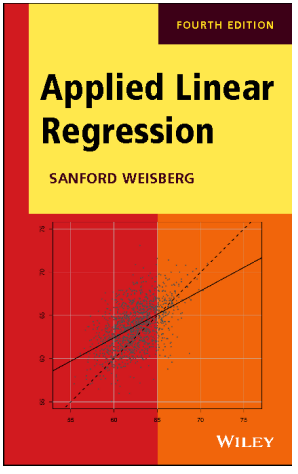


Solutions to Odd-Numbered Problems

Version of October 19, 2015

Sanford Weisberg

School of Statistics
University of Minnesota
Minneapolis, Minnesota 55455

These solutions are best viewed using a pdf viewer such as Adobe Reader with bookmarks showing at the left, and in single page view, selected by `View` → `Page Display` → `Single Page View`. Computer input is indicated by `this font`, while output uses `this font`. The usual command prompt ">" and continuation "+" characters are suppressed so you can cut and paste directly from this document into an R window. Beware, however that a current command may depend on earlier commands in the problem you are reading!

To cite this document, use:
Weisberg, S. (2014). Solutions to odd-numbered problems. Online, http://z.umn.edu/alr4solutions.

Contents

1 Scatterplots **1**

2 Simple Linear Regression **7**

3 Multiple Regression **30**

4 Interpretation of Main Effects **40**

5 Complex Regressors **48**

6 Testing and Analysis of Variance **64**

7 Variances **78**

8 Transformations **84**

9 Regression Diagnostics **108**

10 Variable Selection **128**

11 Nonlinear Regression **133**

12 Binomial and Poisson Regression **142**

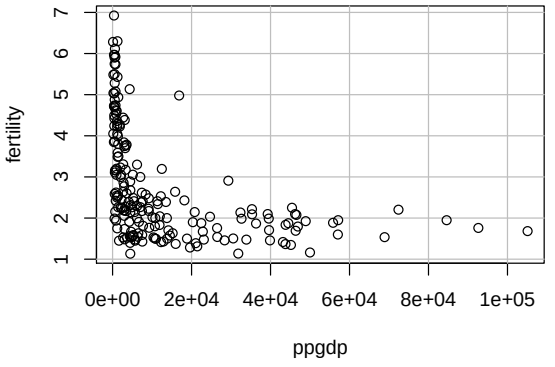
CHAPTER 1

Scatterplots

1.1 1.1.1 Solution:

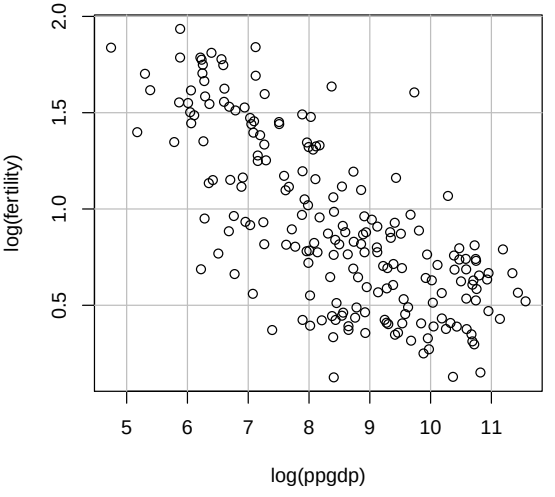
The predictor is a function of `ppgdp`, and the response is a function of `fertility`. □

1.1.2 Solution:



Simple linear regression is not a good summary of this graph. The mean function does not appear to be linear, variance does not appear to be constant. □

1.1.3 Solution:

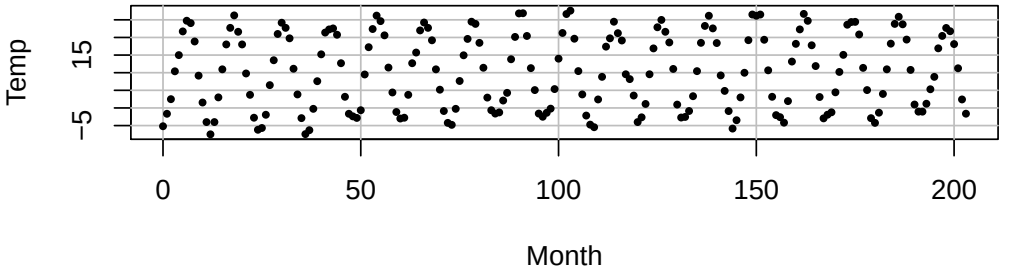


Simple linear regression is much more appropriate in log-scale, as the mean function appears to be linear, and constant variance across the plot is at least plausible, if not completely certain. As one might expect, there may be a few outliers that are localities with either unusually high or low `fertility` for their value of `ppgdp`. □

1.3 1.3.1 Solution:

This appears to be a null plot, with no particularly interesting characteristics. □

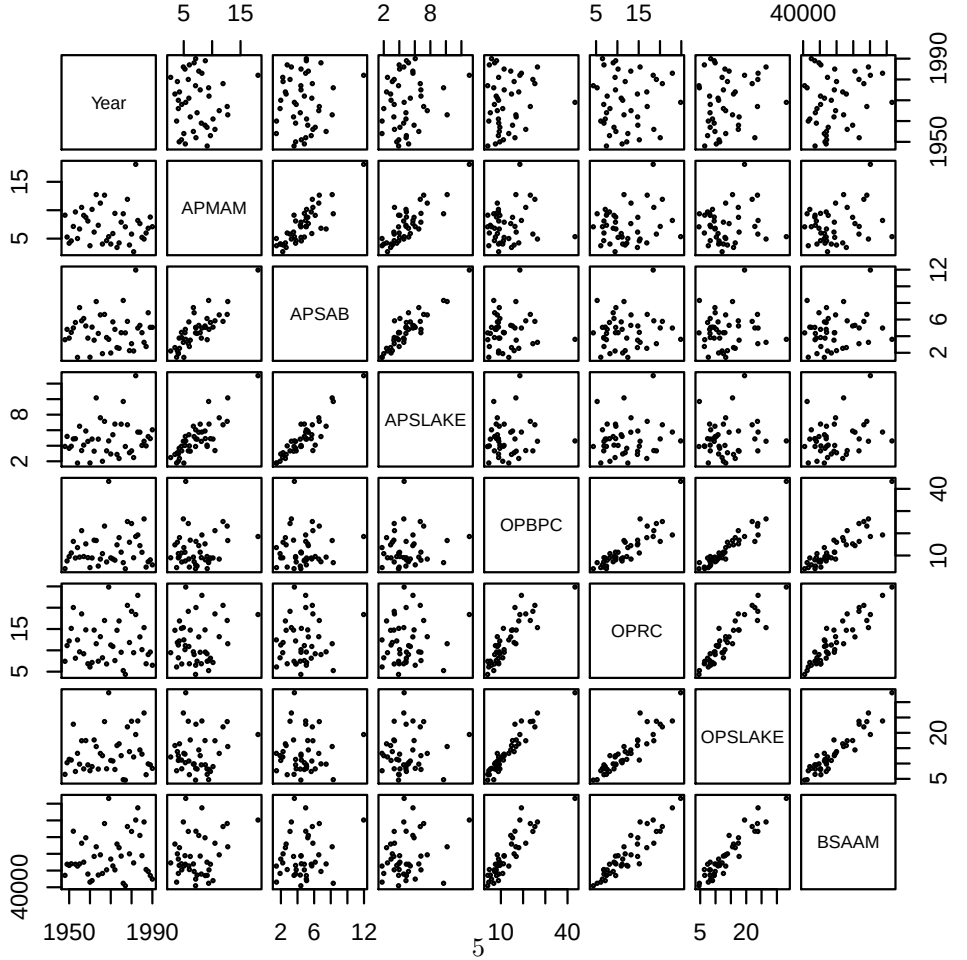
1.3.2 Solution:



Scaling matters! The points could have also been joined with lines to emphasize the temporal pattern in the data: temperature is high in the summer and low in the winter. □

1.5 Solution:

(1) `Year` appears to be largely unrelated to each of the other variables; (2) the three variables starting with "O" seem to be correlated with each other, meaning that all the plot including two of these variables exhibit a dependence between the variables that is stronger than the dependence between the "O" variables and other variables. The three variables starting with "A" also seem to be another correlated group; (3) `BSAAM` is more closely related to the "O" variables than the "A" variables; (4) there is at least 1 separated point with very high run-off. When we continue with this example in later chapters, we will end up taking logs of everything and combining the predictors into to terms.


Year
APMAM
APSAB
APSLAKE
OPBPC
OPRC
OPSLAKE
BSAAM
5 15
2 8
5 15
40000
1950 1990
2 6 12
10 40
5 20

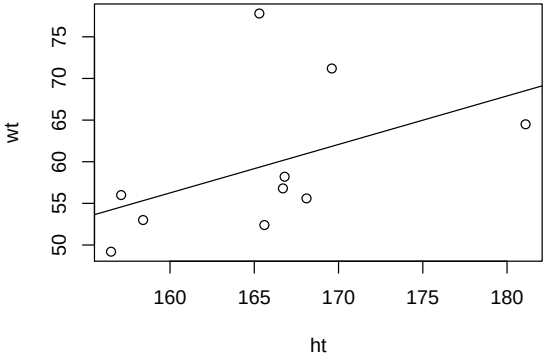
□

CHAPTER 2

Simple Linear Regression

2.1 2.1.1 Solution:

```
plot(wt ~ ht, Htwt)
abline(lm(wt ~ ht, Htwt))
```



With only 10 points, judging the adequacy of the model is hard, but it may be plausible here, as the value of the response is generally increasing from right to left, and and straight line on the plot is visually a plausible summary of this trend. □

2.1.2 Solution:

These computations are straightforward on a calculator, or using a computer language like R. Using a standard computer package, it is easiest to get means and the sample covariance matrix, and then use Table 2.1 to get the summary statistics. In R, the following will do the trick:

```
n <- dim(Htwt)[1]
(ave <- colMeans(Htwt))
```

```
    ht      wt
165.52  59.47
```

```
xbar <- ave[1]
ybar <- ave[2]
print(crossprod <- (dim(Htwt)[1] - 1) * cov(Htwt), digits=5)
```

```
       ht     wt
ht 472.08 274.79
wt 274.79 731.96
```

```
SXX <- crossprod[1, 1]
SYY <- crossprod[2, 2]
SXY <- crossprod[1, 2]
```

The matrix `crossprod` has `SXX` and `SYY` on the diagonal and `SXY` as either off-diagonal entry. The `print` command was used to display this matrix with 5 digits. □

2.1.3 Solution:

Use the computations from the last subproblem. We do the coefficient estimates first:

```
(coefs <- c(Intercept=ybar - (SXY/SXX) * xbar, Slope=SXY/SXX))
```

```
Intercept.wt        Slope
    -36.8759       0.5821
```

Next, the estimate of variance:

```
(s2 <- (SYY - SXY^2/SXX)/(n - 2))
```

```
[1] 71.5
```

Finally, standard errors of the coefficients and t-values:

```
(secoefs <- c(Intercept=sqrt(s2 * (1/n + xbar^2/SXX)),
              Slope=sqrt(s2 * (1/SXX))))
```

```
Intercept.ht        Slope
     64.4728       0.3892
```

```
(cov12 <- - s2 * xbar/SXX)
```

```
    ht
-25.07
```

```
(tvals <- coefs/secoefs)
```

```
Intercept.wt        Slope
      -0.572        1.496
```

□

2.3 2.3.1 Solution:

The log-scale graph appears nearly linear, the distribution of the points on the axes is no longer skewed, and variability appears constant. Also the points at the extreme right are no longer separated from the other points. Vilnius and Budapest still appear to be outliers. □

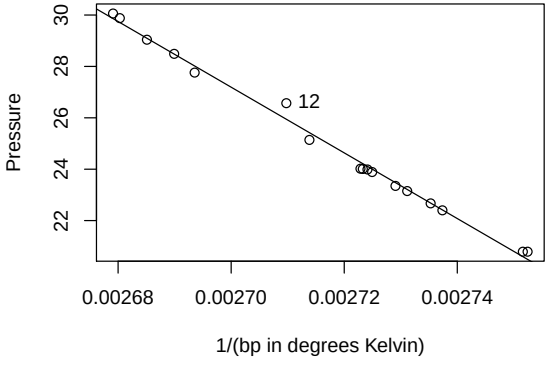
2.3.2 Solution:

The proposed model allows for exponential growth if $\beta_1 > 0$, linear growth is $\beta_1 = 1$ and slower than linear growth if $\beta_1 < 1$ For a fixed value of β_1 the value of γ_0 is essentially a rescaling factor from the scale of x to the scale of y. If $\gamma_0 > 1$, so $\beta_0 > 0$ the fitted curve is shifted up, and if $\gamma_0 < 1$ the fitted curve is shifted down. □

2.5

2.7 2.7.1 Solution:

```
Forbes$u1 <- 1/(255.37 + (5/9)*Forbes$bp)
plot(pres ~ u1, Forbes, ylab="Pressure",
  xlab="1/(bp in degrees Kelvin)")
with(Forbes, text(u1[12], pres[12], "12", pos=4))
abline(lm(pres ~ u1, Forbes))
```



The slope is negative because the inverse transformation was used: large values of `bp` correspond to small values of u_1. □

2.7.2 Solution:

Since the predictor variable has very small values, we should anticipate the results printed by a computer program will use scientific notation. For example, the number `-1.92e+05` corresponds to -1.92×10^5, or -192000, accurate to 3 digits. The number `2e-16` represents a very small number, with 15 zeros to the right of the decimal, followed by a 2.

```
mod.Forbes2 <- lm(pres ~ u1, Forbes)
summary(mod.Forbes2)
```

```
Call:
lm(formula = pres ~ u1, data = Forbes)

Residuals:
    Min      1Q  Median      3Q     Max
```

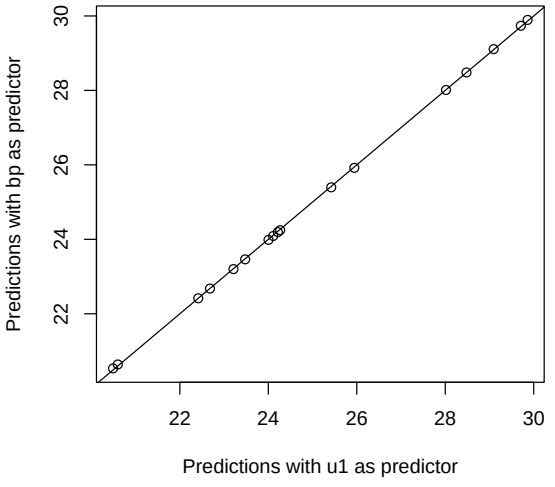
```
-0.2822 -0.1264 -0.0557  0.1711  0.6257

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.72e+02   7.01e+00    53.1   <2e-16
u1          -1.28e+05   2.58e+03   -49.5   <2e-16

Residual standard error: 0.243 on 15 degrees of freedom
Multiple R-squared:  0.994,        Adjusted R-squared:  0.994
F-statistic: 2.45e+03 on 1 and 15 DF,  p-value: <2e-16
```

Apart from case 12, this mean function seems to match the data very well. □

2.7.3 Solution:

The model `mod.Forbes2` fit in the last subproblem is for the Clausius–Clapeyron model, and the model `mod.Forbes1` fit below is for Forbes' model.

```
mod.Forbes1 <- lm(pres ~ bp, Forbes)
plot(predict(mod.Forbes2),predict(mod.Forbes1),
       xlab="Predictions with u1 as predictor",
       ylab="Predictions with bp as predictor")
abline(0, 1)
```



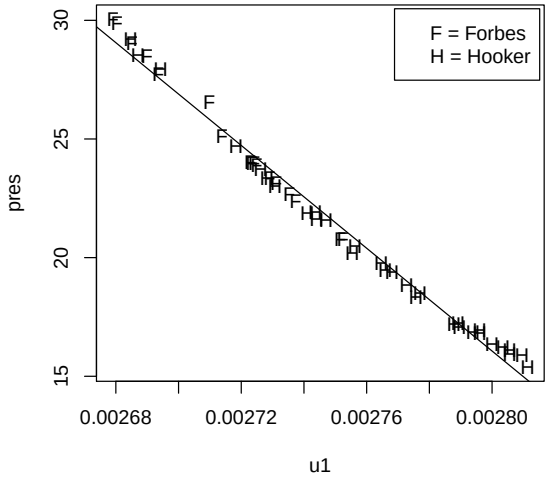
The line shown on the figure is the line $y = x$, and if the 2 models gave the same fitted values, all the points would fall exactly on this line. The deviations from the line are very small, and so the 2 models provide essentially identical fitted values are are therefore essentially indistinguishable based on these data. □

2.7.4 Solution:

We begin by reading the data into R, combining the 2 data sets and drawing a graph:

```
Hooker$u1 <- 1/(255.37 + (5/9) * Hooker$bp)
# create a combined data set for plotting
combined.data <-data.frame(
       u1=c(Forbes$u1, Hooker$u1),
```

```
       pres=c(Forbes$pres, Hooker$pres),
       set=c(rep(c("F","H"), c(17,31))))
plot(pres ~ u1, combined.data, pch=as.character(set))
legend("topright", c("F = Forbes", "H = Hooker"), inset=.01)
abline(lm(pres ~ u1, combined.data))
```



The variable `set` consists of "H" for Hooker and "F" for Forbes. R automatically converted this text variable to a factor, and so to use it to get plotting characters (the `pch=as.character(set)`), we need to convert `set` to a character vector. A legend has been added, and the least squares line. From the graph, we see the 2 sets of data agree very closely, except perhaps at the very largest values of u_1, corresponding to the highest altitudes. Most of Hooker's data was collected at higher

altitudes.

The fitted regression for (2.23) using Hooker's data alone is

```
mod.Hooker2 <- lm(pres ~ u1, data = Hooker)
summary(mod.Hooker2)
```

```
Call:
lm(formula = pres ~ u1, data = Hooker)

Residuals:
   Min     1Q Median     3Q    Max
-0.667 -0.275 -0.148  0.316  0.943

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.09e+02   5.56e+00    55.6   <2e-16
u1          -1.05e+05   2.01e+03   -52.0   <2e-16

Residual standard error: 0.405 on 29 degrees of freedom
Multiple R-squared:  0.989,        Adjusted R-squared:  0.989
F-statistic: 2.7e+03 on 1 and 29 DF,  p-value: <2e-16
```

□

2.9 2.9.1 Solution:

We will do this in two steps. First, turn the second of these equations into the first:

$$\begin{aligned} \mathrm{E}(Y|Z=z) &= \gamma_0 + \gamma_1 z \\ &= \gamma_0 + \gamma_1(ax+b) \\ &= \gamma_0 + \gamma_1 ax + \gamma_1 b \\ &= [\gamma_0 + \gamma_1 b] + [\gamma_1 a]\, x \\ &= \beta_0 + \beta_1 x \end{aligned}$$

and so $\beta_0 = \gamma_0 + \gamma_1 b$ and $\beta_1 = \gamma_1 a$. Now solve for the γs as functions of the βs:

$$\begin{aligned} \gamma_1 &= \beta_1/a \\ \gamma_0 &= \beta_0 - \beta_1 b/a \end{aligned}$$

Multiplying the predictor by a divides the slope by a. Adding b to the predictor doesn't change the slope, but it does change the intercept.

Since the response Y has not changed, the estimate of σ^2 and the value of R^2 will be unchanged. The test of the slope equal to 0 will be unchanged, but the test that the intercept is different because the parameter tested depends on the value of b. □

2.9.2 Solution:

Write

$$\begin{aligned} \mathrm{E}(Y|X) &= \beta_0 + \beta_1 X \\ d\mathrm{E}(Y|X) &= d\beta_0 + d\beta_1 X \\ \mathrm{E}(dY|X) &= d\beta_0 + d\beta_1 X \\ \mathrm{E}(V|X) &= d\beta_0 + d\beta_1 X \end{aligned}$$

and so the slope and intercept and their estimates are all multiplied by d. The variance is also multiplied by d. Scale-free quantities like R^2 and test statistics are unchanged. □

2.11 2.11.1 Solution:

$$\begin{aligned} \sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}\left[(x_i - \bar{x}) - (x_j - \bar{x})\right]^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\left[(x_i - \bar{x})^2 - 2(x_i - \bar{x})(x_j - \bar{x}) + (x_j - \bar{x})^2\right] \\ &= \sum_{j=1}^{n}\left[\sum_{i=1}^{n}(x_i - \bar{x})^2\right] - 2\sum_{i=1}^{n}(x_i - \bar{x})\left[\sum_{j=1}^{n}(x_j - \bar{x})\right] + \sum_{i=1}^{n}\left[\sum_{j=1}^{n}(x_j - \bar{x})^2\right] \end{aligned}$$

The first and third term on the right are each equal to $n\texttt{SXX}$. The term in square brackets in the second term is $\sum(x_j - \bar{x}) = 0$, so the second term is 0. Hence

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2 = 2n\texttt{SXX}$$

The proof is similar for `SXY`. □

2.11.2 Solution:

Using the first part of the problem and the value for w_{ij} given in the problem,

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}b_{ij} &= \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{(x_i - x_j)^2}{2n\texttt{SXX}}\frac{y_i - y_j}{x_i - x_j} \\ &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)(y_i - y_j)}{2n\texttt{SXX}} \\ &= \frac{2n\texttt{SYY}}{2n\texttt{SXX}} = \frac{\texttt{SXY}}{\texttt{SXX}} = \widehat{\beta}_1\end{aligned}$$

□

2.13 2.13.1 Solution:

```
colMeans(Heights)
```

```
mheight dheight
  62.45   63.75
```

```
var(Heights)
```

```
        mheight dheight
mheight   5.547   3.005
dheight   3.005   6.760
```

```
m1 <- lm(dheight ~ mheight, data=Heights)
summary(m1)
```

```
Call:
lm(formula = dheight ~ mheight, data = Heights)

Residuals:
   Min     1Q Median     3Q    Max
-7.397 -1.529  0.036  1.492  9.053

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   29.917      1.623    18.4   <2e-16
mheight        0.542      0.026    20.9   <2e-16

Residual standard error: 2.27 on 1373 degrees of freedom
Multiple R-squared:  0.241,        Adjusted R-squared:  0.24
F-statistic:  435 on 1 and 1373 DF,  p-value: <2e-16
```

The t-statistic for the slope has a p-value very close to 0, suggesting strongly that $\beta_1 \neq 0$. The value of $R^2 = 0.241$, so only about one-fourth of the variability in daughter's height is explained by mother's height. □

2.13.2 Solution:

Although the confidence intervals can be computed from the formulas in the text, most programs will produce them automatically. In R the function `confint` does this:

```
confint(m1, level=0.99)
```

```
              0.5 %  99.5 %
(Intercept) 25.7324 34.1025
mheight      0.4748  0.6087
```

□

2.13.3 Solution:

```
predict(m1, data.frame(mheight=64), interval="prediction",
   level=.99)
```

```
    fit   lwr   upr
1 64.59 58.74 70.44
```

□

2.15 2.15.1 Solution:

```
m1 <- lm(Length ~ Age, wblake)
m1.predict <- predict(m1, data.frame(Age=c(2, 4, 6)), interval="prediction")
m1.predict
```

```
    fit    lwr   upr
1 126.2  69.73 182.6
2 186.8 130.46 243.2
3 247.5 191.05 303.9
```

□

2.15.2 Solution:

The default level for prediction intervals is 95% so we don't need to specify the level we want

```
predict(m1,data.frame(Age=c(9)),interval="prediction")
```

```
    fit   lwr   upr
1 338.4 281.7 395.2
```

This is an extrapolation outside the range of the data, as there were no fish older than 8 years in the sample. We do not know if the straight-line mean function applies to at age 9. □

2.17 2.17.1 Solution:

Differentiate the residual sum of squares function

$$\texttt{RSS}(\beta_1) = \sum (y_i - \beta_1 x_i)^2$$

and set the result to 0:

$$\frac{d\texttt{RSS}(\beta_1)}{d\beta_1} = -2\sum x_i(y_i - x_i\beta_1) = 0$$

or

$$\sum x_i y_i = \beta_1 \sum x_i^2$$

Solving for β_1 gives the desired result. To show unbiasedness,

$$\begin{aligned}
\mathrm{E}(\widehat{\beta}_1|X) &= \mathrm{E}(\sum x_i y_i / \sum x_i^2) \\
&= \sum x_i \mathrm{E}(y_i|X) / \sum x_i^2 \\
&= \sum x_i(x_i\beta_1) / \sum x_i^2 \\
&= \beta_1 \sum x_i^2 / \sum x_i^2 \\
&= \beta_1
\end{aligned}$$

as required. For the variance,

$$\begin{aligned}
\mathrm{Var}(\widehat{\beta}_1) &= \mathrm{Var}(\sum x_i y_i / \sum x_i^2) \\
&= \sum x_i^2 \mathrm{Var}(y_i|X) / (\sum x_i^2)^2 \\
&= \sigma^2 \sum x_i^2 / (\sum x_i^2)^2 \\
&= \sigma^2 / \sum x_i^2
\end{aligned}$$

To estimate variance, we need an expression for the residual sum of squares, which we will call $\texttt{RSS}_0$:

$$\begin{aligned}
\texttt{RSS}_0 &= \sum (y_i - \widehat{\beta}_1 x_i)^2 \\
&= \sum y_i^2 - 2\widehat{\beta}_1 \sum x_i y_i + \widehat{\beta}_1^2 \sum x_i^2 \\
&= \sum y_i^2 - 2(\sum x_i y_i)^2 / \sum x_i^2 + (\sum x_i y_i)^2 / \sum x_i^2 \\
&= \sum y_i^2 - (\sum x_i y_i)^2 / \sum x_i^2
\end{aligned}$$

which is the same as the simple regression formula for RSS except that uncorrected sums of squares and cross-products replace corrected ones. Since the mean function has only 1 parameter, the estimate of σ^2 will have $(n-1)$ *df*, and $\widehat{\sigma}^2 = \texttt{RSS}_0/(n-1)$. □

2.17.2 Solution:

Models are fit in R without the intercept by adding a `-1` to the formula.

```
summary(m0 <- lm(Y ~ X - 1, data=snake))

Call:
lm(formula = Y ~ X - 1, data = snake)

Residuals:
   Min     1Q Median     3Q    Max
-2.421 -1.492 -0.194  1.651  3.077

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
X   0.5204     0.0132    39.5   <2e-16

Residual standard error: 1.7 on 16 degrees of freedom
Multiple R-squared:  0.99,        Adjusted R-squared:  0.989
F-statistic: 1.56e+03 on 1 and 16 DF,  p-value: <2e-16
```

