

Preliminaries

```
> library(DAAG)
> library(splines)
```

Exercise 1

Re-analyze the sugar weight data of Subsection 7.1.1 using the `log(weight)` in place of `weight`.

From the scatterplot in Figure 7.1, it is clear that the treatment variances are not constant. Perhaps a logarithmic transformation will stabilize the variances.

```
> sugarlog.aov <- aov(log(weight) ~ trt, data = sugar)
> summary.lm(sugarlog.aov)
```

Call:

```
aov(formula = log(weight) ~ trt, data = sugar)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.16517	-0.04372	-0.00253	0.03963	0.17069

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.4122	0.0574	76.90	9.1e-13
trtA	-0.2902	0.0811	-3.58	0.0072
trtB	-0.2011	0.0811	-2.48	0.0382
trtC	-0.5229	0.0811	-6.44	0.0002

Residual standard error: 0.0994 on 8 degrees of freedom

Multiple R-Squared: 0.843, Adjusted R-squared: 0.784

F-statistic: 14.3 on 3 and 8 DF, p-value: 0.00141

```
> summary.lm(sugarlog.aov)
```

Call:

```
aov(formula = log(weight) ~ trt, data = sugar)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.16517	-0.04372	-0.00253	0.03963	0.17069

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.4122	0.0574	76.90	9.1e-13
trtA	-0.2902	0.0811	-3.58	0.0072
trtB	-0.2011	0.0811	-2.48	0.0382
trtC	-0.5229	0.0811	-6.44	0.0002

Residual standard error: 0.0994 on 8 degrees of freedom
 Multiple R-Squared: 0.843, Adjusted R-squared: 0.784
 F-statistic: 14.3 on 3 and 8 DF, p-value: 0.00141

On the log scale, the differences from control remain discernible. However the plot should be compared with plots from random normal data. This should be repeated several times. There will be occasional samples that show changes in variability of the observed residuals that are of the extent observed for these data.

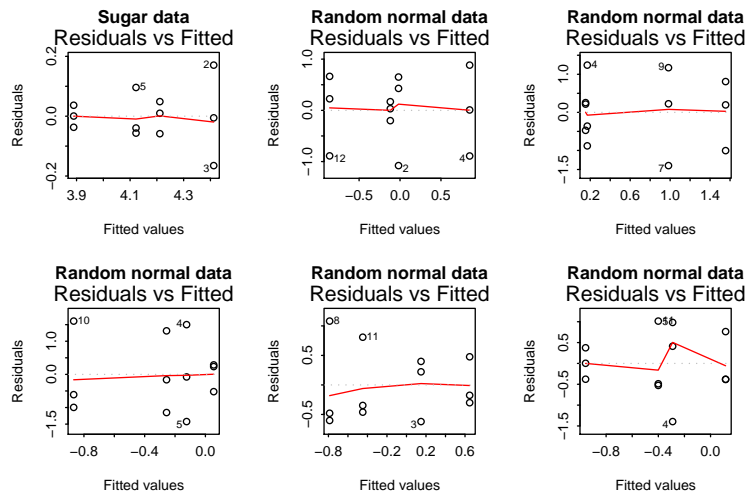


Figure 1: Plot of residuals versus fitted values, for the log(sugar weight) data.

Exercise 3

Use the method of Section 7.3 to determine, formally, whether there should be different regression lines for the two data frames `elastic1` and `elastic2` from Exercise 1 in Section 5.11.

It will be convenient to work with a single data frame:

```
> elastic2$expt <- rep(2, length(elastic2$stretch))
> elastic1$expt <- rep(1, length(elastic1$stretch))
> elastic <- rbind(elastic1, elastic2)
> elastic$expt <- factor(elastic$expt)
```

We fit three models as follows:

```
> e.lm1 <- lm(distance ~ stretch, data = elastic)
> e.lm2 <- lm(distance ~ stretch + expt, data = elastic)
> e.lm3 <- lm(distance ~ stretch + expt + stretch:expt, data = elastic)
```

The following sequential analysis of variance table indicates that there is mild evidence against the two lines having the same intercept.

```
> anova(e.lm1, e.lm2, e.lm3)
```

