3

Multiple linear regression

Multiple linear regression generalizes straight line regression to allow multiple explanatory (or predictor) variables. The focus may be on accurate prediction. Or it may, alternatively or additionally, be on the regression coefficients themselves. Be warned that simple-minded interpretations of regression coefficients can be grossly misleading. Later chapters will elaborate on the ideas and methods developed in this chapter, applying them in new contexts.

Graphical and other diagnostics can be important sources of insight, bringing to attention common types of departure from model assumptions. A check of the residuals may identify one or more influential outliers that unduly skew the model fit and render it unsatisfactory for use for predictions with new data. A plot of partial residuals may show a systematic departure from linearity for a term that has been assumed linear.

3.1 Basic ideas: the `allbacks` book weight data

The data now considered have been put together for primarily didactic purposes. Seven books with hardback covers, together with 8 softbacks, were selected in a relatively haphazard fashion from the first author's shelves. Data are plotted in Figure 3.1 with selected rows printed to the right of the figure. Explanatory variables are the volume of the book ignoring the covers, and the total area of the front and back covers. Assuming that the hard covers are all similar in their construction, we might expect that

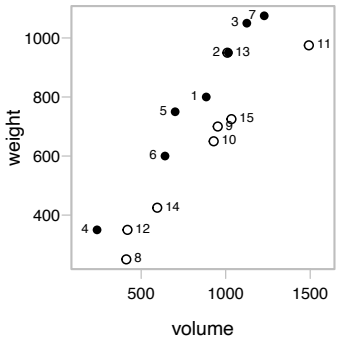
$$\text{weight of book } = b_0 + b_1 \times \text{volume } + b_2 \times \text{area of covers.}$$

For the moment, we will retain the intercept, b_0. It may not be needed.

Code used, with the output that relates to the regression coefficients, is:

```
allbacks <- DAAG::allbacks  # Place the data in the workspace
allbacks.lm <- lm(weight ~ volume+area, data=allbacks)
print(coef(summary(allbacks.lm)), digits=2)
```

```
            Estimate Std. Error t value    Pr(>|t|)
(Intercept)    22.41     58.402    0.38 0.707858178
volume          0.71      0.061   11.60 0.000000071
area            0.47      0.102    4.59 0.000616455
```



	volume (cm^3)	area (cm^2)	weight (g)	cover
1	885	382	800	hb
. . . .				
7	1228	396	1075	hb
8	412	0	250	pb
. . . .				
15	1034	0	725	pb

Figure 3.1 Weight versus volume, for 7 hardback and 8 softback books. Filled dots are hardbacks, while open dots are softbacks. Selected data are shown to the right of the graph.

The correlations between parameter estimates are:

```
## Correlations between estimates -- model with intercept
round(summary(allbacks.lm, corr=TRUE)$correlation, 3)
```

```
            (Intercept) volume   area
(Intercept)       1.000 -0.883 -0.318
volume           -0.883  1.000 -0.002
area             -0.318 -0.002  1.000
```

The intercept is given as $b_0 = 22.4$, with a p-value (=0.7) that suggests that it should be omitted. Other coefficient estimates are $b_1 = 0.708$ and $b_2 = 0.468$, with p-values that are so small that the precise value no longer has much meaning. It makes more sense to focus on the t-statistics, here 11.6 for `volume` and 4.6 for `area`, which convert the coefficients to multiples of their standard error.

The correlation between the coefficient for `volume` and that for `area` is -0.002, so the t-statistics are very nearly independent between coefficients. The final three lines of the default summary output are:

```
Residual standard error: 77.7 on 12 degrees of freedom
Multiple R^2:  0.928,    Adjusted R^2:  0.917
F-statistic: 77.9 on 2 and 12 DF,  p-value: 0.000000134
```

The estimate of the noise standard deviation (the "residual standard error") is 77.7. There are $15 - 3 = 12$ degrees of freedom for the residual; starting with 15 observations and 3 parameters were estimated. In addition, there are two versions of R^2. The F-statistic allows an overall test for the null hypothesis that coefficients other than the intercept are 0.

The 5% critical value for a t-statistic with 12 degrees of freedom, used for calculating 95% confidence intervals for coefficients, is 2.18.[1] Thus, a 95% confidence interval for `volume` is $0.708 \pm 2.18 \times 0.0611$, i.e., it ranges from 0.575 to 0.841. Here, because of the very small correlation between the coefficient estimates, these confidence intervals are for all practical purposes independent.

[1] `## 5% critical value; t-statistic with 12 d.f.`
`qt(0.975, 12)`

The default summary output includes coarse information on the distribution of residuals:

```
Residuals:
   Min     1Q Median     3Q    Max
-104.1  -30.0  -15.5   16.8  212.3
```

Note: The output for the data and associated model considered here, where both b_1 and b_2 are clearly required, contrasts with output where there are there a number of coefficients, with several whose p-values are around the conventional p=0.05 level that has commonly been used to judge 'significance', and with varying amounts of dependence. Omission of one variable will lead to large increases in the t-statistics, and smaller p-values, for variables with which it had a large positive correlation. Additionally, prior probabilities become important when p-values are in the region of 0.05 or somewhat smaller. Here, because of issues in the selection of books, and of available explanatory variables, there is a real question whether the results generalize in any meaningful way to a larger 'population' of books that may be of interest.

3.1.1 A sequential analysis of variance table

The `anova()` function outputs a sequential analysis of variance table that assesses the contribution of each predictor variable to the model in turn, given predictors from earlier rows of the table.

```
anova(allbacks.lm)
```

```
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value     Pr(>F)
volume     1 812132  812132   134.7 0.00000007
area       1 127328  127328    21.1    0.00062
Residuals 12  72373    6031
```

The contribution of `volume` after fitting overall mean is given first, then the contribution of `area` after fitting both the overall mean and `volume`. The p-value for area in the anova table must agree with that in the main regression output, since both these p-values test the contribution of `area` after including `volume` in the model. In general, the p-values for any earlier coefficients will differ from those shown in the table of coefficients and associated statistics shown earlier. Here, because the correlation between `volume` and `area` is so small as to be inconsequential, the difference is not apparent in the printed values.

The model matrix that has been used in the least square calculations is:

```
## Show rows 1, 7, 8 and 15 only
model.matrix(allbacks.lm)[c (1,7,8,15),  ]
```

```
   (Intercept) volume area
1            1    885  382
7            1   1228  396
8            1    412    0
15           1   1034    0
```

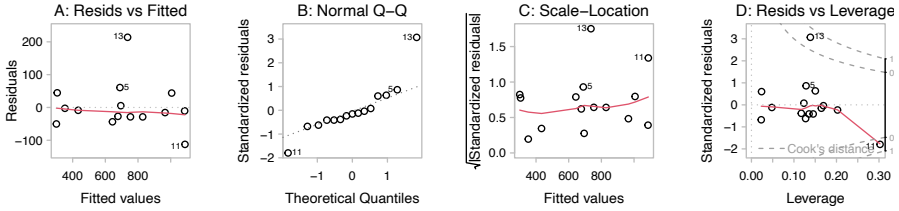


Figure 3.2 Diagnostic plots for the model that fits `weight` as a function of `volume` and `area`, omitting the intercept.

```
## NB, also, code that returns the data frame used
## model.frame(allbacks.lm)
```

Predicted values are given by multiplying the first column by b_0 (=22.4), the second by b_1 (=0.708), the third by b_2 (=0.468), and adding.

Omission of the intercept term

Now investigate the effect of leaving out the intercept, which had $p = 0.7079$ Regression output, when the intercept is omitted, is:

```
allbacks.lm0 <- lm(weight ~ -1+volume+area, data=allbacks)
print(coef(summary(allbacks.lm0)), digits=2)
```

```
       Estimate Std. Error t value         Pr(>|t|)
volume     0.73      0.028    26.3 0.0000000000011
area       0.48      0.093     5.1 0.0001879245500
```

The larger standard errors in the model that included the intercept were a consequence of the substantial negative correlations between the estimates for the intercept and those for `volume` and `area`. The reduction in standard error is greater for the coefficient of `volume`, where the correlation was −0.883, than for `area`, where the correlation was −0.318. (See Section 3.6.3)

Omission of the intercept term results in a substantial increase in the correlation between the coefficients for `volume` and `area`:

```
## Correlations between estimates -- no intercept
print(round(summary(allbacks.lm0, corr=TRUE)$correlation, 3))
```

```
       volume   area
volume  1.000 -0.635
area   -0.635  1.000
```

3.1.2 Diagnostic plots

The plots shown in Figure 3.2 provide graphical checks on the adequacy of the model fit to the `allbacks` data. Simplified code is: [2]

```
## The following has the default captions
plot(allbacks.lm0)
```

[2] `## To show all plots in the one row, precede with`
`par(mfrow=c(1,4))  # Follow with par(mfrow=c(1,1))`

Note the large residual in Panel A for observation 13. In Panel D, it lies outside the 0.5 contour of Cook's distance, well out towards the contour for a Cook's distance of 1. Thus, it is a (mildly) influential point. The Cook's distance measure, which was mentioned in Section 2.5.3, will be discussed in Subsection 3.4.2.

Should we omit observation 13? The first task is make such checks of the data as are possible. In this case, the purpose was served by checking back to the book itself, on the author's shelves. The book was a computing book, with a smaller height to width ratio than any of the other books. It had heavier paper, though differences in the paper were not immediately obvious. If the sample of books had been selected in a way that made generalization to some wider population of books meaningful, it might be legitimate to omit it from the main analysis, but noting that this one book (with a much higher weight to volume ratio than other books) had been omitted for purposes of the analysis. The following omits observation 13:

```
allbacks.lm13 <- lm(weight ~ -1+volume+area, data=allbacks[-13, ])
print(coef(summary(allbacks.lm13)), digits=2)
```

```
       Estimate Std. Error t value Pr(>|t|)
volume     0.69      0.016      43  1.8e-14
area       0.55      0.053      11  2.1e-07
```

The residual standard error is substantially smaller (41 instead of 75.1) in the absence of observation 13. Observation 11 now has a Cook's distance that is close to 1, but does not stand out in the plot of residuals. This is about as far as it is reasonable to go in the investigation of diagnostic plots.

With just 15 points, and books selected that were immediately available from one person's shelves, the small p-values should be treated with skepticism. The fitted model is unlikely to do well at predicting weights for a very different set of books on another set of shelves.

3.2 The interpretation of model coefficients

If an aim is a scientific understanding that involves interpretation of model coefficients, then it is important to fit a model whose coefficients are open to the relevant interpretations. Different formulations of the regression model, or different models, may serve different explanatory purposes. Predictive accuracy is in any case a consideration, and may be the main interest.

Three datasets will be considered. The first dataset has record times, distances, and amounts of climb for Northern Irish hill races. The second has data on book dimensions and weight, from a highly biased sample of books. The third has data on mouse brain weight, litter size, and body weight.

3.2.1 Times for Northern Irish hill races

The dataset `DAAG::nihills`, from which Table 3.1 has selected observations, gives distances (`dist`), heights climbed (`climb`), male record times (`time`) and female record times (`timef`), for 23 Northern Irish hill races. Initially, the interest will be in an equation that predicts `mph`, i.e., miles traversed per hour.

Table 3.1
Distance (`dist`), height climbed (`climb`), and record times (`time`), for four of the 23 Northern Irish hill races.

	Name	`dist` (mi)	`climb` (ft)	`time` (h)	`timef` (h)
1	Binevenagh	7.5	1740	0.86	1.06
2	Slieve Gullion	4.2	1110	0.47	0.62
3	Glenariff Mountain	5.9	1210	0.70	0.89
...	...	...	...	...	...
23	BARF Turkey Trot	5.7	1430	0.71	0.94

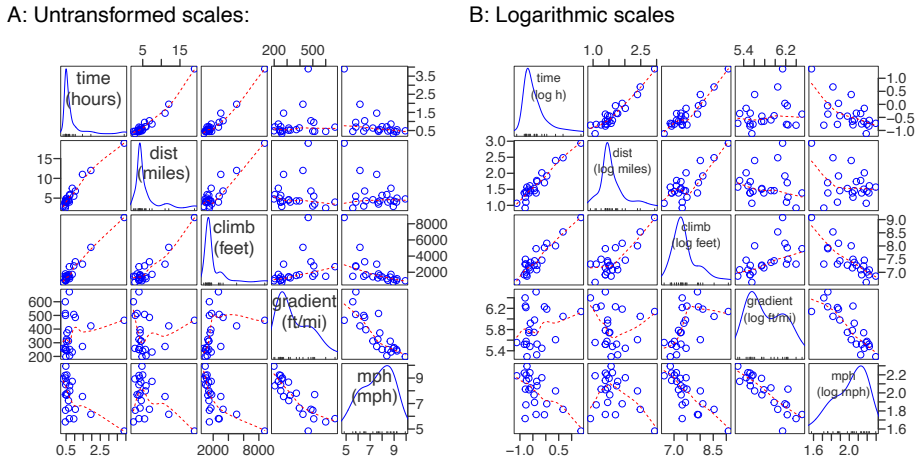


Figure 3.3 Scatterplot matrices for the `nihills` data (Table 3.1), drawn using the function `car::spm()`. Panel A uses untransformed scales, while Panel B uses logarithmic scales. The density plots in the diagonal panels give indications of the extent of distributional asymmetry. Smooth curves have been added to the individual plots. For instructions of how a measure of spread can be added around the smooths, see **Details** for `spread` under `?car::spm`.

Figure 3.3 shows scatterplot matrices, in Panel A for untransformed data, and in Panel B for log transformed data. The diagonal panels give the x-variable names for plots in the column (above or below), and the y-variable names for plots in the same row (left or right). Observe that the vertical axis labels alternate between the axis on the left and the axis on the right, and similarly for the horizontal axis labels. This avoids a proliferation of axis labels on the left and lower axes.

A simplified version of Panels A and B can be obtained by typing

```
nihr <- within(DAAG::nihills, {mph <- dist/time; gradient <- climb/dist})
nihr <- nihr[, c("time","dist", "climb", "gradient", "mph")]
sm <- list(spread=0, col.smooth='red')
car::spm(nihr, regLine=FALSE, col='blue', smooth=sm)
car::spm(log(nihr), regLine=FALSE, col='blue', smooth=sm,
         var.labels=paste("ln", names(nihr)))
```

The logarithmic transformation does make the distributions of `time`, `dist` and

`climb` more symmetric, and that for `gradient` mildly so. The transformation for `mph` pushes its mode somewhat more to the right. It is thus best left untransformed, if the alternative is a logarithmic transformation.

What is special about logarithmic transformations?

The effect of a logarithmic transformation is that values that differ by the same factor on the untransformed scale (e.g, 4/2, 8/4) are then the same absolute distance apart on the log transformed scale.

The following are ways in which a logarithmic transformation may help:

- Where here is a long tail to the right, the transformation can be expected to give a more symmetric distribution, with a much reduced tail to the right.
- In a regression that uses the untransformed variables, point(s) in the extreme right tail will, where the distribution has a long tail to the right, have a larger *leverage* in determining the regression coefficient than those closer to the mean — a point on which Subsection 3.4.2 will comment further. Even after taking logarithms In Figure 3.3B), where values have been log transformed, the leverage of the points with largest values of `time`, `dist` and `climb` remain large, but are less dominating.
- The transformation often makes good scientific sense. Because the physiological demands on the human athlete increase with `time`, it can be expected that `time` will increase more than linearly with `dist`, and similarly for `climb`. Working with logarithmically transformed variables allows for this possibility, though in a specific way.
- Where the variance increases with increasing values of the dependent variable, use of a logarithmic scale may help stabilize the variance.
- Such relationship as is evident between the explanatory variables may be more nearly linear on the logarithmic scale. Linear relationships make diagnostic plots more readily interpretable.
- The ratio of maximum to minimum value varies from 7.6 for distance to 12 for male times and 14.6 for female times. Values of `dist` vary by a factor of 7.56, and those of `climb` by a factor of 11.7. As a general guide, a logarithmic transformation should be considered if the ratio of maximum to minimum value is more than 5.

3.2.2 An equation that predicts `dist/time`

We fit two equations with `log(dist)` as the first of two explanatory variables. In the first of these `log(climb)` is used as a further explanatory variable, while in the second `log(gradient)` = `log(climb/dist)` is used as the second explanatory variable. Code that fits the models, with the two equations obtained, is:

```
##  Hold climb constant at mean on logarithmic scale
mphClimb.lm <- lm(mph ~ log(dist)+log(climb), data = nihr)
## Hold `gradient=climb/dist` constant at mean on logarithmic scale
mphGradient.lm <- lm(mph ~ log(dist)+log(gradient), data = nihr)
```

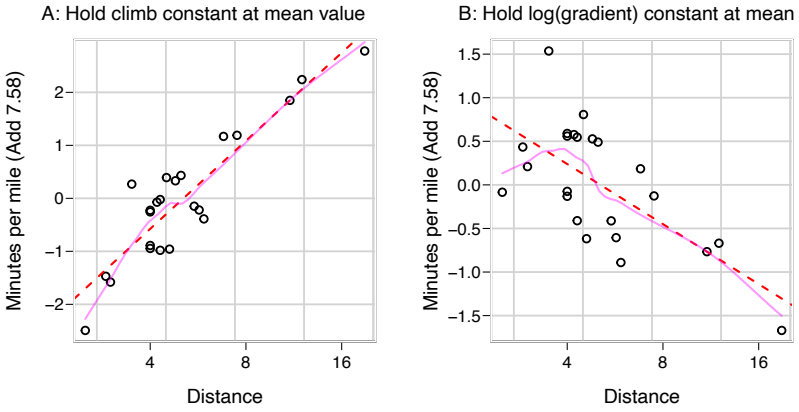


Figure 3.4 Panels A and B show "component plus residual" that relate to the variation in distance per unit time (in miles per hour) with distance. Panel A shows the pattern of change when `log(climb)` is held constant at its mean value, while Panel B shows the pattern of change when `log(climb/dist)` is held constant at its mean value. On the y-axis, tick labels show variation about a mean that equals 7.58.

```
avRate <− mean(nihr$mph)
bClimb <− coef(mphClimb.lm)
```

```
(Intercept)   log(dist)  log(climb)
     28.947       2.397      -3.390
```

```
  (Intercept)     log(dist) log(gradient)
      28.9475       -0.9937       -3.3903
```

The explanatory terms in two models differ only in the parameterization used, and give the same fitted values. The `plot` method for `lm` models gives the same diagnostic plots in the two cases.