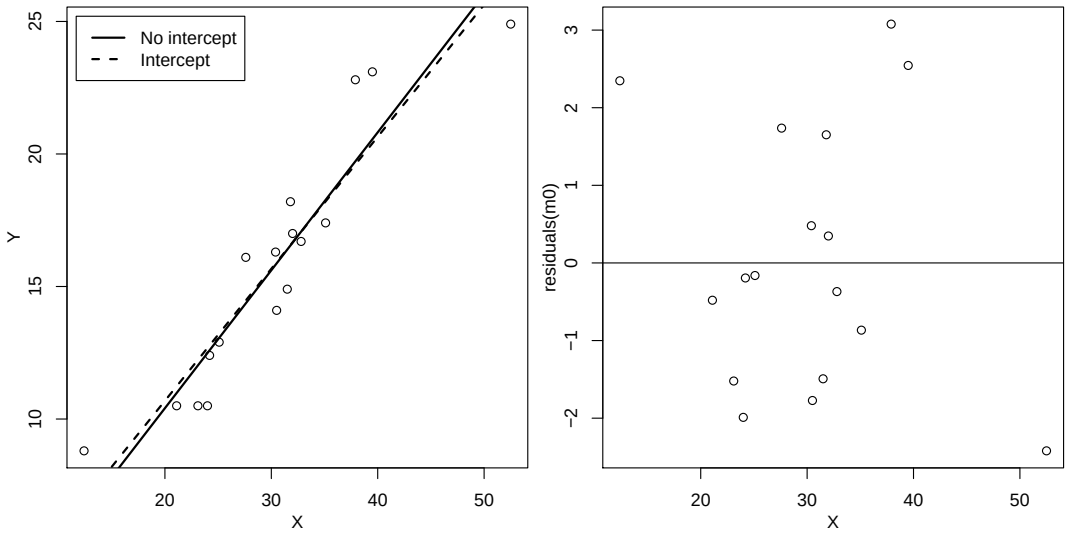
```
tval <- (coef(m0)[1] - 0.49)/ sqrt(vcov(m0)[1,1])
df <- dim(snake)[1] - 1
data.frame(tval = tval, df=df, pval = 1 - pt(abs(tval), df))
```

```
  tval df    pval
X 2.306 16 0.01742
```

Most programs won't automatically provide a test that the slope has any value other than 0, so we need to do the "hand" calculation. The `pt` function computes the area to the left of its argument, which would correspond to the lower tail. We subtract from 1 to get the upper tail. □

2.17.3 Solution:

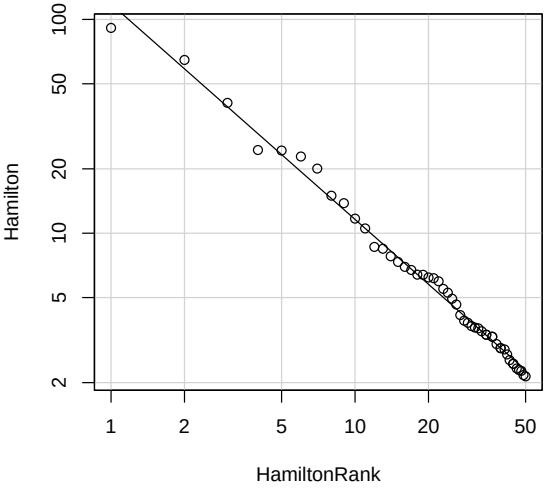
```
par(mfrow=c(1,2),mai=c(.6,.6,.1,.1),mgp=c(2,1,0))
plot(Y ~ X, snake)
m1 <- lm(Y ~ X, snake)
abline(m0, lwd=2)
abline(m1, lty=2, lwd=2)
legend("topleft", c("No intercept", "Intercept"), lty=1:2, inset=0.02, lwd=2)
plot(residuals(m0) ~ X, snake)
abline(h=0)
```



The plot at the left shows both the fit of the through-the-origin model (solid line) and the simple regression model (dashed line), suggesting little difference between them. The residual plot emphasizes the 2 points with the largest and smallest value of X as somewhat separated from the other points, and fit somewhat less well. However, the through-the-origin model seems to be OK here. □

2.19 2.19.1 Solution:

```
scatterplot(Hamilton ~ HamiltonRank, data=MWwords, log="xy",
  subset=HamiltonRank <= 50, smooth=FALSE, boxplots=FALSE)
```



The scatterplot indicates that Zipf's law is remarkably accurate, as the points lie so close to the OLS line. The fitted regression is

```
m1 <- lm(log(Hamilton) ~ log(HamiltonRank), data=MWwords,
         subset= HamiltonRank <= 50)
summary(m1)
```

```
Call:
lm(formula = log(Hamilton) ~ log(HamiltonRank), data = MWwords,
    subset = HamiltonRank <= 50)

Residuals:
```

```
     Min       1Q   Median       3Q      Max
-0.25741 -0.05029 -0.00156  0.04345  0.18827

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.7712     0.0395     121   <2e-16
log(HamiltonRank)  -1.0076     0.0127     -79   <2e-16

Residual standard error: 0.0793 on 48 degrees of freedom
Multiple R-squared:  0.992,        Adjusted R-squared:  0.992
F-statistic: 6.25e+03 on 1 and 48 DF,  p-value: <2e-16
```

□

2.19.2 Solution:

The test of $\gamma = 1$ is equivalent to $\beta_1 = -1$ in simple regression. The test is $t = (-1.00864 - (-1.0))/.01275 = -0.677647$, which can be compared to the $t(48)$ distribution:

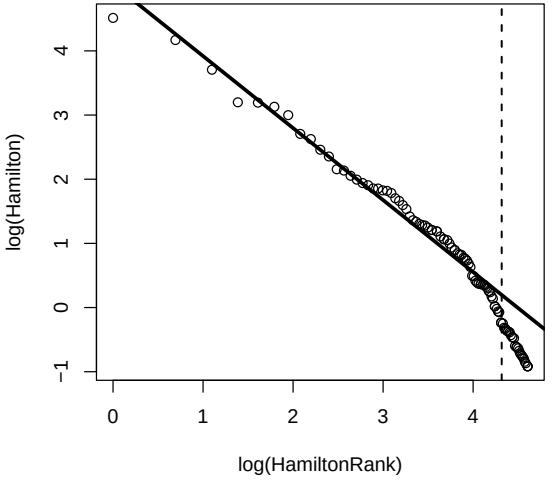
```
2*pt(-.0677647, 48)
```

```
[1] 0.9463
```

and the two-sided p-value is close to 0.95. There is no evidence against $b = 1$. □

2.19.3 Solution:

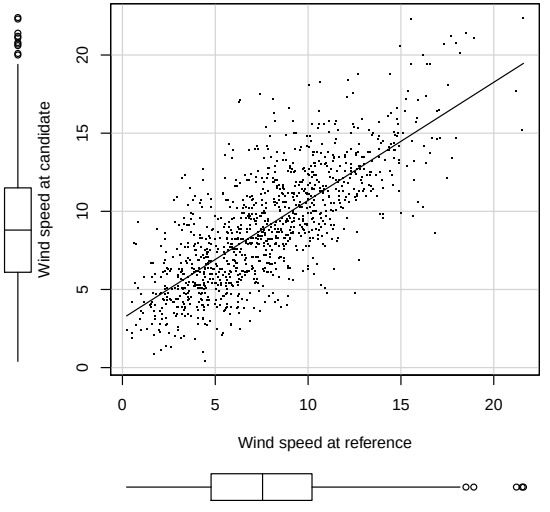
```
plot(log(Hamilton) ~ log(HamiltonRank), MWwords,
  subset=HamiltonRank <= 100)
abline(update(m1, subset=HamiltonRank <=75), lwd=3)
abline(v=log(75), lwd=1.6, lty=2)
```



The thick line is the OLS fit to the first 75 words. The vertical dashed line on the plot has the first 75 words to the left and the last 25 to the right. Zipf's law seems to work for 75 words but does seem less adequate for 100 words. The frequencies of these less frequent words are lower than predicted by Zipf's Law. □

2.21 2.21.1 Solution:

```
scatterplot(CSpd ~ RSpd, wm1, xlab="Wind speed at reference",
       ylab="Wind speed at candidate", pch=".", smooth=FALSE)
```



A straight-line mean function with constant variance seems reasonable here, although there is clearly plenty of remaining variation. As with the heights data, the ranges of the data on the 2 axes are similar. Further analysis might look at the marginal distributions to see if they are similar as well. □

2.21.2 Solution:

```
summary(m1 <- lm(CSpd ~ RSpd, wm1))
Call:
lm(formula = CSpd ~ RSpd, data = wm1)

Residuals:
```

```
   Min     1Q Median     3Q    Max
-7.788 -1.586 -0.199  1.440  9.174

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.1412     0.1696    18.5   <2e-16
RSpd          0.7557     0.0196    38.5   <2e-16

Residual standard error: 2.47 on 1114 degrees of freedom
Multiple R-squared:  0.571,        Adjusted R-squared:  0.571
F-statistic: 1.48e+03 on 1 and 1114 DF,  p-value: <2e-16
```

The value of $R^2 = .57$ indicates that only about half the variation in `CSpd` is explained by `RSpd`. The large value of $\widehat{\sigma}$ also suggests that predictions are likely to be of only modest quality. □

2.21.3 Solution:

The prediction is

$$\widehat{\texttt{CSpd}} = 3.1412 + .7557 \times 6.4285 = 8.7552$$

with standard error given by the square root of $\widehat{\sigma}^2 + \widehat{\sigma}^2(1/1116 + (7.4285 - \overline{\texttt{CSpd}})^2/\texttt{SXX}) = (2.467)^2$. Since the *df* are so large, we can use the normal distribution to get the prediction interval to be from 3.914 to 13.596 meters per second. □

2.21.4 Solution:

For the first result,

$$\frac{1}{m}\sum_{i=1}^{m}\tilde{y}_{*i} = \frac{1}{m}\sum_{i=1}^{m}(\widehat{\beta}_0 + \widehat{\beta}_1 x_{*i}) = \widehat{\beta}_0 + \widehat{\beta}_1\frac{1}{m}\sum_{i=1}^{m}x_{*i} = \widehat{\beta}_0 + \widehat{\beta}_1\bar{x}_*$$

so the average of the predictions is the same as the prediction at the average.

For the second result, we use the results of Appendix **??**. The variance of the average prediction will consist of 2 parts, the estimated error for estimating the coefficients, $\widehat{\sigma}^2(1/n + (\bar{x}_* - \bar{x})^2/\texttt{SXX})$,

and the *average of variance the m independent errors attached to the m future predictions*, with estimated variance $\widehat{\sigma}^2/m$. Adding these two and taking square roots gives (2.29). This standard error is *not* the average of the m standard errors for the m individual predictions, as all the predictions are correlated. □

2.21.5 Solution:

The point estimate is the same as in Problem 2.21.3. To compute the standard error, the first term is replaced by $\widehat{\sigma}^2/m$, given by the square root of $\widehat{\sigma}^2/m + \widehat{\sigma}^2(1/1116 + (7.4285 - \overline{\texttt{CSpd}})^2/\texttt{SXX}) = (0.0748)^2$. If the year 2002 were a typical year, then this standard error would be close to $\widehat{\sigma}/\sqrt{n}$, since the other terms will all be smaller. The 95% prediction interval for the mean wind speed over more than 50 years at the candidate site is from 8.609 to 8.902 meters per second. □

CHAPTER 3

Multiple Regression

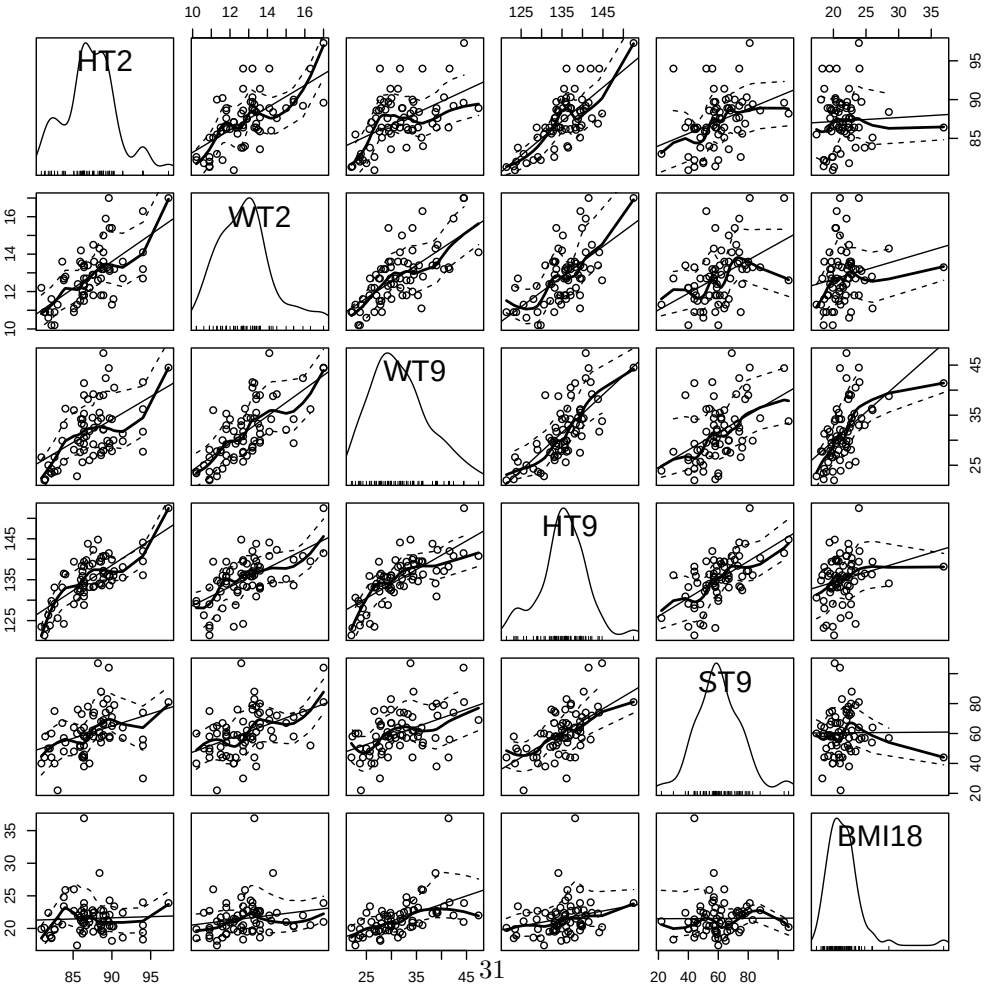
3.1 Solution:

Use any point-identifying software (such as `scatterplot` in the `car` package in R with the argument `id.n` set to about 10) you can discover all the odd points correspond to countries in Africa, apart possibly for Nauru, which is an island nation in the South Pacific. □

3.3 3.3.1 Solution:

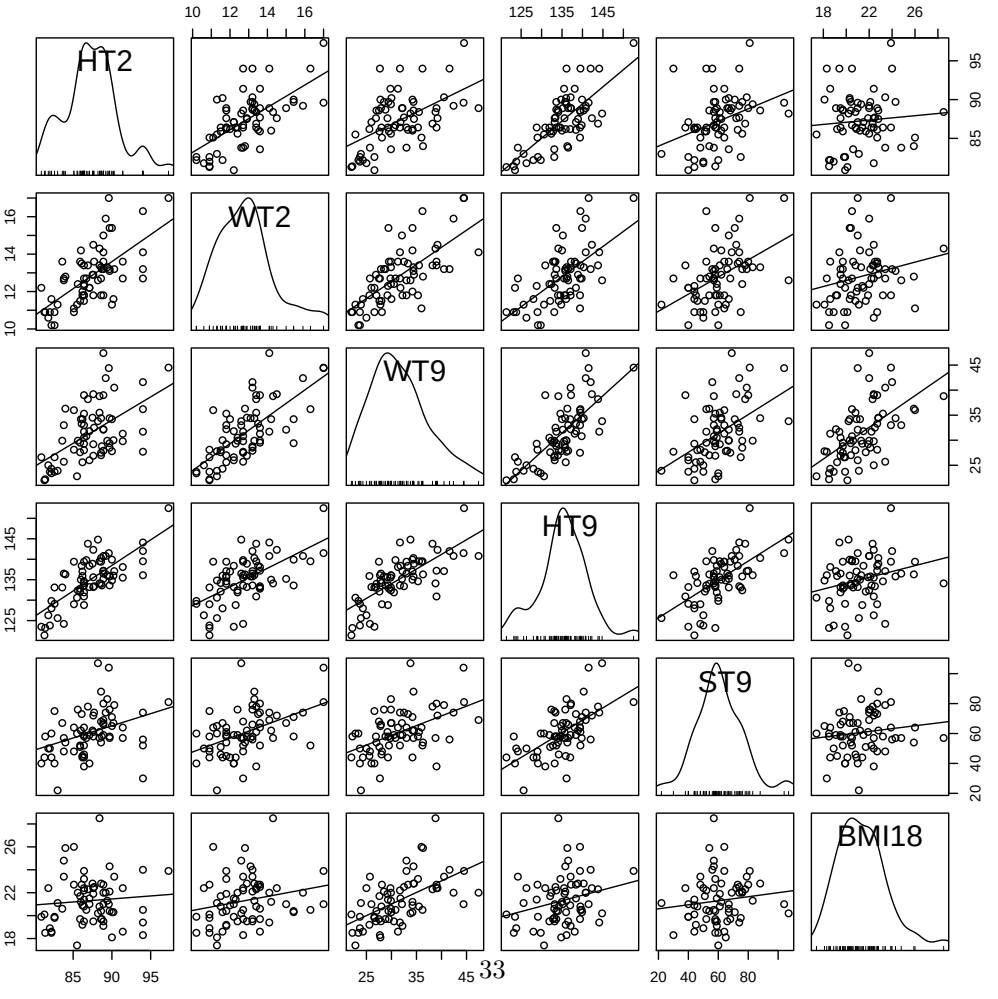
The scatterplot matrix below is enhanced by adding OLS line and a smoother.

```
scatterplotMatrix(~ HT2 + WT2 + WT9 + HT9 + ST9 + BMI18, BGSgirls)
```


HT2
WT2
WT9
HT9
ST9
BMI18
10 12 14 16
125 135 145
20 25 30 35
85 90 95
25 35 45
20 40 60 80

In virtually all of the frames that don't include BMI18, the regressions have linear mean functions, which means that the OLS fit and the smoother agree. This is the ideal case of multiple linear regression. The last row of the scatterplot matrix has the summary plots for the regression of BMI18 on each of the predictors individually. Examining this graphs is difficult because resolution is lost due to a girl with BMI excess of 35 (values above 30 indicate obesity), and so getting a useful visual impression requires removing this point and replotting:

```
scatterplotMatrix(~ HT2 + WT2 + WT9 + HT9 + ST9 + BMI18, BGSgirls,
  smooth=FALSE, subset = BMI18 < 35)
```


HT2
WT2
WT9
HT9
ST9
BMI18
10 12 14 16
125 135 145
18 22 26
85 90 95
10 12 14 16
25 35 45
125 135 145
20 40 60 80
18 22 26

We now see that WT9 is the most closely related to BMI18, and we can't really judge the role of the other predictors in a multiple regression from this plot.

The sample correlation matrix for all the girls.

```
print(cor(BGSgirls[, c("HT2", "WT2", "HT9", "WT9", "ST9", "BMI18")]), digits=3)
```

```
         HT2   WT2   HT9   WT9    ST9  BMI18
HT2   1.0000 0.645 0.738 0.523 0.3617 0.0426
WT2   0.6445 1.000 0.607 0.693 0.4516 0.1909
HT9   0.7384 0.607 1.000 0.728 0.6034 0.2369
WT9   0.5229 0.693 0.728 1.000 0.4530 0.5459
ST9   0.3617 0.452 0.603 0.453 1.0000 0.0056
BMI18 0.0426 0.191 0.237 0.546 0.0056 1.0000
```

From the scatterplot matrix we know to question the usefulness of the correlations with BMI18, because the 1 unusual point could distort the correlations. Deleting the 1 unusual girl:

```
sel <- BGSgirls$BMI18 < 35
print(
 with(BGSgirls[sel ,],
      cor(BMI18, cbind(HT2, WT2, HT9, WT9, ST9))), digits=3)
```

```
        HT2   WT2   HT9   WT9   ST9
[1,] 0.0861 0.223 0.261 0.565 0.129
```

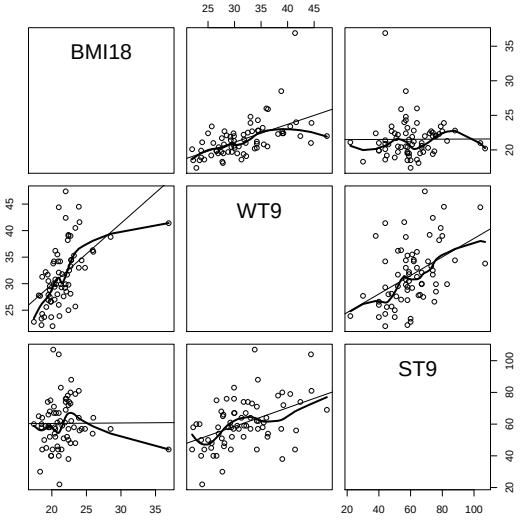
we see that in this particular case since the girl with large BMI18 is near the middle of the range of the other variables it has little influence on the correlation, and so in this case the original corrleation matrix would provide a sensible summary. □

3.3.2 Solution:

The four plots can be drawn in essentially any computer package, since all that is required is two-dimensional scatterplots and saving residuals. Some programs (for example, JMP) draw added-variable plots whenever a multiple linear regression model is fit; others such as R, have pre-written function in the car library for added-variable plots.

The marginal plots can be drawn using a scatterplot matrix,

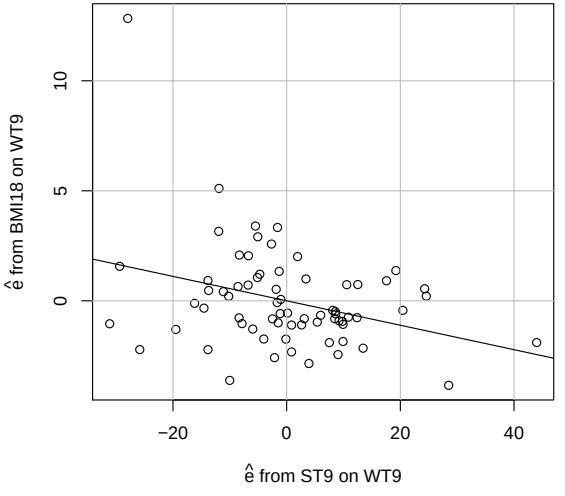
```
scatterplotMatrix(~BMI18 + WT9 + ST9, BGSgirls,
  spread=FALSE, diagonal="none")
```



The marginal response plots are in the first row, and the relevant plot of the predictors is given in the last plot in the second row. We see `BMI18` and `ST9` nearly uncorrelated. The added-variable plot is computed without using special software using:

```
r1 <- residuals(lm(BMI18 ~ WT9, BGSgirls))
r2 <- residuals(lm(ST9 ~ WT9, BGSgirls))
m3 <- lm(r1 ~ r2)
plot(r1 ~ r2,
    xlab=expression(paste(hat(e), " from ST9 on WT9")),
    ylab=expression(paste(hat(e), " from BMI18 on WT9")))
```

```
grid(col="gray", lty="solid")
abline(m3)
```



The added-variable plot for `ST9` after `WT9` shows that after adjustment `BMI18` and `ST9` are negatively related. This relationship is likely due at least in part to the 1 girl with `BMI18` larger than 35. The point corresponding to this girl appears in the upper left-corner of the plot. If this point is deleted the corresponding added-variable plot is

The relationship may persist but it is weaker without this 1 point. □

3.3.3 Solution:

```
Call:
lm(formula = BMI18 ~ HT2 + WT2 + HT9 + WT9 + ST9, data = BGSgirls)

Residuals:
   Min     1Q Median     3Q    Max
-5.095 -1.219 -0.253  1.009 10.495

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.85533    8.78116    3.51  0.00082
HT2         -0.19400    0.13082   -1.48  0.14300
WT2         -0.31778    0.27874   -1.14  0.25850
HT9          0.00806    0.09634    0.08  0.93361
WT9          0.41976    0.07521    5.58  5.2e-07
ST9         -0.04442    0.02222   -2.00  0.04985

Residual standard error: 2.14 on 64 degrees of freedom
Multiple R-squared:  0.443,        Adjusted R-squared:   0.4
F-statistic: 10.2 on 5 and 64 DF,  p-value: 3.29e-07
```

The regression explains about $100 \times R^2 = 44\%$ of the variation in `BMI18`. The hypotheses tested by the t-values are that each of the $\beta_j = 0$ with the other βs arbitrary versus $\beta_j \neq 0$ with all the other βs arbitrary. For this test, only the height variables have t-values with p-values smaller than 0.05. This seems to conflict with the information from the scatterplot matrix, but the scatterplot matrix contains information about *marginal tests* ignoring other variables, while the t-tests are *condition* and correspond to added-variable plots. □

3.5 3.5.1 Solution:

(1) $\widehat{\beta}_1 = \mathtt{SX_1Y}/\mathtt{SX_1X_1}$; (2) $\widehat{\beta}_2 = \mathtt{SX_2Y}/\mathtt{SX_2X_2}$; (3) $\widehat{\beta}_3 = 0$. □

3.5.2 Solution:

(1) $\widehat{e}_{1i} = y_i - \bar{y} - \widehat{\beta}_1(x_{i1} - \bar{x}_1)$; (2) $\widehat{e}_{3i} = x_{i2} - \bar{x}_2$. □

3.5.3 Solution:

Because $\sum \widehat{e}_{3i} = 0$,

$$\begin{aligned}
\text{Slope} &= \sum \widehat{e}_{3i}\widehat{e}_{1i}/\widehat{e}_{3i}^2 \\
&= \sum (x_{i2} - \bar{x}_2)(y_i - \bar{y} - \widehat{\beta}_1(x_{i1} - \bar{x}_1)/(x_{i2} - \bar{x}_2)^2 \\
&= \left(\sum (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) + \mathtt{SX_2Y}\right)/\mathtt{SX_2X_2} \\
&= \mathtt{SX_2Y}/\mathtt{SX_2X_2} \\
&= \widehat{\beta}_2
\end{aligned}$$

The estimated intercept is exactly 0, and the R^2 from this regression is exactly the same as the R^2 from the regression of Y on X_2. □

3.7 Suppose that $\mathbf{A}$ is a $p \times p$ symmetric matrix that we write in partitioned form

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}' & \mathbf{A}_{22} \end{pmatrix}$$

The matrix $\mathbf{A}_{11}$ is $p_1 \times p_1$, so $\mathbf{A}_{22}$ is $(p - p_1) \times (p - p_1)$. One can show that if $\mathbf{A}^{-1}$ exists, it can be written as

$$\mathbf{A}^{-1} = \begin{pmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}' & -\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{12}' & \mathbf{A}_{22}^{-1} \end{pmatrix}$$

Using this result, show that, if $\mathbf{X}$ is an $n \times (p+1)$ data matrix with all 1s in the first column,

$$(\mathbf{X}'\mathbf{X})^{-1} = \begin{pmatrix} \frac{1}{n} + \bar{\mathbf{x}}'(\mathcal{X}'\mathcal{X})^{-1}\bar{\mathbf{x}} & -\bar{\mathbf{x}}'(\mathcal{X}'\mathcal{X})^{-1} \\ -(\mathcal{X}'\mathcal{X})^{-1}\bar{\mathbf{x}} & (\mathcal{X}'\mathcal{X})^{-1} \end{pmatrix}$$

where $\mathcal{X}$ and $\bar{\mathbf{x}}$ are defined in Section 3.4.3.