Analysis of Variance Table

```

Model 1: distance ~ stretch
Model 2: distance ~ stretch + expt
Model 3: distance ~ stretch + expt + stretch:expt
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      14 2549
2      13 2017  1      532 3.22  0.098
3      12 1978  1      39  0.24  0.634

```

Recall, however, from Exercise 5.1, that observation 7 is an influential outlier. Let's check to see what happens to the three models when this observation is deleted.

```

> e.lm1 <- lm(distance ~ stretch, data = elastic[-7, ])
> e.lm2 <- lm(distance ~ stretch + expt, data = elastic[-7, ])
> e.lm3 <- lm(distance ~ stretch + expt + stretch:expt, data = elastic[-7,
+      ])
> anova(e.lm1, e.lm2, e.lm3)

```

Analysis of Variance Table

```

Model 1: distance ~ stretch
Model 2: distance ~ stretch + expt
Model 3: distance ~ stretch + expt + stretch:expt
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      13 1205
2      12 1042  1      162 1.79  0.21
3      11 1000  1      42  0.46  0.51

```

Now, we see that there is really very little evidence of a difference between the two lines. Observation 7 seems different in character from other observations.

Exercise 4

The data frame `toycars` consists of 27 observations on the distance (in meters) traveled by one of three different toy cars on a smooth surface, starting from rest at the top of a 16-inch-long ramp tilted at varying angles (measured in degrees). Because of differing frictional effects for the three different cars, we seek three regression lines relating distance traveled to angle.

- As a first try, fit the model in which the three lines have the same slope but have different intercepts.
- Note the value of R^2 from the summary table. Examine the diagnostic plots carefully. Is there an influential outlier? How should it be treated?
- The physics of the problem actually suggests that the three lines should have the same intercept (very close to 0, in fact), and possibly differing slopes, where the slopes are inversely related to the coefficient of dynamic friction for each car. Fit the model, and note that the value of R^2 is slightly lower than that for the previously fitted model. Examine the diagnostic plots. What has happened to the influential outlier? In fact, we have exhibited an example where taking R^2 too seriously could be somewhat hazardous; in this case, a more carefully thought out model can accommodate all of the data satisfactorily. Maximizing R^2 does not necessarily give the best model!

4

```
> toycars$car <- factor(toycars$car)
> toycars.lm <- lm(distance ~ angle + car, data = toycars)
> summary(toycars.lm)
```

Call:

```
lm(formula = distance ~ angle + car, data = toycars)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.09811	-0.04240	-0.00669	0.01741	0.17251

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.09252	0.03467	2.67	0.0137
angle	0.18854	0.00995	18.96	1.5e-15
car2	0.11111	0.03195	3.48	0.0020
car3	-0.08222	0.03195	-2.57	0.0170

Residual standard error: 0.0678 on 23 degrees of freedom

Multiple R-Squared: 0.945, Adjusted R-squared: 0.938

F-statistic: 132 on 3 and 23 DF, p-value: 1.22e-14

From the diagnostics (below), we see that there is an influential outlier. The model is not fitting all of the data satisfactorily.

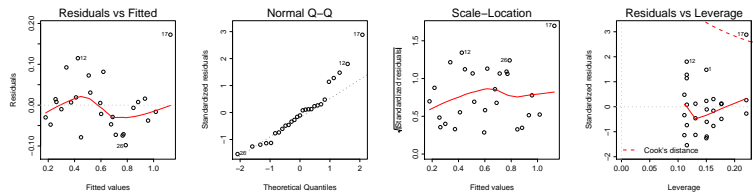


Figure 2: Diagnostic plots for toycars.lm

To fit the model with a constant intercept and possibly differing slopes, we proceed as follows:

```
> toycars.lm2 <- lm(distance ~ angle + angle:car, data = toycars)
> summary(toycars.lm2)
```

Call:

```
lm(formula = distance ~ angle + angle:car, data = toycars)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.1084	-0.0468	-0.0122	0.0697	0.1062

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.1022	0.0304	3.36	0.0027
angle	0.1819	0.0122	14.97	2.4e-13
angle:car2	0.0416	0.0112	3.71	0.0011

```
angle:car3    -0.0217    0.0112    -1.93    0.0654
```

Residual standard error: 0.0701 on 23 degrees of freedom

Multiple R-Squared: 0.941, Adjusted R-squared: 0.934

F-statistic: 123 on 3 and 23 DF, p-value: 2.65e-14

We can see from the diagnostics below that observation 17 is still somewhat influential, but it is no longer an outlier. All of the data are accommodated by this new model reasonably well.