Figure 3.4 uses the function `car::crPlots()` to show in a visually striking way the importance of the choice of the second variable for the interpretation of the coefficient of `log(dist)`. The plots that are produced, termed Component Plus Residual plots, are a variation on termplots, as described in 3.3.2. They have been used here in preference to termplots because they allow the replacement of default x-axis that are centered values of `log(dist)` values by values of `dist`. For each term specified in the `terms` argument, a plot is generated that shows how outcome values change as a function of values of that term when other explanatory variables are held constant.

Code that shows a simplified version of the plot in Figure 3.4A is:

```
car::crPlots(climb.lm, terms = . ~ log(dist), xaxt='n', xlab="Distance")
## `terms = . ~ log(dist)` picks out the part of the model formula for which
## the component+residual plot is required.  The `.` on the left of the `~`
## operator is shorthand for the response on the left of the model formula.
axis(1, at=log(2^(2:5)), labels=paste(2^(2:5)))
```

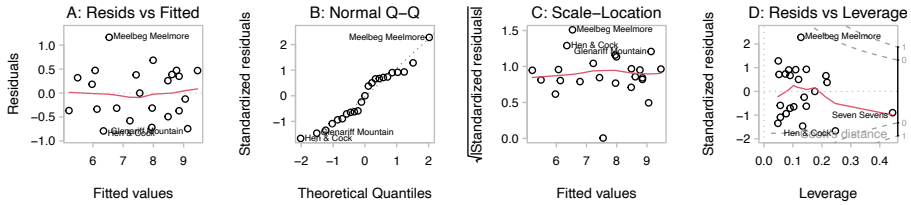


Figure 3.5 Diagnostic plots for the model `mphGradient.lm` that regressed `mph` on `logdist` and `loggradient`. The diagnostics are the same as for the model `mphClimb.lm`

The smooth suggests that the relationship is not quite linear. The difference made by using a model that allows for curvature is minor, as the reader may care to check. See further, Subsection 3.3.1.

This equation implies that for a given height of climb, the average rate (`mph`) at which the route is traversed increases with increasing distance. To understand what is happening, think carefully about the implications of holding `climb` constant. For a given value of `climb`, short races will be steep while for long races the gradient will be relatively gentle. Thus, the increase in average rate is not altogether surprising.

Correlations between the coefficients for the two models are:

```
summary(mphClimb.lm, corr=T)$correlation["log(dist)", "log(climb)"]
```

```
[1] -0.7801
```

```
summary(mphGradient.lm, corr=T)$correlation["log(dist)", "log(gradient)"]
```

```
[1] 0.06529
```

Thus, a side benefit of working with `mphGradient.lm` is that the two coefficients have negligible correlation.

Diagnostic plots are given in Figure 3.5

```
## Show the plots, with default captions
plot(mphClimb.lm, fg='gray')
```

The importance of contextual information for the interpretation of regression results will be a common theme in later examples.

3.2.3 Equations that predict `log(time)`

Models will be fitted with the same two respective choices of explanatory variables as before, but now with `log(time)` as the dependent variable. This would make good sense if the interest is in predicting the likely best time for a race that is run over a new route. The first equation to be fitted is:

$$\log(\texttt{time}) = a + b_1 \log(\texttt{dist}) + b_2 \log(\texttt{climb}),$$

```
lognihr <- setNames(log(nihr), paste0("log", names(nihr)))
timeClimb.lm <- lm(logtime ~ logdist + logclimb, data = lognihr)
```

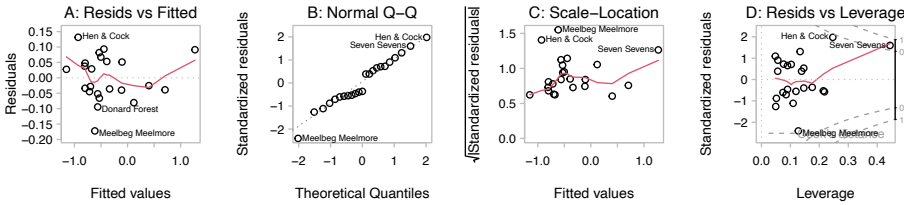


Figure 3.6 Diagnostic plots for the regression of `logtime` on `logdist` and `logclimb`.

Figure 3.6 shows the diagnostic plots:

```
## Show the plots, with default captions
plot(timeClimb.lm, fg='gray')
```

The diagnostic plots do not indicate any major issue. The Meelbeg Meelmore race has a moderately large residual. There is a hint of saucer-shaped curvature in the plot of residuals against fitted values. We will look at this in more detail shortly.

The estimates of the coefficients (a, b_1 and b_2) are:

```
print(coef(summary(timeClimb.lm)), digits=2)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    -4.96       0.274     -18  7.1e-14
logdist         0.68       0.055      12  8.2e-11
logclimb        0.47       0.045      10  2.0e-09
```

The fitted equation is

$$\underset{}{\texttt{log(time)} = } \underset{[\text{SE}=0.27]}{-4.96} \underset{[\text{SE}=0.055]}{+0.68} \times \texttt{log(dist)} + \underset{[\text{SE}=0.045]}{0.47} \times \texttt{log(climb)}$$

Exponentiating both sides of this equation, and noting that $\exp(-4.96) = 0.0070$, gives

$$\texttt{time} = 0.0070 \times \texttt{dist}^{0.68} \times \texttt{climb}^{0.0.47}.$$

This equation implies that for a given height of climb, the time taken to traverse a given distance is less for longer races. The relative rate of increase in time is 68% of the relative rate of increase in distance. As before, the issue is that short races will be steep while for long races the gradient will be relatively gentle.

Now regress on `logdist` and `log(climb/dist)`:

```
timeGradient.lm <- lm(logtime ~ logdist + loggradient, data=lognihr)
print(coef(summary(timeGradient.lm)), digits=3)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -4.961     0.2739   -18.1 7.09e-14
logdist        1.147     0.0346    33.2 5.90e-19
loggradient    0.466     0.0453    10.3 1.98e-09
```

The coefficient of `logdist` is now, reassuringly, greater than 1. A related benefit

is that the correlation between `logdist` and `logradient` is -0.065, which is negligible relative to the correlation of 0.78 between `logdist` and `logclimb`.[3] Because the correlation between `logdist` and `loggradient` is so small, the coefficient of `logdist` (=1.124) in the regression on `logdist` alone is almost identical to the coefficient of `logdist` (=1.147) in the regression on `logdist` and `logGradient`.

The standard error of the coefficient of `logdist` is smaller, 0.035 as against 0.055, when the second explanatory variable is `logGradient` rather than `logclimb`. Note that the predicted values do not change. The models `timeClimb.lm` and `timeGradient.lm` are different parameterizations of the same underlying model.

3.2.4 Book dimensions — the `oddbooks` dataset

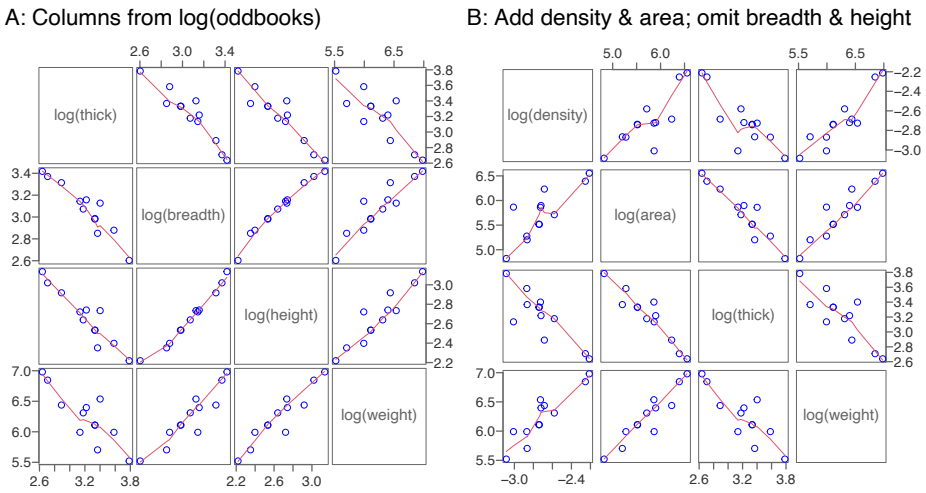


Figure 3.7 Panel A shows the scatterplot matrix for the logarithms of the variables in the `oddbooks` data frame. Panel B has the derived variables `log(density)` and `log(area)`, plus original variables `log(thick)` and `log(weight)`. These plots used the function `pairs()` from the base R *graphics* package.

The way that data are sampled can lead to large biases in the estimated coefficients. Figure 3.7A shows a scatterplot matrix for logged measurements, from the data frame `oddbooks`, on twelve soft-cover books.[4] The books were taken from one particular section of one particular bookshelf.

Figure 3.7B repeats two of the variables from Panel A, adding two derived vari-

[3]
```
## Correlations of logGrad and logclimb with logdist
with(lognihr, cor(cbind(loggradient, logclimb), logdist))
```

[4]
```
## Code for Panel A, omitting the title; use 'pairs()'
oddbooks <- DAAG::oddbooks
pairs(log(oddbooks), lower.panel=panel.smooth, upper.panel=panel.smooth,
```

ables:[5] Books were selected in such a way that weight increased with decreasing thickness, reflected in the strong negative correlation between log(`weight`) and log(`thick`).

The interest will be in what the regression could tell us if the dimensions (`thick`, `breadth`, `height`) were all the information available that might explain the weight, and this was a context where these did not multiply to give a volume to which the weight could be directly related.

We will start by fitting all three explanatory variables, then using the smallest AIC statistic from `drop1()` to identify the 'best' of the two variable equations:

```
lob3.lm <- lm(log(weight) ~ log(thick)+log(breadth)+log(height),
              data=oddbooks)
# coef(summary(lob3.lm))
```

We leave readers to examine the table of coefficients and standard errors, with p-values that, rounded to 2 decimal places, range from 0.12 for `log(breadth)` to 0.91 for `log(height)`. The AIC statistics for single term deletions are:

```
setNames(drop1(lob3.lm)$AIC, rownames(drop1(lob3.lm)))
```

```
      <none>   log(thick) log(breadth)  log(height)
      -40.68        -41.07        -38.77        -42.66
```

The resulting equation is:

```
lob2.lm <- lm(log(weight) ~ log(thick)+log(breadth), data=oddbooks)
coef(summary(lob2.lm))
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.5028     2.5202 -0.1995 0.846285
log(thick)     0.4507     0.3950  1.1411 0.283297
log(breadth)   1.9904     0.4845  4.1084 0.002643
```

Output from `add1()` with the model that has only the constant term will be used to choose between the three models with a single explanatory variable. We will then fit the model that appears preferred:

```
lob0.lm <- lm(log(weight) ~ 1, data=oddbooks)
add1(lob0.lm, scope=~log(breadth) + log(thick) + log(height))
```

```
Single term additions

Model:
log(weight) ~ 1
             Df Sum of Sq   RSS   AIC
<none>                    2.033 -19.3
log(breadth)  1      1.79 0.238 -43.0
log(thick)    1      1.43 0.598 -32.0
log(height)   1      1.73 0.302 -40.2
```

```
lob1.lm <- update(lob0.lm, formula=. ~ .+log(breadth))
```

[5]
```
## Panel B, omitting the title
oddothers <- with(oddbooks,
  data.frame(density = weight/(breadth*height*thick),
  area = breadth*height, thick=thick, weight=weight))
pairs(log(oddothers), lower.panel=panel.smooth, upper.panel=panel.smooth, gap=0.5)
```

Coefficients in the fitted equations, with SEs in square brackets underneath, are:

$$\begin{aligned}
\log(\text{weight}) &= \underset{[3.2]}{-0.72} + \underset{[0.43]}{0.46}\log(\text{thick}) + \underset{[1.07]}{1.88}\log(\text{breadth}) + \underset{[1.27]}{0.15}\log(\text{height}) \\
\log(\text{weight}) &= \underset{[2.52]}{-0.5} + \underset{[0.49]}{0.35}\log(\text{thick}) + \underset{[0.48]}{1.99}\log(\text{breadth}) \\
\log(\text{weight}) &= \underset{[SE=0.45]}{2.33} + \underset{[0.17]}{1.47}\log(\text{breadth})
\end{aligned}$$

The predicted values for the three very different models are very similar:

```
round(rbind("lob1.lm"=predict(lob1.lm), "lob2.lm"=predict(lob2.lm),
            "lob3.lm"=predict(lob3.lm)),2)
```

```
          1    2    3    4    5    6    7    8    9   10   11   12
lob1.lm 6.94 6.77 6.62 6.33 6.21 6.36 6.06 6.06 5.79 6.35 5.86 5.59
lob2.lm 6.93 6.73 6.61 6.33 6.18 6.40 6.04 6.04 5.70 6.47 5.89 5.62
lob3.lm 6.92 6.73 6.61 6.33 6.18 6.41 6.04 6.04 5.70 6.47 5.89 5.61
```

Figure 3.7A made it clear that `weight`) increases with increasing values of `density`. In essence, it is the omission of `log(density)` from the regression equations that has skewed the regression coefficient estimates. The plot of `log(weight)` against `log(density)` in Panel B shows the strong relationship. The regression equation is:

```
oddbooks <- within(oddbooks, density <- weight/(thick*breadth*height))
lm(log(weight) ~ log(density), data=oddbooks) |> summary() |> coef() |>
  round(3)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    10.284      0.586  17.555        0
log(density)    1.492      0.215   6.924        0
```

As a check, the following code generates a message to say that the equation gives what is essentially a perfect fit:

```
## Not run. It is left to the reader to run this code.
lm(log(weight) ~ log(thick)+log(breadth)+log(height)+log(density),
   data=oddbooks) |> summary() |> coef() |> round(3)
```

The `oddbooks` dataset was contrived to give a skewed picture of the way that book weight varies with dimensions. Correlations between `area` and `thick`, and between both `area` and `thick` and `density`, make it impossible to separate the effects of these three variables. Solid results require use of equations that capture the relevant physical relationships. Where the best that can be done is to guess, it is hazardous to try to attach a causal interpretation to regression coefficients.

For the `oddbooks` data, we know how the relevant variables (`thick`, `breadth`, `height`, `density`) drive and in that sense 'cause' 'weight', and could verify that leaving `log(density)` out of the equation gives coefficients that are uninterpretable. With observational data, there will in general be no way to know for sure how results may be affected by variables that have been left out or incorrectly modeled. For the `oddbooks` data, it was important that effects were additive on a logarithmic scale.

Observational data is very susceptible to such bias. For example, solar radiation, windspeed, temperature and rainfall may change systematically with distance up

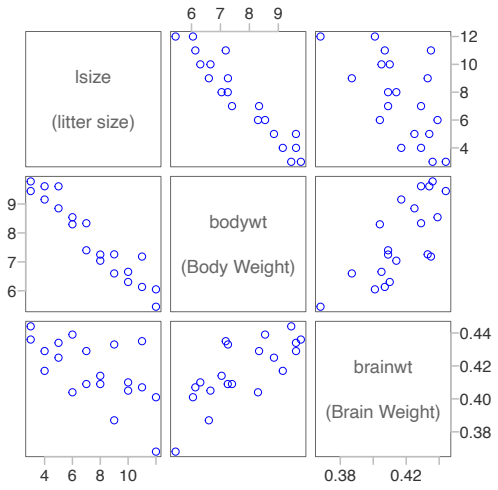


Figure 3.8 Scatterplot matrix for the litters dataset. Data relate to Wainright et al. (1989).

a hillside, making it impossible to distinguish the effects of the different factors on plant growth or on the ecology. There may be effects, crucial for making sense of the data, that are not obvious from the data themselves.