CHAPTER 4

Interpretation of Main Effects

4.1 Solution:
The original regressors give:

```
summary(m1 <- lm(BMI18 ~ WT2 + WT9 + WT18, BGSgirls))

Call:
lm(formula = BMI18 ~ WT2 + WT9 + WT18, data = BGSgirls)

Residuals:
   Min     1Q Median     3Q    Max
-3.104 -0.743 -0.124  0.832  4.348
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.3098     1.6552    5.02  4.2e-06
WT2          -0.3866     0.1515   -2.55    0.013
WT9           0.0314     0.0494    0.64    0.527
WT18          0.2874     0.0260   11.04  < 2e-16

Residual standard error: 1.33 on 66 degrees of freedom
Multiple R-squared:  0.777,        Adjusted R-squared:  0.767
F-statistic: 76.7 on 3 and 66 DF,  p-value: <2e-16
```

The revised regressors give:

```
BGSgirls$ave <- with(BGSgirls, (WT2 + WT9 + WT18)/3)
BGSgirls$lin <- with(BGSgirls, WT18 - WT2)
BGSgirls$quad <- with(BGSgirls, WT2 - 2*WT9 + WT18)
summary(m2 <- lm(BMI18 ~ ave + lin + quad, BGSgirls))
```

```
Call:
lm(formula = BMI18 ~ ave + lin + quad, data = BGSgirls)

Residuals:
   Min     1Q Median     3Q    Max
-3.104 -0.743 -0.124  0.832  4.348

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.3098     1.6552    5.02  4.2e-06
ave          -0.0678     0.1275   -0.53      0.6
lin           0.3370     0.0747    4.51  2.7e-05
```

```
quad         -0.0270     0.0398   -0.68      0.5

Residual standard error: 1.33 on 66 degrees of freedom
Multiple R-squared:  0.777,        Adjusted R-squared:  0.767
F-statistic: 76.7 on 3 and 66 DF,  p-value: <2e-16
```

(1) The summary statistics R^2 and $\widehat{\sigma}^2$ are identical. (2) All residuals are identical. (3) Intercepts are the same. The mean function for the first model is

$$\mathrm{E}(\texttt{BMI18}|\text{Weights}) = \beta_0 + \beta_1\texttt{WT2} + \beta_2\texttt{WT9} + \beta_3\texttt{WT18}$$

Substituting the definitions of `ave`, `lin` and `quad`, the mean function for the second model is

$$\begin{aligned}
\mathrm{E}(\texttt{BMI18}|\text{Weights}) &= \eta_0 + \eta_1\texttt{ave} + \eta_2\texttt{lin} + \eta_3\texttt{quad} \\
&= \eta_0 + \eta_1(\texttt{WT2} + \texttt{WT9} + \texttt{WT18})/3 \\
&\quad +\eta_2(\texttt{WT2} - \texttt{WT18}) + \eta_3(\texttt{WT2} - 2\texttt{WT9} + \texttt{WT18}) \\
&= \eta_0 + (\eta_1/3 + \eta_2 + \eta_3)\texttt{WT2} + (\eta_1/3 - 2\eta_3)\texttt{WT9} \\
&\quad +(\eta_1/3 - \eta_2 + \eta_3)\texttt{WT18}
\end{aligned}$$

which shows the relationships between the βs and the ηs (for example, $\widehat{\beta}_1 = \widehat{\eta}_1/3 + \widehat{\eta}_2 + \widehat{\eta}_3$). The interpretation in the transformed scale may be a bit easier, as only the linear trend has a small p-value, so we might be willing to describe the change in `BMI18` over time as increasing by the same amount each year. □

4.3 4.3.1 Solution:

$$x_i = \mu_x + \frac{1}{\rho_{xy}}\frac{\sigma_x}{\sigma_y}(y_i - \mu_y)$$

This is undefined if $\rho_{xy} = 0$. □

4.3.2 Solution:

Simply reverse the role of x and y in (4.14) to get

$$x_i|y_i \sim \mathrm{N}\left(\mu_x + \rho_{xy}\frac{\sigma_x}{\sigma_y}(y_i - \mu_y), \sigma_x^2(1-\rho_{xy}^2)\right)$$

These produce the same fitted line if and only if the correlation is equal to $+1$, -1 or 0. □

4.5 Solution:

Changing the base of logs would multiply the equations shown by a constant, but the value of β_1 will be divided by the same constant, resulting in no effect on the results. For the second part, the coefficient has to be multiplied by a factor $\log_e(10)$ that converts from base-10 to base-e. □

4.7 Solution:

Compute $100 \times (\exp(log(1.25)\widehat{\beta}_1) - 1)$, where $\widehat{\beta}_1$ is the estimate for $\log(\texttt{ppgdp})$. □

4.9 4.9.1 Solution:

The intercept is \$24,697, which is the estimated salary for a male faculty members. Female faculty members have expected salaries that are \$3340 lower. □

4.9.2 Solution:

Using Section 4.2, given (4.22), we get to (4.21) by replacing `Years` by the conditional expectation of `Years` given the other 3 regressors,

$$\mathrm{E}(\widehat{\texttt{Salary}|\texttt{Sex}}) = 18,065 + 201\texttt{Sex} + 759\mathrm{E}(\texttt{Years}|\texttt{Sex})$$

Equating the right side of this last equation with the right side of (4.22), we can solve for $\mathrm{E}(\texttt{Years}|\texttt{Sex})$,

$$\begin{aligned}\mathrm{E}(\texttt{Years}|\texttt{Sex}) &= \frac{24697-18065}{759} - \frac{3340+201}{759}\texttt{Sex} \\ &\approx 8.7 - 4.7\texttt{Sex}\end{aligned}$$

The two mean functions are consistent if the average male has about 8.7 years of experience but the average female has only about $8.7 - 4.7 = 4.0$ years of experience. □

4.11 4.11.1 Solution:

$x \sim \mathrm{N}(0,1)$ and $e \sim \mathrm{N}(0,1)$. Hence y is also normal with mean 0 and variance $\mathrm{Var}(y) = \mathrm{Var}(2x + e) = 4 + 1 = 5$. The covariance between x and y is $\mathrm{Cov}(x, y) = \mathrm{Cov}(x, 2x + e) = 2\mathrm{Cov}(x, x) = 2$, and so

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}\right)$$

The squared correlation between x and y is $\rho^2_{xy} = (2/\sqrt{5})^2 \approx 0.8$, and $\sigma^2_y(1-\rho^2_{xy}) = 5(1-0.8) = 1$. From (4.14–4.15),

$$y|x \sim \mathrm{N}(2x, 1)$$

□

4.11.2 Solution:

```
set.seed(1000)
x <- rnorm(10000)
e <- rnorm(10000)
y <- 2*x + e
summary(m <- lm(y ~ x))
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.00360    0.00991   -0.36     0.72
x            2.00444    0.00997  201.15   <2e-16

Residual standard error: 0.991 on 9998 degrees of freedom
Multiple R-squared: 0.802
F-statistic: 4.05e+04 on 1 and 9998 DF,  p-value: <2e-16
```

The results are as expected. □

4.11.3 Solution:

```
data.frame(tval = t1 <- (coef(m)[2] - 2)/sqrt(vcov(m)[2, 2]),
  df = m$df.residual,
  pval = 2 * (1 - pt(t1, m$df.residual)))
```

```
   tval   df   pval
x 0.446 9998 0.6556
```

Since the NH is true by construction, the chance of rejecting at the 5% level is by definition 0.05. This does not depend on the sample size. □

4.11.4 Solution:

```
sumry <- function(m){
  c(coef=m$coef, sigmahat=sigmaHat(m), r2=summary(m)$r.squared)}
rbind(
  sumry(m),
  sumry(update(m, subset = abs(x) < 2/3)),
  sumry(update(m, subset = abs(x) > 2/3)),
  sumry(update(m, subset = x < 0)))
```

```
     coef.(Intercept) coef.x sigmahat     r2
[1,]       -0.0036048  2.004   0.9914 0.8019
[2,]        0.0009264  2.047   0.9875 0.3789
[3,]       -0.0079601  2.001   0.9953 0.8802
[4,]       -0.0082501  2.003   0.9988 0.5884
```

In all cases the estimated intercept, slope and σ are close to the population values but R^2 depends on the sampling. □

4.11.5 Solution:

```
rbind(
  sumry(m),
```

```
   sumry(update(m, subset = abs(y) < 1.5)),
   sumry(update(m, subset = abs(y) > 1.5)),
   sumry(update(m, subset = y < 0)))
```

```
      coef.(Intercept) coef.x sigmahat     r2
[1,]        -0.003605 2.0044   0.9914 0.8019
[2,]        -0.005126 0.9148   0.6603 0.3801
[3,]         0.003805 2.2134   1.0381 0.8828
[4,]        -0.711432 1.4961   0.8597 0.5923
```

The results here are quite different from the last subproblem. If we include only cases with $|y| < 1.5$ then the estimated slope is considerably smaller than the population value of $\beta_1 = 2$ and $\widehat{\sigma}$ is also too small. If we only see the cases with extreme values of y then the estimates are closer to the population values. The last case in which we only see the cases with smaller values of y gives an answer that is not obviously relevant to estimating the population values. Without knowledge of the plan for removing an observation the regression may not provide useful information. □

4.13 Solution:

```
MinnWater$perCapitaUse <- with(MinnWater, 10^6 * muniUse/muniPop)
m0 <- lm(log(perCapitaUse) ~ year, MinnWater)
m1 <- update(m0, ~ . + muniPrecip)
round(compareCoefs(m0, m1), 6)
```

```
Call:
1:"lm(formula = log(perCapitaUse) ~ year, data = MinnWater)"
2:"lm(formula = log(perCapitaUse) ~ year + muniPrecip, data = MinnWater)"
               Est. 1      SE 1    Est. 2      SE 2
(Intercept)  3.62e+00  3.72e+00  3.50e+00  2.59e+00
year         5.63e-05  1.86e-03  2.16e-04  1.30e-03
muniPrecip                     -1.03e-02  2.08e-03
```

```
              Est. 1     SE 1    Est. 2     SE 2
(Intercept) 3.617979 3.716861  3.504036 2.592557
year        0.000056 0.001859  0.000216 0.001297
muniPrecip        NA       NA -0.010259 0.002084
```

□

CHAPTER 5

Complex Regressors

5.1 5.1.1 Solution:

If X is at its lowest level, $U_2 = \cdots = U_d = 0$, and substituting into (5.17), $\mathrm{E}(Y|U_2 = 0, \ldots, U_d = 0) = \beta_0$. If X is at level j, for any $j \in \{2, \ldots, d\}$, $U_j = 1$ while $U_k = 0$ for $k \neq j$. Consequently, $\mathrm{E}(Y|U_j = 1, U_k = 0, k \neq j) = \beta_0 + \beta_d$. □

5.1.2 Solution:

Let μ_j be as defined in Problem 5.1.1, and recalling that for fixed j, only $U_j \neq 0$,

$$\texttt{RSS}(\boldsymbol{\beta}) = \sum_{j=1}^{d}\sum_{i=1}^{n_j}(y_{ji} - \beta_0 - \beta_2 U_2 - \cdots - \beta_d U_d)^2 \tag{5.1}$$

$$= \sum_{j=1}^{d}\sum_{i=1}^{n_j}(y_{ij} - \mu_j)^2 \tag{5.2}$$

For each j we will then have $\widehat{\mu}_j = \bar{y}_j$ which can be verified either by recalling that the least squares estimate of a population mean is the sample mean, or by differentiating $\sum_i (y_{ij} - \mu_j)^2$ with respect to μ_j, setting the result to zero and solving for μ_j. By invariance of least squares estimates under linear transformation, $\widehat{\beta}_0 = \widehat{\mu}_1 = \bar{y}_1$, and $\widehat{\beta}_j = \widehat{\mu}_j - \widehat{\mu}_1 = \bar{y}_j - \bar{y}_1$ for $j > 1$. □

5.1.3 Solution:

The residual sum of squares function evaluated at the ols estimates is

$$\texttt{RSS}(\widehat{\boldsymbol{\beta}}) = \sum_{j=1}^{d}\left[\sum_{i=1}^{n_j}(y_{ij} - \widehat{\mu}_j)^2\right]$$

For fixed j, the quantity in square brackets is $n_j - 1)\text{SD}_j^2$. □

5.1.4 Solution:

If all the n_j are equal to the same value, say n_1, then

$$\begin{aligned}
\text{se}(\widehat{\beta}_0|X)^2 &= \text{se}(\widehat{\mu}_1|X)^2 \\
&= \widehat{\sigma}^2/n_1 \\
se(\widehat{\beta}_j|X)^2 &= \text{se}(\widehat{\mu}_j - \widehat{\mu}_1|X)^2 \\
&= \text{se}(\widehat{\mu}_j|X)^2 + \text{se}(\widehat{\mu}_1|X)^2 \\
&= \widehat{\sigma}^2(1/n_1 + 1/n_1)
\end{aligned}$$

The estimated group means are uncorrelated because they are computed from different observations, and so there is no covariance term to consider. □

5.3 5.3.1 Solution:

This uses the `lsmeans` package in R.

```
library(lsmeans)
m1 <- lm(lifeExpF ~ group, UN11)
lsmeans(m1, pairwise ~ group)
```

```
$lsmeans
 group  lsmean     SE  df lower.CL upper.CL
 oecd    82.45 1.1279 196    80.22    84.67
 other   75.33 0.5856 196    74.17    76.48
 africa  59.77 0.8626 196    58.07    61.47

Confidence level used: 0.95

$contrasts
 contrast       estimate    SE  df t.ratio p.value
 oecd - other       7.12 1.271 196   5.602  <.0001
 oecd - africa     22.67 1.420 196  15.968  <.0001
 other - africa    15.55 1.043 196  14.918  <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

The `lsmeans` function takes the name of a regression model as its first argument. The second argument `pairwise ~ group` tells R to do pairwise comparisons of the levels of `group`, as required. The second part of the output is the relevant part for this problem. □

5.3.2 Solution:

```
m2 <- lm(lifeExpF ~ group + log(ppgdp), UN11)
lsmeans(m2, pairwise ~ group)[[2]]
```

```
contrast       estimate     SE  df t.ratio p.value
oecd - other      1.535 1.1737 195   1.308  0.3927
oecd - africa    12.170 1.5574 195   7.814  <.0001
other - africa   10.636 0.9792 195  10.862  <.0001

P value adjustment: tukey method for comparing a family of 3 estimates
```

The baffling `[[2]]` printed only the second part of the output from the `lsmeans` command. The main change is the adjusted means for `oecd` and `other` do not appear to be different ($p \approx .39$). □

5.5 5.5.1 Solution:

`Y ~ A + B + A:B` □

5.5.2 Solution:

We can write out:

$$\begin{aligned}
\mu_{11} &= \mathrm{E}(Y|A = a_1, B = b_1) = \beta_0 \\
\mu_{12} &= \mathrm{E}(Y|A = a_1, B = b_2) = \beta_0 + \beta_2 B_2 \\
\mu_{13} &= \mathrm{E}(Y|A = a_1, B = b_3) = \beta_0 + \beta_3 B_3 \\
\mu_{21} &= \mathrm{E}(Y|A = a_2, B = b_1) = \beta_0 + \beta_1 A_2 \\
\mu_{22} &= \mathrm{E}(Y|A = a_2, B = b_2) = \beta_0 + \beta_1 A_2 + \beta_2 B_2 + \beta_4 A_2 B_2 \\
\mu_{23} &= \mathrm{E}(Y|A = a_2, B = b_3) = \beta_0 + \beta_1 A_2 + \beta_3 B_3 + \beta_5 A_2 B_3
\end{aligned}$$

This gives 6 equations in 6 unknowns, which can be solved:

$$\begin{aligned}
\beta_0 &= \mu_{11} \\
\beta_1 &= -\mu_{11} + \mu_{21} \\
\beta_2 &= -\mu_{11} + \mu_{12} \\
\beta_3 &= -\mu_{11} + \mu_{13} \\
\beta_4 &= +\mu_{11} - \mu_{21} - \mu_{12} + \mu_{22} \\
\beta_5 &= +\mu_{11} - \mu_{13} - \mu_{21} + \mu_{23}
\end{aligned}$$

Only the intercept is directly interpretable as a mean for a combination of factor levels. All the main effects are differences between a mean and μ_{11}. The interactions are relatively complicated linear combinations of 4 of the 6 cell means. □

5.5.3 Solution:

The 6 equations for the main-effects only model are

$$\begin{aligned}
\mu_{11} &= \mathrm{E}(Y|A = a_1, B = b_1) = \beta_0 \\
\mu_{12} &= \mathrm{E}(Y|A = a_1, B = b_2) = \beta_0 + \beta_2 B_2 \\
\mu_{13} &= \mathrm{E}(Y|A = a_1, B = b_3) = \beta_0 + \beta_3 B_3 \\
\mu_{21} &= \mathrm{E}(Y|A = a_2, B = b_1) = \beta_0 + \beta_1 A_2 \\
\mu_{22} &= \mathrm{E}(Y|A = a_2, B = b_2) = \beta_0 + \beta_1 A_2 + \beta_2 B_2 \\
\mu_{23} &= \mathrm{E}(Y|A = a_2, B = b_3) = \beta_0 + \beta_1 A_2 + \beta_3 B_3
\end{aligned}$$

This gives 6 equations but only 4 unknowns. This equations are consistent, however, because $\mu_{22} = \mu_{12} + \mu_{21} - \mu_{11}$ and $\mu_{32} = \mu_{13} + \mu_{21} - \mu_{11}$. This means that only 4 of the μ_{ij} are needed as

the remaining 2 are just functions of them. Thus

$$\begin{aligned}\beta_0 &= \mu_{11}\\ \beta_1 &= -\mu_{11} + \mu_{21}\\ \beta_2 &= -\mu_{11} + \mu_{12}\\ \beta_3 &= -\mu_{11} + \mu_{13}\end{aligned}$$

This is the same as for the interaction-included model. □

5.5.4 Solution:

For the model of Problem 5.5.2,

$$\begin{aligned}\mu_{+1} &= \beta_0 + \beta_1/2\\ \mu_{+2} &= (\beta_0 + \beta_2) + (\beta_1 + \beta_4)/2\\ \mu_{+3} &= (\beta_0 + \beta_3) + (\beta_1 + \beta_5)/2\end{aligned}$$

Main-effects are generally not recommended with interactions present because they depend on the interaction parameters β_4 and β_5. In the no-interaction model of Problem 5.5.3 set $\beta_4 = \beta_5 = 0$. □

5.5.5 Solution:

There are now only 5 equations and 5 parameters because we have no data for 1 of the cells. For those 5 cells the solution is the same as if all 6 cells are observed.

The main effects of factor B are not now well defined because we can't average over A for all the levels of B. □

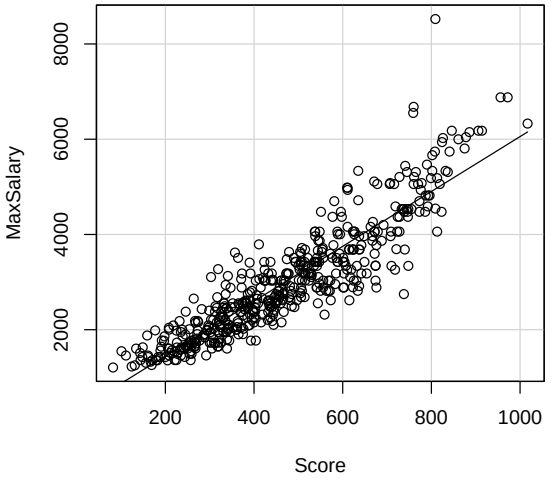
5.7 Solution:

(5.19) is a model of parallel regressions: the slope for each level of F is β_1, and the intercepts are β_0, $\beta_0 + \beta_2$ and $\beta_0 + \beta_3$.

(5.20) is a model of common intercept: The lines cross at the intercept β_0 for all three levels of F, and the slopes are β_1, $\beta_1 + \beta_2$ and $\beta_1 + \beta_3$.

(5.21) is similar to (5.20), except that the three lines cross at $x_2 = \delta$ rather than at $x_1 = 0$. □

5.9 5.9.1 Solution:

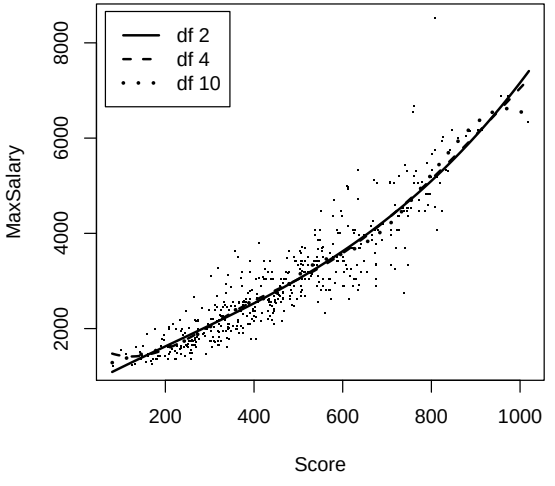


The mean function is clearly curved and variability increases from left to right. □

5.9.2 Solution:

```
library(splines)
m3 <- lm(MaxSalary ~ bs(Score), salarygov)
m5 <- lm(MaxSalary ~ bs(Score, 5), salarygov)
m10 <- lm(MaxSalary ~ bs(Score, 10), salarygov)
plot(MaxSalary ~Score, salarygov, pch=".")
```

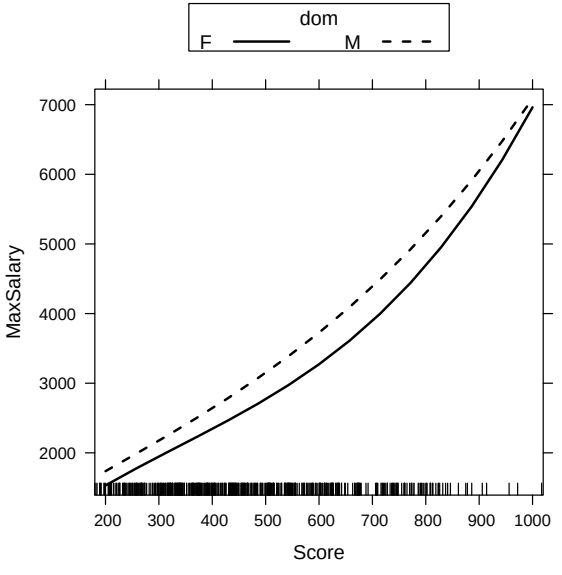
```
snew <- 80:1020
lines(snew, predict(m3, data.frame(Score=snew)), lwd=2, lty=1)
lines(snew, predict(m5, data.frame(Score=snew)), lwd=2, lty=2)
lines(snew, predict(m10, data.frame(Score=snew)), lwd=3, lty=3)
legend("topleft", paste("df", c(2, 4, 10)), lty=1:3, lwd=c(2,2,3), inset=.02)
```



The default for `bs` is three basis vectors, as used in `m3`. All three fits match the data fairly well, with the larger values of *df* providing rather unbelievable results at the boundaries of the range. The $df = 3$ solution seems to match the data well. □

5.9.3 Solution:

```
                 Estimate Std. Error t value  Pr(>|t|)
(Intercept)         955.8      222.6  4.2942 2.117e-05
bs(Score)1         1707.7      736.1  2.3199 2.076e-02
bs(Score)2         1676.5      823.0  2.0371 4.218e-02
bs(Score)3         6251.5     1371.7  4.5575 6.554e-06
domM                267.6      290.9  0.9198 3.582e-01
bs(Score)1:domM    -332.3      872.9 -0.3807 7.036e-01
bs(Score)2:domM     804.8      860.7  0.9351 3.502e-01
bs(Score)3:domM    -197.6     1418.9 -0.1393 8.893e-01
```



The effects plot suggests that the female-dominated fit is consistently below the male-dominated fit, as confirmed by the t-test for the factor `dom`. The estimated difference is \$268 in favor of males.

There is little visual evidence that the splines have different shapes in the two groups, which we will confirm with a test in Problem 6.11. □

5.11 5.11.1 Solution:

```
m5 <- update(m4, ~ . + financing)
confint(m5)["financingseller_financed" ,]
```

```
   2.5 %    97.5 %
-0.11466 -0.07088
```

The negative sign suggests seller financed sales lower than other types of sales, with seller financed sales estimated to be between 11% lower and 7% lower. □

5.11.2 Solution:

The first statement implies causation, and therefore cannot be supported by this observational study. The second statement is consistent with the data, but this is not the only possible explanation of the outcome. □

5.13 Solution:

The tricky bit here is combining the two data files into one.