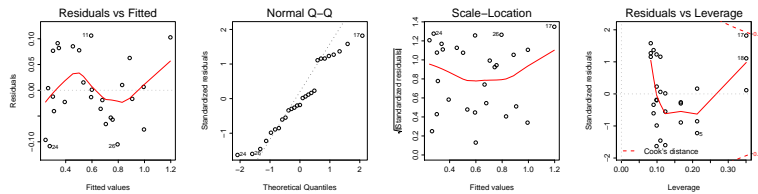


Figure 3: Diagnostic plots for toycars.lm2

Exercise 5

The data frame `cuckoos` holds data on the lengths and breadths of eggs of cuckoos, found in the nests of six different species of host birds. Fit models for the regression of length on breadth that have:

- A: a single line for all six species.
- B: different parallel lines for the different host species.
- C: separate lines for the separate host species.

Use the `anova()` function to print out the sequential analysis of variance table. Which of the three models is preferred? Print out the diagnostic plots for this model. Do they show anything worthy of note? Examine the output coefficients from this model carefully, and decide whether the results seem grouped by host species. How might the results be summarized for reporting purposes?

```
> cuckoos.lm <- lm(length ~ breadth, data = cuckoos)
> cuckoos.lm2 <- lm(length ~ breadth + species, data = cuckoos)
> cuckoos.lm3 <- lm(length ~ breadth + species + species:breadth,
+ data = cuckoos)
> anova(cuckoos.lm, cuckoos.lm2, cuckoos.lm3)
```

Analysis of Variance Table

Model 1: `length ~ breadth`

Model 2: `length ~ breadth + species`

Model 3: `length ~ breadth + species + species:breadth`

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	118	101.9				
2	113	79.1	5	22.8	6.57	2.2e-05
3	108	75.0	5	4.1	1.19	0.32

From the anova summary, we see that the second model is preferable. The standard diagnostics are given below.

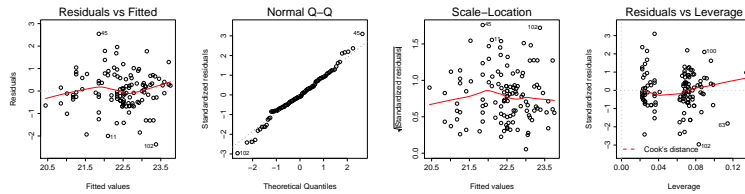


Figure 4: Diagnostic plots for cuckoos.lm2

There is nothing on these plots that calls for especial attention.

```
> summary(cuckoos.lm2)
```

Call:

```
lm(formula = length ~ breadth + species, data = cuckoos)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.3734	-0.4911	-0.0682	0.5298	2.5447

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	9.5156	3.0177	3.15	0.00207
breadth	0.8112	0.1795	4.52	1.5e-05
speciesmeadow.pipit	-0.8013	0.2561	-3.13	0.00223
speciespied.wagtail	-0.0132	0.3145	-0.04	0.96650
speciesrobin	-0.3031	0.3114	-0.97	0.33241
speciestree.pipit	0.0449	0.3114	0.14	0.88562
specieswren	-1.2391	0.3530	-3.51	0.00064

Residual standard error: 0.837 on 113 degrees of freedom

Multiple R-Squared: 0.419, Adjusted R-squared: 0.388

F-statistic: 13.6 on 6 and 113 DF, p-value: 1.44e-11

The baseline species is hedge sparrow, and we see some groupings among the host species.

The relation between length and breadth of the eggs is similar when the host species are hedge sparrow, pied wagtail and tree pipit. Even when the robin is the host species, there is little evidence of a difference in the way in which length and breadth are related. However, the linear relation between length and breadth has a smaller intercept when the host species is either the meadow pipit or the wren.

Exercise 8

Apply spline regression to the `geophones` data frame. Specifically, regress thickness against distance, and check the fits of 4-, 5- and 6-degree-of-freedom cases. Which case gives the best fit to the data? How does this fitted curve compare with the polynomial curves obtained in the previous exercise? Calculate pointwise confidence bounds for the 5-degree-of-freedom case.

We fit the 4-, 5-, and 6-degree-of-freedom spline models to the geophones data as follows:

```
> geo.spl4 <- lm(thickness ~ ns(distance, df = 4), data = geophones)
> geo.spl5 <- lm(thickness ~ ns(distance, df = 5), data = geophones)
> geo.spl6 <- lm(thickness ~ ns(distance, df = 6), data = geophones)
```

The fitted curves are plotted thus:

```
> plot(geophones)
> lines(spline(geophones$distance, predict(geo.spl4)), col = 1)
> lines(spline(geophones$distance, predict(geo.spl5)), col = 2,
+       lty = 2)
> lines(spline(geophones$distance, predict(geo.spl6)), col = 3,
+       lty = 4)
> bottomleft <- par()$usr[c(1, 3)]
> legend(bottomleft[1], bottomleft[2], lty = c(1:2, 4), col = 1:3,
+       legend = c("4 df", "5 df", "6 df"), xjust = 0, yjust = 0)
```

The 6-degree-of-freedom case gives the best fit to the data; it captures some of the curvature at the large distance values, while retaining smoothness in other regions. The 5-degree-of-freedom case is smoother than the quartic, while capturing similar amounts of curvature in the large distance region.

The 95% confidence bounds for the 5-degree-of-freedom case can be obtained and plotted as follows:

```
> plot(geophones)
> lines(spline(geophones$distance, predict(geo.spl5), col = 2)
> lines(geophones$distance, predict(geo.spl5, interval = "confidence"),
+       "lwr"), col = 2)
> lines(geophones$distance, predict(geo.spl5, interval = "confidence"),
+       "upr"), col = 2)
```

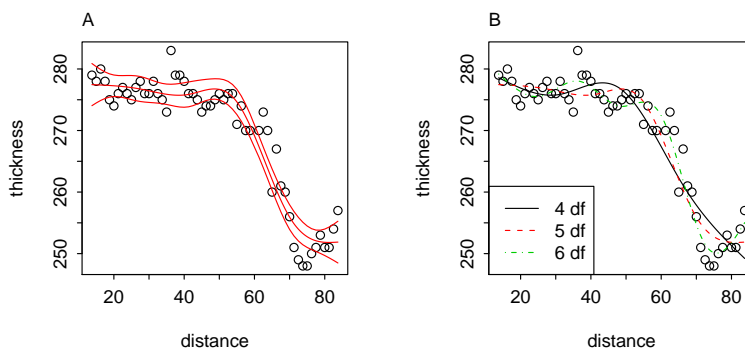


Figure 5: Panel A shows 4, 5 and 6 df spline curves fitted to the geophones data. Panel B shows confidence bounds for expected thickness, for the 5df fit.

Exercise 10

Check the diagnostic plots for the results of exercise 8 for the 5-degree-of-freedom case. Are there any influential outliers?

The standard diagnostics for the 5-degree-of-freedom spline model fit to the geophones data can be plotted using

```

> par(mfrow = c(1, 4))
> plot(geo.sp15)
> par(mfrow = c(1, 1))

```

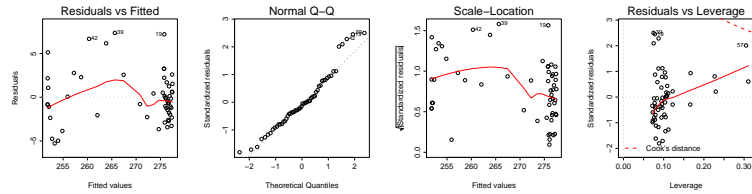


Figure 6: Diagnostic plots for 5df spline model.

There are no extreme outliers. Observation 19 is a mild outlier which exerts moderate influence. This should not be of major concern. The plot of the residuals versus the fitted values does indicate that some of the nonlinearity has not been satisfactorily modeled.

Exercise 11

Continuing to refer to exercise 8, obtain plots of the spline basis curves for the 5-degree-of-freedom case. That is, plot the relevant column of the model matrix against y .

The first basis function is a constant, to include an intercept in the model. (Note that this implies that there are actually 6 degrees of freedom in the model.) The remaining basis functions are plotted as follows:

```

> X5 <- model.matrix(geo.sp15)
> plot(X5[, 2] ~ geophones$distance, type = "l")
> lines(X5[, 3] ~ geophones$distance, col = 3)
> lines(X5[, 4] ~ geophones$distance, col = 4)
> lines(X5[, 5] ~ geophones$distance, col = 5)
> lines(X5[, 6] ~ geophones$distance, col = 6)

```

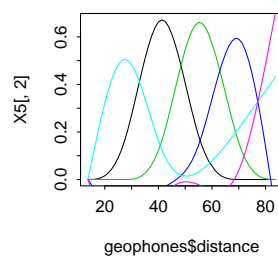


Figure 7: Spline basis functions.

We could also use `matplot()` for this problem.

```
matplot(geophones$distance, X5[,-1], type="l")
```

Exercise 13

The `ozone` data frame holds data, for nine months only, on ozone levels at the Halley Bay station between 1956 and 2000. (See Christie (2000) and Shanklin (2001) for the scientific background. Up to date data are available from the web page <http://www.nerc-bas.ac.uk/public/icd/jds/ozone/>.) Replace zeros by missing values. Determine, for each month, the number of missing values. Plot the October levels against Year, and fit a smooth curve. At what point does there seem to be clear evidence of a decline? Plot the data for other months also. Do other months show a similar pattern of decline?

A simple way to replace 0's by missing value codes is the following:

```
> names(ozone)

[1] "Year"  "Aug"   "Sep"   "Oct"   "Nov"   "Dec"   "Jan"   "Feb"
[9] "Mar"   "Apr"   "Annual"

> Ozone <- ozone
> for (i in 2:11) {
+   Ozone[ozone[, i] == 0, i] <- NA
+ }
```

One way to count up the monthly missing values is the following:

```
> sapply(Ozone[, -c(1, 11)], function(x) sum(is.na(x)))

Aug Sep Oct Nov Dec Jan Feb Mar Apr
  21  8  0  0  0  0  0  0  11
```

A plot of the October ozone levels against Year can be obtained as follows:

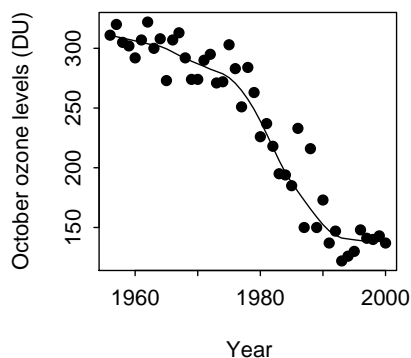


Figure 8: Lowess curve fitted to the ozone data.

We see that ozone level is decreasing throughout the period, but there is an acceleration in the mid- to late-1970s.

To plot the data for the other months, we can do the following:

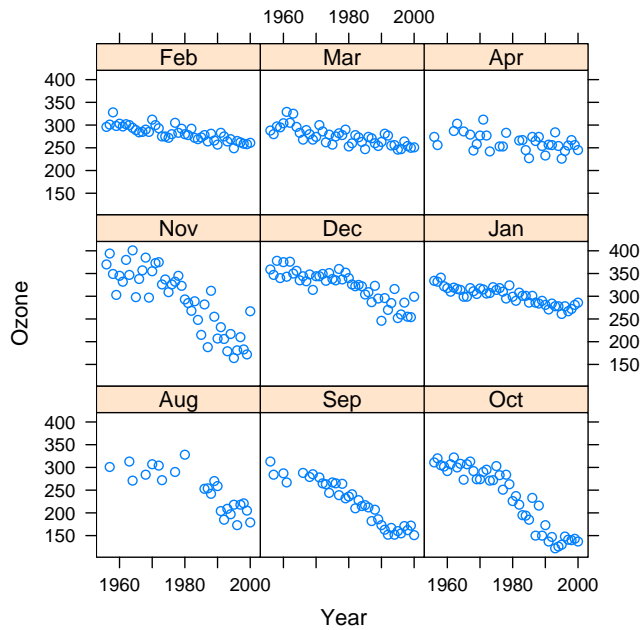


Figure 9: Change in ozone levels over time, by month.

Similar declines are evident in several of the other months. The decline is less steep in some of the other months.

Exercise 15*

*Compare the two results