3.2.5 Mouse brain weight example

The `DAAG::litters` data frame has values of brain weight, body weight, and litter size for each of 20 mice. As Figure 3.8 makes clear, the explanatory variables `lsize` and `bodywt` are strongly correlated. Stripped down code for Figure 3.8 is:

```
litters <- DAAG::litters
pairs( litters )
```

Observe now that, in a regression with `brainwt` as the response variable, the coefficient for `lsize` has a different sign (−ve versus +ve) depending on whether `bodywt` also appears as an explanatory variable. Here are the calculations:

```
## Regression of brainwt on lsize
summary(lm(brainwt ~ lsize, data = litters),  digits=3)$coef
```

```
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  0.447000   0.009625  46.443 3.391e-20
lsize       -0.004033   0.001198  -3.366 3.445e-03
```

```
## Regression of brainwt on lsize and bodywt
summary(lm(brainwt ~ lsize + bodywt, data = litters), digits=3)$coef
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.17825   0.075323   2.366 0.030097
lsize        0.00669   0.003132   2.136 0.047513
bodywt       0.02431   0.006779   3.586 0.002278
```

The coefficients have different interpretations in the two cases:

- In the first regression, variation in `brainwt` depends on `lsize`, regardless of `bodywt`. No adjustment has been made for the increase in `bodywt` as `lsize` decreases. Individuals from small litters (small `lsize`) have, on average, large large `bodywt` and large `brainwt`. Individuals from large litters have, on average, low `bodywt` and low `brainwt`.
- In the multiple regression, the coefficient for `lsize` is a measure of the change in `brainwt` with `lsize`, when `bodywt` is held constant. For any particular value of `bodywt`, `brainwt` increases with `lsize`. This was a noteworthy finding for the purposes of the study.

The results are consistent with the biological concept of brain sparing, whereby the nutritional deprivation that results from large litter sizes has a proportionately smaller effect on brain weight than on body weight.

3.2.6 Issues for causal interpretation

The literature on conditions and checks that are needed in contexts where the hope is to give a regression coefficient a causal interpretation is large and growing. In an introduction titled "Towards Less Casual Causal Inferences", Hernán and Robins (2020) comment on the need to bring together evidence from multiple sources (in what is termed 'triangulation') and consider multiple methodological approaches. They go on to comment:

> No book can possibly provide a comprehensive description of methodologies for causal inference across the sciences.

Note also the influential Bradford Hill criteria, discussed at length in Höfler, 2005, with extensive accompanying commentary. With the limited attention given in this present text to issues of causation, the best that can be done is to highlight some of the important themes.

There may be variables whose effects are not of primary interest, but which nonetheless have important explanatory power. These are termed *covariates*. Where they affect the both the outcome and the variable of interest, they are known as *confounders*. As the number of variates and covariates increases, it becomes an increasing challenge to find a model that effectively accounts for observed outcomes, and to convincingly justify causal interpretations.

The following are contexts where coefficients may, with reasonable confidence, be interpretable:

- The coefficient of interest is not much affected by the choice of covariates, once all clearly relevant covariates are included. This will happen if the variable of interest is independent of such covariates.
- The effect is so large relative to other influences (as, e.g., in the effect from smoking in many health studies) that its contribution is not in doubt.
- The regression model reflects well-understood scientific laws.

- Accidents of nature or society have created conditions that, it can be argued, closely mirror the requirements for a randomized experiment. For example, in a large-scale study of a chemical pollutant, it may be possible to identify cases where one of a pair of identical twins has been exposed to the pollutant, while the other has not.
- Different sources of evidence, with different biases, all point to the same conclusion. Thus, it was a confluence of evidence, including the development of an understanding of the mechanisms involved, that settled debate on the link between smoking and lung cancer.
  - Subgroups can in some instances be analyzed separately in a manner that provides what are effectively independent sources of evidence. There may be grounds for arguing that any biases are likely to go in different directions.

Where the groups that are compared may differ on more than one or two covariates, any use of regression methods that claims to identify a causative effect has to meet stringent requirements:

- All relevant covariates have been taken into account;
- There must have been checks that allow for the possibility of nonlinear effects and/or interactions;
- It has to be established that causation goes from from the variable or factor to the dependent variable, conditional on other covariates being held constant.

These are difficult to demonstrate convincingly, unless the groups are already closely matched on everything except the variable or factor whose effect it is hoped to demonstrate. A plot akin to Figure 9.24 should be provided as a matter of course. Propensity scores, described in Subsection 9.7.2, provide a more limited one-dimensional comparison. Matching approaches, described in Section 9.7, can be an effective complement to regression methods. There remain issues of how close the matches need to be.

Where there is complexity in the causal pathways, graphs in the style of "directed acyclic graphs" (DAGs) can be helpful in making clear what is known and what is assumed. Cunningham (2021, pp 96-118) is a helpful overview of basic concepts and terminology. Vignettes that accompany the *ggdag* package provide a brief introduction to some of the important ideas.

Effects of lifestyle on health

Does cutting down on sugar reduce risk to health, in particular from Type 2 diabetes? Is it important whether sugar comes from fruit, or from cooked or processed foods that contain large amounts of refined sugar? Are fats a major problem? If so, which fats? Are ketogenic diets – low carbohydrate and high fat – effective as claimed? Lichtenstein et al. (2021) is an assessment of the evidence on effects of diet on cardiovascular health. How important is regular exercise? What is the effect of environmental pollution? What is the effect of moderate alcohol consumption.[6]

[6] See the fact sheet at
`https://www.cdc.gov/alcohol/fact-sheets/moderate-drinking.htm`

It is relatively easy to demonstrate that a given factor is associated with good health but difficult to know the extent the health effects are caused by the factor rather than just an indicator of a generally healthy diet and lifestyle.

Mokdad et al. (2018) used results from multiple studies to bring together and balance carefully critiqued evidence from a wide range of studies on human health effects, for the population of the United States between 1990 and 2016. Their assessments (notably their Figure 2) suggested that, depending on the measure used, dietary effects on risk of death were around six times those from low physical activity, and similar to those of tobacco use. Air pollution appeared a somewhat greater risk than low physical activity.

Contrast the assessments in Mokdad et al. (2018) with those in Paluch et al. (2021). Paluch et al report on the relationship between steps per day and all-cause mortality for 5115 adults, aged 18 to 35 when recruited in 1985 and 1986 for a prospective study. Participants were from four locations in the US, with the sample balanced to reflect the wider population in race (black/white), sex, and age distribution. They wore an accelerometer over 2005 to 2006, and were then followed for a mean of 10.8 years, with deaths recorded through to 2018. Adjustments were made for age, sex, race, accelerometer wear time, education, center, BMI[7], smoking, and alcohol (with or without diet variable).

Three categories were compared: under 7000 steps per day (used as baseline), 7000 to <10,000, and ≥10,000. The hazard rate, measuring risk of death given survival up to a point in time, were reported as reduced, for the two higher step rate groups relative relative to baseline, as:

7000 to <10,000 steps per day: 0.22 (95% CI 0.11 - 0.43)
≥10,000 steps per day: 0.29 (0.15 - 0.54)

These estimates and confidence intervals (see Paluch et al's Table 6) were reported as exactly the same irrespective of whether a "healthy eating index" (a diet variable) was included in the model.[8]

The results are not readily reconciled with those of Mokdad et al. (2018). Was number of steps per day serving as a measure of general fitness rather than measuring a direct effect on the risk of death?

The studies mostly agree. But what do they say?

A number of studies have found that, as one might expect, mortality in the first year after birth was higher for babies of mothers who smoked during pregnancy than for non-smokers. On the other hand, if attention is limited to low birthweight babies, here defined as weighing less than 2.5kg at birth, the risk was lower when the mother was a smoker. The likely explanation is that a baby may have a low birthweight for reasons other than having a mother who smokes. The other factors, whatever they are, bring a risk of death that is greater than that of smoking as the primary cause of low birthweight. See Hernández-Díaz et al. (2006), and the

[7] Body Mass Index, used as a measure of obesity
[8] This seems unlikely. Was this a mistake? Presumably they were not much different.

commentary in Wilcox (2006). The authors state that, for babies with a birthweight of less than 2kg, the mortality for non-smoking mothers relative to smokers is 0.79. This appears to contradict a graph where the ratio is never less than around 0.96. Just possibly, the 0.79 figure is an artifact of very different distributions of numbers within the under 2kg category. In any case, the 0.96 figure is the more pertinent.

Adjusting for confounders

Hernández-Díaz et al. (2006) comment

> It is the mantra of observational studies that we can never rule out unobserved confounding. Perhaps we need a second mantra: Never adjust for covariates just because they are handy.

The birthweight paradox provides a further example of the need to think carefully about the causal pathways involved when interpreting regression coefficients. For the `nihills` data, it was necessary to recognize the very different consequences that would flow from holding `climb`, as opposed to `gradient`, constant. Especially where a number of covariates are involved, the physical processes determine the causal pathways may, as with the birthweight data, be difficult or impossible to identify. The key requirement is that, as noted in Imbens (2015) "the comparison of units with different treatments but identical pretreatment variables can be given a causal interpretation." All confounders must be identified, and their role correctly modeled. Effects may not be linear, and interactions must be accounted for. Glass et al. (2013) discuss the issues involved in detail. Because the causal pathways are usually not well understood, regression coefficients will often be suggestive rather than definitive.

Where there are two groups to be compared, a propensity score may sometimes be effective. The score, measuring the 'propensity' of observations to belong to one group rather than the other, is used to replace the explanatory terms in the model that would otherwise be needed to account for group differences other than the factor (e.g., a treatment effect) of interest. While it will sometimes be possible to show that a propensity score is not doing the task required of it, it is in general not possible to verify empirically that a propensity score has been effective in accounting for all 'nuisance' differences. See further Subsection 9.7.2.

3.3 Choosing the model, and checking it out

As a preliminary to setting out a general strategy for fitting multiple regression models, we describe the assumptions that underpin multiple regression modeling. Can the equation be trusted to give reliable predictions for a suitably defined population from which the data have been taken? The meaning that can be attached to individual coefficients are not, for purposes of the checks that will be described here, an immediate concern.

We have explanatory variables $x_1, x_2, \ldots, x_p$. Or, more generally, the x_i may be columns of the model matrix:

- The expectation $E[y]$ is some linear combination of $x_1, x_2, \ldots, x_p$:

$$E[y] = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p. \tag{3.1}$$

- The distribution of y is normal with mean $E[y]$ and constant variance, independently between observations.

Independence and constant variance are crucial. The role of normality can be overstated. For inference regarding model coefficients, it is enough that the sampling distributions of the coefficients are close to normal.[9]

The assumption that $E[y]$ is a linear combination of the explanatory variables is a common starting point for investigation. Diagnostic plots, and other checks, are important, both in detecting common types of failure of assumptions and in indicating the nature of the failure. Assumptions may be hard to fault if the noise component of the variation in y is a substantial fraction of the variation for which the model is able to account. It becomes, for example, harder to detect any nonlinearity in the effects of explanatory variables.

3.3.1 Criteria for model choice

Criteria that may influence the choice of model include:

1. The model should do an effective job in accounting for the bulk of the data. Models where a few very large points (or, just possibly, a few very small points) determine the form of the fitted model are unsatisfactory.
2. Equation 3.1 implies that, if other variables in the model are held constant, the relationship between y and x_i will be a straight line, with independent and normally distributed errors. That is, the relationship between y and x_i is linear, conditional on $x_1, \ldots x_{i-1}, x_i, \ldots, x_p$. If transformations can be found such that most of the p pairwise scatterplots of y against x_i appear linear, that will be a good starting point for identifying an appropriate linear model. It is then likely, though not guaranteed, that the unconditional relationships will be linear or close to linear,
3. Models should, where relevant, be scientifically meaningful.

In Subsection 3.2.1 (Figure 3.3), a comparison of the scatterplot matrix for the untransformed data with that for the log transformed data was used to justify the use of log transformed variables. Figure 3.3B is preferable to Figure 3.3A on both of the criteria 1 and 2. A more general approach, in which logarithmic transformations are viewed as a special case of power transformations, will be discussed in Subsection 3.3.3

Linear relations between y and individual x_i imply, also, that there will be linear relations between the x_i. One possibility is to look for transformations, for each variable separately, that give a distribution that is close to symmetric. For power

[9] Strictly, the requirement is that the joint distribution is close to multinormal. Normality of the individual distributions will in most practical contexts ensure this.

transformations, the function `car::powerTransform()` can be used to check for transformations for all variables at the same time. See Subsection 3.3.3.

When considering possible fine tuning of an initially chosen model that makes reasonable scientific and statistical sense, care is needed that such fine tuning does bring with it selection effects that may in fact reduce predictive power.

3.3.2 Plots that show the contribution of individual terms

Termplots, implemented using R's function `termplot()`) or as *component + residual plots* by the function `car::crPlots()`, take advantage of a point noted in connection with Equation 3.1 — conditional on all variables except x_i, the plot of y against x_i should exhibit random normal scatter about a straight line.

For simplicity, assume a model with three explanatory variables: x_1, x_2, and x_3, as in the `oddbooks` data. The interest is in examining the contribution of each in turn to the model. The fitting of a regression model makes it possible to write:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + e \tag{3.2}$$

$$= \widehat{y} + e, \quad \text{where } \widehat{y} = b_0 + b_1 x_1 + b_2 x_2 \tag{3.3}$$

Another way to write the model that is to be fitted is:

$$y - b_0 = b_1(x_1 - \bar{x}_1) + b_2(x_2 - \bar{x}_2) + b_3(x_3 - \bar{x}_3) + e$$

It is a fairly straightforward algebraic exercise to show that $b_0 = \bar{y}$. Thus, we have:

$$y - \bar{y} = t_1 + t_2 + t_3 + e$$

where $t_1 = b_1(x_1 - \bar{x}_1)$, $t_2 = b_2(x_2 - \bar{x}_2)$, $t_3 = b_3(x_3 - \bar{x}_3)$.

Termplots are an exercise in leaving out each of t_1, t_2 and t_3 in turn. Omission of t_1 from $t_1 + t_2 + t_3 + e$ leaves $t_2 + t_3 + e$ to be explained by $t_1 = b_1(x_1 - \bar{x}_1)$. The quantity $r_1 = t_2 + t_3 + e$ is the partial residual for x_1. Then each of the following plots should show random variation about a line:

1. $r_1 = t_2 + t_3 + e$ against x_1
2. r_2 against x_2, which leaves t_2 out
3. r_3 against x_3, which leaves t_3 out

If one or more of the plots shows clear nonlinearity in the scatter of points, this is an indication that a more complex model is needed.

In the terminology of *Component + Residual plots*, the quantities r_i have the form *component + residual*. Thus, for $r_1 = t_2 + t_3 + e$, $t_2 + t_3$ is the component, while e is the residual; the sum r_1 is plotted against x_1. The argument generalizes in the obvious way when there are more than three terms.

The `predict()` function has an option (`type="terms"`) that gives the t_i, here for the model fitted to the `oddbooks` data.

```
oddbooks.lm <- lm((weight) ~ log(thick)+log(height)+log(breadth),
data=DAAG::oddbooks)
yterms <- predict(oddbooks.lm, type="terms")
```

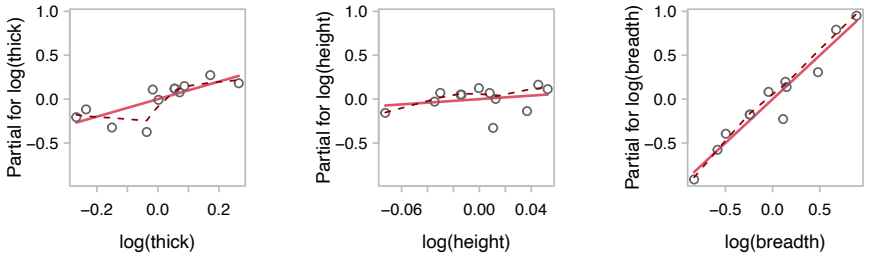


Figure 3.9 The solid lines in the termplots show the respective contribution of the model term, in the regression of `log(weight)` on `log(thick)`, `log(height)` and `log(breadth)`. Partial residuals (specify `partial.resid=TRUE`), and an associated smooth curve (specify `smooth=panel.smooth`) have been added. With `transform.x=TRUE` the fitted responses appear as straight lines.

The first column of `yterms` has values of t_1, the second has values of t_2, and so on. This information is used to construct the *component plus residual* plots that are given by R's function `termplot()`. (The function does not return anything sensible for terms where interactions are involved, e.g., `x1 + x1:fac` where `fac` is a factor.)

Figure 3.9 shows the contributions of the individual terms to the model. The solid line in left panel plots $b_1(x_1 - \bar{x}_1)$ against x_1, while the solid line in right panel plots $b_2(x_2 - \bar{x}_2)$ against x_2. Here is code that may be used to create the plots:

```
## To show points, specify partial.resid=TRUE
## For a smooth curve, specify smooth=panel.smooth
termplot(oddbooks.lm, partial.resid=TRUE, smooth=panel.smooth,
 col.res ="gray30", transform.x=TRUE)
```

If new log transformed variables are created and used in the model formula, i.e., `logthick = log(thick)`, etc., argument `transform.x = TRUE` is unnecessary.

3.3.3 A more formal approach to the choice of transformation*

Subsection 2.5.6 drew attention to the power family of transformations, defined for this purpose by:

$$\tilde{y} = \frac{y^\lambda}{\lambda - 1} \tag{3.4}$$

This formulation, used in place of the simpler $\tilde{y} = y^\lambda$, has the advantage that the logarithmic transformation then corresponds to $\lambda = 0$; it is the limiting transformation as λ goes to zero. Values of λ between 0 and 1 give transformations that lie, in a mathematically meaningful sense, between the logarithmic and no transformation. If evident right skewness remains after taking logarithms, something "stronger" than a log transformation is required. A negative λ may be effective.