```
combined.data <- data.frame(bp = c(Forbes$bp, Hooker$bp),
                  pres=c(Forbes$pres, Hooker$pres),
                  Source = factor(rep(c("F", "H"), c(dim(Forbes)[1], dim(Hooker)[1])))
                  )
m1 <- lm(log(pres) ~ Source + bp + Source:bp, combined.data)
summary(m1)$coef
```

```
              Estimate Std. Error  t value  Pr(>|t|)
(Intercept) -0.9708662  0.0749676 -12.9505 1.304e-16
SourceH     -0.0512712  0.0823661  -0.6225 5.368e-01
bp           0.0206224  0.0003692  55.8502 1.713e-42
SourceH:bp   0.0002474  0.0004098   0.6037 5.491e-01
```

The significance level for the interaction, which corresponds to the hypothesis of interest, is slightly larger than 0.05. If the likely outlier in Forbes' data is deleted:

```
summary(update(m1, subset=-12))$coef
```

```
              Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.9517662  0.0622593 -15.287 5.421e-19
SourceH     -0.0703712  0.0683775  -1.029 3.092e-01
bp           0.0205186  0.0003068  66.879 4.240e-45
SourceH:bp   0.0003512  0.0003403   1.032 3.078e-01
```

The significance level decreases to about 0.04, not much of a change. A small difference in slope between the two sources may be apparent. □

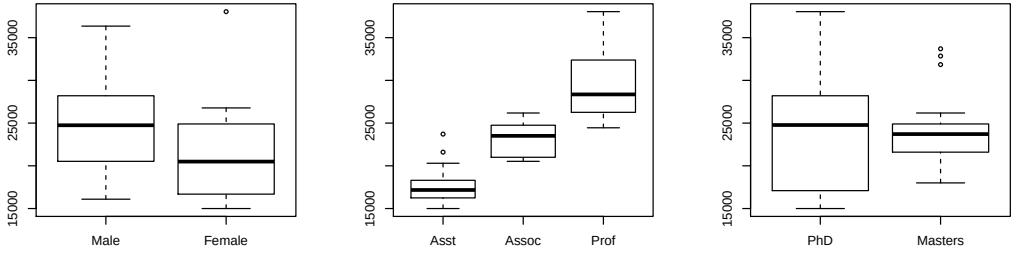
5.15 Solution:

Model (a) the `HT2` effect and the `HT9` to be the same for each level of `Sex`. Model (b) allows the `HT2` effect and the `HT9` effects to vary separately by level of `Sex`. Model (c) allows the `HT2` effect and the `HT9` effects to vary jointly by level of `Sex`: the effect of `HT9`, for example, depends on both the value of `Sex` and on `HT2`. □

5.17 5.17.1 Solution:

Let's start with the factors. These can be displayed in boxplots:

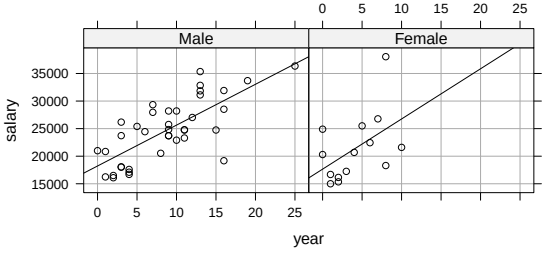
```
par(mfrow=c(1, 3))
boxplot(salary~sex, salary)
boxplot(salary~rank, salary)
boxplot(salary~degree, salary)
```



Female salaries appear to be generally lower than male salaries, salary increases with rank. Faculty with a Masters degree have much more variable salaries. The boxplots don't have anything to say about interactions.

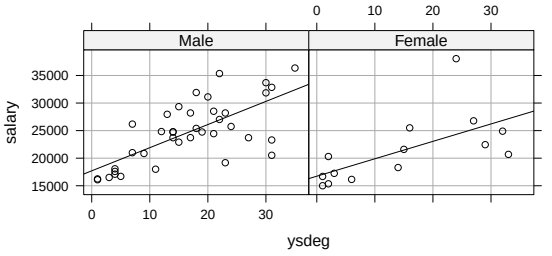
Turn next to the two continuous variables:

```
xyplot(salary~year|sex, data=salary, type=c("p", "g", "r"))
```



Females generally have fewer years in rank, and while for males salary clearly increases with `year`, this is not so clear for females.

```
xyplot(salary~ysdeg|sex, data=salary, type=c("p", "g", "r"))
```



Interestingly, the females are more variable on `ysdeg` than on `year`. For this variable it does appear that `salary` increases with `ysdeg` for both sexes. □

5.17.2 Solution:

This is simply a two-sample t-test, which can be computed using regression software by fitting an intercept and a dummy variable for `sex`. Using regression software:

```
summary(m0 <- lm(salary ~ sex, salary))$coef
```

```
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)     24697        938  26.330 5.762e-31
sexFemale       -3340       1808  -1.847 7.060e-02
```

The significance level is 0.07 two-sided, and about 0.035 for the one-sided test that women are paid less. The point estimate of the `Sex` effect is $3340 in favor of men. □

5.17.3 Solution:

```
m2 <- lm(salary ~ ., data=salary)
summary(m2)
```

```
Call:
lm(formula = salary ~ ., data = salary)
```

```
Residuals:
   Min     1Q Median     3Q    Max
 -4045  -1095   -362    813   9193

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    15746.0      800.2   19.68  < 2e-16
degreeMasters   1388.6     1018.7    1.36     0.18
rankAssoc       5292.4     1145.4    4.62  3.2e-05
rankProf       11118.8     1351.8    8.23  1.6e-10
sexFemale       1166.4      925.6    1.26     0.21
year             476.3       94.9    5.02  8.7e-06
ysdeg           -124.6       77.5   -1.61     0.11

Residual standard error: 2400 on 45 degrees of freedom
Multiple R-squared:  0.855,        Adjusted R-squared:  0.836
F-statistic: 44.2 on 6 and 45 DF,  p-value: <2e-16
```

```
confint(m2)["sexFemale", , drop=FALSE]
```

```
           2.5 % 97.5 %
sexFemale -697.8   3031
```

Adjusting for the other predictors, the `Sex` effect is for *higher salaries for females*, although the difference has a corresponding large p-value. □

5.17.4 Solution:

```
summary(update(m2, ~ . - rank))$coef
```

```
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)   17183.6    1147.94 14.9690 1.659e-19
degreeMasters -3299.3    1302.52 -2.5331 1.470e-02
```

```
sexFemale      -1286.5    1313.09 -0.9798 3.322e-01
year             352.0     142.48  2.4703 1.719e-02
ysdeg            339.4      80.62  4.2098 1.144e-04
```

If we ignore `rank`, then the coefficient for `Sex` is again negative, indicating an advantage for males, but the p-value is .33 (or .165 for a one-sided test), indicating that the difference is not significant.

One could argue that other variables in this data set are tainted as well, so using data like these to resolve issues of discrimination will never satisfy everyone. □

5.19 5.19.1 Solution:

Using the parameterization used by default by R, for $i \in (\texttt{len}, \texttt{amp}, \texttt{load})$, let U_{ij} be the dummy variable for level j for variable i, $j = 2, 3$. This parameterization has a dummy variable for the middle and high level of each factor, dropping the low level. The two mean functions in R notation are

$$\begin{aligned}
\mathrm{E}(\log(\texttt{cycles})|\text{First-order}) &= \beta_0 + \sum_{i=1}^{3}\sum_{j=2}^{3}\beta_{ij}U_{ij} \\
\mathrm{E}(\log(\texttt{cycles})|\text{Second-order}) &= \beta_0 + \sum_{i=1}^{3}\sum_{j=2}^{3}\beta_{ij}U_{ij} + \\
&\quad \sum_{i=1}^{2}\sum_{k=i+1}^{3}\sum_{j=2}^{3}\beta_{ikj}U_{ij}U_{kj}
\end{aligned}$$

Most computer programs have a simple way of writing these mean functions. First, declare `len`, `amp`, and `load` to be factors. The two mean functions are then just:

$$\begin{aligned}
\log(\texttt{cycles}) &\sim \texttt{len} + \texttt{amp} + \texttt{load} \\
\log(\texttt{cycles}) &\sim (\texttt{len} + \texttt{amp} + \texttt{load})^2
\end{aligned}$$

The computer program is responsible for creating the correct dummy variables and products. □

5.19.2 Solution:

For the first-order model using the R parameterization, the change is $\beta_{33} - \beta_{32}$. Using the second-order mean function, the change is $\beta_{33} - \beta_{32} + \beta_{133} - \beta_{132} + \beta_{233} - \beta_{232}$. □

CHAPTER 6

Testing and Analysis of Variance

6.1 Solution:

```
Analysis of Variance Table

Model 1: lifeExpF ~ 1
Model 2: lifeExpF ~ group
  Res.Df   RSS Df Sum of Sq   F Pr(>F)
1    198 20293
2    196  7730  2     12563 159 <2e-16
```

In R, the `anova` function when applied to two fitted models computes the desired F test. In this case the p-value is essentially 0 suggesting log(`ppgdp`) should not be removed. □

6.3 Solution:

```
u3 <- lm(lifeExpF ~ group + log(ppgdp), UN11)
u1 <- update(u3, ~ . - log(ppgdp))
anova(u1, u3)
```

```
Analysis of Variance Table

Model 1: lifeExpF ~ group
Model 2: lifeExpF ~ group + log(ppgdp)
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1    196 7730
2    195 5090  1      2640 101 <2e-16
```

In R, the `anova` function when applied to two fitted models computes the desired F test. In this case the p-value is essentially 0 suggesting log(`ppgdp`) should not be removed. □

6.5 6.5.1 Solution:

Since `oecd` is the baseline level for the factor `group`, this hypothesis can be tested with the t-test for the level `other` of `group`

```
u3 <- lm(lifeExpF ~ group + log(ppgdp) + group:log(ppgdp), UN11)
summary(u3)$coef
```

```
                      Estimate Std. Error t value  Pr(>|t|)
(Intercept)            59.2137     15.220  3.8904 0.0001377
groupother            -11.1731     15.595 -0.7165 0.4745723
groupafrica           -22.9848     15.784 -1.4562 0.1469536
log(ppgdp)              2.2425      1.466  1.5292 0.1278438
```

```
groupother:log(ppgdp)    0.9294      1.518  0.6124 0.5409862
groupafrica:log(ppgdp)   1.0950      1.578  0.6937 0.4887032
```

The significance level is 0.193 suggesting no difference in intercept for these groups. □

6.5.2 Solution:

This is harder than the last problem. Here are three ways to do this in R.

```
linearHypothesis(u3, "groupother - groupafrica")
```

```
Linear hypothesis test

Hypothesis:
groupother - groupafrica = 0

Model 1: restricted model
Model 2: lifeExpF ~ group + log(ppgdp) + group:log(ppgdp)

  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1    194 5204
2    193 5078  1       126 4.81  0.029
```

```
deltaMethod(u3, "groupother-groupafrica")
```

```
                       Estimate    SE
groupother-groupafrica    11.81 5.386
```

```
group1 <- relevel(UN11$group, "other")
u4 <- update(u3, ~  log(ppgdp)*group1)
summary(u4)$coef
```

```
                        Estimate Std. Error t value  Pr(>|t|)
(Intercept)              48.0406     3.3971 14.1418 1.263e-31
log(ppgdp)                3.1720     0.3910  8.1131 5.638e-14
group1oecd               11.1731    15.5948  0.7165 4.746e-01
```

```
group1africa             -11.8117     5.3860 -2.1930 2.950e-02
log(ppgdp):group1oecd     -0.9294     1.5177 -0.6124 5.410e-01
log(ppgdp):group1africa    0.1655     0.7028  0.2355 8.140e-01
```

The first approach uses the `linearHypothesis` function in the `car` package, and the second uses the `deltaMethod` function. For this second you need to compute the test yourself by dividing the estimate by its standard error. The third approach is to change the baseline level of the factor and get the t-test of interest automatically. □

6.7 6.7.1 Solution:

```
Analysis of Variance Table

Response: Fuel
           Df Sum Sq Mean Sq F value  Pr(>F)
Tax         1  26635   26635    6.33  0.0155
Dlic        1  79378   79378   18.85 7.7e-05
Income      1  61408   61408   14.58  0.0004
log(Miles)  1  34573   34573    8.21  0.0063
Residuals  46 193700    4211
```

```
Analysis of Variance Table

Response: Fuel
           Df Sum Sq Mean Sq F value  Pr(>F)
log(Miles)  1  70478   70478   16.74 0.00017
Income      1  49996   49996   11.87 0.00123
Dlic        1  63256   63256   15.02 0.00034
Tax         1  18264   18264    4.34 0.04287
Residuals  46 193700    4211
```

Type I ANOVA provides sequential tests and are order dependent. □

6.7.2 Solution:

```
Anova Table (Type II tests)

Response: Fuel
           Sum Sq Df F value  Pr(>F)
Tax         18264  1    4.34 0.04287
Dlic        56770  1   13.48 0.00063
Income      32940  1    7.82 0.00751
log(Miles)  34573  1    8.21 0.00626
Residuals  193700 46
```

```
Anova Table (Type II tests)

Response: Fuel
           Sum Sq Df F value  Pr(>F)
log(Miles)  34573  1    8.21 0.00626
Income      32940  1    7.82 0.00751
Dlic        56770  1   13.48 0.00063
Tax         18264  1    4.34 0.04287
Residuals  193700 46
```

The Type I tests for the last regressor added are equivalent to Type II tests. All other tests are different. □

6.9 Solution:

```
m1 <- lm(Y ~ X1 + I(X1^2) + X2 + I(X2^2) + X1:X2, cakes)
m2 <- update(m1, ~ . - X1:X2)
m3 <- update(m1, ~ . - I(X1^2))
```

```
m4 <- update(m1, ~ . - X1 - I(X1^2) - X1:X2)
anova(m2, m1)
```

```
Analysis of Variance Table

Model 1: Y ~ X1 + I(X1^2) + X2 + I(X2^2)
Model 2: Y ~ X1 + I(X1^2) + X2 + I(X2^2) + X1:X2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      9 4.24
2      8 1.47  1      2.77 15.1 0.0047
```

```
anova(m3, m1)
```

```
Analysis of Variance Table

Model 1: Y ~ X1 + X2 + I(X2^2) + X1:X2
Model 2: Y ~ X1 + I(X1^2) + X2 + I(X2^2) + X1:X2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      9 4.38
2      8 1.47  1      2.91 15.8 0.0041
```

```
anova(m4, m1)
```

```
Analysis of Variance Table

Model 1: Y ~ X2 + I(X2^2)
Model 2: Y ~ X1 + I(X1^2) + X2 + I(X2^2) + X1:X2
  Res.Df   RSS Df Sum of Sq    F  Pr(>F)
1     11 11.47
2      8  1.47  3        10 18.1 0.00063
```

In R, the anova function when applied to two fitted models computes the desired F test. In all three cases the p-value < 0.01 suggesting against all three null hypotheses. □

6.11 Solution:

```
salarygov$dom <- factor(with(salarygov,
           ifelse(NW/NE >= 0.70, "F", "M")))
library(splines)
m1 <- lm(MaxSalary ~ bs(Score)*dom, salarygov)
Anova(m1)
```

```
Anova Table (Type II tests)

Response: MaxSalary
               Sum Sq  Df F value  Pr(>F)
bs(Score)     4.88e+08   3  764.05 < 2e-16
dom           1.03e+07   1   48.35 1.2e-11
bs(Score):dom 9.64e+05   3    1.51    0.21
Residuals     1.04e+08 487
```

```
m2 <- update(m1, ~. - bs(Score):dom)
c(PointEst=coef(m2)[5], confint(m2)["domM",])
```

```
PointEst.domM         2.5 %        97.5 %
        321.6         230.6         412.6
```

Since separate spline fits for each level of the factor are clearly not needed, we refit with a common spline fit. The point estimate of \$321 for the advantage of male dominated job classes has a 95% confidence interval of \$230 to \$412. □

6.13 Solution:

```
m1 <- lm(log(cycles) ~ load + len:amp, Wool)
lsmeans(m1, pairwise ~ load)

$`load lsmeans`
 load lsmean      SE df lower.CL upper.CL
   40  6.705 0.04686 16    6.606    6.804
   45  6.380 0.04686 16    6.280    6.479
   50  5.920 0.04686 16    5.820    6.019

$`load pairwise differences`
        estimate      SE df t.ratio p.value
40 - 45   0.3253 0.06628 16   4.908 0.00044
40 - 50   0.7852 0.06628 16  11.848 0.00000
45 - 50   0.4599 0.06628 16   6.940 0.00001
    p values are adjusted using the tukey method for 3 means

lsmeans(m1, pairwise ~ amp|len)

$`amp:len lsmeans`
 amp len lsmean      SE df lower.CL upper.CL
   8 250  6.034 0.08117 16    5.862    6.207
   9 250  5.585 0.08117 16    5.412    5.757
  10 250  4.802 0.08117 16    4.630    4.974
   8 300  6.932 0.08117 16    6.759    7.104
   9 300  6.480 0.08117 16    6.308    6.653
  10 300  5.764 0.08117 16    5.592    5.936
   8 350  7.955 0.08117 16    7.783    8.127
   9 350  6.891 0.08117 16    6.718    7.063
  10 350  6.570 0.08117 16    6.398    6.742
```

```
$`amp:len pairwise differences`
             estimate     SE df t.ratio p.value
8 - 9 | 250    0.4499 0.1148 16   3.920 0.00331
8 - 10 | 250   1.2324 0.1148 16  10.736 0.00000
9 - 10 | 250   0.7825 0.1148 16   6.816 0.00001
8 - 9 | 300    0.4511 0.1148 16   3.929 0.00324
8 - 10 | 300   1.1674 0.1148 16  10.170 0.00000
9 - 10 | 300   0.7164 0.1148 16   6.241 0.00003
8 - 9 | 350    1.0646 0.1148 16   9.274 0.00000
8 - 10 | 350   1.3854 0.1148 16  12.068 0.00000
9 - 10 | 350   0.3207 0.1148 16   2.794 0.03295
    p values are adjusted using the tukey method for 3 means
```

□

6.15 Solution:

```
m3 <- lm(log(acrePrice) ~ year + fyear, MinnLand)
summary(m3)
```

```
Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.53e+02   7.87e+00  -19.45  < 2e-16
year         8.01e-02   3.92e-03   20.44  < 2e-16
fyear2003   -8.16e-02   2.94e-02   -2.78  0.00547
fyear2004   -1.22e-02   2.64e-02   -0.46  0.64350
fyear2005    1.20e-01   2.46e-02    4.87  1.1e-06
fyear2006    7.36e-02   2.34e-02    3.15  0.00163
fyear2007    7.64e-02   2.23e-02    3.43  0.00062
fyear2008    2.03e-01   2.17e-02    9.36  < 2e-16
```

```
fyear2009    1.54e-01   2.48e-02    6.18  6.4e-10
fyear2010    1.17e-01   2.46e-02    4.75  2.1e-06
fyear2011          NA         NA      NA       NA

Residual standard error: 0.678 on 18690 degrees of freedom
Multiple R-squared: 0.129
F-statistic:  308 on 9 and 18690 DF,  p-value: <2e-16
```

```
anova(m3)
```

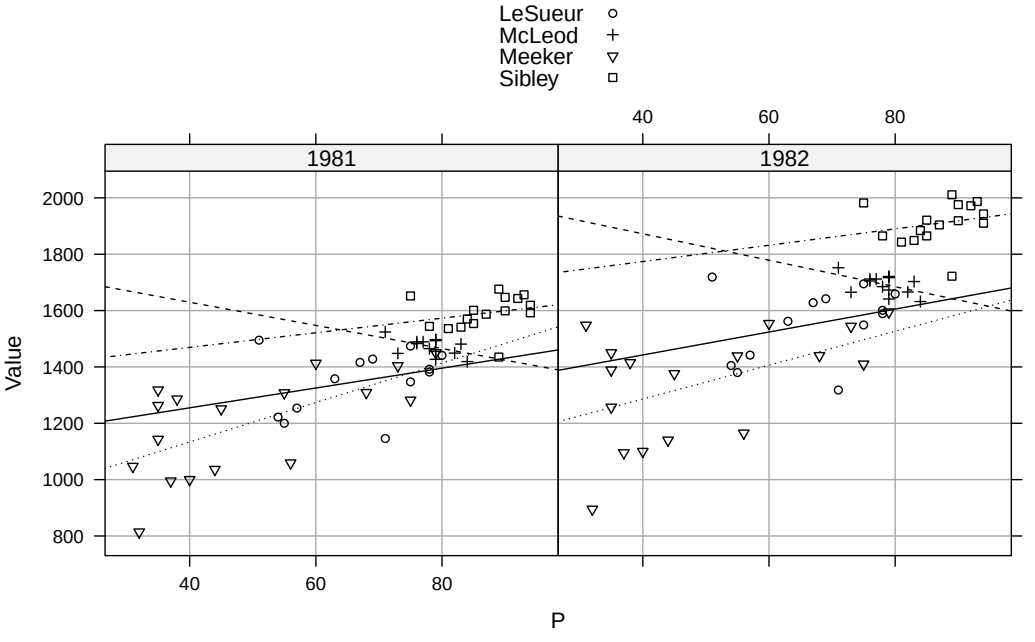
```
Analysis of Variance Table

Response: log(acrePrice)
             Df Sum Sq Mean Sq F value Pr(>F)
year          1   1187    1187  2585.4 <2e-16
fyear         8     88      11    23.9 <2e-16
Residuals 18690   8579       0
```

At least R fits this model without a whimper. One of the regressors for `fyear` is aliased (has an NA in the estimate column) reflecting that `year` is a linear combination of the regressors for `fyear`. Also the F test from `fyear` is identical to the lack-of-fit test computed in the last problem. □

6.17 Solution:

From (6.20) the SD for a randomly assigned treatment indicator is 1/4. Using the calculator at **?**, choose the "Linear regression" platform, and set the number of predictors to 1, the SD of x to .25, the error SD to .5, the detectable beta to 1, and the power to 0.90. The resulting sample size is $n = 44$ or 22 per group. To detect a treatment effect of size 0.5 requires a sample of $n = 170$, while $n = 13$ is adequate to detect a treatment effect of 2.0. □

6.19 Solution:



The figure shows plots of `Value` versus `P` separately for each year, with a separate symbol and regression line for each county. Ignoring counties, the mean functions appear to be straight for each year, with similar scatter for each year. The range of P is very different in each county; for example in McLeod county where P is mostly in the 70s. As a result, the within county regressions are relatively poorly estimated. Thus, we suspect, but are not certain, that the variation between the fitted lines in the graph may be due to very small range in P within county.

Given this preliminary, we turn to models for help. We begin by fitting the largest possible model with all interactions, and then examining the type II ANOVA table.

```
m1 <- lm(Value ~ P * Year * County, data=prodscore)
Anova(m1)
```

```
Anova Table (Type II tests)

Response: Value
               Sum Sq  Df F value  Pr(>F)
P              413397   4    8.70 4.3e-06
Year          1464788   1  123.27 < 2e-16
County         546844   6    7.67 7.8e-07
P:Year            999   1    0.08    0.77
P:County        55189   3    1.55    0.21
Year:County     63417   3    1.78    0.16
P:Year:County    1340   3    0.04    0.99
Residuals     1235843 104
```

None of the interactions seem to be important, so we fit a main effects only model:

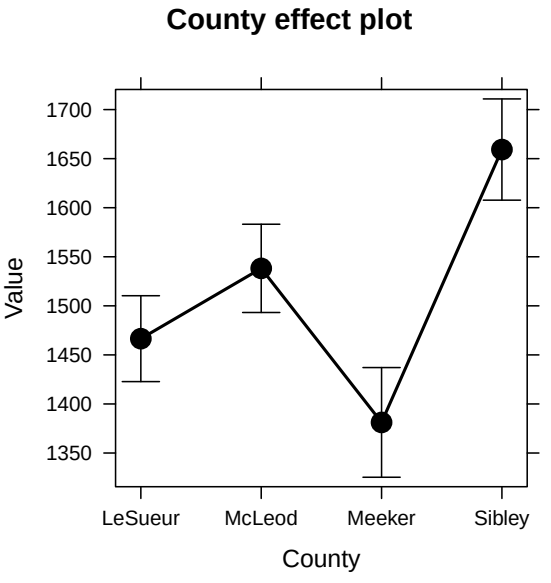
```
summary(m2 <- update(m1, ~ P + Year + County))
```

```
Call:
lm(formula = Value ~ P + Year + County, data = prodscore)

Residuals:
   Min     1Q Median     3Q    Max
-392.8  -53.9   12.2   61.7  265.6

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.37e+05   4.04e+04  -10.80  < 2e-16
P             5.38e+00   1.00e+00    5.36  4.5e-07
```

```
Year          2.21e+02   2.04e+01   10.83  < 2e-16
CountyMcLeod  7.16e+01   3.24e+01    2.21    0.029
CountyMeeker -8.53e+01   3.42e+01   -2.50    0.014
CountySibley  1.93e+02   3.55e+01    5.43  3.2e-07

Residual standard error: 112 on 114 degrees of freedom
Multiple R-squared:  0.805,        Adjusted R-squared:  0.796
F-statistic: 93.8 on 5 and 114 DF,  p-value: <2e-16
```

Each increase in P of 1 point is associated with a \$5.38 increase in assessed value; the increase from 1981 to 1982 was \$221. The country differences are most easily seen with an effects plot:

```
plot(effect("County", m2))
```


County effect plot
Value
1700
1650
1600
1550
1500
1450
1400
1350
LeSueur
McLeod
Meeker
Sibley
County

□

CHAPTER 7

Variances

7.1 Solution:
The estimate of σ^2 is multiplied by 2 in Joe's analysis. All other summaries are unaffected. □

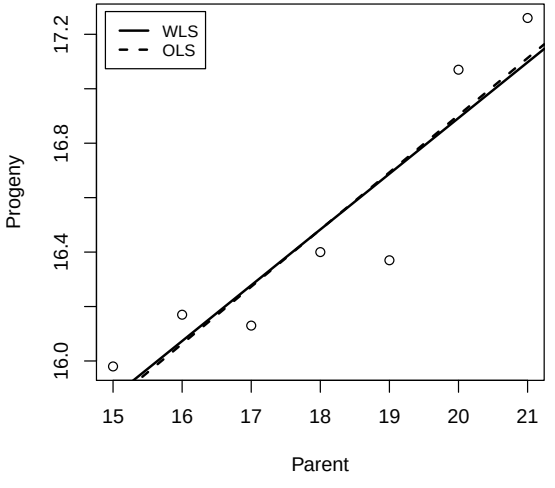
7.3 7.3.1 Solution:
Oversampled means having a higher inclusion probability, and therefore a lower sampling weight. □

7.3.2 Solution:
Each observation represents more of the population, so the weight increases. □

7.5

7.7 7.7.1 Solution:

```
plot(Progeny ~ Parent, galtonpeas)
abline(m.weighted <- lm(Progeny ~ Parent,
   data=galtonpeas, weights= 1/SD^2), lwd=2)
abline(m.unweighted <- lm(Progeny ~ Parent,
   data=galtonpeas), lty=2, lwd=2)
legend("topleft", c("WLS", "OLS"), lty=1:2 , lwd=2,
   cex=.8, inset=.02)
```



□

7.7.2 Solution:

```
compareCoefs(m.weighted, m.unweighted)
```

```
Call:
1:"lm(formula = Progeny ~ Parent, data = galtonpeas, weights = 1/SD^2)"
2:"lm(formula = Progeny ~ Parent, data = galtonpeas)"
             Est. 1    SE 1  Est. 2    SE 2
(Intercept) 12.7964  0.6811 12.7029  0.6993
Parent       0.2048  0.0382  0.2100  0.0386
```

The OLS line is virtually identical to the WLS line. □

7.7.3 Solution:

This should decrease the slope, and it could increase variances, making differences more difficult to detect. □

7.9 7.9.1 Solution:

In R the function `t.test` can be used to get the confidence interval:

```
interval <- t.test(log(UN11$fertility))$conf.int
out <- rbind(interval, exp(interval))
colnames(out) <- c("2.5 %", "97.5 %")
rownames(out) <- c("log(fertility)", "fertilty")
out
```

```
                2.5 % 97.5 %
log(fertility)   0.85 0.9744
fertilty         2.34 2.6497
```

□

7.9.2 Solution:

Depending on your software, this problem may require writing your own computer program using the algorithm outlined in Section 7.7. If you are using R, this is easily done using the `boot` function in the package of the same name. First, write a function that will return the statistic of interest, from the data:

```
get.median <- function(data, indices) {
   median(data[indices])
   }
```

The `boot` function will create the vector of `indices` for each bootstrap sample. The statement `median(data[indices])` computes the median of the bootstrap sample of `data`, where `data` is the original data for which you want the median. The call to `boot` to get 999 bootstrap samples is[1]

```
library(boot)
set.seed(12345)
b1 <- boot(UN11$fertility, get.median, R=999)
boot.ci(b1, type=c("norm", "perc", "bca"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 999 bootstrap replicates

CALL :
boot.ci(boot.out = b1, type = c("norm", "perc", "bca"))

Intervals :
Level      Normal              Percentile            BCa
95%   ( 2.088,  2.412 )   ( 2.146,  2.430 )   ( 2.142,  2.410 )
Calculations and Intervals on Original Scale
```

The function `boot.ci` returns confidence intervals at 95% by default. Intervals can be computed in several ways for the bootstrap samples: `norm` assumes the bootstraped values are approximately normal and independent and computes the interval using the bootstrap mean plus or minus 1.96 times the bootstrap standard error; `perc` uses the percentile bootstrap described in Section 7.7 and `bca` uses the bias corrected and accelerated (BCa) method also mentioned. All three give

[1] In the text the number of bootstrap replications is B, but in the `boot` function it is R. The change was made in the text to avoid confusion between the number of bootstrap replications and the multiple correlation coefficient.

different intervals and the BCa method is generally more accurate and preferred. The intervals from the three methods overlap, with the BCa method producing the narrowest interval. The interval computed in Problem 7.9.1 includes values well above the upper endpoint of the BCa method, reflecting the bias inherent in exponentiating the endpoints of an interval. □

7.11 Solution:

This is likely to be a very difficult problem for most students. Differentiate with respect to both X_1 and X_2:

$$\begin{aligned}\frac{d\mathrm{E}(Y|X)}{dX_1} &= \beta_1 + 2\beta_3 X_1 + \beta_5 X_2 \\ \frac{d\mathrm{E}(Y|X)}{dX_2} &= \beta_2 + 2\beta_4 X_2 + \beta_5 X_1\end{aligned}$$

Set the two derivatives equal to zero, and then solve for X_1 and X_2,

$$\begin{aligned}\tilde{X}_1 &= \frac{\beta_2\beta_5 - 2\beta_1\beta_4}{4\beta_3\beta_4 - \beta_5^2} \\ \tilde{X}_2 &= \frac{\beta_1\beta_5 - 2\beta_2\beta_3}{4\beta_3\beta_4 - \beta_5^2}\end{aligned}$$

We can now use the delta method to get estimates and standard errors (using R), replacing the βs by their OLS estimates:

```
m1 <- lm(Y ~ X1 + X2 + I(X1^2) + I(X2^2) + X1:X2, data=cakes)
```

The names of the coefficients of this fit are hard to type, so we will use the `parameterNames` argument to `deltaMethod` to use simpler names:

```
param.names <- paste("b", 0:5, sep="")
x1.max <- "(b2*b5-2*b1*b4)/(4*b3*b4-b5^2)"
deltaMethod(m1, x1.max,
     parameterNames=param.names)
```

```
                                  Estimate     SE
(b2*b5-2*b1*b4)/(4*b3*b4-b5^2)    35.83 0.4331
```

Then, repeat for X_2:

```
x2.max <- "(b1*b5-2*b2*b3)/(4*b3*b4-b5^2)"
deltaMethod(m1, x2.max,
     parameterNames=param.names)
```

```
                                  Estimate    SE
(b1*b5-2*b2*b3)/(4*b3*b4-b5^2)    352.6 1.203
```

□

7.13 Solution:

```
m1 <- lm(time  ~  t1  + t2, Transact)
deltaMethod(m1, "t1/t2")
```

```
      Estimate    SE
t1/t2    2.685 0.319
```

```
b1 <- bootCase(m1, coef, B=999)
set.seed(24)
data.frame(mean=mean(b1[, 2]/b1[, 3]), sd=sd(b1[, 2]/b1[, 3]))
```

```
   mean     sd
1 2.737 0.5214
```

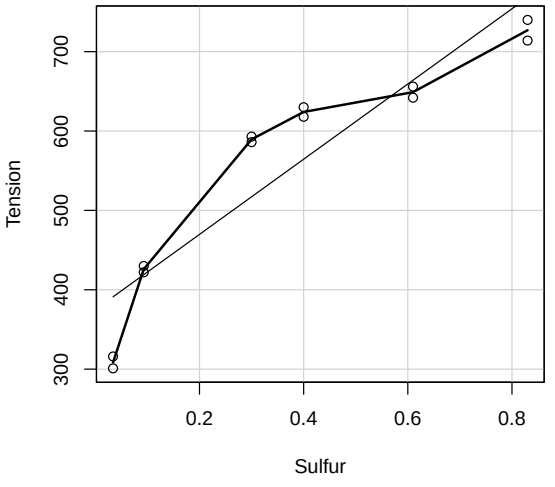
While the means agree reasonably closely, the standard deviation computed by the deltaMethod is about 40% too small, so confidence intervals computed from the deltaMethod will be too short. □

CHAPTER 8

Transformations

8.1 8.1.1 Solution:

```
scatterplot(Tension ~ Sulfur, baeskel,
            smooth=TRUE, boxplots=FALSE, spread=FALSE)
```

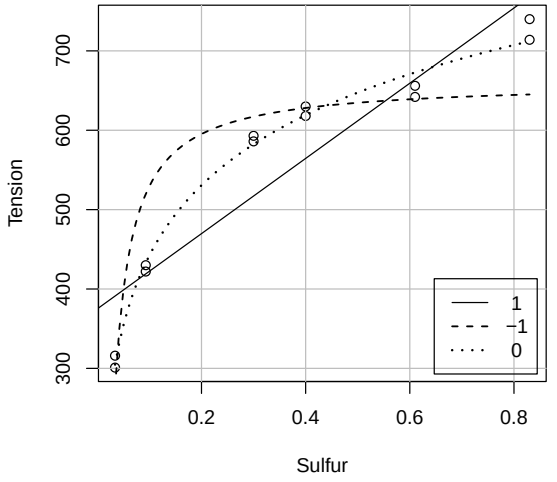


The points appear to fall on a curve, not a straight line. □

8.1.2 Solution:

```
plot(Tension ~ Sulfur, baeskel)
grid(col="gray", lty="solid")
m1 <- lm(Tension ~ Sulfur, baeskel)
abline(m1, lwd=1)
new <- with(baeskel, seq(min(Sulfur), max(Sulfur), length=100))
m2 <- update(m1, ~ I(1/Sulfur))
with(baeskel, lines(new, predict(m2, data.frame(Sulfur=new)), lwd=1.5, lty=2))
```

```
m2 <- update(m2, ~ log(Sulfur))
with(baeskel, lines(new, predict(m2, data.frame(Sulfur=new)), lwd=2, lty=3))
legend("bottomright", c(" 1", "-1", " 0"), lwd=c(1, 1.5, 2), lty=1:3, inset=0.02)
```

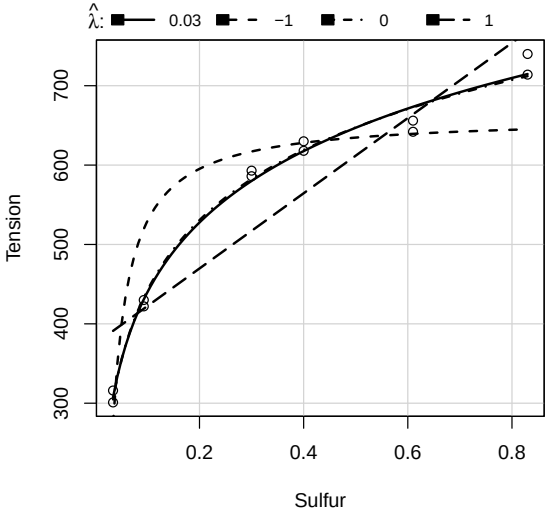


From the above figure, only the log transformation closely matches the data. The `invTranPlot` function in the `car` package in Rautomates this plot, and in addition shows the `RSS` for each λ and displays $\widehat{\lambda}$:

```
with(baeskel, invTranPlot(Sulfur, Tension))
```

```
    lambda   RSS
1  0.03442  2484
2 -1.00000 35692
```

```
3  0.00000  2536
4  1.00000 35824
```



□

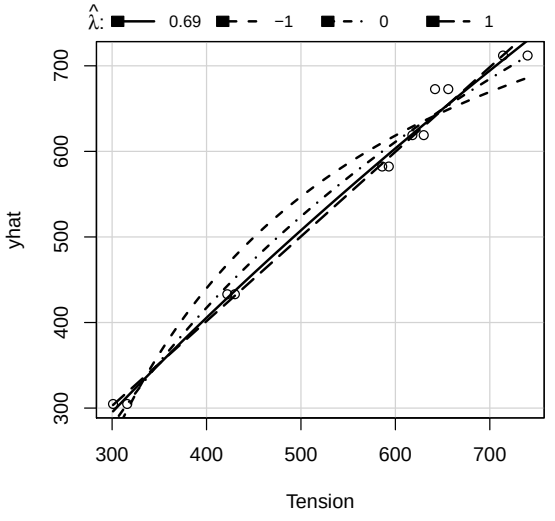
8.1.3 Solution:

`invResPlot` is a convenience function that calls `invTranPlot`.

```
invResPlot(lm(Tension ~ log(Sulfur), baeskel))
```

```
   lambda   RSS
1  0.6861  2202
2 -1.0000 10594
3  0.0000  3658
```

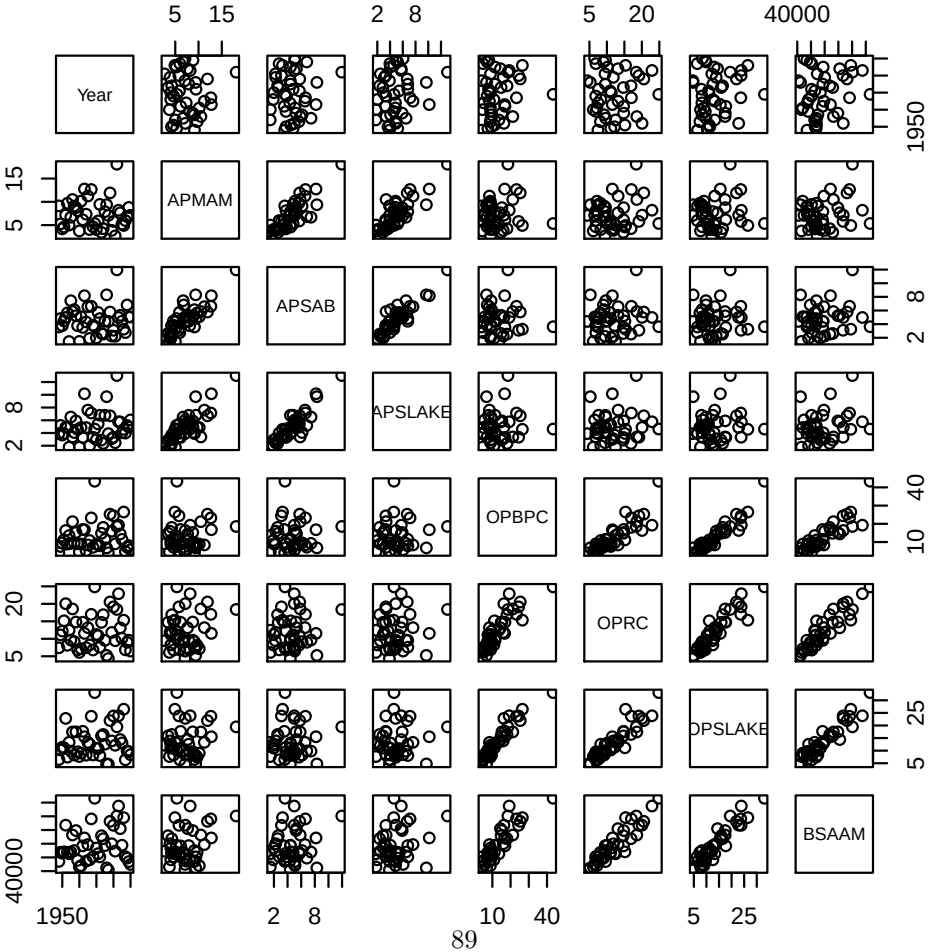
```
4  1.0000  2510
```



Untransformed, $\lambda = 1$, matches well, suggesting no further need to transform. □

8.3 Solution:

```
pairs(water)
```


Year
APMAM
APSAB
APSLAKE
OPBPC
OPRC
OPSLAKE
BSAAM
1950
5 15
2 8
5 20
40000
10 40
5 25

(1) The "O" measurements are very highly correlated, but the "A" measurements are less highly correlated, and correlations between the `O` and `A` variables are small; (2) there is no apparent time trend in the predictors; (3) at least marginally the `O` variables appear to be more highly correlated with the response than are the `A` variables. □

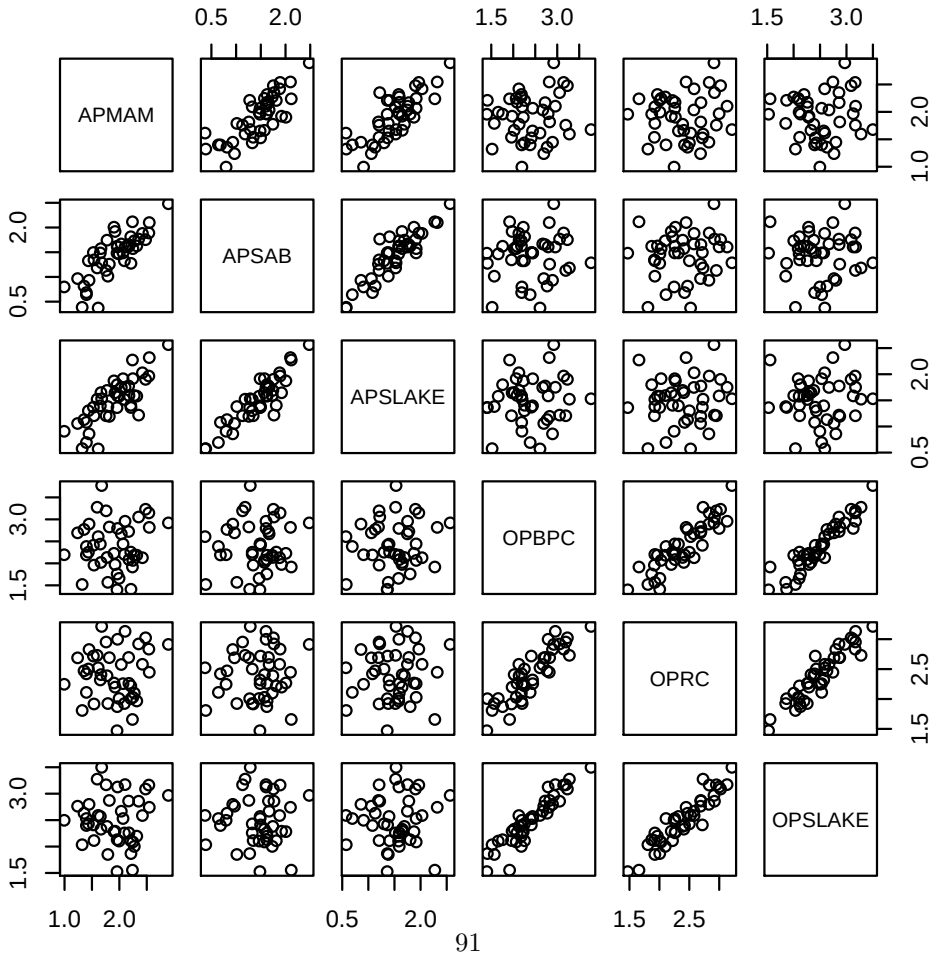
Solution:
Code for the automatic choice of a transformation is available in at least two sources: in the program Arc described by **?**, and in the `car` package for R (**?**). Using R,

```
summary(ans <- powerTransform( as.matrix(water[ , 2:7]) ~ 1))
```

```
bcPower Transformations to Multinormality

        Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
APMAM      0.0982   0.2861          -0.4625           0.6589
APSAB      0.3450   0.2032          -0.0533           0.7432
APSLAKE    0.0818   0.2185          -0.3466           0.5101
OPBPC      0.0982   0.1577          -0.2109           0.4073
OPRC       0.2536   0.2445          -0.2255           0.7328
OPSLAKE    0.2534   0.1763          -0.0921           0.5988

Likelihood ratio tests about transformation parameters
                                    LRT df      pval
LR test, lambda = (0 0 0 0 0 0)  5.453  6 4.872e-01
LR test, lambda = (1 1 1 1 1 1) 61.203  6 2.563e-11
```

The indication is to transform all the predictors to log scale, since the p-value for the LR test is about .49.

```
pairs(log(water[ , 2:7]))
```


APMAM
APSAB
APSLAKE
OPBPC
OPRC
OPSLAKE
0.5
2.0
1.5
3.0
1.0
2.5

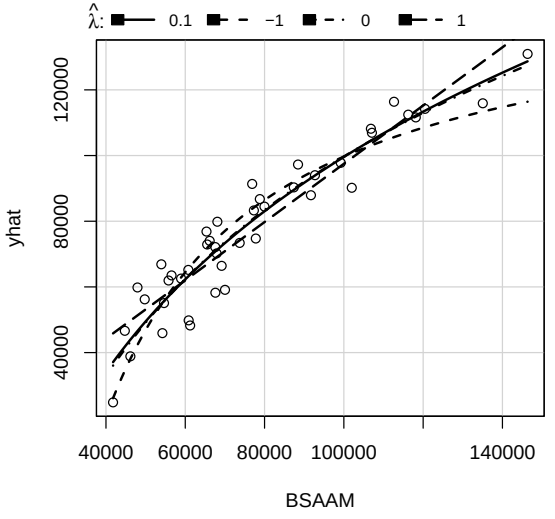
□

8.3.12 Solution:

Either the Box–Cox method or the inverse response plot method will indicate that the log transformation matches the data. Here is the inverse response plot:

```
m4 <- lm(BSAAM ~ log(APMAM) + log(APSAB) + log(APSLAKE) +
                 log(OPBPC) + log(OPRC) + log(OPSLAKE), water)
invResPlot(m4)
```

```
   lambda       RSS
1  0.1048 2.257e+09
2 -1.0000 3.009e+09
3  0.0000 2.264e+09
4  1.0000 2.745e+09
```



The nonlinear LS estimate of $\widehat{\lambda} = .10$, and the fitted line matches the fitted line for $\lambda = 0$ almost perfectly. Log scale is nearly identical and is indicated. □

8.3.3 Solution:

```
m5 <- update(m4, log(BSAAM) ~ .)
summary(m5)
```

```
Call:
lm(formula = log(BSAAM) ~ log(APMAM) + log(APSAB) + log(APSLAKE) +
    log(OPBPC) + log(OPRC) + log(OPSLAKE), data = water)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.18671 -0.05264 -0.00693  0.06130  0.17698

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.4667     0.1235   76.63   <2e-16
log(APMAM)    -0.0203     0.0660   -0.31   0.7597
log(APSAB)    -0.1030     0.0894   -1.15   0.2567
log(APSLAKE)   0.2206     0.0896    2.46   0.0187
log(OPBPC)     0.1113     0.0817    1.36   0.1813
log(OPRC)      0.3616     0.1093    3.31   0.0021
log(OPSLAKE)   0.1861     0.1314    1.42   0.1652

Residual standard error: 0.102 on 36 degrees of freedom
Multiple R-squared:  0.91,        Adjusted R-squared:  0.895
F-statistic: 60.5 on 6 and 36 DF,  p-value: <2e-16
```

The negative coefficients are for two of the (nonsignificant) `A` regressors. The negative signs are due to correlations with other regressors already included in the mean function. □

8.3.4 Solution:

For the first test, fit two models, one with six regressors plus the intercept, the other replacing the logarithms of the "O" regressors by their sum.

```
water$logOsum <- rowSums(log(water[, 5:7]))
m6 <- lm(log(BSAAM) ~ log(APMAM) + log(APSAB) + log(APSLAKE) +
          logOsum, water)
anova(m6, m5)
```

```
Analysis of Variance Table

Model 1: log(BSAAM) ~ log(APMAM) + log(APSAB) + log(APSLAKE) + logOsum
```

```
Model 2: log(BSAAM) ~ log(APMAM) + log(APSAB) + log(APSLAKE) + log(OPBPC) +
    log(OPRC) + log(OPSLAKE)
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1     38 0.405
2     36 0.372  2    0.0329 1.59   0.22
```

Repeat for the `A` regressors:

```
water$logAsum <- rowSums(log(water[, 2:4]))
m7 <- lm(log(BSAAM) ~ log(OPBPC) + log(OPRC) + log(OPSLAKE) +
         logAsum, water)
anova(m7, m5)
```

```
Analysis of Variance Table

Model 1: log(BSAAM) ~ log(OPBPC) + log(OPRC) + log(OPSLAKE) + logAsum
Model 2: log(BSAAM) ~ log(APMAM) + log(APSAB) + log(APSLAKE) + log(OPBPC) +
    log(OPRC) + log(OPSLAKE)
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1     38 0.419
2     36 0.372  2    0.0468 2.26   0.12
```

Both tests suggest the sum is as good as the individual measurements. This suggests the average snow depth represents its valley as well as do the individual measurements.

These tests can also be done using the `linearHypothesis` method in R to get a Wald test. For the hypothesis concerning the `O` variables,

```
(mat <- rbind(c(0, 0, 0, 0, 1, -1, 0), c(0, 0, 0, 0, 1, 0, -1)))
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    0    0    0    0    1   -1    0
[2,]    0    0    0    0    1    0   -1
```

```
linearHypothesis(m5, mat)
```

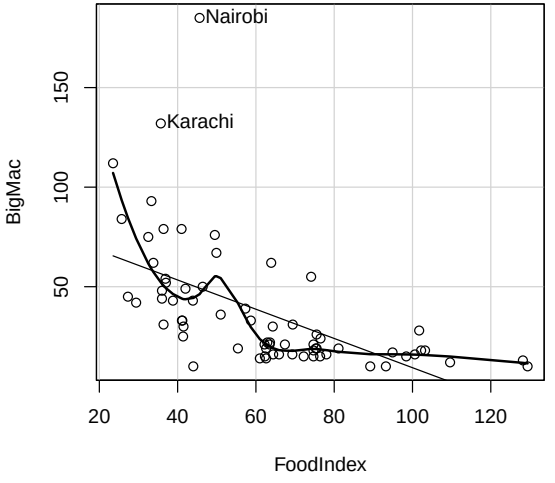
```
Linear hypothesis test

Hypothesis:
log(OPBPC) - log(OPRC) = 0
log(OPBPC) - log(OPSLAKE) = 0

Model 1: restricted model
Model 2: log(BSAAM) ~ log(APMAM) + log(APSAB) + log(APSLAKE) + log(OPBPC) +
    log(OPRC) + log(OPSLAKE)

  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1     38 0.405
2     36 0.372  2    0.0329 1.59   0.22
```

The hypothesis is specified by the matrix `mat`. The first row of this matrix specifies that the coefficient for the fifth regressor log(`OPBPC`) is equal the coefficient for the sixth regressor log(`OPRC`), and the second row specifies the fifth regressor is equal to the regressor for log(`OPSLAKE`), effectively specifying all three coefficients are equal. □

8.5 8.5.1 Solution:

```
scatterplot(BigMac ~ FoodIndex, BigMac2003, boxplots=FALSE,
     spread=FALSE, id.n=2)
```

```
Karachi Nairobi
     26      46
```



The scatterplot indicates that the real cost of a Big Mac, the amount of work required to buy one, declines with overall food prices; the Big Mac is cheapest, for the local people, in countries with high `FoodIndex`. The cost in Nairobi and Karachi were relatively very high. In Nairobi 185 minutes of labor are required by the typical worker to buy a Big Mac. □

8.5.2 Solution:

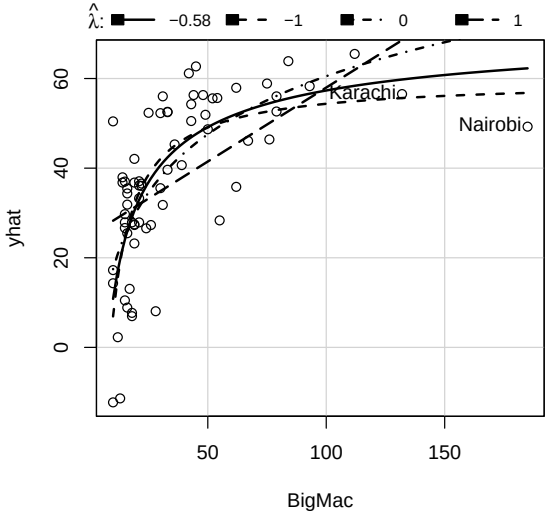
```
m1 <- lm(BigMac ~ FoodIndex, BigMac2003)
summary(powerTransform(m1))
```

```
bcPower Transformation to Normality
```

```
   Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
Y1   -0.4471   0.1534          -0.7478          -0.1463

Likelihood ratio tests about transformation parameters
                         LRT df     pval
LR test, lambda = (0)  8.853  1 0.002925
LR test, lambda = (1) 93.463  1 0.000000
```