```
seedrates.lm <- lm(grain ~ rate + I(rate^2),
                  data=seedrates)
seedrates.pol <- lm(grain ~ poly(rate,2),
                  data=seedrates)
```

Check that the fitted values and residuals from the two calculations are the same, and that the t -statistic and p -value are the same for the final coefficient, i.e., the same for the coefficient labeled `poly(rate, 2)` in the polynomial regression as for the coefficient labeled `I(rate^2)` in the regression on `rate` and `rate^2`.

Regress the second column of `model.matrix(seedrates.pol)` on `rate` and `I(rate^2)`, and similarly for the third column of `model.matrix(seedrates.pol)`. Hence express the first and second orthogonal polynomial terms as functions of `rate` and `rate^2`.

The following shows that the fitted values and residuals are the same for the two calculations. The t -statistic and p -value are also the same for the final coefficient.

```
> seedrates.lm <- lm(grain ~ rate + I(rate^2), data = seedrates)
> seedrates.pol <- lm(grain ~ poly(rate, 2), data = seedrates)
> fitted(seedrates.lm) - fitted(seedrates.pol)
```

```
1 2 3 4 5
0 0 0 0 0
```

```
> resid(seedrates.lm) - resid(seedrates.pol)
```

```

      1      2      3      4      5
-6.939e-17  1.804e-16  0.000e+00 -1.318e-16  6.245e-17

```

```
> summary(seedrates.lm)
```

```
Call:
```

```
lm(formula = grain ~ rate + I(rate^2), data = seedrates)
```

```
Residuals:
```

```

      1      2      3      4      5
0.04571 -0.12286  0.09429 -0.00286 -0.01429

```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	24.060000	0.455694	52.80	0.00036
rate	-0.066686	0.009911	-6.73	0.02138
I(rate^2)	0.000171	0.000049	3.50	0.07294

```
Residual standard error: 0.115 on 2 degrees of freedom
```

```
Multiple R-Squared: 0.996, Adjusted R-squared: 0.992
```

```
F-statistic: 256 on 2 and 2 DF, p-value: 0.00390
```

```
> summary(seedrates.pol)
```

```
Call:
```

```
lm(formula = grain ~ poly(rate, 2), data = seedrates)
```

```
Residuals:
```

```

      1      2      3      4      5
0.04571 -0.12286  0.09429 -0.00286 -0.01429

```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19.3200	0.0513	376.8	7e-06
poly(rate, 2)1	-2.5614	0.1146	-22.3	0.002
poly(rate, 2)2	0.4009	0.1146	3.5	0.073

```
Residual standard error: 0.115 on 2 degrees of freedom
```

```
Multiple R-Squared: 0.996, Adjusted R-squared: 0.992
```

```
F-statistic: 256 on 2 and 2 DF, p-value: 0.00390
```

From the following output, we can infer that the first orthogonal polynomial is

$$p_1(x) = -1.265 + .01265x$$

and the second orthogonal polynomial is

$$p_2(x) = 3.742 - .08552x + .0004276x^2$$

```
> attach(seedrates)
```

```
> y <- model.matrix(seedrates.pol)[, 2]
```

```
> y.lm <- lm(y ~ rate + I(rate^2))
```

```
> coef(y.lm)
```

(Intercept)	rate	I(rate^2)
-1.265e+00	1.265e-02	3.917e-20

12

```
> y <- model.matrix(seedrates.pol)[, 3]
> y.lm <- lm(y ~ rate + I(rate^2))
> coef(y.lm)
```

```
(Intercept)      rate  I(rate^2)
 3.7416574 -0.0855236  0.0004276
```

Among other things, the polynomials given above have the property that

$$p_1(50)p_2(50) + p_1(75)p_2(75) + p_1(100)p_2(100) + p_1(125)p_2(125) + p_1(150)p_2(150)$$

since the values of the predictor are:

```
> rate
```

```
[1]  50  75 100 125 150
```

```
> detach(seedrates)
```