The function `car::powerTransform()` is designed, when applied to values of a single variable, to guide the choice of a power transformation that brings the distribution as close as possible to normal. Optimizing for closeness to normality

can be expected to yield, where the data allow it, an approximately symmetric and unimodal distribution.[10] The following are possible alternative modes of use:

- If applied to a matrix or data frame, a transformation will be found for each of the columns, designed to bring the joint distribution as close as possible to multivariate normality. The indicated transformations for the explanatory variables may be different, depending on whether the outcome variable is included.
- If applied to a regression model object, it finds a transformation of the outcome variable such that the distribution of residuals is as close as possible to normality.

Application of the function to the columns `mph`, `dist` and `gradient` of the data frame `nihr` with which we worked in Section 3.2.1 yields:

```
## Use car::powerTransform
nihr <- within(DAAG::nihills, {mph <- dist/time; gradient <- climb/dist})
summary(car::powerTransform(nihr[, c("dist", "gradient")]),  digits=3)
```

```
bcPower Transformations to Multinormality
         Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
dist       -0.9709          -1       -1.782       -0.1594
gradient   -0.4833           0       -1.830        0.8631

Likelihood ratio test that transformation parameters are equal to 0
 (all log transformations)
                          LRT df  pval
LR test, lambda = (0 0) 6.533  2 0.038

Likelihood ratio test that no transformations are needed
                          LRT df       pval
LR test, lambda = (1 1) 30.78  2 0.00000021
```

For `dist` and `gradient`, the intervals with endpoints defined by the Wald lower and upper 95% confidence interval bounds contains 0, suggesting that a log transformation is a reasonable choice, while `mph` might,be left untransformed.

If we allow `powerTransform()` to choose the transformation that does best in providing normally distributed residuals, we find:

```
form <- mph ~ log(dist) + log(gradient)
summary(car::powerTransform(form, data=nihr))
```

```
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
Y1     1.009           1      -0.4183        2.437

Likelihood ratio test that transformation parameter is equal to 0
 (log transformation)
                        LRT df pval
LR test, lambda = (0) 1.985  1 0.16

Likelihood ratio test that no transformation is needed
                          LRT df pval
LR test, lambda = (1) 0.000157  1 0.99
```

[10] In fact, `powerTransform()` implements two families of power transformations — the Box-Cox family as defined in Equation 3.4, and the more general Yeo-Johnson family of transformations that will be used in Subsection 3.3.5.

GAM models, discussed in Subsection 4.4.2, provide a more general context in which to examine whether a parametric model adequately captures the contributions of the several terms. GAM models are able, in principle, to capture arbitrary curvilinear contributions to the model fit.

The use of transformations — further comments

Often there are scientific reasons for transformations. Thus, suppose we have weights w of individual apples, but the effects under study are more likely to be related to surface area. It then makes sense to consider using $x = w^{\frac{2}{3}}$ as the explanatory variable. If the interest is in studying relative, rather than absolute, changes, consider working with the logarithms of measurements.

A logarithmic transformation may both remove an interaction and give more nearly normal data. It can on the other hand introduce an interaction where there was none before. Or a transformation can reduce skewness while increasing heterogeneity. The availability of direct methods for fitting special classes of model with non-normal errors, for example the generalized linear models that we will discuss in Chapter 5, has reduced the need for transformation prior to analysis.

3.3.4 Accuracy estimates, for fitted values and for new observations

The interest may be in accuracy assessments for one or more fitted values. Or it may be in the accuracy with which the fitted value predicts the value for a new observation. The following table gives both 95% coverage (confidence) intervals and 95% prediction intervals for `time`, for the first 4 observations. For predicting future observations, scatter about the fitted values needs to be taken into account, resulting in much wider intervals. Note the change from the default `interval="confidence"` to `interval="prediction"` in the call to `predict()`.

```
lognihr <- log(DAAG::nihills)
names(lognihr) <- paste0("log", names(lognihr))
timeClimb.lm <- lm(logtime ~ logdist + logclimb, data = lognihr)
## Coverage intervals; use exp() to undo the log transformation
citimes <- exp(predict(timeClimb.lm, interval="confidence"))
## Prediction intervals, i.e., for new observations
pitimes <- exp(predict(timeClimb.lm, newdata=lognihr, interval="prediction"))
## fit ci:lwr ci:pwr pi:lwr pi:upr
ci_then_pi <- cbind(citimes, pitimes[,2:3])
colnames(ci_then_pi) <- paste0(c("", rep(c("ci-","pi-"), c(2,2))),
                               colnames(ci_then_pi))
## First 4 rows
print(ci_then_pi[1:4,], digits=2)
```

```
                   fit ci-lwr ci-upr pi-lwr pi-upr
Binevenagh        0.89   0.85   0.94   0.75   1.06
Slieve Gullion    0.49   0.47   0.51   0.41   0.58
Glenariff Mountain 0.64   0.60   0.68   0.54   0.76
Donard & Commedagh 1.13   1.07   1.19   0.95   1.33
```

Figure 3.10A shows these confidence and prediction intervals graphically. The narrower 95% confidence bands are in red, while the wider 95% prediction bands

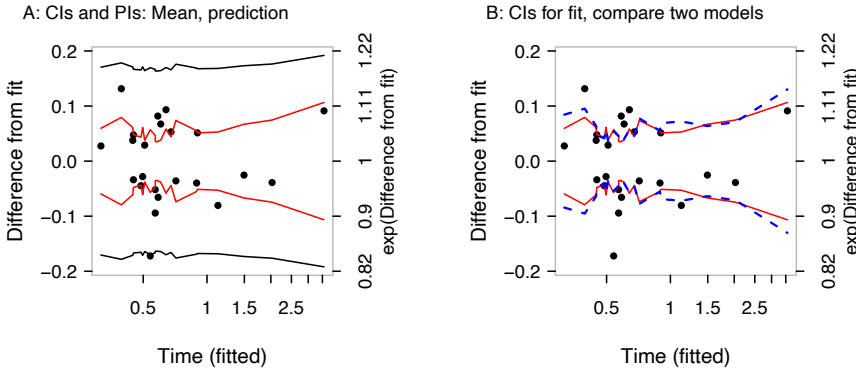


Figure 3.10 Both panels show differences of observed values (black dots) and differences of predicted confidence intervals (in red), from fitted values, with fitted values on the x-axis. As these differences are on a logarithmic scale, the back transformed values (take exponents) are Observed/Fit, where 'Fit' is on the back transformed scale. In both panels, pointwise 95% confidence intervals for fitted values are in red, with the fitted values subtracted off. Panel A has, in addition, the wider 95% prediction intervals, shown in black. Panel B adds confidence intervals for the model that has a quadratic term in `log(dist)` (blue dashes).

are shown in black. The prediction bands are relevant for calculating bounds for new races, possibly as here on the same courses. Note that:

- Because the combinations of distance and time vary from race to race, the bounds are not smooth functions of the predicted times.
- The model that replaces `log(climb)` with `log(gradient)` gives the same predicted values and the same intervals.

Figure 3.10B repeats the confidence bound information from Figure 3.10A (but omits the prediction bound information), now adding confidence bound information for a model that has a quadratic term in `log(dist)`:

```
timeClimb2.lm <- update(timeClimb.lm, formula = . ~ . + I(logdist^2))
```

[The dots (`"."`) on the left and right of the argument `new` repeat, respectively, the left and right parts of the model formula used on `timeClimb.lm`.]

Note that we had to write `I(logdist^2)`. The wrapper function `I()` ensures that its argument is handled as an arithmetic expression rather than as a model formula. (Within a model formula, `log(dist)^2` is interpreted as `log(dist)`. More helpfully, `(a+b+c)^2` is shorthand for `a+b+c+a:b+b:c+c:a`. Terms such as `a:b` are, technically, interactions.)

Observe that the bounds for the model `timeClimb2.lm` (dashed lines) are similar to those for `timeClimb.lm` (solid lines) within the main body of the data, but fan out at either extreme. The extra degree of freedom for the quadratic term leads to bounds that may better reflect the uncertainly in the predictions for large times.

There is a tradeoff between bias for `timeClimb.lm`, and increased variance for `timeClimb2.lm`.

These confidence and prediction bound estimates have important limitations:

1. Failure of assumptions of independence of the data points and homogeneity of variance, perhaps because of clustering or other forms of dependence, will bias or invalidate both types of interval.
2. The normality assumption is crucial for prediction intervals. Normality plays a less directly important role in the accuracy of the confidence bands.
3. Both types of bands apply only to the population from which the data have been sampled. It might be hazardous to use the above model to predict winning times for hill races in England or Mexico or Tasmania.

Point 3 can be addressed by testing the model against data from these other locations. Subsection 3.5 will compare results from the Scottish `hills2000` data with results from the Northern Irish `nihills` data. The results are broadly comparable.

3.3.5 Choosing the model — deaths from Atlantic hurricanes

The dataset `DAAG::hurricNamed` has data on fatalities caused in the United States by the 94 Atlantic hurricanes that made landfall over 1950—2012 inclusive. Explanatory variables that will be used here are `BaseDam2014` (damage converted to 2014 dollars) and `LF.PressureMB` (barometric pressure at the time of landfall in the US; if more than one landfall, then the minimum was taken), with `deaths` as the dependent variable.[11]

The analysis now given will treat `deaths` as a continuous variable. It will be desirable to check, to the extent possible, that this does not have any effect of consequence on the model choice and model fit. Figure 3.11 shows power transformed values of `deaths`, `LF.PressureMB`, and `BaseDam2014`, as suggested by output from the code:

```
hurric <− DAAG::hurricNamed[,c("LF.PressureMB", "BaseDam2014", "deaths")]
thurric <− car::powerTransform(hurric, family="yjPower")
transY <− car::yjPower(hurric, coef(thurric, round=TRUE))
smoothPars <− list(col.smooth='red', lty.smooth=2, lwd.smooth=1, spread=0)
car::spm(transY, lwd=0.5, regLine=FALSE, oma=rep(2.5,4), gap=0.5,
         col="blue", smooth=smoothPars, cex.labels=1)
```

As `deaths` has some zero values, the Yeo-Johnson family of transformations, described in Section 2.5.6, has been specified. The range of values of `LF.PressureMB` (909 – 1003) and of `BaseDam2014` (1.04 – 98195.4) is in each case so large that adding 1 to their values makes little difference to the suggested transformation.

Notice that `LF.PressureMB`, which has been left untransformed, has a bimodal

[11] The controversial PNAS article article (Jung et al., 2014) used the estimate of the dollar damage for a comparable hurricane in 2013, i.e., the 2013 equivalent of `NDAM2014`, rather than the more relevant inflation adjusted damage at the time measure. It claimed that hurricanes with female sounding names were, because treated less seriously, more dangerous than those with male names.

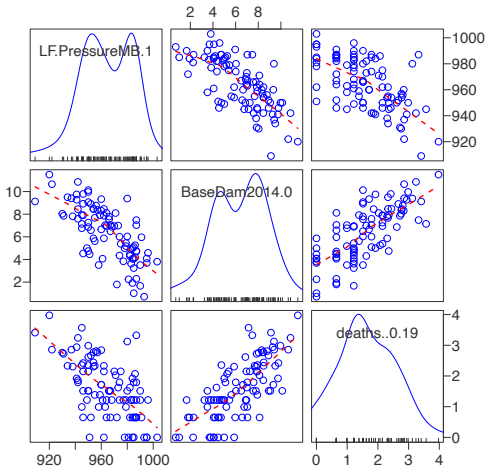


Figure 3.11 Scatterplot matrix, with automatically chosen power transformations, for hurricane death data.

distribution. Its relationship with the transformed values of `BaseDam2014` is nonlinear. The choice $\lambda = 0$ for `BaseDam2014` implies a logarithmic transformation. We then work with `log(BaseDam2014` and with `LF.PressureMB`.

The primary concern is for the distribution of the outcome variable to be as close to normal after accounting for the explanatory terms. A further use of the function `powerTransform()`, directly with the relevant model formula to suggest a transformation, then gives:

```
modelform <− deaths ∼ log(BaseDam2014) + LF.PressureMB
powerT <− car::powerTransform(modelform, data=as.data.frame(hurric),
                              family="yjPower")
summary(powerT, digits=3)
```

```
yjPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
Y1    -0.2033         -0.2       -0.3111       -0.0955

 Likelihood ratio test that transformation parameter is equal to 0
                          LRT df     pval
LR test, lambda = (0) 15.41   1 0.000087
```

This suggests use of a power transformation with $\lambda = -0.20$. This is not much different from that used in Figure 3.11, for the unconditioned values of `deaths+1`. Now fit a line, using the function `car::yjPower()` to create the transformed values of the outcome variable:

```
deathP <− with(hurric, car::yjPower(deaths, lambda=−0.2))
power.lm <− MASS::rlm(deathP ∼ log(BaseDam2014) + LF.PressureMB, data=hurric)
print(coef(summary(power.lm)),digits=2)
```

```
                Value Std. Error t value
```

```
## Use (deaths+1)^(-0.2) as outcome variable
plot(power.lm, cex.caption=0.85, fg="gray",
  caption=list('A: Resids vs Fitted', 'B: Normal Q−Q', 'C: Scale−Location', '',
               'D: Resids vs Leverage'))
```

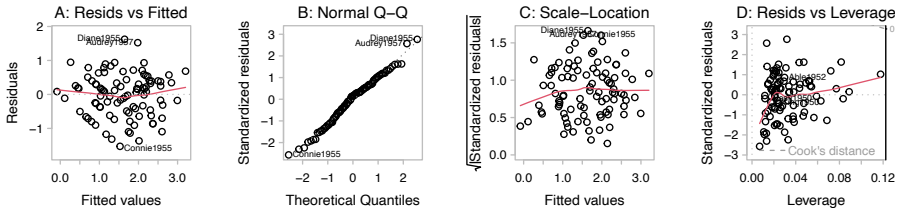


Figure 3.12 Diagnostic plots for a model that used power transformed values of `deaths` ($\lambda = -0.20$) for the outcome variable.

```
(Intercept)        9.6289      4.978      1.9
log(BaseDam2014)   0.2247      0.040      5.6
LF.PressureMB     -0.0098      0.005     -2.0
```

Figure 3.12 shows the diagnostic plots: Notice the slight curvature in the plot of residuals versus fitted values. Diane and Audrey stand out in several of the plots, but not to an extent that identifies them as outliers. The points with low leverage, where the smooth dips down in Panel D, are likely to be for data where there were no deaths. This is not a particular concern, as interest is primarily in points for cases where deaths occurred. Otherwise, these plots appear unexceptional. This dataset will be examined further in Subsection 5.4.4.

3.3.6 Strategies for fitting models — suggested steps

- Examine the distribution of each of the explanatory variables, and of the response variable. Look for any instances where distributions are highly skew, or where there are outlying values. Check whether any outlying values may be mistakes.
- Where a variable has a skewed distribution, consider whether a transformation may give a more symmetric distribution. Surprisingly often, logarithmic transformation of one or more explanatory variables gives a more symmetric distribution, leads to a regression relationship on the transformed scale(s) that are more nearly linear, and makes it easier to identify a regression equation that has good predictive power.
- Examine the scatterplot matrix involving all the explanatory variables. (Including the response is, at this point, optional.)
  - Look for values that appear as outliers in any of the pairwise scatterplots.
  - Look for evidence of nonlinearity in the plots of explanatory variables against each other. Such nonlinearity is often a result of skewness in one of the distributions, or of differences in the extent of skewness. In such cases consider transformation of one or both variables.

- Note the ranges of each of the explanatory variables. Do they vary sufficiently to affect values of the response variable?
- How accurately are each of the explanatory variables measured? At worst, the inaccuracy may be so serious that coefficients of other explanatory variables will be seriously in error and/or that any effect is unlikely to be detected. On implications for estimating coefficients of other variables, see Section 3.7.
- Look for pairs of explanatory variables that are so highly correlated that they appear to give the same information. Do scientific considerations help judge whether both variables should be retained? For example, the two members of the pair may measure what is essentially the same quantity. Note however that the difference, although small, can be important. Section 5.2 has an example, where the replacing of variables x_1 and x_2 by new variables $x_1 + x_2$ and $x_1 - x_2$ gave a better and more usable summary of the information in the data.

Note that if relationships between explanatory variables are nonlinear, diagnostic plots may be misleading. See Cook and Weisberg (1999).

Diagnostic checks

Checks should include:

- Plot residuals against fitted values. For initial checks, consider the use of residuals from a resistant regression model. Check for patterns in the residuals, and for the fanning out (or in) of residuals as the fitted values change. (Do not plot residuals against observed values. This is potentially deceptive; there is an inevitable positive correlation.)
- Examine the Cook's distance statistics.
- If it seems helpful, examine standardized versions of the drop-1 coefficients that are available using the function `dfbetas()`. See Subsection 3.4.2. It may be necessary to delete influential data points and refit the model.
- For each explanatory variable, construct a component plus residual plot, to check whether any of the explanatory variables require transformation.
- If observations follow a time sequence, use the function `acf()` to check for sequential correlation in the residuals. (See Section 9.2.)

The *dr* package implements and demonstrates a suite of tools that use ideas of "structural dimension" to guide the choice of a suitable form of nonlinear equation. This methodology is beyond the scope of this text.