```
invResPlot(m1, id.n=2, id.method="x")
```

```
   lambda   RSS
1 -0.5841 10252
2 -1.0000 10528
3  0.0000 10907
4  1.0000 14846
```



Both methods suggest using the inverse square root scale for `BigMac`, although the improvement over the logarithmic transformation is small. □

8.5.3 Solution:

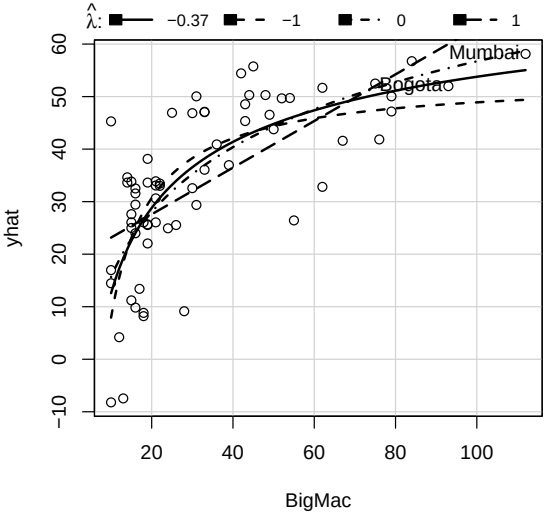
```
sel <- match(c("Karachi", "Nairobi"), rownames(BigMac2003))
m2 <- update(m1, subset=-sel)
invResPlot(m2, id.n=2, id.method="x")
```

```
   lambda  RSS
1 -0.3671 7272
2 -1.0000 7617
```

```
3  0.0000 7395
4  1.0000 8754
```

```
summary(powerTransform(m2))
```

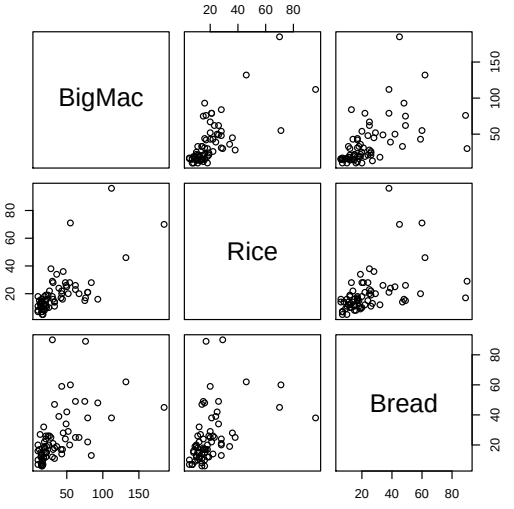
```
bcPower Transformation to Normality

   Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
Y1   -0.3342   0.1774           -0.682           0.0136

Likelihood ratio tests about transformation parameters
                          LRT df      pval
LR test, lambda = (0)  3.564  1 5.904e-02
LR test, lambda = (1) 54.013  1 1.992e-13
```



Although the estimated power is now close to the inverse cube root, both methods suggest little difference between the best estimate and logarithms. Logs are much easier to interpret and should be used in this problem. □

8.5.4 Solution:

```
pairs(~ BigMac + Rice + Bread, BigMac2003)
```



The scatterplot matrix indicates the need to transform because the points are clustered on the lower-left corners of the plots, the variables range over several orders of magnitude, and curvature is apparent. The results of the multivariate Box–Cox procedure are

```
summary(pows <- powerTransform(cbind(BigMac, Rice, Bread) ~ 1, BigMac2003))
```

```
bcPower Transformations to Multinormality

       Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
BigMac   -0.3035   0.1503          -0.5980          -0.0089
Rice     -0.2406   0.1345          -0.5043           0.0230
Bread    -0.1566   0.1466          -0.4439           0.1307
```

```
Likelihood ratio tests about transformation parameters
                                LRT df    pval
LR test, lambda = (0 0 0)     7.683  3 0.05303
LR test, lambda = (1 1 1)   204.556  3 0.00000
LR test, lambda = (-0.5 0 0)  6.605  3 0.08560
```

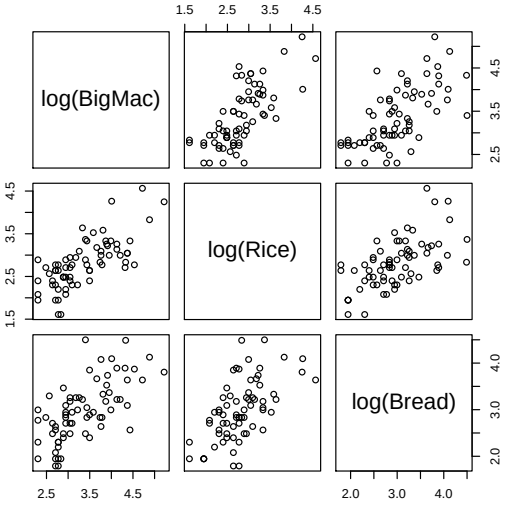
Removing two cases:

```
summary(pow1s<-powerTransform(cbind(BigMac, Rice, Bread) ~ 1, BigMac2003,
    subset=-c(26, 46)))
```

```
bcPower Transformations to Multinormality

       Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
BigMac   -0.2886   0.1742          -0.6301           0.0529
Rice     -0.2465   0.1413          -0.5235           0.0305
Bread    -0.1968   0.1507          -0.4922           0.0986

Likelihood ratio tests about transformation parameters
                             LRT df    pval
LR test, lambda = (0 0 0)   7.084  3 0.06927
LR test, lambda = (1 1 1) 181.891  3 0.00000
```

The resulting transformations are not very different from the transformations using all the data, and logs of all three seem to be appropriate. The scatterplot matrix for the transformed variables is

```
pairs(~ log(BigMac) + log(Rice) + log(Bread), BigMac2003)
```

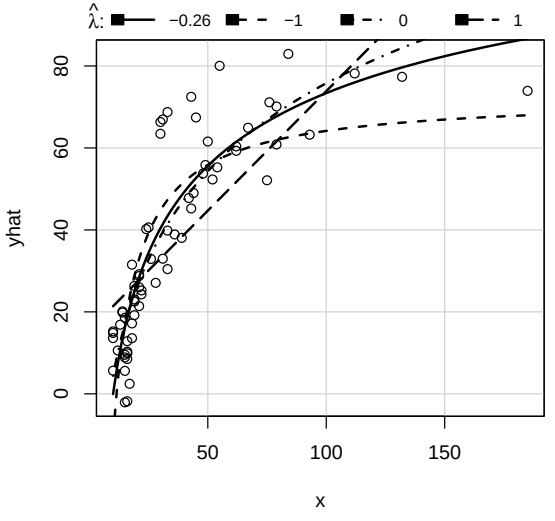


□

8.5.5 Solution:

```
m2 <- lm(BigMac ~ I(Apt^(1/3)) + log(Bread) + log(Bus) +
        log(TeachGI), BigMac2003)
invResPlot(m2, xlab="(a) BigMac")
```

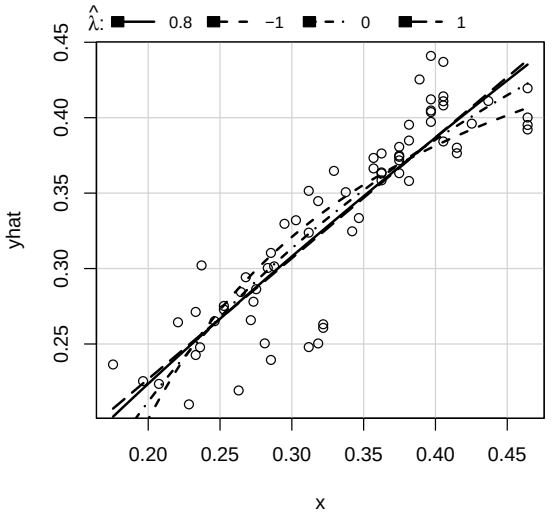
```
   lambda   RSS
1 -0.2569  7864
2 -1.0000 10341
3  0.0000  8228
4  1.0000 16321
```



This suggest a negative cube root, or perhaps a logarithm.

```
m3 <- update(m2, I(BigMac^(-1/3)) ~ .)
invResPlot(m3, xlab=expression(paste("(b) ", BigMac^(-1/3))))
```

```
   lambda     RSS
1  0.7998 0.05646
2 -1.0000 0.07123
3  0.0000 0.05931
4  1.0000 0.05663
```



No further transformation seems necessary as the inverse response plot is nearly linear. □

8.7 Solution:

The range for `Miles` is from 1,534 to 300,767, and according to the log rule, transformation of `Miles` to log scale is justified as a starting point because the range is about two orders of magnitude. We can see if further transformation is desirable using the multivariate Box–Cox method:

```
fuel2001$Dlic <- 1000 * fuel2001$Drivers/fuel2001$Pop
summary(b1 <- powerTransform(cbind(Tax, Dlic, Income,
         logMiles=log(Miles)) ~ 1, data=fuel2001))

bcPower Transformations to Multinormality
```

```
          Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
Tax          1.8493   0.4803           0.9079            2.791
Dlic         2.2669   1.3671          -0.4127            4.946
Income      -0.5105   0.8432          -2.1632            1.142
logMiles     6.4715   1.4063           3.7151            9.228

Likelihood ratio tests about transformation parameters
                                    LRT df      pval
LR test, lambda = (0 0 0 0)      47.182  4 1.398e-09
LR test, lambda = (1 1 1 1)      25.420  4 4.142e-05
LR test, lambda = (1 1 1 6.47)    7.682  4 1.040e-01
```

The suggested transformation parameter for log(`Miles`) is well outside the usual range of -2 to 2, and so we would conclude that no further transformation is needed.

If you start with the Box–Cox method before replacing `Miles` with log(`Miles`), a square root transformation is suggested as better than the logarithmic. However, changes in scale for the predictors are less important than changes in scale for the response, and there is little difference between using these two transformations. The logarithmic is preferred because it is easier to interpret. □

CHAPTER 9

Regression Diagnostics

9.1 9.1.1 Solution:

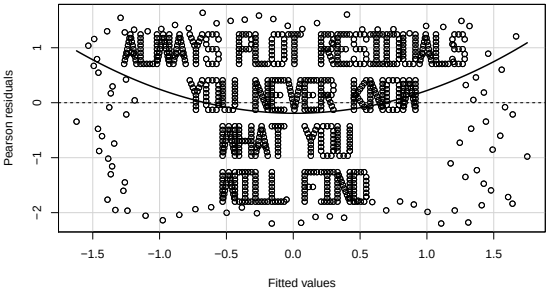
Nothing strange. □

9.1.2 Solution:

Nothing strange. □

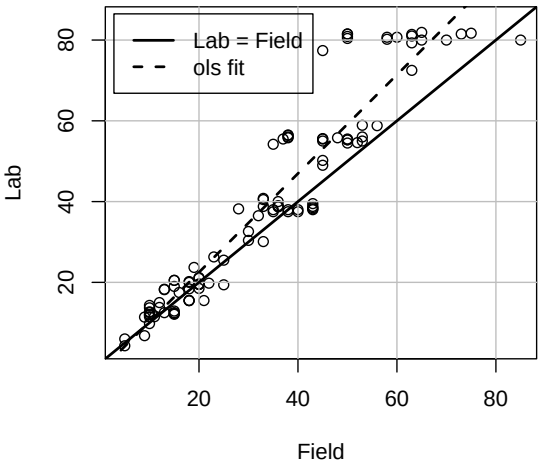
9.1.3 Solution:

```
residualPlot(lm(y ~ ., Rpdata))
```



□

9.3 9.3.1 Solution:



If on the average `Lab` and `Life` measured the same quantity, the 45° line, shown as a solid line, should match the data. Most of the points are above this line. The dashed OLS line; it appears that the field measurement underestimates depth for the deeper faults. □

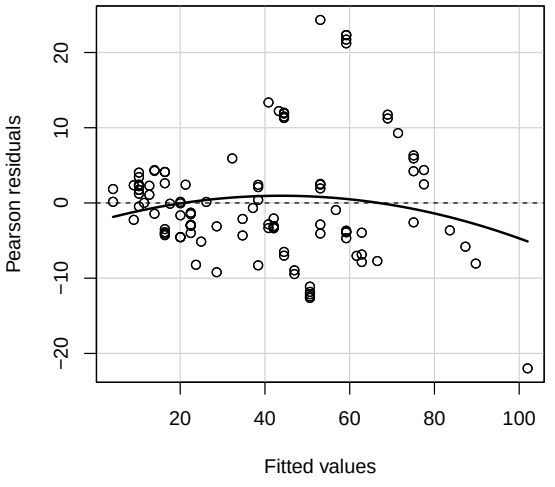
9.3.2 Solution:

Here is the computer output for this problem using the `car` package R::

```
summary(m1 <- lm(Lab ~ Field, pipeline))

Call:
lm(formula = Lab ~ Field, data = pipeline)

Residuals:
```

```
   Min     1Q Median     3Q    Max
-21.98  -4.07  -1.43   2.50  24.33

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.9675     1.5748   -1.25     0.21
Field         1.2230     0.0411   29.78   <2e-16

Residual standard error: 7.86 on 105 degrees of freedom
Multiple R-squared:  0.894,        Adjusted R-squared:  0.893
F-statistic:  887 on 1 and 105 DF,  p-value: <2e-16
```

```
residualPlot(m1)
```



The fitted model has $R^2 = 0.89$. The slope estimate $\widehat{\beta}_1 = 1.22$ is considerably larger than 1, suggesting that the `Field` measurements underestimate `Lab` measurements, particularly for larger `Field` measurements. The residual plot suggests nonconstant variance as the larger residuals are most at the right-end of the plot.

```
ncvTest(m1)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 29.59    Df = 1     p = 5.35e-08
```

The score test for variance as a function of `Field` is $S = 29.59$ with 1 *df*, for a very small p-value. The conclusion is that variance increases with `Field`; deeper faults are less well measured. □

9.3.3 Solution:

```
set.seed(1234)
b1 <- Boot(m1)
d1 <- deltaMethod(m1, "Field", vcov = hccm)
print(out <- rbind(ols = summary(m1)$coef[2, 1:2],
            bootstrap = summary(b1)[2, c(2, 4), drop=TRUE],
            wls = summary(update(m1, weights=1/Field))$coef[2, 1:2],
            hcorrected = c(d1$Estimate, d1$SE)
            ), digits=5)
```

```
           Estimate Std. Error
ols        1.223    0.041069
bootstrap  1.223    0.043883
wls        1.2118   0.035265
hcorrected 1.223    0.047506
```

The slope estimate is the same for all methods except WLS where the difference is small. The bootstrap and OLS standard errors are nearly the same, while the WLS estimate is 20% smaller, so ignoring the weights underestimates precision. The corrected standard error is about 10% larger than the OLS standard error. In this example the correction seems to be in the wrong direction. □

9.5 Solution:

The QR factorization is $\mathbf{X} = \mathbf{QR}$ where $\mathbf{Q}$ is $n \times p'$ with orthogonal columns and $\mathbf{R}$ is a $p' \times p'$ upper triangular matrix. Then $(\mathbf{X}'\mathbf{X})^{-1} = (\mathbf{R}'\mathbf{Q}'\mathbf{QR})^{-1} = (\mathbf{R}'\mathbf{R})^{-1} = \mathbf{R}^{-1}(\mathbf{R}')^{-1}$, and so

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \mathbf{QRR}^{-1}(\mathbf{R}')^{-1}\mathbf{R}'\mathbf{Q}' = \mathbf{QQ}'$$

This is what we set out to prove. □

9.7 Solution:

$$\mathbf{H}(\mathbf{I} - \mathbf{H}) = \mathbf{H} - \mathbf{H}^2 = \mathbf{H} - \mathbf{H} = \mathbf{0}$$

so $\mathbf{H}$ and $\mathbf{I}-\mathbf{H}$ are orthogonal. The numerator of the OLS slope in the simple regression of $\widehat{\mathbf{e}}$ on $\widehat{\mathbf{Y}}$ is $(\widehat{\mathbf{e}}-\bar{\widehat{e}}\mathbf{1})'(\widehat{\mathbf{Y}}-\bar{\widehat{\mathbf{Y}}}\mathbf{1})$, where $\mathbf{1}$ is a column of 1s. As long as the intercept is in the mean function, $\bar{\widehat{e}}=0$, and the numerator reduces to $\widehat{\mathbf{e}}'\widehat{\mathbf{Y}}=\mathbf{Y}(\mathbf{I}-\mathbf{H})\mathbf{H}\mathbf{Y}=0$.

The slope of the regression of $\widehat{\mathbf{e}}$ on $\mathbf{Y}$ is $\mathbf{Y}'(\mathbf{I}-\mathbf{H})\mathbf{Y}/(\mathbf{Y}-\bar{y}\mathbf{1})'(\mathbf{Y}-\bar{y}\mathbf{1})=$ RSS/SYY $=1-R^2$. □

9.9 Solution:

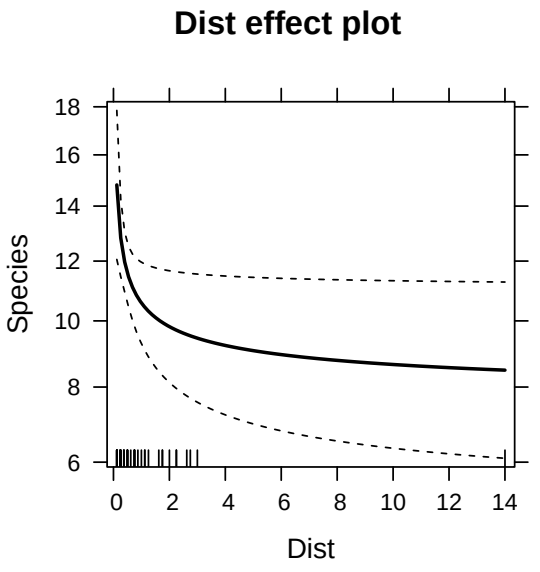
The predictors should mostly be transformed, using logs of everything except `Photo`, `Dist`, `Long` and `Lat` (I added 2 to `Elev` because one lake had elevation -1). I transformed `Photo` to $\texttt{Photo}^{-.33}$ and `Dist` to $\text{Dist}^{.33}$. Transforming `Long` and `Lat` doesn't make much sense. The Box–Cox method does not suggest further transforming the response.

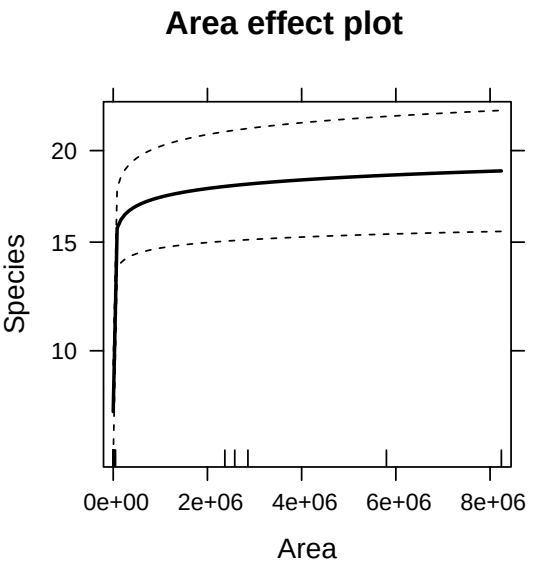
Only $\texttt{Dist}^{-.33}$, log(`Area`), log(`NLakes`), and $\texttt{Photo}^{-.33}$ appear to be important. There is some nonconstant variance; the score test has p-value of about 0.04. One might expect nonconstant variance because the response is a count. One approach at this point is to use Poisson regression, as will be pursued in Chapter 12. Another alternative is to use a variance stabilizing transformation, probably the square root. The concern is that stabilizing variance may destroy linearity of the mean function. We fit in both the untransformed scale and in square root scale. Using marginal model plots, both seem to match the data equally well, but the square root scale also seems to have reasonably constant variance, since the p-value for the score test is about 0.67. The residual plots appear to be a little better in square root scale as well. The regression summary is

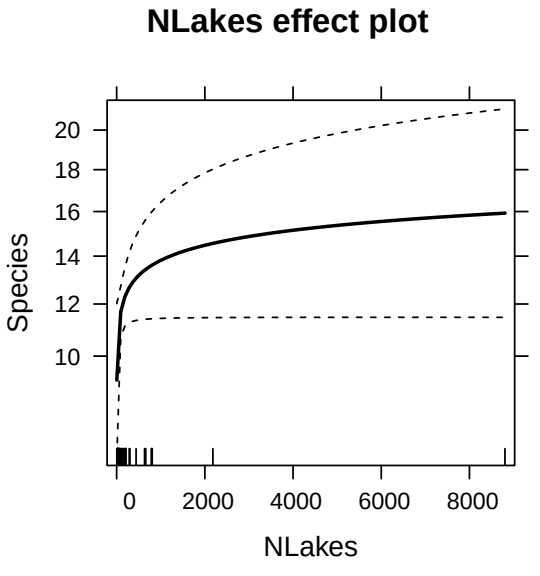
```
summary(m5 <- lm(sqrt(Species) ~ I(Dist^(-1/3)) + log(Area) +
         log(NLakes) + I(Photo^(1/3)), lakes))

Call:
lm(formula = sqrt(Species) ~ I(Dist^(-1/3)) + log(Area) + log(NLakes) +
    I(Photo^(1/3)), data = lakes)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0774 -0.3218  0.0662  0.3581  0.9580
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.1234     0.3224    3.48  0.00117
I(Dist^(-1/3))   0.5797     0.2349    2.47  0.01774
log(Area)        0.0815     0.0162    5.02  9.9e-06
log(NLakes)      0.1258     0.0646    1.95  0.05824
I(Photo^(1/3))   0.0995     0.0257    3.87  0.00037

Residual standard error: 0.505 on 42 degrees of freedom
  (22 observations deleted due to missingness)
Multiple R-squared:  0.72,        Adjusted R-squared:  0.693
F-statistic:   27 on 4 and 42 DF,  p-value: 4.05e-11
```

```
library(effects)
plot(allEffects(m5, default.levels=100,
  transformation=list(inverse=function(x) x^2)), ylab="Species")
```


Dist effect plot
Species
Dist
18
16
14
12
10
8
6
0
2
4
6
8
10
12
14


Area effect plot
Species
Area
20
15
10
0e+00
2e+06
4e+06
6e+06
8e+06


NLakes effect plot
Species
NLakes
20
18
16
14
12
10
0
2000
4000
6000
8000

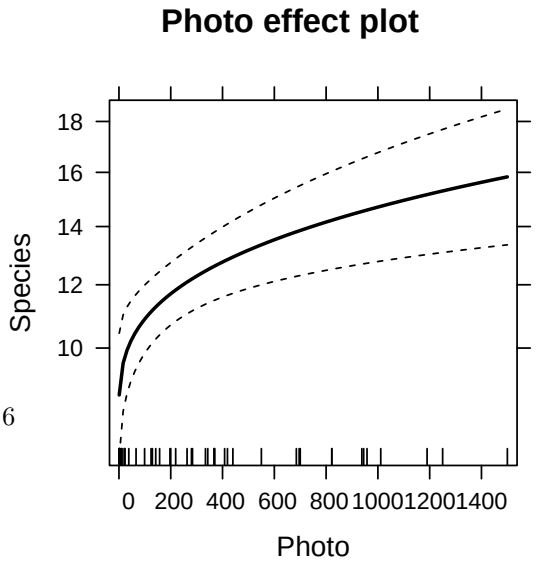


Photo effect plot
Species
Photo
18
16
14
12
10
0
200
400
600
800
1000
1200
1400

The variable `Photo` is missing for about one third of the lakes, so one might want to examine models that ignore `Photo`. The analysis given is reasonable if a missing at random assumption is tenable here; we don't really have enough information to decide if it is tenable or not. □

9.11 Solution:

```
                    y     ehat      r      t     h     D
Alaska          514.279 -163.145 -2.915 -3.193 0.256 0.585
New_York        374.164 -137.599 -2.317 -2.438 0.162 0.208
Hawaii          426.349 -102.409 -1.771 -1.814 0.206 0.162
Wyoming         842.792  183.499  2.954  3.246 0.084 0.160
Dist._of_Col.   317.492  -49.452 -0.996 -0.996 0.415 0.141
```

The largest outlier test is 3.246, and the Bonferroni p-values are, for all five states,

```
> pmin(51*2*pt(-abs(out$Ti[subset]),46),1)
       Alaska      New_York        Hawaii       Wyoming Dist._of_Col.
      0.12958       0.95272       1.00000       0.11145       1.00000
```

None would be declared outliers. Alaska has the largest influence on the regression. □

9.13 Solution:

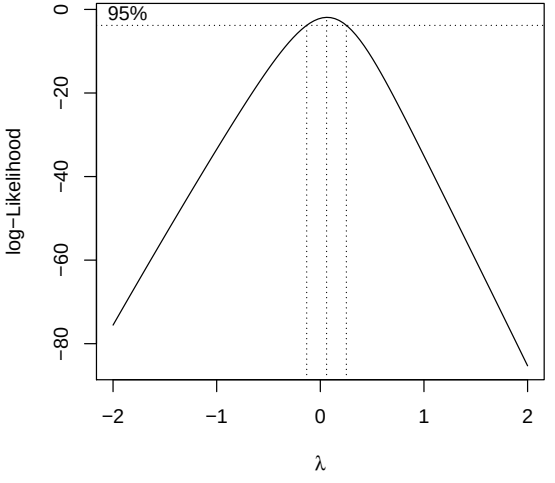
Using the appendix,

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_{(i)} &= \widehat{\boldsymbol{\beta}} - \frac{(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i\widehat{e}_i}{1-h_{ii}} \\
y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}_{(i)} &= y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}} + \frac{\mathbf{x}_i'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i\widehat{e}_i}{1-h_{ii}} \\
&= \widehat{e}_i + \frac{h_{ii}}{1-h_{ii}}\widehat{e}_i \\
&= \frac{\widehat{e}_i}{1-h_{ii}}
\end{aligned}$$

□

9.15 9.15.1 Solution:

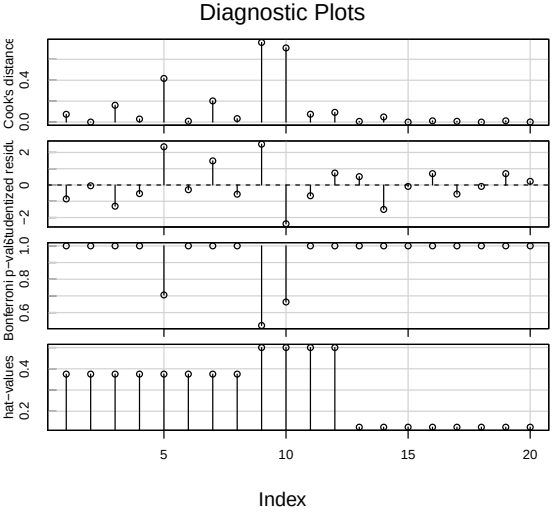
```
m1 <- lm(Life ~ poly(Speed, 2) + poly(Feed, 2) + Speed:Feed, lathe1)
boxcox(m1)
```



This is the graph of the profile log-likelihood for the transformation parameter using the Box–Cox method for the second-order lathe model. The confidence interval for λ is very narrow and includes zero, suggesting a log transformation. □

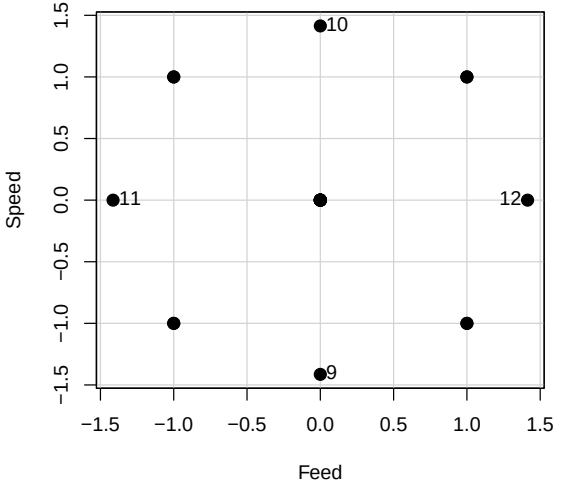
9.15.2 Solution:

```
m2 <- update(m1, log(Life) ~ .)
influenceIndexPlot(m2)
```



Observation 9–12 have large leverages by design in this planned experiment. These were "star points", unreplicated observations at relatively extreme values of the predictors to model curvature. Two of these, 9 and 10, also have large residuals and these in combination give large values for Cook's Distance as well.

