 4.8e-13 |
| age | -0.01352 | 0.00515 | -2.63 | 0.0087 |
| educ | 0.05832 | 0.02607 | 2.24 | 0.0254 |
| log(re74 + 1) | 0.23005 | 0.02485 | 9.26 | < 2e-16 |
| log(re75 + 1) | 0.44831 | 0.02359 | 19.01 | < 2e-16 |
| black | -0.02357 | 0.12735 | -0.19 | 0.8532 |
| hisp | 0.62958 | 0.28356 | 2.22 | 0.0265 |
| nodeg | -0.11205 | 0.16780 | -0.67 | 0.5043 |

Residual standard error: 2.61 on 2666 degrees of freedom
Multiple R-Squared: 0.384, Adjusted R-squared: 0.382
F-statistic: 207 on 8 and 2666 DF, p-value: <2e-16

Using logarithms of the income variables makes a big difference. Now, age, educ and re74 appear to be important variables, whereas they did not seem necessary in the original analysis.

Furthermore, now that the predicted income is expressed on the log scale, we must take exponentials in order to obtain predicted values on the usual scale. Also, the treatment effect is now multiplicative on the usual scale instead of additive as in the original model. Thus, in order to make a comparison now between control and treatment groups, we must multiply a predicted income for the control group by 6.07 (see below) to get a predicted income for the experimental group, assuming covariate values are the same.

> exp(1.804)

[1] 6.074

Alternatively, we can eliminate all 0 earnings:

```r
> attach(nsw74psid1, pos = 1)
> nsw74psid2 <- nsw74psid1[((re78 > 0) & (re75 > 0)) & (re74 > + 0), ]
> detach(nsw74psid1)
> nsw74psid2$log74 <- log(nsw74psid2$re74)
> nsw74psid2$log75 <- log(nsw74psid2$re75)
> nsw74psid2$log78 <- log(nsw74psid2$re78)
```

We then fit the log model as follows:

```r
> nsw.lm <- lm(log78 ~ trt + (age + educ + log74 + log75) + (black + + hisp + nodeg), data = nsw74psid2)
> summary(nsw.lm)
```

Call:

```
  lm(formula = log78 ~ trt + (age + educ + log74 + log75) + (black + + hisp + nodeg), data = nsw74psid2)
```

Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
-5.3961 -0.1477 0.0421 0.2336 2.4025

Coefficients:

|          | Estimate | Std. Error | t value |  Pr(>|t|) |
|----------|----------|------------|---------|----------|
| (Intercept) | 3.30605  | 0.19197    | 17.22   | < 2e-16  |
| trt       | -0.31008 | 0.08847    | -3.51   | 0.00047  |
| age       | -0.00181 | 0.00124    | -1.46   | 0.14392  |
| educ      | 0.02523  | 0.00585    | 4.31    | 1.7e-05  |
| log74     | 0.33309  | 0.02602    | 12.80   | < 2e-16  |
| log75     | 0.32240  | 0.02528    | 12.76   | < 2e-16  |
| black     | -0.09683 | 0.02768    | -3.50   | 0.00048  |
| hisp      | 0.06301  | 0.06342    | 0.99    | 0.32057  |
| nodeg     | -0.08399 | 0.03744    | -2.24   | 0.02497  |

Residual standard error: 0.51 on 2109 degrees of freedom
Multiple R-Squared: 0.508, Adjusted R-squared: 0.506
F-statistic: 272 on 8 and 2109 DF, p-value: <2e-16

The treatment effect factor changes from 6.04 to \( \exp(-0.31) \) or 0.73.

---

Exercise 6

The following is a rewording, intended to avoid ambiguity.

Section 6.1 used \( \text{lm()} \) to analyze the `allbacks` data that are presented in Figure 6.1. Repeat the analysis using (1) the function `rlm()` in the `MASS` package, and (2) the function `lqs()` in the `lqs` package. Compare the two sets of results with the results in Section 6.1.