3.4 Robust regression, outliers, and influence

This section gives additional detail on diagnostic graphs and statistics that may be useful in critiquing and/or interpreting regression models. There will be an especial focus on tools that can be effective in drawing attention to outliers and in assessing their effect, if included on the model.

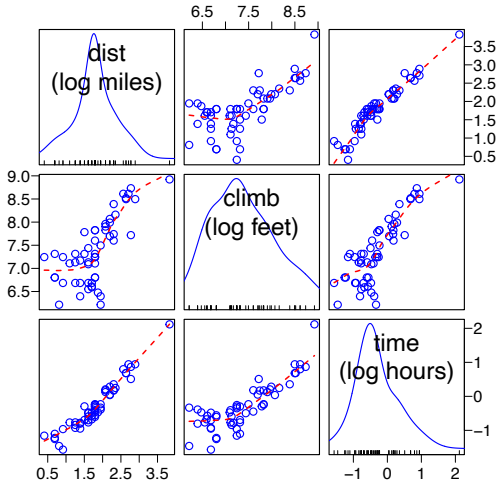


Figure 3.13 Scatterplot matrix for the `hills2000` data, with logarithmic scales.

3.4.1 Making outliers obvious — robust regression

Outliers can be hard to detect. Two (or more) outliers that are influential may mask each other. If this seems a possible issue, a useful recourse is to work with residuals from a resistant fit. A resistant fit is a type of robust fit that aims to completely ignore the effect of outliers, to give a fitted model against which outliers stand out.

The Scottish hillrace dataset `hills2000` is directly comparable with the Northern Irish dataset `nihills`. Again, we will use logarithmic scales and limit attention to the male results. Figure 3.13 shows the scatterplot matrix. Apart from a possible outlier, the relationship between `dist` and `climb` seems approximately linear on the log scale. Code is a ready adaptation of the code for Figure 3.3B.

The help page for `races2000` (`hills2000` is a subset of `races2000`) suggests uncertainty about the distance for the Caerketton race in row 42. We will include Caerketton during our initial analysis and check whether it appears to be an outlier.

Figure 3.14 shows residuals (A) from a least squares (`lm`) fit and (B) from a resistant `lqs` fit, in both cases plotted against fitted values. By default, even if almost half the observations are outliers, a resistant fit should ensure that the effect on the fitted model will be small. See the help page for `lqs` for details. The code for the regression fits is:

```
## Panel A
lhills2k.lm  <− lm(log(time) ∼ log(climb) + log(dist), data = hills2000)
## Panel B
 lhills2k.lqs   <− MASS::lqs(log(time) ∼ log(climb) + log(dist), data = hills2000)
reres  <− residuals( lhills2k.lqs )
```

Caerketton shows up clearly as an outlier, in both panels. Its residual in panel 1 is -0.356 (visual inspection might suggest -0.35). The predicted `logtime` for this

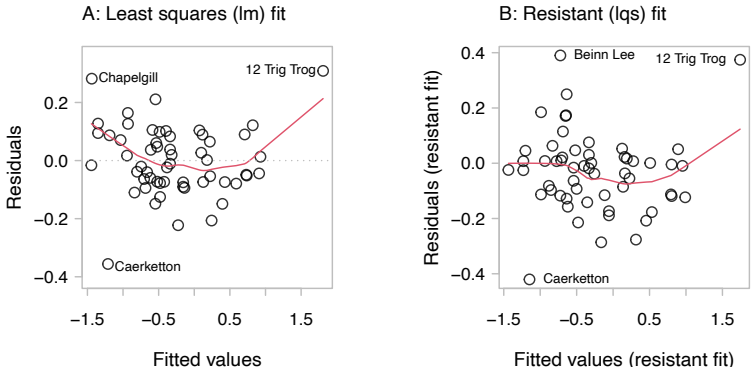


Figure 3.14 Plots of residuals against fitted values from the regression of `logtime` on `logclimb` and `logdist`. Panel A is from the least squares (`lm`) fit, while Panel B is for a resistant fit that uses `MASS::lqs()`. Note that the resistant fit relies on repeated sampling of the data, and will differ slightly from one run to the next.

race is conveniently written as $\log(\widehat{\text{time}})$. Then

$$\log(\text{time}) - \log(\widehat{\text{time}}) = -0.356$$

Thus

$$\log(\frac{\text{time}}{\widehat{\text{time}}}) = -0.356, \text{ i.e., } \frac{\text{time}}{\widehat{\text{time}}} \simeq \exp -0.356 = 0.7$$

The time given for this race is 70% of that predicted by the regression equation. The standardized difference is -3; this can be seen by use of

```
plot( lhills2k.lm ,  which=2)
```

The resistant fit in Panel B suggests that Beinn Lee and 12 Trig Trog are also outliers. These outliers may be a result of nonlinearity.

Dynamic graphic exploration can be helpful. See further, Figure 3.15A in the subsection that follows.

Outliers, influential or not, should be taken seriously

Outliers, influential or not, should never be disregarded. Careful scrutiny of the original data may reveal an error in data entry. Alternatively, use of an inappropriate model may result in one or more outliers. If apparently genuine outliers remain excluded from the final fitted model, they should be noted in the eventual report or paper. They should be included, separately identified, in graphs.

3.4.2 Leverage, influence, and Cook's distance

This extends earlier discussions of diagnostics in Subsections 2.5.3 and 3.1.2.

What difference does replacing y_i by $y_i + \Delta_i$, while leaving other y-values unchanged, make to the fitted surface. There is a straightforward answer; the fitted value changes from $\widehat{y}_i$ to $\widehat{y}_i + h_{ii}\Delta_i$, where h_{ii} is the *leverage* for that point.

* *Leverage and the hat matrix — technical details*

In a regression with outcome variable y and model matrix X, the *hat* matrix is

$$\mathrm{H} = \mathrm{X}(\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T$$

The leverage values h_{ii} are the diagonal elements of the hat matrix H that can be derived from the model matrix. They sum to give the number p of columns of the model matrix. (The vector $\widehat{\mathrm{y}}$ of fitted or *hat* values is given by the matrix calculation $\widehat{\mathrm{y}} = \mathrm{Hy}$. The hat matrix puts the hat on y.) A large h_{ii} gives the ith observation high leverage. Use the function `hatvalues()` to obtain the leverages, thus:

```
round(unname(hatvalues(timeClimb.lm)),2)
```

```
 [1] 0.12 0.07 0.13 0.11 0.09 0.15 0.05 0.15 0.25 0.22 0.09 0.06 0.06
[14] 0.13 0.05 0.10 0.18 0.21 0.44 0.09 0.14 0.05 0.08
```

The largest leverage, for observation 19, is 0.44. As this is more than three times the average value of 0.13, it may call for attention. (The model matrix has $p = 3$ columns. With $n = 23$ observations, the average is $p/n = 0.13$.)

Influential points and Cook's distance

The Cook's distance statistic is a commonly used measure of 'influence', i.e. of the combined effect of the size of the residual and its leverage in determining fitted values. Recall the guideline given in Section 2.5.3, that any Cook's distance that is 1.0 or more, or that is substantially larger than other Cook's distances, warrants careful examination.

Any serious distortion of the fitted response may lead to residuals that are hard to interpret or even misleading. It is wise to check the effect of removing any highly influential data points before proceeding far with the analysis.

Dynamic graphics

The left panel of Figure 3.15 is a plot of residuals against leverage values, with the Cook's distances are shown as contours. The right panel is a snapshot, created using the abilities of the *rgl* package, of a three-dimensional dynamic graphic plot that shows the regression plane in the regression of `log(time)` on `log(dist)` and `log(climb)`. The two points that in the perspective shown appear most extreme have been labeled. The left panel can be obtained thus:

```
## Residuals versus leverages
plot(timeClimb.lm, which=5, add.smooth=FALSE)
## The points can alternatively be plotted using
## plot(hatvalues(model.matrix(timeClimb.lm)), residuals(timeClimb.lm))
```

The right panel can be obtained thus:

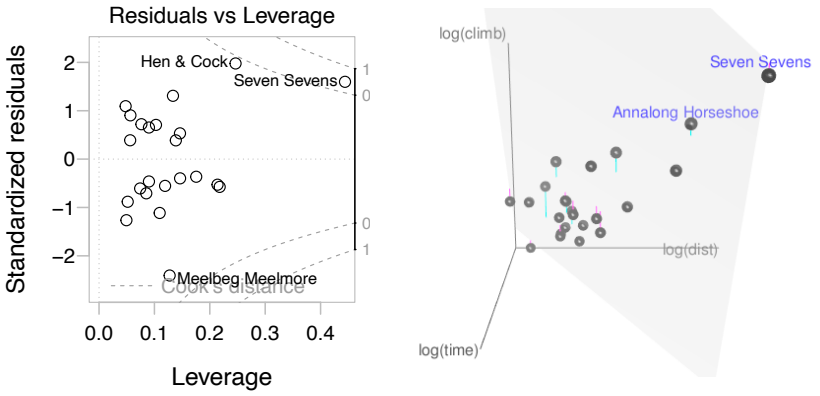


Figure 3.15 In the left panel, standardized residuals are plotted against leverages. Contours are shown for Cook's distances of 0.5 and 1.0. The right panel is a snapshot of a three-dimensional dynamic graphic plot.

```
with( nihills , scatter3d(x=log(dist), y=log(climb), z=log(time), grid=FALSE,
                          point.col ="black", surface.col ="gray60",
                          surface.alpha=0.2, axis.scales =FALSE))
with( nihills , Identify3d(x=log(dist), y=log(climb), z=log(time),
                           labels=row.names(DAAG::nihills), minlength=8), offset=0.05)
## To rotate display, hold down the left mouse button and move the mouse.
## To put labels on points, right-click and drag a box around them, perhaps
## repeatedly.  Create an empty box to exit from point identification mode.
```

Dynamic graphic 3-dimensional plots can be useful in calling attention to any clustering in the points, or points that lie away from the main body of points.

The abilities of the *rgl* package can be conveniently accessed from the 3D graph submenu of the R Commander GUI's graphics pull-down menu.

Influence on the regression coefficients

In addition to diagnostic plots, it is useful to investigate the effect of each observation on the estimated regression coefficients. The function `dfbetas()` calculates the differences in the coefficient estimates obtained with and without each observation, then dividing by the respective standard error estimate to give a standardized es-

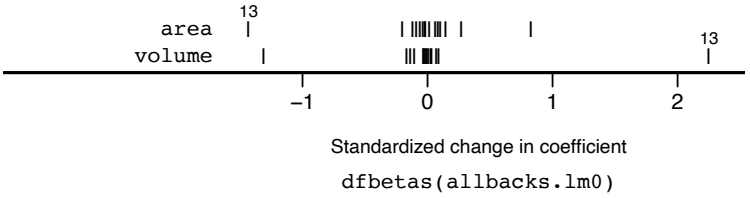


Figure 3.16 Standardized changes in regression coefficients, for the model fitted to the `allbacks` dataset. The points for the one row (row 13) where the change for one of the coefficients was greater than 2 in absolute value are labeled with the row number.

timate. Figure 3.16 shows values for the regression model fit to the `allbacks` data that excluded the intercept term.

Under the distributional assumptions, standardized changes that are larger than 2 can be expected, for a specified coefficient, in about 1 observation in 20. Here, the only change that seems worthy of note is for `volume` in observation 13. For absolute changes, should they be required, use the function `lm.influence()`.

** Additional diagnostic plots*

The functions in the *car* package, designed to accompany Fox and Weisberg (2018), greatly extend the range of diagnostic plots. See the examples and references in the help pages for this package. As an indication of what is available, try

```
car :: influencePlot (allbacks.lm)
```

3.5 Assessment and comparison of regression models

The R^2 and adjusted R^2 statistics (discussed in Subsection 2.5.2) are included in the default output from `summary.lm()`. Note also the F-statistic that compares the *mean square* explained by the model with the *residual mean square* that is used as a variance estimate. Other statistics that may be used appear in output from functions that will be discussed shortly – `add1()`, `drop1()`, and `anova()`. The use of such statistical measures should be accompanied by careful scrutiny of diagnostic plots, term plots, and other relevant graphical checks. Finally, we will look briefly at the use of Bayes Factors to compare regression models.

While adjusted R^2 is in general a more meaningful statistic, it is not a good basis for model comparison. The AIC (or AICc) and BIC *information* statistics that were introduced in Subsection 1.7.1 are preferred for this purpose. These are among measures that are designed to choose, from among a small number of alternatives, the model with the best predictive power. While predictive accuracy is not necessarily the only or the most important consideration, it is always an important consideration.

*3.5.1 * AIC, AICc, BIC, and Bayes Factors for normal theory regression models*

Consider now the use of a regression model that relates `brainwt` (brain weight), in the dataset `DAAG::litters`, to `lsize` (litter size) and `bodywt` (body weight). How strong is the case for including `lsize` as an explanatory factor? The following fits models that do and do not include `lsize`:

```
## Calculations using mouse brain weight data
mouse.lm <− lm(brainwt ∼ lsize+bodywt, data=DAAG::litters)
mouse0.lm <− update(mouse.lm, formula = . ∼ . − lsize)
```

Now look at alternative ways to compare the two models:

```
aicc <− sapply(list(mouse0.lm, mouse.lm), AICcmodavg::AICc)
infstats <− cbind(AIC(mouse0.lm, mouse.lm), AICc=aicc,
                  BIC=BIC(mouse0.lm, mouse.lm)[,−1])
print(rbind( infstats , "Difference"=apply(infstats,2, diff )), digits=3)
```

```
            df      AIC     AICc      BIC
mouse0.lm    3 -112.81 -111.31 -109.82
mouse.lm     4 -115.57 -112.90 -111.58
Difference   1   -2.76   -1.59   -1.76
```

More generally, how do the statistics compare when the number of parameters estimated is a modest proportion, perhaps 5% or more, of the number of observations? Figure 3.17 looks, for a range of values of n (number of observations), at how the AIC, AICc and BIC statistics compare. The lines all show the increase in the penalty term difference for an increase of one in the number of parameters estimated. The change (negative if the statistic decreases) in the relevant statistic in moving from a model with log likelihood L_1 to a log likelihood L_2 that results from adding one more degree of freedom is:

$$-2L_2 + 2L_1 + \text{ penalty increase}$$

where the penalty difference is that shown in Figure 3.17.

```
df <- data.frame(n=5:35, AIC=rep(2,31), BIC=log(5:35))
cfAICc <- function(n,p,d) 2*(p+d)*n/(n-(p+d)-1) - 2*p*n/(n-p-1)
df <- cbind(df, AICc12=cfAICc(5:35,1,1), AICc34=cfAICc(5:35,3,1))
labs <- sort(c(2^(0:6),2^(0:6)*1.5))
xyplot(AICc12+AICc34+AIC+BIC ~ n, data=df, type='l', auto.key=list(columns=4),
       scales=list(y=list(log=T, at=labs, labels=paste(labs))),
 par.settings=simpleTheme(lty=c(1,1:3), lwd=2, col=rep(c('gray','black'), c(1,3))))
```

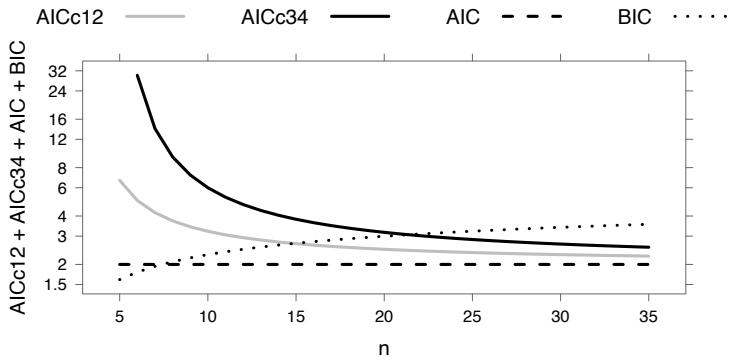


Figure 3.17 The increase in the penalty term difference is shown for an increase of one in the number of parameters p. For the AIC statistic, this equals the constant 2. For the BIC statistic, this depends only on n. For the AICc statistic, the number of parameters p makes a huge difference when n is small, as shown by the difference between the black (4 vs 3) and gray (2 vs 1) solid lines

There is a strong case for consistently using AICc in place of AIC. Once n/p is small enough that the BIC difference is smaller than the AICc difference, this may reasonably be taken as evidence that the asymptotic results no longer apply, whether for BIC as well as for AIC. Note that the sampling variation is large for all three statistics when n/p is small.

The functions `drop1()` *and* `add1()`

Readers may care to check that the AIC and BIC information can be obtained by using `drop1()` with `mouse.lm` as argument, or `add1()` with the *scope* extended to include `lsize`. The default is to show AIC statistics. Setting `k=log(n)`, where `n=nrow(DAAG::litters)`, gives the BIC statistics.

```
n <- nrow(DAAG::litters)
drop1(mouse.lm, scope=~lsize)              # AIC, with/without `lsize`
drop1(mouse.lm, scope=~lsize, k=log(n))    # BIC, w/wo `lsize`
add1(mouse0.lm, scope=~bodywt+lsize)     # AIC, w/wo `lsize`, alternative
```

In this context where the number of observations to maximum number of parameters ratio is $20/4 = 5$, neither of these statistics makes much sense.