```
scatterplot(Speed ~ Feed, lathe1, id.method=9:12, smooth=FALSE, reg.line=FALSE,
   boxplots=FALSE, cex=2, pch=20)
```



With all the data, the analysis of variance table is

```
Anova(m2)
```

```
Anova Table (Type II tests)

Response: log(Life)
               Sum Sq Df F value  Pr(>F)
poly(Speed, 2)  31.02  2  175.59 1.2e-10
poly(Feed, 2)    9.02  2   51.06 3.7e-07
Speed:Feed       0.04  1    0.48     0.5
Residuals        1.24 14
```

With the two cases deleted, the interaction is no longer important.

```
m3 <- update(m2, subset=-c(9, 10))
Anova(m3)
```

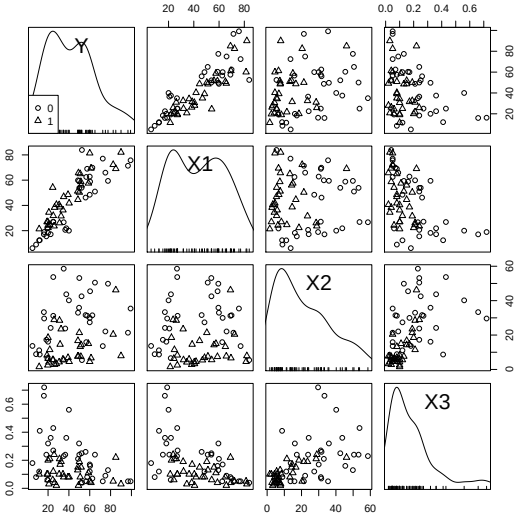
```
Anova Table (Type II tests)

Response: log(Life)
               Sum Sq Df F value  Pr(>F)
poly(Speed, 2)  16.71  2  153.75 2.8e-09
poly(Feed, 2)    8.63  2   79.46 1.2e-07
Speed:Feed       0.04  1    0.78    0.39
Residuals        0.65 12
```

In this example deleting the points is probably not called for; after all, this were designed to be influential cases, so there is a hint of an interaction here. □

9.17 Solution:

As usual, we begin with a scatterplot matrix. We use `X4`, which is a dummy variable, as a marking variable.

```
scatterplotMatrix(~Y + X1 + X2 + X3|X4,
   landrent, reg.line=FALSE, smooth=FALSE)
```



The mean functions in each of the plots of predictors versus other predictors, either conditioning on point color or ignoring it, seems to be somewhat curved, so transformations of the predictors seem likely to be useful. The results of the multivariate Box–Cox method are:

```
summary(b1 <- powerTransform(cbind(X1, X2, X3) ~ 1,
     data=landrent))
```

```
bcPower Transformations to Multinormality

   Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
X1    0.7903   0.2030           0.3924           1.1882
X2    0.2371   0.1218          -0.0016           0.4759
```

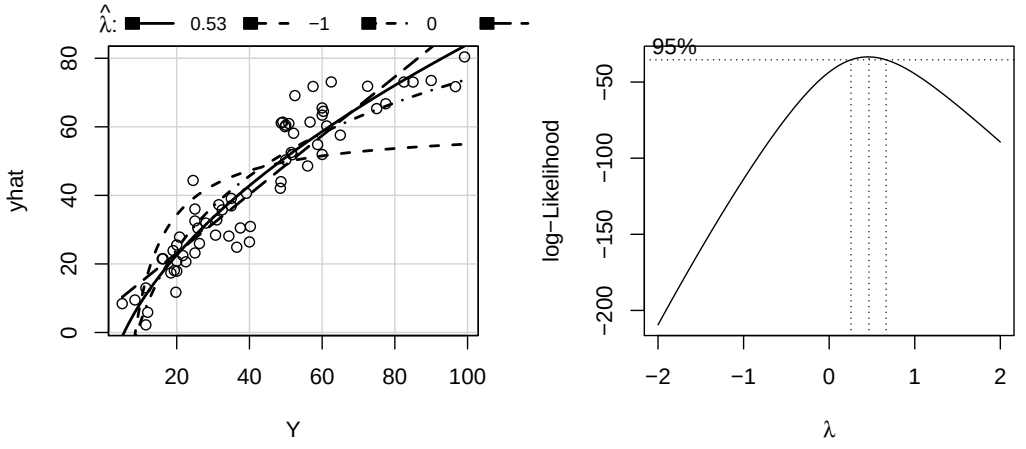
```
X3    0.0825   0.0991          -0.1118           0.2768

Likelihood ratio tests about transformation parameters
                               LRT df      pval
LR test, lambda = (0 0 0)  23.156  3 3.748e-05
LR test, lambda = (1 1 1) 102.374  3 0.000e+00
LR test, lambda = (1 0 0)   5.254  3 1.541e-01
```

which suggests replacing `X1` and `X2` by their logarithms. Ignoring for the moment the indicator `X4`, we now turn to transforming `Y`. Given below are both the inverse response plot for the mean function `Y ~ X1 + log(X2) + log(X3)`

```
m1 <- lm(Y ~ X1 + log(X2) + log(X3) + X4, landrent)
par(mfrow=c(1, 2))
inverseResponsePlot(m1)
```

```
   lambda   RSS
1  0.5299  3457
2 -1.0000 13701
3  0.0000  4624
4  1.0000  4138
```

```
boxcox(m1)
```



Both figures suggest using a transformation of Y close to the square root. The inverse response plot suggests that the improvement of the square root over untransformed is relatively small, and the decision not to transform may be reasonable. In this solution, however, we use the square root transformations for the response.

```
summary(m3 <- lm(sqrt(Y) ~ X1 + log(X2) + log(X3) + X4, landrent))
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.64775    0.62656    1.03     0.31
X1           0.06893    0.00541   12.74  < 2e-16
log(X2)      0.78613    0.15674    5.02  4.7e-06
log(X3)     -0.15899    0.17254   -0.92     0.36
```

```
X4           0.29375    0.19576    1.50     0.14

Residual standard error: 0.626 on 62 degrees of freedom
Multiple R-squared: 0.884
F-statistic:  118 on 4 and 62 DF,  p-value: <2e-16
```

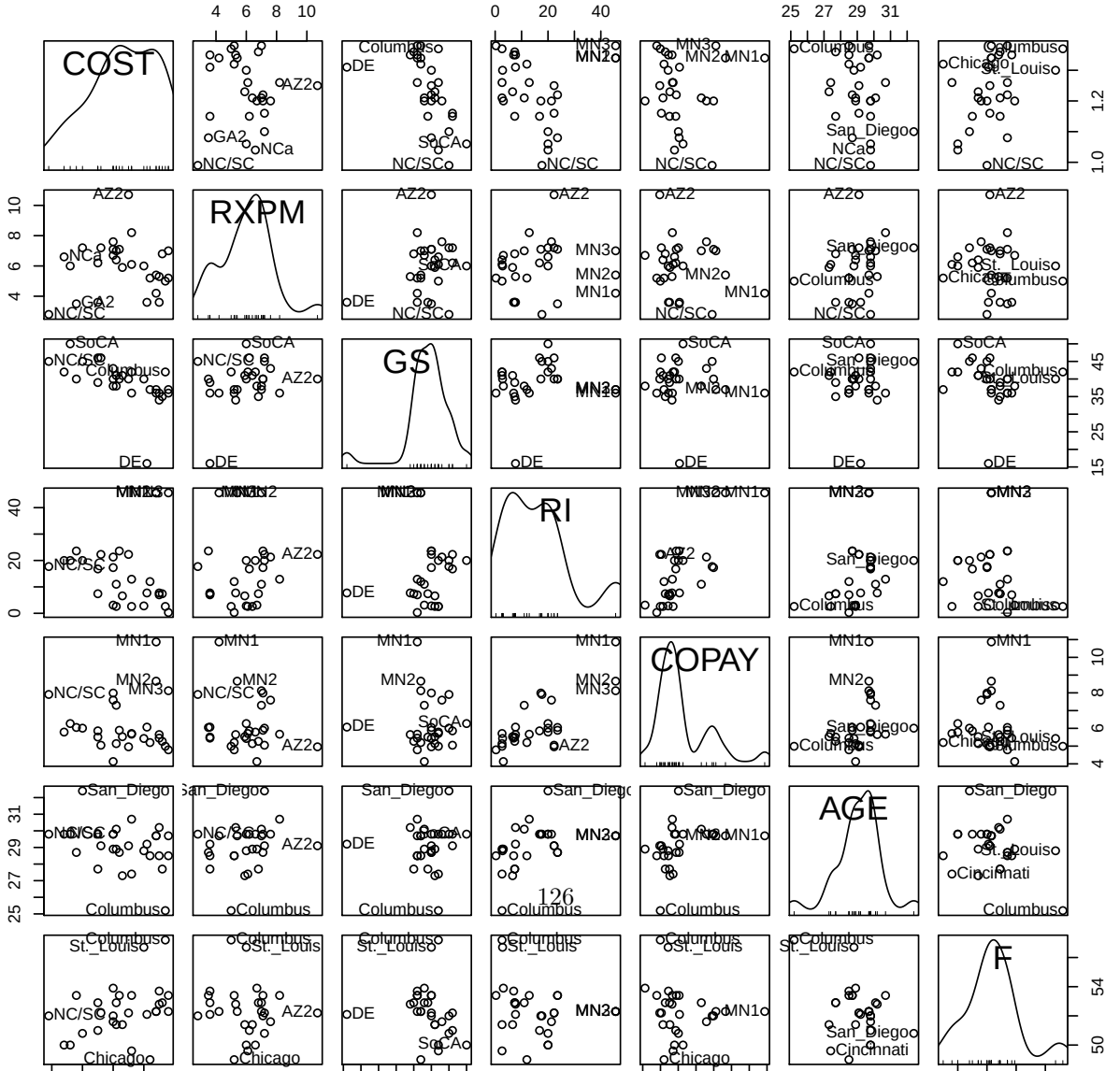
We turn to model checking, which would suggest looking for influential observations, outliers, and lack-of-fit of the mean function, but no unusual results are found.

In summary, rent paid increases with `X1` = average rent paid in the county and `X2` = density of dairy cows. Neither liming nor amount of pasture in the county are of any importance. □

9.19 Solution:

Nearly all the variables have a very restricted range, so transformations are likely to be of little value. Also this is an ecological regression problem, with data for clinics when it is individuals who get medical care. Weighting by clinic size `MM` could be appropriate. Using the plan as the unit of analysis is appropriate for a policy maker interested in understanding how plans cope with prescription costs. Using the member as the unit of analysis might be appropriate for a consumer or someone studying how the health *community* pays for drugs delivered to individuals.

View the scatterplot matrix of the data.


COST
RXPM
GS
RI
COPAY
AGE
F
AZ2
NC/SC
SoCA
DE
MN1
MN2
MN3
San_Diego
Columbus
St._Louis
Chicago
Cincinnati

Three of the plans (MN1, MN2, and MN3) have very high values of RI and also very high costs. One plan, DE, is much lower of GS than all the other plans. At the first stage, these four plans are removed. In this scaling, there is no need for transformations (MM is not used as a predictor, although it appears to be irrelevant anyway).

All indicators suggest linearly related predictors, mostly because there is so much noise in the data that we cannot really detect anything else.

The unweighted analysis is particularly straightforward. Increasing GS by 10% will lower prescription per day cost by around \$0.09 to \$0.11, depending on the status of the four separated points. The restricted formulary RI is more complex. Without the three Minnesota clinics, increasing RI by 10% will decrease costs by about \$0.04. With these clinics included, we get the result that costs are high if RI is either too low or too high. Overall, the regression appears to match the data quite well: the standard deviation in drug cost between clinics is about \$0.11, and after fitting the model the residual standard deviation is about \$0.06.

A weighted analysis, using MM as weights, is also appropriate for these data, but interpretation of variances is harder. Using MM as weights means that the unit of analysis is the patient month, not the clinic. When we say that $\text{Var}(y|x) = \sigma^2/\texttt{MM}$, we really mean that we have MM units, all with the same value of x, and the reported value of y is the average of those MM units. □

CHAPTER 10

Variable Selection

10.1 10.1.1

10.1.2

10.3 Solution:

Using the `step` method in R, here is the result for forward selection:

```
m0 <- lm(Y ~ 1, data=mantel)
step(m0,scope= ~ X1 + X2 + X3, direction="forward")
```

```
Start:  AIC=9.59
Y ~ 1
```

```
       Df Sum of Sq   RSS   AIC
+ X3    1     20.69  2.11 -0.31
+ X1    1      8.61 14.19  9.22
+ X2    1      8.51 14.29  9.25
<none>              22.80  9.59

Step:  AIC=-0.31
Y ~ X3

       Df Sum of Sq  RSS    AIC
<none>              2.11 -0.309
+ X2    1    0.0663 2.05  1.532
+ X1    1    0.0645 2.05  1.536

Call:
lm(formula = Y ~ X3, data = mantel)

Coefficients:
(Intercept)           X3
      0.798        0.695
```

Using backward elimination,

```
m1 <- lm(Y ~ X1 + X2 + X3, data=mantel)
step(m1,scope=~1, direction="backward")
```

```
Start:  AIC=-314.8
Y ~ X1 + X2 + X3

       Df Sum of Sq  RSS    AIC
```

```
- X3    1      0.00 0.00 -316.2
<none>              0.00 -314.8
- X1    1      2.05 2.05    1.5
- X2    1      2.05 2.05    1.5

Step:  AIC=-316.2
Y ~ X1 + X2

       Df Sum of Sq  RSS    AIC
<none>               0.0 -316.2
- X2    1      14.2 14.2    9.2
- X1    1      14.3 14.3    9.3

Call:
lm(formula = Y ~ X1 + X2, data = mantel)

Coefficients:
(Intercept)           X1           X2
      -1000            1            1
```

It appears that the backward elimination algorithm selects to remove *none* of the regressors, as `AIC` is lowest for the mean function will all terms. *However, the residual sum of squares for both the full mean function, and the mean function without X_3, are zero, within rounding error.* Consequently, the difference in `AIC` between the full mean function and the mean function without X_3 is due to rounding error only. Consequently, X_3 can be deleted, and still give an exact fit. Using backward elimination, therefore, $X_{\mathcal{A}} = \{X_1, X_2\}$.

These two computational algorithms give different answers. We would certainly prefer the choice $X_{\mathcal{A}} = \{X_1, X_2\}$ from backward elimination because it gives an exact fit. □

10.5 Solution:

As usual, we begin with a scatterplot matrix.

Comment: *Figure dwaste1 missing*

We have replaced O2UP by its logarithm based solely on the range of this variable. There are several separated points in the graph, which we would like to identify. Rdoes not permit identifying points in a scatterplot matrix, a facility that is greatly missed. The case with the very low value of TVS is case 17; deleting this case from the data, we get

```
summary(b1 <- powerTransform(cbind(BOD, TKN, TS, TVS, COD) ~ 1,
  data=dwaste, subset=-17))
```

```
bcPower Transformations to Multinormality

    Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
BOD    0.6749   0.2469           0.1909            1.159
TKN   -0.5903   1.0466          -2.6416            1.461
TS     0.0668   0.4764          -0.8669            1.000
TVS    2.3332   3.7079          -4.9342            9.601
COD    0.2722   0.5866          -0.8776            1.422

Likelihood ratio tests about transformation parameters
                                LRT df    pval
LR test, lambda = (0 0 0 0 0) 11.12  5 0.04900
LR test, lambda = (1 1 1 1 1) 10.88  5 0.05381
```

The p-values for both all logarithms and all untransformed are very close to 0.05. We interpret this to mean that there is very little information about the choice of transformation. We tentatively decide to continue without any further transformation. We can then justify the log-transform to the response using either the Box–Cox method or using an inverse response plot.

We next turn to residuals and influence. Examining residuals plots, and in particular using Tukey's test for nonadditivity, suggest that the mean function with predictors untransformed and the log of O2UP as

the response appears to be inadequate. An index plot of the influence statistics suggests that case #1 is highly influential for estimating coefficients; when case #1 is deleted, the resulting fit appears to be adequate. Using backward elimination, we are led to using only `TS` as the single active predictor. As a check, the plot of the fitted values from the mean function with all predictors versus the fitted values from the regression of with `TS` as the only term in the mean function is a straight line with relatively little scatter. We are led to include that `TS` might well be the only active term in the mean function.

We should now consider the deleted cases, seventeen and one. Case seventeen would have little impact on the mean function with `TS` as the only active term, since it was not unusual on `TS`. Case one is a little different because the data were ordered in time, and this day might well represent a different process that stabilized after a few hours. □

CHAPTER 11

Nonlinear Regression

11.1 11.1.1 Solution:

The mean function is nonlinear because γ multiplies β_{ij}. It describes a straight-line mean function for each level of G. Each group has its own slope β_{1j}, but all lines are concurrent at $x = \gamma$. □

11.1.2 Solution:

```
m0 <- lm(TS ~ log(BodyWt):factor(D), sleep1)
summary(m0)$coef
```

```
                        Estimate Std. Error t value  Pr(>|t|)
(Intercept)              11.6259     0.5457 21.3048 2.372e-27
log(BodyWt):factor(D)1  -0.2892     0.2794 -1.0351 3.054e-01
```

```
log(BodyWt):factor(D)2  -0.5930     0.6996 -0.8477 4.005e-01
log(BodyWt):factor(D)3  -0.9325     0.3521 -2.6480 1.069e-02
log(BodyWt):factor(D)4  -0.6414     0.3019 -2.1250 3.836e-02
log(BodyWt):factor(D)5  -1.6585     0.3321 -4.9936 7.044e-06
```

This was to get starting values only. Here is the nonlinear fit

```
m1 <- nls(TS~ b0 + b11*((D==1)*(log(BodyWt) - gamma))
               + b12*((D==2)*(log(BodyWt) - gamma))
               + b13*((D==3)*(log(BodyWt) - gamma))
               + b14*((D==4)*(log(BodyWt) - gamma))
               + b15*((D==5)*(log(BodyWt) - gamma)),
   data=sleep1,
   start=list(b0=11,b11=-.3,b12=-.6,b13=-.9,b14=-.6,
              b15=-1.6, gamma=0))
summary(m1)
```

```
Formula: TS ~ b0 + b11 * ((D == 1) * (log(BodyWt) - gamma)) + b12 * ((D ==
    2) * (log(BodyWt) - gamma)) + b13 * ((D == 3) * (log(BodyWt) -
    gamma)) + b14 * ((D == 4) * (log(BodyWt) - gamma)) + b15 *
    ((D == 5) * (log(BodyWt) - gamma))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
b0      49.372    192.659    0.26  0.79877
b11     -0.590      0.258   -2.29  0.02610
b12     -0.630      0.167   -3.76  0.00044
b13     -0.650      0.192   -3.38  0.00138
b14     -0.652      0.191   -3.41  0.00128
b15     -0.705      0.388   -1.82  0.07492
gamma  -60.130    305.083   -0.20  0.84454
```

```
Residual standard error: 3.37 on 51 degrees of freedom

Number of iterations to convergence: 14
Achieved convergence tolerance: 9.5e-06
  (4 observations deleted due to missingness)
```

The estimate of γ has such a large variance that there is no reason to include γ in the mean function. □

11.3 Solution:

This requires specifying a sequence of models corresponding to the choices of mean function to be prepared. We considered five such mean functions, although many more are possible:

```
LI <- max(walleye$length)+1
m0 <- lm(log(1-length/LI) ~ age, walleye)
K <- -coef(m0)[2]
t0 <- coef(m0)[1]/coef(m0)[2]
c1 <- nls(length~LI*(1-exp(-K*(age-t0))),
          start=list(LI=LI, K=K, t0=t0),
          data=walleye)
c2 <- nls(length~(period==1)*LI1*(1-exp(-K1*(age-t01))) +
                 (period==2)*LI2*(1-exp(-K2*(age-t02))) +
                 (period==3)*LI3*(1-exp(-K3*(age-t03))),
                 start=list(LI1=LI,LI2=LI,LI3=LI,
                            K1=K,K2=K,K3=K,
                            t01=t0,t02=t0,t03=t0),
                 data=walleye)
c3 <- nls(length~(period==1)*LI*(1-exp(-K1*(age-t01))) +
                 (period==2)*LI*(1-exp(-K2*(age-t02))) +
                 (period==3)*LI*(1-exp(-K3*(age-t03))),
                 start=list(LI=LI,
                            K1=K,K2=K,K3=K,
```

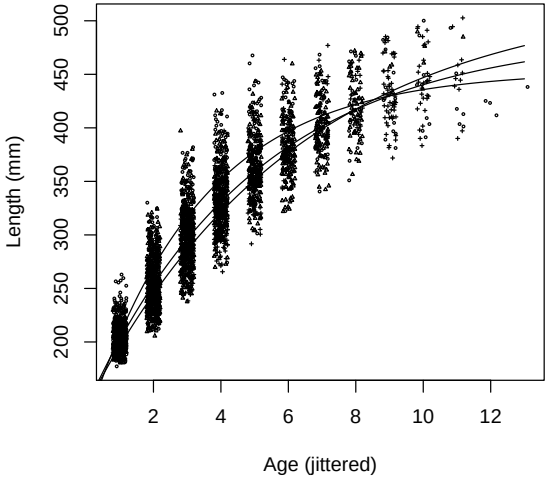
```
                          t01=t0,t02=t0,t03=t0),
                 data=walleye)
 c4 <- nls(length~(period==1)*LI1*(1-exp(-K*(age-t01))) +
                 (period==2)*LI2*(1-exp(-K*(age-t02))) +
                 (period==3)*LI3*(1-exp(-K*(age-t03))),
                 start=list(LI1=LI,LI2=LI,LI3=LI,
                          K=K,
                          t01=t0,t02=t0,t03=t0),
                 data=walleye)
 c5 <- nls(length~(period==1)*LI1*(1-exp(-K1*(age-t0))) +
                 (period==2)*LI2*(1-exp(-K2*(age-t0))) +
                 (period==3)*LI3*(1-exp(-K3*(age-t0))),
                 start=list(LI1=LI,LI2=LI,LI3=LI,
                          K1=K,K2=K,K3=K,
                          t0=t0),
                 data=walleye)
 anova(c1,c3,c2)

Analysis of Variance Table

Model 1: length ~ LI * (1 - exp(-K * (age - t0)))
Model 2: length ~ (period == 1) * LI * (1 - exp(-K1 * (age - t01))) + (period == 2) * LI *
Model 3: length ~ (period == 1) * LI1 * (1 - exp(-K1 * (age - t01))) + (period == 2) * LI2
  Res.Df Res.Sum Sq Df Sum Sq F value  Pr(>F)
1   3195    2211448
2   3191    1994577  4 216871    86.7 < 2e-16
3   3189    1963513  2  31064    25.2 1.3e-11

 anova(c1,c4,c2)

Analysis of Variance Table
```

```
Model 1: length ~ LI * (1 - exp(-K * (age - t0)))
Model 2: length ~ (period == 1) * LI1 * (1 - exp(-K * (age - t01))) + (period == 2) * LI2 *
Model 3: length ~ (period == 1) * LI1 * (1 - exp(-K1 * (age - t01))) + (period == 2) * LI2
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1   3195    2211448
2   3191    2014863  4 196585    77.8 <2e-16
3   3189    1963513  2  51350    41.7 <2e-16
```

```
anova(c1,c5,c2)
```

```
Analysis of Variance Table

Model 1: length ~ LI * (1 - exp(-K * (age - t0)))
Model 2: length ~ (period == 1) * LI1 * (1 - exp(-K1 * (age - t0))) + (period == 2) * LI2 *
Model 3: length ~ (period == 1) * LI1 * (1 - exp(-K1 * (age - t01))) + (period == 2) * LI2
  Res.Df Res.Sum Sq Df Sum Sq F value  Pr(>F)
1   3195    2211448
2   3191    1989989  4 221458    88.8 < 2e-16
3   3189    1963513  2  26476    21.5 5.3e-10
```

The model `c1` ignores the period effect. `c5` has separate parameters for each period, and is the most general. Models `c2`–`c4` are intermediate, setting either the asymptote, rate or start parameters equal. In each case, we use the method suggested in previous problems to get starting values. The five models can be compared using analysis of variance. The most general model seems appropriate, so all three parameters differ in each period. Sample sizes here are very large, so the tests are very powerful and may be detecting relatively unimportant differences.



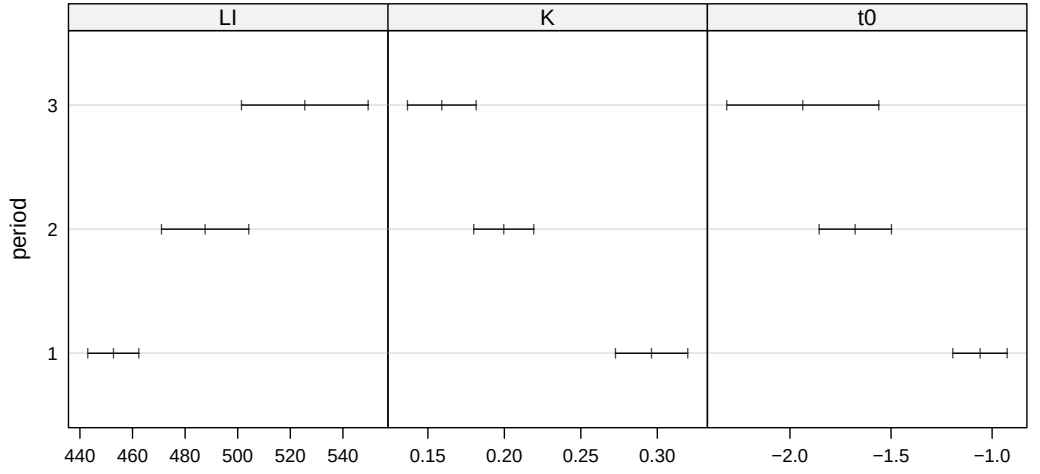
The R package `nlme` has a function `nlsList` that simplifies much of the preceding at the cost of generality:

```
library(nlme)
l1 <- nlsList(length~LI*(1-exp(-K*(age-t0)))|period,
          start=list(LI=LI, K=K, t0=t0),
          data=walleye)
intervals(l1)
```

```
, , LI
```

```
  lower  est. upper
1 443.1 452.8 462.5
2 471.1 487.7 504.3
3 501.5 525.6 549.7

, , K

   lower   est.  upper
1 0.2728 0.2963 0.3199
2 0.1800 0.1997 0.2194
3 0.1367 0.1591 0.1816

, , t0

   lower   est.   upper
1 -1.193 -1.059 -0.9252
2 -1.855 -1.676 -1.4970
3 -2.311 -1.936 -1.5600
```

```
print(plot(intervals(l1), layout=c(3, 1)))
```



□

11.5 Solution:

```
psi.s <- function(x, lambda) bcPower(x, lambda)
# starting values
bstart <- coef(lm(log(rate) ~ psi.s(len, 1) + psi.s(adt, 1),
          data=Highway))
m2 <- nls(log(rate) ~ b0 + b1*psi.s(len, lam1) + b2*psi.s(adt, lam2),
          data=Highway, start=list(b0=bstart[1], b1=bstart[2],
          b2=bstart[3], lam1=1, lam2=1))
summary(m2)
```

```
Formula: log(rate) ~ b0 + b1 * psi.s(len, lam1) + b2 * psi.s(adt, lam2)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
b0      3.530      1.180    2.99   0.0051
b1     -1.159      1.365   -0.85   0.4018
b2     -0.392      0.480   -0.82   0.4193
lam1   -0.352      0.544   -0.65   0.5218
lam2   -0.693      0.879   -0.79   0.4363

Residual standard error: 0.379 on 34 degrees of freedom

Number of iterations to convergence: 23
Achieved convergence tolerance: 8.92e-06
```

The function `psi.s` matches the definition of ψ_S in the text, and it uses the `bcPower` function in `car`. We get starting values by fitting via OLS assuming that $\lambda_1 = \lambda_2 = 1$. The nonlinear mean function is then specified using the starting values just obtained. The methods in Chapter 7 either transform one variable at a time for linearity in the regression of the response on the predictor, or else use the multivariate Box–Cox method to transform for multivariate normality. This method simultaneously transforms two predictors for linearity, and so is different from the other methods. The suggested transformations are $\lambda_1 \approx -1/3$ and $\lambda_2 \approx -2/3$, but both are within one standard error of zero for a log-transformation. □

CHAPTER 12

Binomial and Poisson Regression

12.1 12.1.1 Solution:

```
(t1 <- xtabs(~ spp + y, Blowdown))
```

```
                y
spp               0   1
  aspen         130 306
  balsam fir     69   6
  black spruce  426 233
  cedar         438 532
  jackpine      311  44
```

```
 paper birch   89 413
 red pine     407  90
 red maple    101  22
 black ash     11  38
```

The number of trees of each species is

```
rowSums(t1)
```

```
      aspen   balsam fir black spruce        cedar     jackpine  paper birch
        436           75          659          970          355          502
   red pine    red maple    black ash
        497          123           49
```

and the number that died and survived is

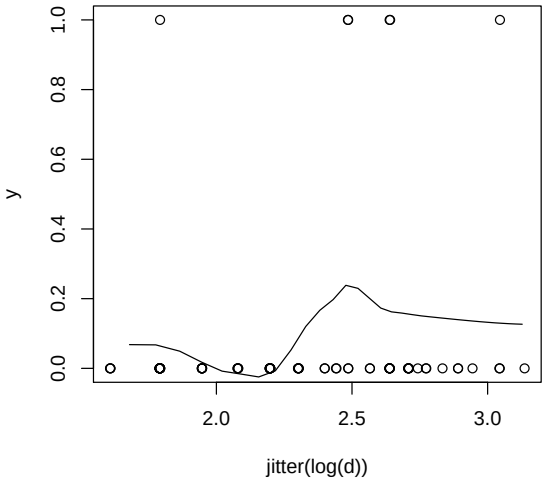
```
colSums(t1)
```

```
   0    1
1982 1684
```

□

12.1.2 Solution:

```
plot(y ~ jitter(log(d)), data=Blowdown, subset=spp=="balsam fir")
sm <- loess(jitter(y) ~ log(d), data=Blowdown, subset=spp=="balsam fir")
d1 <- seq(1,55, length=100)
lines(log(d1), predict(sm, data.frame(d=d1)))
```



Without the smoother the plot is uninformative. The decline in the probability of might suggest that fitting log(d) alone might not be adequate because that would not permit a decline in blow down probability for larger trees. □

12.1.3 Solution:

```
summary(g1 <- glm(y ~ log(d), family=binomial,
  data=Blowdown, subset=spp=="balsam fir"))$coef
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.857      2.805  -2.088  0.03681
log(d)         1.422      1.111   1.280  0.20051
```

□

12.1.4 Solution:

The **anova**, with a small "a", is used to get the change in deviance between two or more models.

```
summary(g2 <- update(g1, ~ . + I(log(d)^2)))$coef
```

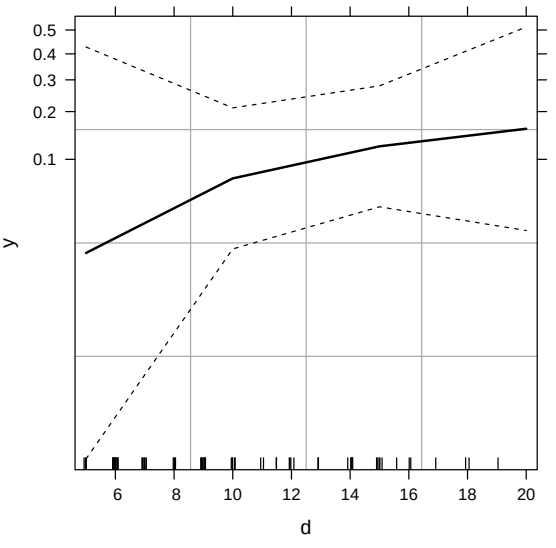
```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -8.3644     16.052 -0.5211   0.6023
log(d)        3.5534     13.447  0.2643   0.7916
I(log(d)^2)  -0.4411      2.771 -0.1592   0.8735
```

```
print(anova(g1, g2, test="Chisq"), digits=5)
```

```
Analysis of Deviance Table

Model 1: y ~ log(d)
Model 2: y ~ log(d) + I(log(d)^2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        73     40.053
2        72     40.027  1 0.025893   0.8722
```

$z^2 = (-0.159)^2 = 0$, which is close to, but not exactly the same as the change in $G^2 = 4.02$, but both tests would conclude modest evidence that the probability of blow down declines for the largest trees.

```
plot(Effect("d", g2), main="", grid=TRUE)
```



Decline in probability is plausible, but the estimated curve does not show decline. □

12.3 12.3.1 Solution:

```
(t1 <- xtabs( ~ outcome + myopathy, Downer))
```

```
          myopathy
outcome    absent present
  died         78      89
  survived     49       6
```

To get the survival fraction, divide the second row of the table by the column sums:

```
t1[2,]/colSums(t1)
```

```
 absent present
0.38583 0.06316
```

□

12.3.2 Solution:

```
m1 <- glm(outcome ~ myopathy, data=Downer,
          family=binomial)
summary(m1)
```

```
Call:
glm(formula = outcome ~ myopathy, family = binomial, data = Downer)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.987  -0.987  -0.361  -0.361   2.350

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.465      0.182   -2.55    0.011
myopathypresent   -2.232      0.459   -4.86  1.2e-06

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 248.57  on 221  degrees of freedom
Residual deviance: 214.14  on 220  degrees of freedom
  (213 observations deleted due to missingness)
AIC: 218.1

Number of Fisher Scoring iterations: 5
```

```
exp(coef(m1))
```

```
    (Intercept) myopathypresent
         0.6282          0.1073
```

The intercept is the estimated log-odds of survival when `myopathy` = 0. The coefficient for `myopathy` is the estimated increase in log-odds when `myopathy` is present. Changing to odds case, when `myopathy` is present the odds of survival are multiplied by about 0.107, a huge change. The 95% confidence intervals are

```
confint(m1)
```

```
                 2.5 %  97.5 %
(Intercept)     -0.828 -0.1115
myopathypresent -3.232 -1.4016
```

or in the odds scale,

```
exp(confint(m1))
```

```
                  2.5 % 97.5 %
(Intercept)     0.43691 0.8945
myopathypresent 0.03946 0.2462
```

The predicted values in the scale of the response will give the fitted probability of survival for the two classes,

```
predict(m1,
   data.frame(myopathy=factor(levels(Downer$myopathy))),
   type="response")
```

```
      1       2
0.38583 0.06316
```

The estimated survival probabilities match the observed survival rates for the two conditions. □

12.3.3 Solution:

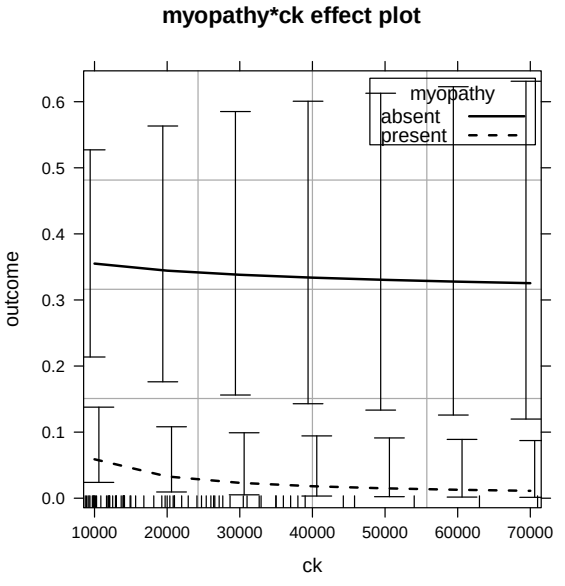
summary(g2 <- glm(outcome log(ck), binomial, Downer)) When `ck` increases by 10%, the odds of survival decline by about $.1 \times (-.61) = -0.061$, or by about 6%. □

12.3.4 Solution:

```
g3 <- glm(outcome ~ myopathy + log(ck) + myopathy:log(ck),
       binomial, Downer)
Anova(g3)
```

```
Analysis of Deviance Table (Type II tests)

Response: outcome
                LR Chisq Df Pr(>Chisq)
myopathy           12.62  1    0.00038
log(ck)             1.32  1    0.24999
myopathy:log(ck)    3.42  1    0.06439
```

```
plot(Effect(c("myopathy", "ck"), g3), grid=TRUE,
    multiline=TRUE, ci.style="bars", rescale.axis=FALSE,
    key.args=list(corner=c(.98,.98)) )
```



The presence of myopathy clearly decreases survival probability, but the effect of `ck` is much smaller, and only for smallest values of `ck` does there seem to be an effect.

□

12.5 12.5.1 Solution:

```
 (t2 <- xtabs(~ y + sex, Donner))
          sex
y          Female Male
  died         10   32
  survived     25   24
```

```
 (totals <- colSums(t2))
```

```
Female   Male
    35     56
```

```
 (freqs <- t2[2, ]/totals)
```

```
Female   Male
0.7143 0.4286
```

There were 56 males and 35 females. The survival rate for females was about 71% and about 43% for males. We test for equality of rates using Pearson's X^2; the uncorrected test (not corrected for continuity) has p-value of about 0.008, so we reject the hypothesis that the survival rate was the same for the two sexes. □

12.5.2 Solution:

```
 summary(m1 <- glm(y ~ age, data=Donner, family=binomial()))
```

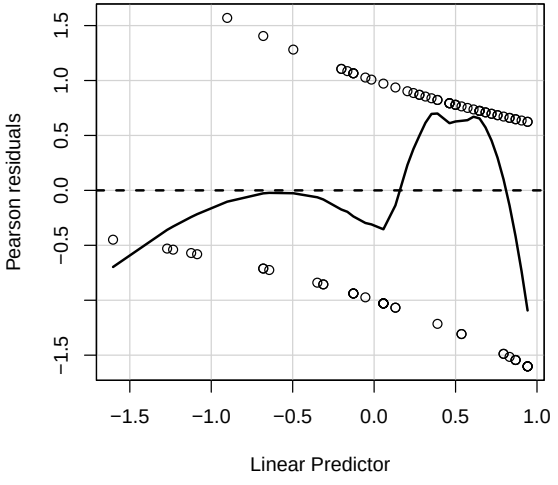
```
Call:
glm(formula = y ~ age, family = binomial(), data = Donner)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.595  -1.202   0.844   0.988   1.577

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.9792     0.3746    2.61    0.009
age          -0.0369     0.0149   -2.47    0.013

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 120.86  on 87  degrees of freedom
Residual deviance: 114.02  on 86  degrees of freedom
  (3 observations deleted due to missingness)
AIC: 118

Number of Fisher Scoring iterations: 4
```

The coefficient for `age` is negative, suggesting that survival probability decreased with age, and a year increase in `age` corresponds to about a -3.7% decrease in the odds of survival □

12.5.3 Solution:

```
residualPlot(m1, grid=TRUE)
```



The graph is not very satisfactory, but the curve in the smoother does suggest the possibility that survival probability is overestimated for the older ages. Refit with a quadratic in age,

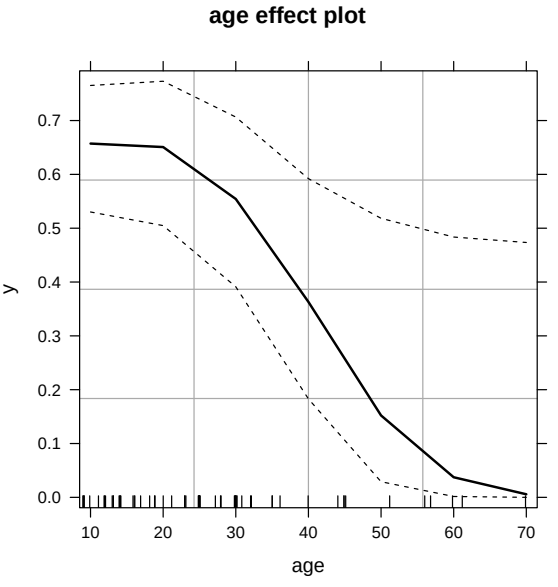
```
summary(m2 <- update(m1, ~ poly(age, 2), data=Donner[ !is.na(Donner$age), ]))
```

```
Call:
glm(formula = y ~ poly(age, 2), family = binomial(), data = Donner[!is.na(Donner$age),
    ])

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.476  -1.332   0.905   0.952   2.000
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      0.138      0.245    0.57    0.572
poly(age, 2)1   -7.618      3.298   -2.31    0.021
poly(age, 2)2   -5.635      3.364   -1.68    0.094

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 120.86  on 87  degrees of freedom
Residual deviance: 110.24  on 85  degrees of freedom
AIC: 116.2

Number of Fisher Scoring iterations: 5
```

Three of the cases have `age` missing and the `poly` fails with missing data, so these 3 need to be deleted. The significance level of the quadratic term is close to the 5% level. Here is the effects plot:

```
plot(allEffects(m2), rescale.axis=FALSE, grid=TRUE)
```



Perhaps both young and old had lower survival probabilities. □

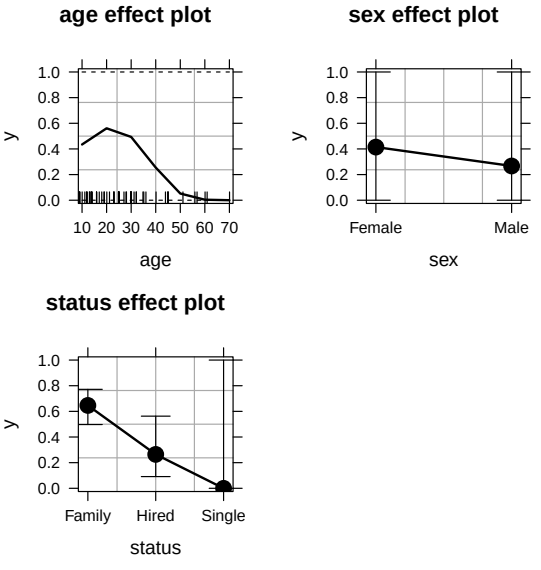
12.5.4 Solution:

```
m3 <- glm(y ~ poly(age, 2) + sex + status, binomial,
      data=na.omit(Donner))
Anova(m3)
```

```
Analysis of Deviance Table (Type II tests)

Response: y
            LR Chisq Df Pr(>Chisq)
```

```
poly(age, 2)    14.29  2    0.00079
sex              1.44  1    0.23094
status          11.58  2    0.00306
```

```
plot(allEffects(m3), grid=TRUE, rescale.axis=FALSE)
```



The quadratic effect of `age` is much more pronounced in this larger model that explains more variability than is explained by `age` alone: the very young and the very old were less likely to survive. There was no clear difference due to `sex`. Hired men were less likely to survive than family members. The single men who were not hired really did poorly:

```
xtabs( ~ y + status, Donner)
          status
y          Family Hired Single
```

```
  died         25    12      5
  survived     43     6      0
```

All 5 single men died, and this is the reason for the wide confidence interval for single men.

As an additional check, one could refit with all two-factor interactions. and then do a test:

```
m4 <- update(m3, ~ (.)^2)
anova(m3, m4, test="Chisq")
```

```
Analysis of Deviance Table

Model 1: y ~ poly(age, 2) + sex + status
Model 2: y ~ poly(age, 2) + sex + status + poly(age, 2):sex + poly(age,
    2):status + sex:status
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        82       92.4
2        75       80.9  7     11.5     0.12
```

The interactions are not needed. □

12.7 12.7.1 Solution:

```
summary(m1 <- glm(cbind(surv, m - surv) ~ class + age + sex,
    binomial, data=Whitestar))
```

```
Call:
glm(formula = cbind(surv, m - surv) ~ class + age + sex, family = binomial,
    data = Whitestar)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-4.136  -1.713   0.781   2.680   4.383
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.186      0.159    7.48  7.4e-14
classfirst     0.858      0.157    5.45  5.0e-08
classsecond   -0.160      0.174   -0.92     0.36
classthird    -0.920      0.149   -6.19  5.9e-10
agechild       1.062      0.244    4.35  1.4e-05
sexmale       -2.420      0.140  -17.24  < 2e-16

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 671.96  on 13  degrees of freedom
Residual deviance: 112.57  on  8  degrees of freedom
AIC: 171.2

Number of Fisher Scoring iterations: 5
```

From Table 12.8, nearly all females survived, except in third class, where female survival was much lower. This implies a `class` × `sex` interaction. Other interactions might exist as well. □

12.7.2 Solution:

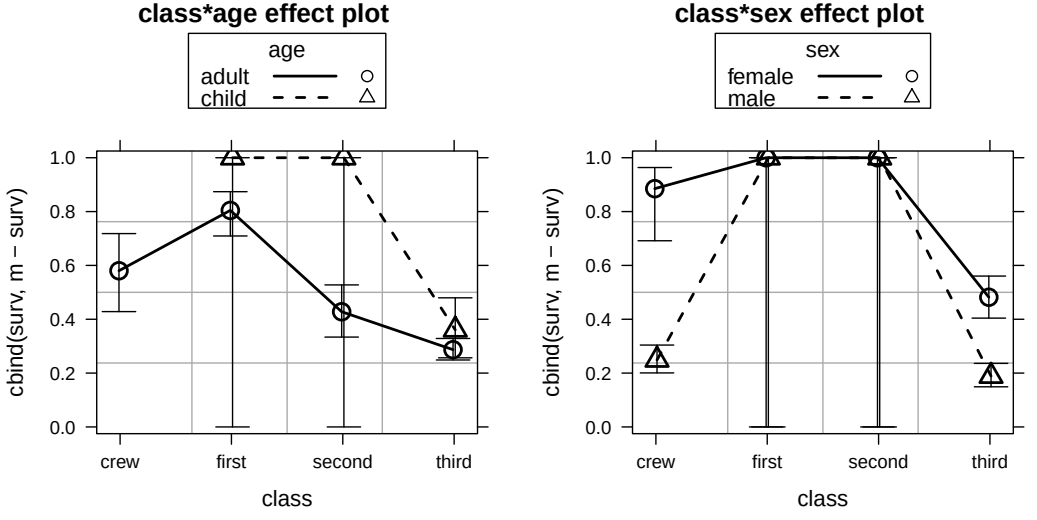
```
m2 <- update(m1, ~(class + age + sex)^2)
Anova(m2)
```

```
Analysis of Deviance Table (Type II tests)

Response: cbind(surv, m - surv)
          LR Chisq Df Pr(>Chisq)
class          121  3    < 2e-16
age             20  1    6.5e-06
sex            359  1    < 2e-16
```

```
class:age       37  2    8.1e-09
class:sex       65  3    5.0e-14
age:sex          2  1       0.19
```

The `age` × `sex` interaction can apparently be dropped, but the other two interactions are required.

```
m3 <- update(m2, ~ . - age:sex)
plot(allEffects(m3), rescale.axis=FALSE, grid=TRUE,
  multiline=TRUE, ci.style="bars")
```



Although not covered in the text, this is a model of conditional independence: given `class`, `age` and `sex` are independent, meaning that within a fixed class survival does not depend on age or sex. Survival rates were highest for first class, lowest for third class. Overall, men were much less likely to survive than women. □

12.9 12.9.1

12.9.2 Solution:

```
AMS1 <- reshape(AMSsurvey, varying=c("count", "count11"), v.names="y",
      direction="long", times=c("08-09", "11-12"), timevar="year")
AMS1$type <- factor(AMS1$type, levels=levels(AMS1$type)[order(xtabs(y~type, AMS1))])
AMS1$year <- factor(AMS1$year)
p1 <- glm(y ~ (type + sex + citizen + year)^4, poisson, AMS1)
Anova(p1)
```

```
Analysis of Deviance Table (Type II tests)

Response: y
                     LR Chisq Df Pr(>Chisq)
type                      534  5    < 2e-16
sex                       449  1    < 2e-16
citizen                     7  1    0.00681
year                       15  1    0.00011
type:sex                  104  5    < 2e-16
type:citizen               65  5    1.1e-12
type:year                  12  5    0.03886
sex:citizen                 2  1    0.12841
sex:year                    2  1    0.12185
citizen:year                0  1    0.51845
type:sex:citizen            3  5    0.70774
type:sex:year               8  5    0.18084
type:citizen:year           4  5    0.50807
sex:citizen:year            0  1    0.68396
type:sex:citizen:year       1  5    0.92579
```

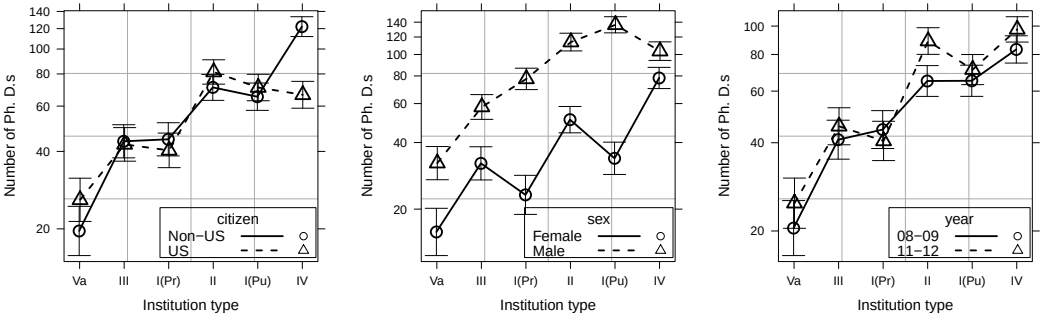
The third-order model is the biggest that can be fit. From the Type II analysis of deviance, starting at the bottom, only two-factor interactions wtih `type` are important.

```
p2 <- update(p1, ~ type*(sex + citizen + year))
Anova(p2)
```

```
Analysis of Deviance Table (Type II tests)

Response: y
             LR Chisq Df Pr(>Chisq)
type              534  5    < 2e-16
sex               449  1    < 2e-16
citizen             7  1    0.00681
year               15  1    0.00011
type:sex          108  5    < 2e-16
type:citizen       70  5    1.2e-13
type:year          11  5    0.05056
```

Here are the effects plots:

```
plot(Effect(c("type", "citizen"), p2), multiline=TRUE, ci.style="bars",
   main="", xlab="Institution type", ylab="Number of Ph. D.s",
   row = 1, col = 1, nrow = 1, ncol = 3, more = TRUE, grid=TRUE,
   key.args=list(corner=c(.98,.02)))
plot(Effect(c("type", "sex"), p2), multiline=TRUE, ci.style="bars",
   main="", xlab="Institution type", ylab="Number of Ph. D.s",
   row = 1, col = 2, nrow = 1, ncol = 3, more = TRUE, grid=TRUE,
   key.args=list(corner=c(.98,.02)))
plot(Effect(c("type", "year"), p2), multiline=TRUE, ci.style="bars",
   main="", xlab="Institution type", ylab="Number of Ph. D.s",
   row = 1, col = 3, nrow = 1, ncol = 3, more = FALSE, grid=TRUE,
   key.args=list(corner=c(.98,.02)))
```


Number of Ph. D.s
Institution type
Va
III
I(Pr)
II
I(Pu)
IV
citizen
Non−US
US
sex
Female
Male
year
08−09
11−12

□