Here are fits, w/wo intercept, using `rlm()`

```r
> data(allbacks)
> allbacks.rlm <- rlm(weight ~ volume + area, data = allbacks)
> summary(allbacks.rlm)
Call: rlm(formula = weight ~ volume + area, data = allbacks)
Residuals:
   Min     1Q Median     3Q    Max
-80.86  -22.18  -9.58  34.54 232.26
Coefficients:
             Value  Std. Error  t value
(Intercept) 9.239     40.316     0.229
volume  0.701     0.042    16.641
area   0.514     0.070     7.311

Residual standard error: 39.4 on 12 degrees of freedom
Correlation of Coefficients:

          (Intercept) volume
volume -0.883
area  -0.318

> allbacks.rlm0 <- rlm(weight ~ volume + area - 1, data = allbacks)
> summary(allbacks.rlm0)
```

---
Call: rlm(formula = weight ~ volume + area - 1, data = allbacks)

Residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residuals</td>
<td>-86.0</td>
<td>-20.6</td>
<td>-10.3</td>
<td>36.1</td>
<td>231.8</td>
</tr>
</tbody>
</table>

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume</td>
<td>0.711</td>
<td>0.018</td>
<td>38.511</td>
</tr>
<tr>
<td>area</td>
<td>0.517</td>
<td>0.062</td>
<td>8.288</td>
</tr>
</tbody>
</table>

Residual standard error: 39.7 on 13 degrees of freedom

Correlation of Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>volume</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume</td>
<td>-0.635</td>
<td></td>
</tr>
<tr>
<td>area</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Here are plots of residuals against fitted values, for the two models.

> par(mfrow = c(1, 2))
> plot(allbacks.rlm, which = 1)
> mtext(side = 3, line = 1, "rlm(), intercept included")
> plot(allbacks.rlm0, which = 1)
> mtext(side = 3, line = 1, "rlm(), no intercept")
> par(mfrow = c(1, 2))

Figure 12: Residuals vs fitted values, for the rlm() models with & without intercept.

Comparison of the coefficients of the intercept and no-intercept with the lm() counterparts reveals larger differences in coefficient estimates for the intercept models. The robust method has given smaller coefficient standard errors than lm().

The influence of the outlying observation (the 13th) is reduced using the robust method; therefore, on the residual plots we see this observation featured even more prominently as an outlier than on the corresponding plots for the lm() fits.

We next consider the lqs() approach. By default, lqs() employs a resistant regression method called least trimmed squares regression (lts), an idea due to Rousseeuw (1984) (“Least median of squares regression.” Journal of the American Statistical Association 79: 871–888). The method minimizes the sum of the $k$ smallest squared residuals, where $k$ is usually taken to be slightly larger than 50% of the sample size. This approach removes all of the influence of outliers on the fitted regression line.

> library(lqs)
> allbacks.lqs <- lqs(weight ~ volume + area, data = allbacks)
> allbacks.lqs$coefficients
(Intercept)     volume     area  
-59.6232       0.7737     0.4709  

> allbacks.lqs0 <- lqs(weight ~ volume + area - 1, data = allbacks)
> coefficients(allbacks.lqs0)

volume     area  
0.7117     0.4849  

The robust coefficient estimates of volume and area are similar to the corresponding
coefficient estimates for the \texttt{lm()} fit.

Here are plots of residuals against fitted values, for the two models.

> par(mfrow = c(1, 2))
> plot(allbacks.lqs$residuals ~ allbacks.lqs$fitted.values)
> mtext(side = 3, line = 1, "lqs(), intercept included")
> plot(allbacks.lqs0$residuals ~ allbacks.lqs0$fitted.values)
> mtext(side = 3, line = 1, "lqs(), no intercept")
> par(mfrow = c(1, 1))

![Figure 13: Residuals vs fitted values, for the \texttt{lqs()} models with \& without intercept.]

Because the outlying observation (13) is now not used at all in the final regression
coefficient estimates, it has no influence. Neither does observation 11, another outlier.
Both points plot farther away from the reference line at 0 than in the corresponding \texttt{lm()} residual plots.

\textit{Exercise 8}

Check the variance inflation factors for \texttt{bodywt} and \texttt{lsize} for the model
\texttt{brainwt \sim bodywt + lsize}, fitted to the \texttt{litters} data set. Comment.

We can use the function \texttt{vif()} to determine the variance inflation factors for the litters
data as follows:

> data(litters)
> litters.lm <- lm(brainwt ~ bodywt + lsize, data = litters)
> vif(litters.lm)

\texttt{bodywt  \hspace{2em} lsize  
11.33  \hspace{2em} 11.33  

These factors are fairly large, indicating fairly strong multicollinearity. A scatterplot
of litter size versus body weight would confirm that the two variables have a relation which
is close to linear. The effect is to give inflated standard errors in the above regression,
though not enough to obscure the relationship between brain weight and body weight and litter size completely.

It is hazardous to make predictions of brain weight for pigs having body weight and litter size which do not lie close to the line relating these variables.