The use of `BayesFactor::lmBF` *to compare the two models*

A further possibility is to use a *BayesFactor* package style Bayes Factor to compare the two models. Separate factors `bf1` and `bf2` are first calculated that are for comparisons with the model that has only the constant term. The ratio `bf2/bf1` is then used to compare the two models.

```
suppressPackageStartupMessages(library(BayesFactor))
bf1 <- lmBF(brainwt ~ bodywt, data=DAAG::litters)
bf2 <- lmBF(brainwt ~ bodywt+lsize, data=DAAG::litters)
bf2/bf1
```

```
Bayes factor analysis
--------------
[1] bodywt + lsize : 1.512 ś0%

Against denominator:
  brainwt ~ bodywt
---
Bayes factor type: BFlinearModel, JZS
```

The `bf1/bf0` ratio of 1.51 can be compared with a Bayes Factor type relative preference statistic that is calculated from the BIC, thus:

```
## Relative support statistics
setNames(exp(-apply(infstats[,-1],2,diff)/2), c("AIC","AICc","BIC"))
```

```
  AIC  AICc   BIC
3.965 2.213 2.410
```

Note also the C_p statistic. The `anova()` method for models fitted using `lm()` sets the C_p statistic to be:[12]

$$C_p = \text{RSS} + 2p\sigma^2$$

In the linear regression context, with the smallest value preferred, this will choose the same model as the AIC statistic.

[12] The function `olsrr:ols_mallows_cp()` implements, instead, the more usual: $C_p = \frac{\text{RSS}}{\sigma^2} + 2p - n$

3.5.2 Using `anova()` to compare models — the `nihills` data

The function `anova()`, used in Subsection 2.5.2 and Section 3.1 to give a sum of squares breakdown for a regression model, can be used to compare models. The sum of squares breakdown, and the use of an F-statistic to compare models, makes sense only if the models are *nested*, i.e., the more complex model can be formed by adding a term or terms to a simpler model.

If as usually happens the variance has to be estimated, σ^2 is replaced by the estimated variance s^2 for the 'largest' model considered. An alternative to the default `test="F"` is `test="Cp"`, which (as with AIC, AICc and BIC), does not require nested models.[13]

Figure 3.6 suggested some curvature in the contribution of the term `logdist` that was fitted to the `nihills` data. The following investigates adding a squared term, giving the model `nihills2.lm`.

```
lognihr <- log(DAAG::nihills)
lognihr <- setNames(log(nihr), paste0("log", names(nihr)))
timeClimb.lm <- lm(logtime ~ logdist + logclimb, data = lognihr)
timeClimb2.lm <- update(timeClimb.lm, formula = . ~ . + I(logdist^2))
print(anova(timeClimb.lm, timeClimb2.lm, test="F"), digits=4)
```

```
Analysis of Variance Table

Model 1: logtime ~ logdist + logclimb
Model 2: logtime ~ logdist + logclimb + I(logdist^2)
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1     20 0.1173
2     19 0.0999  1   0.01744 3.318 0.0843
```

The result from the alternative argument `test="Cp"` is:

```
print(anova(timeClimb.lm, timeClimb2.lm, test="Cp"), digits=3)
```

```
Analysis of Variance Table

Model 1: logtime ~ logdist + logclimb
Model 2: logtime ~ logdist + logclimb + I(logdist^2)
  Res.Df    RSS Df Sum of Sq   Cp
1     20 0.1173             0.15
2     19 0.0999  1    0.0174 0.14
```

```
## Compare with the AICc difference
sapply( list (timeClimb.lm, timeClimb2.lm), AICcmodavg::AICc)
```

```
[1] -45.9 -46.3
```

Both suggests a slight preference for the model with the squared term.

Now investigate adding, in turn, each of `I(logdist^2)` and `logdist:logclimb`. The interaction term `logdist:logclimb` may equivalently (as these are continuous variables) be written `I(logdist*logclimb)`:

[13] For comparing nested models in a context where the argument `scale` is used to specify a known residual variance, specify `test="Chisq"`. This is not relevant here.

```
form1 <- update(formula(timeClimb.lm), ~ . + I(logdist^2) + logdist:logclimb)
addcheck <- add1(timeClimb.lm, scope=form1, test="F")
print(addcheck, digits=4)
```

```
Single term additions

Model:
logtime ~ logdist + logclimb
                 Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>                       0.1173 -115.4
I(logdist^2)      1   0.01744 0.0999 -117.1   3.318 0.0843
logdist:logclimb  1   0.01172 0.1056 -115.8   2.108 0.1628
```

The model formula `form1` includes, in addition to the terms already in the model, the terms `I(logdist^2)` and `logdist:logclimb`. The function `add1()` checks the effect of adding each of these additional terms, one at a time, to the model.

3.5.3 Training/test approaches, and cross-validation

The surest check is to use the chosen model to make predictions for "test" data that is separate from the "training" data used to fit the model, and that reflects the conditions under which predictions will be used. For testing the accuracy of predictions when the model is applied one year into the future, a useful check is to compare past predictions made one year into the future with what eventuated. (In times of relative stability, what this gives is a best case scenario, with no accounting for what a pandemic, or a war, or a freak of nature, may throw up.)

As noted earlier, the *DAAG* package has two hill race datasets — `nihills` for Northern Ireland, and `hills2000` for Scotland. The square of the "residual standard error" from `timeClimb.lm` is a mean squared error (MSE) estimate, with `nihills` in the role of training data. We will use the dataset `hills2000` as test data, but omitting the seemingly erroneous Caerketton observation. The mean square prediction error (MSPE) estimate for `hills2000` as test data is the mean squared difference between predictions for that data from the model `timeClimb.lm`, and observed values for the Scottish hill race data.

If the two datasets can be treated as different random samples from the combined dataset, the expected values of the MSE and the MSPE will be the same. Degrees of freedom are number of rows (23) minus 3 for the MSE, and number of of rows (55, after omitting Caerketton) for the MSPE estimate. As the `hills2000` data are independent of the data used to fit the model `timeClimb.lm`, there is no adjustment for number of parameters estimated.

```
## Check how well timeClimb.lm model predicts for hills2000 data
timeClimb.lm <- lm(logtime ~ logdist + logclimb, data = lognihr)
logscot <- log(subset(DAAG::hills2000,
                      !row.names(DAAG::hills2000)=="Caerketton"))
names(logscot) <- paste0("log", names(hills2000))
scotpred <- predict(timeClimb.lm, newdata=logscot, se=TRUE)
trainVar <- summary(timeClimb.lm)[["sigma"]]^2
trainDF <- summary(timeClimb.lm)[["df"]][2]
mspe <- mean((logscot[,'logtime']-scotpred[['fit']])^2)
```

```
mspeDF <- nrow(logscot)
```

The two estimates are, to four significant figures:

- Training data (`nihr`): 0.0059 on 20 degrees of freedom. Note that degrees of freedom are number of rows (23), less number of parameters estimated.
- Test data (`hills2000`): 0.0173 on 55 degrees of freedom. Degrees of freedom are number of rows (55) of test data.

The two mean square error estimates are independent, and can be compared using an F-test:

```
pf(mspe/trainVar, mspeDF, trainDF, lower.tail=FALSE)
```

```
[1] 0.004789
```

The increase in the mean square error arises, in part, because the squared "residual standard error" when the model is fitted to the `hills2000` data is larger than for the `timeClimb.lm` fitted model. There is more inherent variability in the `hills2000` data.

```
scot.lm <- lm(logtime ~ logdist+logclimb, data=logscot)
signif(summary(scot.lm)[['sigma']]^2, 4)
```

```
[1] 0.01228
```

3.5.4 Further points and issues

Patterns in the diagnostic plots – are they more than hints?

The smooths added to the plots of residuals against fitted values in Figures 3.6 and 3.12 had saucer-like shapes. Were these likely to be a result of statistical noise? As a guide to judgment, we can use the function `DAAG::plotSimDiags()` to simulate from the fitted model and examine the diagnostic plots. If plots from simulated data give comparable curvature with modest frequency, the hints can be ignored.

Thus, consider again the model for which Figure 3.6 gives the diagnostic plots. Code that generates 8 simulated sets of y-values, then showing the 8 sets of plots of residuals against fitted values, is:

```
set.seed(91)          # Reproduce plots as shown here
plotSimDiags(timeClimb.lm, layout=c(4,2), which=1, caption=list(""))
```

Figure 3.18 shows the resulting plots. Panels 1 and 5 show saucer-shaped curvature that is comparable to that in the first of the plots in Figure 3.6. A hint that the model is not quite capturing the trend in the response should probably be ignored.

What is the scatter about the fitted response

When there is extensive data, it can happen that the scatter about the fitted model is so large that the relatively small amount of the total variation explained by the model makes it of limited practical use. See Soyer and Hogarth (2012) for extended

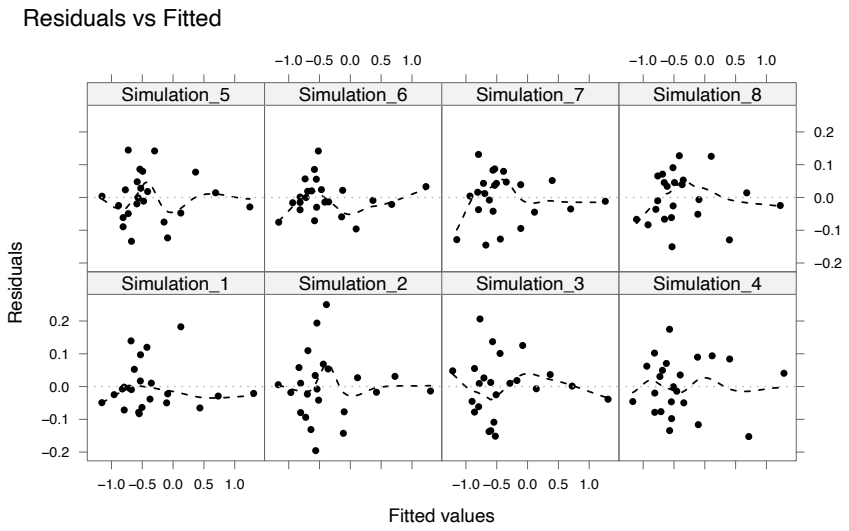


Figure 3.18 Eight sets of simulated time values were generated by adding random normal noise, with standard deviation equal to the fitted model error root mean square, to the fitted `log(time)` (`ltime`) values for the model. Panels show diagnostic plots of residuals against fitted values.

commentary. Termplots are a good way to check how this feeds into the contribution of individual variables when other variables are held constant.

In the Component plus Residual plots in Figure 3.4, the scatter was small relative to the explanatory power of the terms individually. Figure 2.8B related to data (the dataset `HistData::GaltonFamilies`) where, notwithstanding a t-statistic of 9.3 for the gradient of the fitted line, the line explained only 15% of the variation.

Model selection and tuning risks

Selection bias is a concern whenever a model performance measure is used to choose from a number of alternative models. Criteria that suggest transformations for explanatory variables, if this is done independently of the effect on the model fit, do not introduce this bias. They become an increasing concern as, using a model with linear terms as a starting point, ever more modifications or alternatives are considered. These may include the addition of one or more or other quadratic terms, or the transformation of one or more variables. Excessive fine tuning is both a waste of time and counter-productive.

As discussed in Subsection 3.5.3, a safe way to proceed is to fit the model to data that are separate from the data used to develop the model. In principle, cross-validation can be used to address the issue, but that requires the model development steps to be repeated for each cross-validation fold.

Generalization to new contexts requires a random sample of contexts

Chapter 7 discusses models that in principle allow generalization to new contexts, accounting for variation between contexts (perhaps countries) as well as variation within countries. Development of such a model for hillraces would require data from more than two countries, preferably at least 5 or 6.

What happens if we do not transform the hillrace data?

If we avoid transformation and do not allow for increasing variability for the longer races (see further, Exercise 8 at the end of the chapter), several observations appear outliers, with the race that has the longest time highly influential.

Venables and Ripley (2002, p. 154) point out (in connection with the Scottish hillrace data) that it is reasonable to expect that variances will be larger for longer races. Using `dist` as a surrogate for time, they give observations weights of `1/dist^2`. This is roughly equivalent, in its effect on the variance, to our use of `logtime` as the response variable.

Are "errors in x" an issue?

As explained in Subsection 3.7, random errors in the measured values of the explanatory variables can bias the model coefficients, and/or lead to spurious indications that one or more other variable is having an important effect. In the dataset `nihills`, most distances are given to the nearest half mile, and may in any case not be known at all accurately. The error is however likely to be small relative to the range of values of distances, so that the attenuation effects that will be discussed in Subsection 3.7 are likely to be small and of minor consequence.

3.6 Problems with many explanatory variables

Variable selection is an issue when the aim is to obtain the best prediction possible. If instead the interest is in which variables have useful explanatory power, then the choice of variables will in general depend on which variables are to be held constant when the effects of other variables are estimated. In unusually favorable circumstances, it may turn out that one or perhaps two variables stand out as having a dominant effect, with coefficients not much affected by what other variables are included in the regression equation. There should in any case be an initial exploratory investigation of explanatory variables, as described in Subsection 3.3.6, leading perhaps to transformation of one or more of the variables.

One suggested rule is that there should be at least 10 times as many observations as variables, before any use of variable selection takes place. For any qualitative factor, subtract one from the number of levels, and count this as the number of variables contributed by that factor. While this may be a reasonable working rule when working with relatively noisy data where none of the variables have a dominant effect, there are important contexts where it is clearly inapplicable.

The following strategies, individually or in combination, can help keep to a minimum the number of different models that are to be compared:

1. Start with an informed guess on what variables/factors are likely to be important. Where there are many explanatory variables, consider classifying them into groups according to an assessment of scientific 'importance'. Fit the most important variables first, then add the next set as a group, checking whether this improves the fit.
2. Use an omnibus check to compare a model that has, e.g., all first order interaction terms (i.e., model formulae terms of the form `x1:x2`), against a main effects model, rather than checking for interaction effects one at a time.
3. Principal components analysis is one of several methods that may be able to identify a few components, i.e., combinations of the explanatory variables, that together account for most of the variation in the explanatory variables. In favorable circumstances, one or more of the first few principal components will prove to be useful explanatory variables, or may suggest useful simple forms of summary of the original variables. In unfavorable circumstances, the components will prove irrelevant! See Section 9.2 and Harrell (2015, Sections 4.7 and 8.6) for further commentary and examples. See Section 9.7 for examples.
4. Discriminant analysis can sometimes be used to identify a summary variable. There is an example in Section 9.7.

3.6.1 Variable selection issues

We caution against giving much credence to p-values that appear in output from conventional automatic variable selection techniques – various forms of stepwise regression, and best subsets regression. The resulting regression equation may have poorer genuine predictive power than the regression that includes all explanatory variables. The standard errors and t-statistics typically ignore the effects of the selection process; estimates of standard errors, p-values and F-statistics will be optimistic. Estimates of regression coefficients are biased upwards in absolute value – positive coefficients will be larger than they should be, and negative coefficients will be smaller than they should be. See Harrell (2015) for further discussion.

Selection effects of various types arise in an increasing variety of contexts. In part this is a consequence of new automated technologies for data collection, and of the ease with which large data collections can be exposed to automated scrutiny.

Variable selection – a simulation with random data

Repeated simulation of a regression problem where the data consist entirely of noise will demonstrate the extent of the problem. In each regression there are 41 vectors of 100 numbers that have been generated independently and at random from a normal distribution. In these data:[14]

1. The first vector is the response variable y
2. The remaining 40 vectors are the variables x_1, x_2,..., x_{40}.

[14]
```
## Generate a 100 by 40 matrix of random normal data
y <- rnorm(100)
xx <- matrix(rnorm(4000), ncol = 40)
dimnames(xx)<- list(NULL, paste("X",1:40, sep=""))
```

If we find any regression relationships in these data, this will indicate faults with our methodology. (In computer simulation, we should not however totally discount the possibility that a quirk of the random number generator will affect results. This is unlikely to be an issue for the present simulation!)

The footnote has code that performs a best subsets regression that looks for the three x-variables that best explain y.[15] The function `DAAG::bestsetNoise()` automates the calculation:

```
## DAAG::bestsetNoise(m=100, n=40)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1438     0.0957    1.50   0.1363
xV4          -0.1918     0.0992   -1.93   0.0561
xV10         -0.3075     0.0948   -3.24   0.0016
xV14          0.2643     0.0961    2.75   0.0071
```

When repeated ten times, the outcomes were as follows. Categories are exclusive:

	Instances
All three variables selected had $p < 0.01$	1
All three variables had $p < 0.05$	3
Two out of three variables had $p < 0.05$	3
One variable with $p < 0.05$	3
Total	10

In the modeling process there are two steps:

1. Select variables.
2. Do a regression and determine SEs and p-values, etc.

The p-value calculations have taken no account of step 1. Such finding of 'significance' in datasets that consist only of noise is evidence of a large bias.

The extent of selection effects – a detailed simulation:

As above, datasets of random normal data were created, always with 100 observations and with the number of variables varying between 3 and 50. For three variables, there was no selection, while in other cases an exhaustive search selected the 'best' three variables. Figure 3.19 plots the p-values for the 3 variables that were selected against the total number of variables. The fitted line estimates the median p-value.

When all 3 variables are taken, the p-values are expected to average 0.5. Notice that, for selection of the best 3 variables out of 10, the median p-value has reduced to about 0.1. Code for Figure 3.19 is:

[15]
```
## Find the best fitting model. (The 'leaps' package must be installed.)
xx.subsets <- leaps::regsubsets(xx, y, method = "exhaustive", nvmax = 3, nbest = 1)
subvar <- summary(xx.subsets)$which[3,-1]
best3.lm <- lm(y ~ -1+xx[, subvar])
print(summary(best3.lm, corr = FALSE))
```

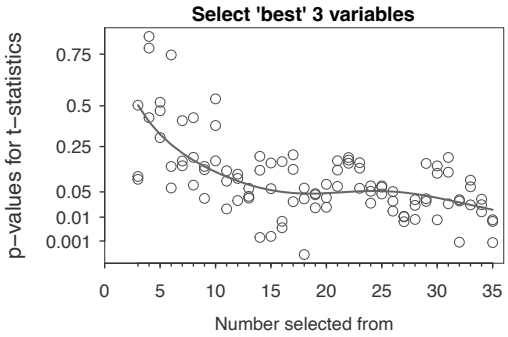


Figure 3.19 *P*-values, versus number of variables available for selection, when an exhaustive search selected the 'best' three variables. The fitted line estimates the median *p*-value. The function `bsnVaryNvar()` makes repeated calls to `bestsetNoise()`. Results will vary somewhat from one run to another..

```
library ( splines )
DAAG::bsnVaryNvar(nvmax=3, nvar = 3:35, xlab="Number selected from")
```

Cross-validation that accounts for the variable selection process

Subsection 2.6.1 introduced the use of cross-validation. In the variable selection context it is important, at each fold, to repeat both the variable selection step and the calculation of the sum of squares of the outcome values for the test data for that fold about the fitted values for a model fit that used the training data for that fold. The sums of squares for the separate folds are added and divided by the total number of observations to give an overall variance estimate.

** Regularization approaches*

Regularization approaches are available that add to the sum of squares a penalty term that is a multiple λ of a penalty that is a function of the model coefficients. See Breheny and Huang (2011) and Breheny (2019) and the vignette that accompanies the *ncvreg* package. A number of different types of penalty have been proposed, with lasso perhaps the best known (Efron, Hastie, et al., 2003). Methods of this type are a resort when there are too many possible subsets to allow a best subsets approach. Cross-validation can be used to choose the value of λ that gives the lowest error mean square, or when there are more variables than observations.

3.6.2 Multicollinearity

Some explanatory variables may be linearly related to combinations of one or more of the other explanatory variables. Technically, this is known as multicollinearity. For each multicollinear relationship, there is one redundant variable.

The approaches that we have advocated – careful thinking about the background science, careful initial scrutiny of the data, and removal of variables whose effect is

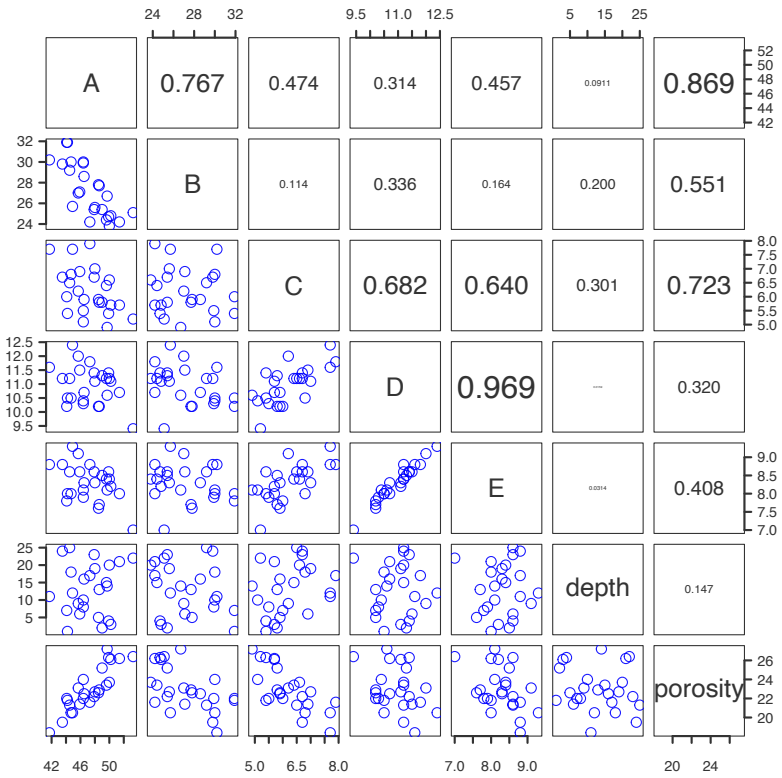


Figure 3.20 Scatterplot matrix for the variables in the `Coxite` data, with absolute values of pairwise correlations shown in the upper panel.

already accounted for by other variables – will generally avoid the more extreme effects of multicollinearity that we will illustrate. Milder consequences are pervasive, especially for observational data.

An example – compositional data

The matrix `compositions::Coxite` has the mineral compositions of 25 coxite type rock specimens. Each composition consists of the percentage by weight of each of five minerals (`A` = albite, `B` = blandite, `C` = cornite, `D` = daubite and `E` = endite), the depth of location, and porosity. The analysis that follows is a relatively crude use of these data. For an analysis that uses a method that is designed for compositional data, see Aitchison (2003).

Figure 3.20 shows the scatterplot matrix.[16] The relationship between `D` and `E` appears close to linear.

[16]
```
## Simplified plot
data(Coxite, package="compositions")  # Places Coxite in the workspace
  # NB: Proceed thus because `Coxite` is not exported from `compositions`
coxite <- as.data.frame(Coxite)
pairs(coxite)
```

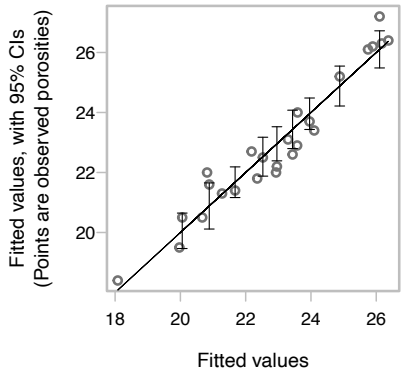


Figure 3.21 Line $y = x$, with 95% pointwise confidence bounds for fitted values shown at several locations along the range of fitted values. The points show the observed porosities at each of the fitted values.

We will look for a model that explains `porosity` as a function of mineral composition. Inclusion of all six explanatory variables in the model formula gives:

```
coxiteAll.lm <- lm(porosity ~ A+B+C+D+E+depth, data=coxite)
print(coef(summary(coxiteAll.lm)), digits=2)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -217.747    253.444  -0.859     0.40
A              2.649      2.483   1.067     0.30
B              2.191      2.601   0.842     0.41
C              0.211      2.227   0.095     0.93
D              4.949      4.672   1.059     0.30
depth          0.014      0.033   0.435     0.67
```

The percentages of the five minerals sum, for each observation, to very close to 100. The very slight differences from 100, for some of the rows, are enough that the model fits without complaint. Observe that:

- The variable `E`, because it is a linear combination of earlier variables, has been 'aliased', i.e., left out. Effectively, its coefficient has been set to zero.
- None of the individual coefficients comes anywhere near the usual standards of statistical significance.
- For the overall regression fit, $p = 1.18 \times 10^{-10}$.

The overall regression fit has good predictive power, notwithstanding the inability to tease out the contributions of the individual coefficients. Figure 3.21 shows 95% pointwise confidence intervals for fitted values at several points within the range.

Pointwise confidence bounds can be obtained thus:

```
hat <- predict(coxiteAll.lm, interval="confidence", level=0.95)
```

The object that is returned is a matrix, with columns `fit` (fitted values), `lwr` (lower confidence limits) and `upr` (upper confidence limits). Data points that distort the fitted response are "influential".

3.6.3 The variance inflation factor (VIF)

The variance inflation factor (VIF) measures the effect of correlation with other variables in increasing the standard error of a regression coefficient. If x_j, with values x_{ij} $(i = 1, \ldots, n)$ is the only variable in a straight line regression model, and b_j is the estimated coefficient, then:

$$\text{var}(b_j) = \frac{\sigma^2}{s_{jj}}, \quad \text{where } s_{jj} = \sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2.$$

and σ^2 is the variance of the error term in the model. When further terms are included in the regression model, this variance is inflated, as a multiple of σ^2, by the variance inflation factor. Notice that the VIF depends only on the model matrix. It does not reflect changes in the residual variance.

Because the model `coxiteAll.lm` was singular (one variable was a linear combination of earlier variables), VIFs are not available. To demonstrate the calculation of VIFs, we explicitly omit `E` from the model (omitting one of `A`, `B`, `C`, or `D` would also allow the calculation of VIFs), thus:

```
print(DAAG::vif(lm(porosity ~ A+B+C+D+depth, data=coxite)), digits=2)
```

```
     A      B      C      D  depth
2717.8 2485.0  192.6  566.1    3.4
```

The size of these factors has made it impossible to obtain meaningful estimates of the individual coefficients.

We now investigate the use of the function `leaps::regsubsets()` to choose the "best" subsets with 1, 2, 3 and 4 of the explanatory variables:

```
b <- leaps::regsubsets(porosity ~ ., data=coxite, nvmax=4, method='exhaustive')
## The calculation fails for nvmax=5
inOut <- summary(b)[["which"]]
## Extract and print the coefficents for the four regressions
dimnam <- list(rep("",4),c("Intercept", colnames(coxite)[-7]))
cmat <- matrix(nrow=4, ncol=7, dimnames=dimnam)
for(i in 1:4)cmat[i,inOut[i,]] <- signif(coef(b,id=1:4)[[i]],3)
outMat <- cbind(cmat," "=rep(NA,4),
as.matrix(as.data.frame(summary(b)[c("adjr2", "cp", "bic")])))
print(signif(outMat,3),na.print="")
```

```
 Intercept     A       B      C     D E   depth     adjr2    cp    bic
     -10.6 0.71                                    0.745 50.60 -28.7
      52.3       -0.575 -2.19                      0.924  1.40 -56.9
    -184.0 2.33   1.840        4.27               0.925  2.15 -55.3
    -194.0 2.41   1.950        4.51    0.0122     0.923  3.86 -52.4
```

Observe that

- `E` does not appear in any of the models.
- Both for the *cp* (or C_p) criterion and and the *bic* (or *BIC*) criterion, the model with the smallest value is preferred. (Refer back to Section 3.5). Both choose the model with the two explanatory variables `B` and `C`.

The preferred model is, noting also the variance inflation factors:

```
BC.lm <- lm(porosity ~ B+C, data=coxite)
print( signif (coef(summary(BC.lm)), digits=3))
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    52.300     1.7800    29.3 4.01e-19
B              -0.575     0.0508   -11.3 1.19e-10
C              -2.190     0.1560   -14.1 1.81e-12
```

```
car :: vif (BC.lm)
```

```
    B     C
1.013 1.013
```

The variance inflation factors are $(1 - r^2)^{-1}$, where $r = 0.114$ is the correlation between the two variables.

Readers may care to check the diagnostic plots.[17]

Numbers that do not quite add up

Now round all the percentages to whole numbers, and repeat the analysis that uses all six available explanatory variables.

```
coxiteR <- coxite
coxiteR[,  1:5]  <- round(coxiteR[, 1:5])
coxiteR.lm <- lm(porosity ~ ., data=coxiteR)
print(coef(summary(coxiteR.lm)), digits=2)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    -1.425     23.404  -0.061  0.95212
A               0.560      0.246   2.278  0.03515
B               0.016      0.240   0.065  0.94865
C              -1.318      0.294  -4.484  0.00029
D               0.975      0.422   2.309  0.03305
E              -0.553      0.524  -1.055  0.30543
depth          -0.015      0.022  -0.667  0.51310
```

```
print(DAAG::vif(lm(porosity ~ .-E, data=coxiteR)), digits=2)
```

```
    A     B     C     D depth
 17.0  16.5   3.3   4.7   1.3
```

This result may seem surprising. Adding noise has reduced the correlation, so that `C` now appears significant even when all other explanatory variables are included. Perhaps more surprising is the $p = 0.033$ for `D`.

While this is contrived, we have from occasionally seen comparable effects in computer output that researchers have brought for scrutiny.

[17] `plot(BC.lm)`

Remedies for multicollinearity

As noted at the beginning of the section, careful initial choice of variables, based on scientific knowledge and careful scrutiny of relevant exploratory plots, will often avert the problem. Occasionally, it may be possible to find or collect additional data that will reduce correlations among the explanatory variables.

A variety of 'regularization' methods have been proposed for alleviating the effects of multicollinearity, in inflating coefficients and their standard errors. There is an overlap with variable selection issues. A good place to start is Breheny and Huang (2011) and the vignette that accompanies the *ncvreg* package.

3.7 Errors in x

The discussion so far has assumed, either that the explanatory variables are measured with negligible error, or that the interest is in the regression relationship given the observed values of explanatory variables. The present section is designed to draw attention to the major effect that errors in the explanatory variables can have on the regression gradients. The implications of the theoretical results, for particular practical circumstances, can be quite different from what might be intuitively expected. Discussion will mainly focus on the 'classical' errors in x model.

With a single explanatory variable, the effect under the classical "errors in x" model is to reduce the expected magnitude of the gradient. The attenuated gradient is less likely, relative to use of an x that is measured without error, to be distinguishable from statistical noise.

Measurement of dietary intake

The 36-page Diet History Questionnaire is a Food Frequency Questionnaire (FFQ) that was developed and evaluated at the U.S. National Cancer Institute. In large-scale trials that look for dietary effects on cancer and on other diseases, it has been important to have an instrument for measuring food intake that is relatively cheap and convenient. (Some trials have cost US$100 000 000 or more.)

The Schatzkin et al. (2003) study used the FFQ to ask for details of food intake over the previous year for 124 food items. It queried frequency of intake and, for most items, portion sizes. They investigated, also, the supplementary use of an instrument that questioned participants on their dietary intake in the previous 24 hours, then using the 24-hour dietary recall to calibrate the FFQ assessments.

Schatzkin et al. (2003) then compared FFQ measurements with those from Doubly Labeled Water, used as an accurate but expensive biomarker. They concluded that the FFQ was too inaccurate for its intended purpose. The 24-hour dietary recall, although better, was still seriously inaccurate. In some instances, the standard deviation for estimated energy intake was seven times the standard deviation, between different individuals, of the reference. A bias in the relationship between FFQ and reference further reduced the attenuation factor, to 0.04 for women and to 0.08 for men. For the relationship between the 24 hour recalls and the reference, the attenuation factors were 0.1 for women and 0.18 for men, though these could be improved by use of repeated 24-hour recalls.

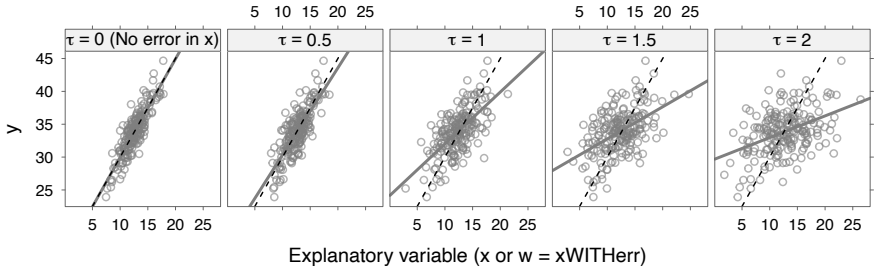


Figure 3.22 The fitted solid lines show how the change in the regression line as the error in x changes. The underlying relationship, shown with the dashed line, is in each instance $y = 15 + 1.5\,x$. For the definition of τ, see the text.

These results raise serious questions about what such studies can achieve, using presently available instruments whose cost and convenience makes them viable for use in large studies. Carroll (2004) gives an accessible summary of the issues. Subsection 7.9.6 has further brief comment on the modeling issues.

Simulations of the effect of measurement error

Suppose that the underlying regression relationship that is of interest is

$$y_i = \alpha + \beta x_i + \epsilon_i, \text{ where } \mathrm{var}(\epsilon_i) = \sigma^2 \ (i = 1, \ldots, n)$$

Let $s_x = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2/(n-1)}$ be the standard deviation of the values that are measured without error. Take the measured values as

$$w_i = x_i + \eta_i; \text{ where } \mathrm{var}(\eta_i) = s_x^2\,\tau^2,$$

The η_i are assumed independent of the ϵ_i.

Figure 3.22 shows results from a number of simulations that use the w_i as the explanatory values. If $\tau = 0.4$ (the added error has a variance that is 40% of s_x), the effect on the gradient is modest. If $\tau = 2$, the attenuation is severe. The function `DAAG::errorsINx()` can be used for additional simulations such as in Figure 3.22.

Estimation of the magnitude of the error and consequent attenuation of the slope requires a direct comparison with values that are measured with negligible error. More is required than the w and y values used to create scatterplots such as in Figure 3.22.

An estimate of the attenuation in the gradient is, to a close approximation:

$$\lambda = \frac{1}{1+\tau^2}, \text{ where } \lambda \text{ has the name reliability ratio.}$$

If for example $\tau = 0.4$, then $\lambda \simeq 0.86$. The study context will be important in deciding whether a reduction in the estimated gradient by a factor of 0.86 is of consequence. There may be more important concerns. Very small attenuation factors (large attenuations), e.g., less than 0.1 such as were found in the Schatzkin et al. (2003) study, are likely to seriously compromise the use of analysis results. Points to note are:

- From the data used in the panels of Figure 3.22, it is impossible to estimate τ, or to know the underlying x_i values. For this, an investigation is required that compares the w_i with an accurate, i.e., for all practical purposes error-free, determination of the x_i.
- A test for $\beta = 0$ can be undertaken in the usual way, but with reduced power to detect an effect that may be of interest.
- The t-statistic for testing $\beta = 0$ is affected in two ways; the numerator reduces by an expected factor of λ, while the standard error that appears in the numerator increases. Thus, if $\lambda = 0.1$, the sample size required to detect a non-zero gradient increases by more than the factor of 100 that is suggested by the effect on the gradient alone.

* *Two explanatory variables*

Consider first the case where one predictor is measured with error, and others without error. The coefficient of the variable that is measured with error is attenuated, as in the single variable case. The coefficients of other variables may be reversed in sign, or show an effect when there is none. See Carroll et al. (2006, pp. 52-55) for summary comment.

Suppose that

$$y = \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

If w_1 is unbiased for x_1 and the measurement error η is independent of x_1 and x_2, then least squares regression with explanatory variables w_1 and x_2 yields an estimate of $\lambda\beta_1$, where if ρ is the correlation between x_1 and x_2

$$\lambda = \frac{1-\rho^2}{1-\rho^2+\tau^2}$$

A new feature is the bias in the least squares estimate of β_2. The naive least squares estimator estimates

$$\beta_2 + \beta_1(1-\lambda)\,\gamma_{12}, \text{ where } \gamma_{12} = \rho\frac{s_1}{s_2} \tag{3.5}$$

Here γ_{12} is the coefficient of x_2 in the least squares regression of x_1 on x_2, $s_1 = \mathrm{SD}[x_1]$, $s_2 = \mathrm{SD}[x_2]$. The estimate of β_2 may be quite different from zero, even though $\beta_2 = 0$. Where $\beta_2 \neq 0$, the least squares estimate can be reversed in sign from β_2. Some of the effect of x_1 is transferred to the estimate of the effect of x_2.

Two explanatory variables, one measured without error – a simulation

The function `DAAG::errorsINx()`, when supplied with a non-zero value for the argument `gpdiff`, simulates the effect when the variable that is measured without error codes for a categorical effect. Figure 3.23 had `gpdiff=1.5`. Two lines appear, suggesting a 'treatment' effect where there was none.

The function `errsINseveral()` simulates a model where there are two continuous variables x_1 and x_2. The default choice of arguments has

$$\beta_1 = 1.5,\ \beta_1 = 0,\ \rho = -0.5,\ s_1 = s_2 = 2,\ \tau = 1.5,\ \mathrm{var}[\epsilon] = 0.25$$

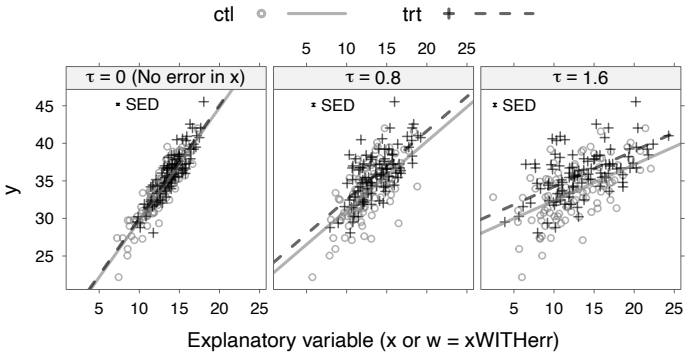


Figure 3.23 Errors in x may readily generate spurious differences between groups. For the simulations whose results are shown, y is a linear function of x. The mean value of x is 12.5 for the first group level (`"ctl"`), and 14.0 for the second (`"trt"`) level. In the left panel, values of x are measured without error. In the middle and right panels, independent errors have been added to x from distributions with SDs that are respectively 0.8 and 1.6 times that of the within group standard deviation of x. The SEDs are conditional on $w =$ `xWITHerr`.

Measurement error variances are x_1: $s_1^2\tau^2$, x_2: 0. Then $\lambda = 0.25$, $\gamma_{12} = -0.5$, and the expected value for the naive least squares estimator of β_2 is

$$\beta_2 + \beta_1(1-\lambda) = 0 + 1.5 \times 0.75 \times (-0.5) = -0.5675$$

An arbitrary number of variables

Where two or more exploanatory variables are measured with substantial error, this widens the range of possibilities for transferring some part or parts of effects between variables. The function `DAAG::errorsINseveral()` can be used for simulations with an arbitrary correlation structure for explanatory variables, and with an arbitrary variance-covariance matrix for the added errors.[18]

**The classical error model versus the Berkson error model*

In the classical model $\mathrm{E}[w_i|x_i] = x_i$. In the Berkson model $\mathrm{E}[x_i|w_i] = w_i$. The Berkson model may be a realistic model in an experiment where w_i is an instrument setting, but the true value varies randomly about the instrument setting. For example, the temperature in an oven or kiln may be set to w_i, but the resulting (and unknown) actual temperature is x_i. In straight line regression, the coefficient is then unbiased, but the variance of the estimate of the coefficient is increased.

Zeger et al. (2000) discuss the practical consequences of both types of error,

[18] If $\boldsymbol{\beta}$ is the vector of coefficients in the model without errors in the measured values, V corresponds in the obvious way to `V`, and U to `xerrV`, then an estimate for the resulting least squares estimates when for regression on the values that are measured with error is $\boldsymbol{\beta}' V(V+U)^{-1}$. Note that Zeger et al. (2000, p.421) have an initial T (our U), where V is required.

though giving most of their attention to the classical model. In their context, realistic models may have elements of both the classical and Berkson models.

Using missing value approaches to address measurement error

Errors that arise from multiple imputation, to be discussed in Section 9.8, can be treated as a form of measurement error. Blackwell et al. (2017) offer a methodology.

3.8 Multiple regression models – additional points

The notes that follow should dispel any residual notion that this chapter's account of multiple regression models has covered everything of importance.

3.8.1 Confusion between explanatory and response variables

As an example, we return to the `allbacks` data. We compare the coefficients in the equation that predicts `area` given `volume` and `weight`, with the rearranged coefficients from the equation that predicts `weight` given `volume` and `area`:

```
coef(lm(area ~ volume + weight, data=allbacks))
```

```
(Intercept)      volume      weight
    35.4587     -0.9636      1.3611
```

```
b <- as.vector(coef(lm(weight ~ volume + area, data=allbacks)))
c("_Intercept_"=-b[1]/b[3], volume=-b[2]/b[3], weight=1/b[3])
```

```
_Intercept_      volume      weight
    -47.847      -1.512       2.135
```

Only if the relationship is exact, so that predicted time is the same as observed time, will the equations be the same. Williams (1983) gives examples from the earth sciences literature.

Unintended correlations

Suppose that x_i $(i = 1, 2, \ldots, n)$ are results from a series of controls, while y_i $(i = 1, 2, \ldots, n)$ are the results from the corresponding treated group. It is tempting to plot $y - x$ versus x. Unfortunately, there is likely to be a negative correlation between $y - x$ and x, though this is not inevitable. This emphasizes the desirability of maintaining a clear distinction between explanatory and response variables. See the example in Sharp et al. (1996).

3.8.2 Missing explanatory variables

Here the issue is use of the wrong model for the expected value. With the right 'balance' in the data, the expected values are unbiased or nearly unbiased. Where there is serious imbalance, the bias may be large.

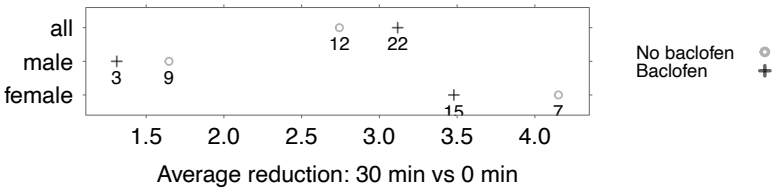


Figure 3.24 Does pre-operative baclofen (additional to earlier painkiller), reduce pain? Subgroup numbers, shown below each point, weight the overall averages when sex is ignored.

Figure 3.24 relates to data collected in an experiment on the use of painkillers (Gordon et al., 1995). Pain was measured as a VAS (Visual-Analogue Scale) score. Researchers were investigating differences in the pain score between the two analgesic treatments, without and with baclofen.

Notice that the overall comparison (average for baclofen versus average for no baclofen) goes in a different direction from the comparison for the two sexes separately. As the two treatment groups had very different numbers of men and women, and as there was a strong sex effect, an analysis that does not account for the sex effect gives an incorrect estimate of the treatment effect (Cohen, 1996).

The overall averages in Figure 3.24 reflect the following subgroup weighting effects (f is shorthand for female and m for male):

Baclofen: 15f to 3m, i.e., $\frac{15}{18}$ to $\frac{3}{18}$ (a little less than f average)
No baclofen: 7f to 9m, i.e., $\frac{7}{16}$ to $\frac{9}{16}$ ($\approx \frac{1}{2}$-way between m & f)

There is a sequel. More careful investigation revealed that the response to pain has a different pattern over time. For males, the sensation of pain declined more rapidly over time.

Strategies

(i) Simple approach Calculate means for each subgroup separately.
Overall treatment effect is average of subgroup differences.
Effect of baclofen (reduction in pain score from time 0) is:

Females: 3.479 - 4.151 = -0.672 (-ve, therefore an increase)
Males: 1.311 - 1.647 = -0.336
Average over male and female = -0.5 × (0.672+0.336) = -0.504

(ii) Fit a model that accounts for sex and baclofen effects

$$y = \text{overall mean} + \text{sex effect} + \text{baclofen effect} + \text{interaction}$$

(At this point, we are not including an error term).

When variables or factors are omitted from models, values of the outcome variable are as far as possible accounted for using those that remain. The mouse brain weight example in Subsection 3.2.5 can be understood in this way. Bland and Altman (2005) give several examples of published results where conclusions have been vitiated by effects of this type.

Another example of this same type, albeit in the context of contingency tables, was discussed in Subsection 2.1.2. The analysis of the UCB admissions data in Section 5.3 formulates the analysis of contingency table data as a regression problem.

3.8.3* Added variable plots

Added variable plots are a partial alternative to the use of termplots, described in Section 3.3. They can be created using the function `avPlots()` in the *car* package. Such plots are designed to examine the contribution of a variable z, given variables already in the model. Here, variables already in the model will be collectively represented by the symbol x.

As a starting point for understanding added variable plots, observe that a multiple regression calculation can be handled as a sequence of straight line regressions. As an example, consider calculations based on the `nihills` dataset, with $y =$ `ltime = log(time)`, $x =$ `lclimb = log(climb)` and $z =$ `ldist = log(dist)`. As x is here a single variable, comprising a single column of the model matrix, the individual calculation steps can be performed using straight line regression.

The sequence of steps is:

1. Regress y on x, with vector of residuals $e_{y|x}$. For the example data:

```
yONx.lm <- lm(logtime ~ logclimb, data=lognihr)
e_yONx <- resid(yONx.lm)
print(coef(yONx.lm), digits=4)
```

```
(Intercept)    logclimb
    -7.1047      0.9021
```

2. Regress z on x, with vector of residuals $e_{z|x}$

```
zONx.lm <- lm(logdist ~ logclimb, data=lognihr)
e_zONx <- resid(zONx.lm)
print(coef(yONx.lm), digits=4)
```

```
(Intercept)    logclimb
    -7.1047      0.9021
```

3. Regress $e_{y|x}$ (from 1 above) on $e_{z|x}$ (from 2 above), with vector of residuals $e_{y|xz}$.

```
ey_xONez_x.lm <- lm(e_yONx ~ 0+e_zONx)
e_yONxz <- resid(ey_xONez_x.lm)
print(coef(ey_xONez_x.lm), digits=4)
```

```
e_zONx
0.6814
```

Note that as $e_{y|x}$ and $e_{z|x}$ have mean 0, the constant term in the regression equation is zero. The effect is to reduce the residual sum of squares from $\sum_{i=1}^{n} e_{y|z}(i)^2$ to $\sum_{i=1}^{n} e_{y|xz}(i)^2$. (Use of $e_{z|x}$ as an explanatory variable can only decrease the sum of squares, or perhaps leave it unchanged.)

The coefficients for the regression of y on x and z can be recovered by putting together the results from the regressions in items 1 – 3 above. Details of the algebra that recovers the regression parameters, which are a little intricate, are given below.

For each of the regressions 1, 2 and 3, there is a plot of residuals against fitted values. In the simplest case to consider, all three relationships are linear. Or if one of them is not linear there must, to obtain a multiple linear regression, be a compensating nonlinear trend in one or both of the other two. The following shows the plots of residuals against fitted values, for each of the three regressions.

The plot of $e_{y|x}$ against $e_{z|x}$ has the name *added variable plot*, here for $z =$ `ldist`. Panel A in Figure 3.25 has the added variable plot for `ldist`, while Panel B has the added variable plot for `lclimb`. Both panels use the function `car::avPlots()`, which is designed to leave out one variable (or, more generally, term) only at a time:

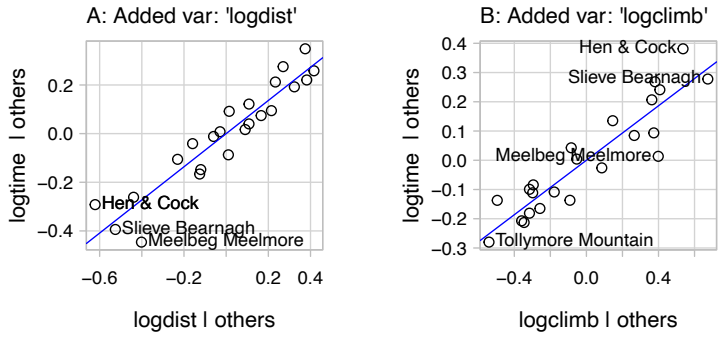


Figure 3.25 Added variable plots, A: for `ldist` (after fitting `lclimb`), and B: for `lclimb` (after fitting `ldist`). Panel A provides a check whether, given that $x =$ `lclimb` is included as an explanatory variable, $z =$ `ldist` should be also included. In Panel B, the roles of the two variables are reversed.

```
## Code for added variable plots
logtime.lm <- lm(logtime ~ logclimb+logdist, data=lognihr)
car::avPlots(logtime.lm, lwd=1, terms="logdist", fg="gray")
car::avPlots(logtime.lm, lwd=1, terms="logclimb", fg="gray")
```

The following single call gives both plots:

```
car::avPlots(timeClimb.lm, terms=~.)
```

The first of the plots can alternatively be obtained, based on the straight line calculations given above, from:

```
plot(e_yONx ~ e_zONx)
```

Figure 3.26 has plots of residuals against fitted values, for each of the three regressions. The final plot is for the residuals and fitted values from the added variable plot for `lclimb`. This is interesting because apparent nonlinearities in the first two plots have, in the final plot, largely canceled out.
Slightly modified code is:

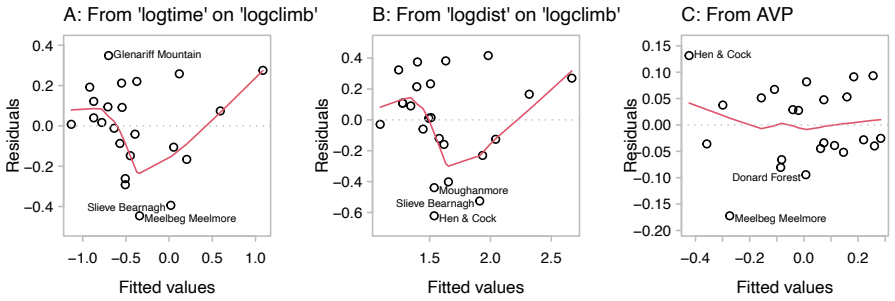


Figure 3.26 The vertical axis shows, respectively, residuals from the regressions, A: `logtime` on `logclimb`; B: `logtime` on `logdist`; and C: Residuals from A vs residuals from B (AVP = added variable plot). These are in each case plotted against fitted values from the corresponding regression.

```
plot(yONx.lm, which=1, caption="A: ... 'logtime' on 'logclimb'")
plot(zONx.lm, which=1, caption="B: ... 'logdist' on 'logclimb'")
plot(ey_xONez_x.lm, which=1, caption="C: From Added Variable plot")
```

Alternatives to Added Variable Plots

The plot of $e_{y|x}$ against z, i.e., regress y on x and plot the residuals against z, is not satisfactory for judging whether z should be included in the regression. Observe that residuals from the regression of y on x take the form:

$$e_{y|x} = y - a_{y|x} - b_{y|x}\, x$$

If x and z have a non-zero correlation, this will lead to a non-zero correlation between z and the term $b_{y|x}\, x$) in $e_{y|x}$. Termplots, also known as 'component plus residual' plots, avoid this issue because the residuals in the 'component plus residual' are residuals from the model with all terms fitted.

Termplots may on the whole be more informative than added variable plots. While systematic departures from linearity in one or more of the added variable plots may indicate the need for transforming one or more variables, the fact that both ordinates represent residuals from a regression calculation can make it difficult to establish a direct connection back to the original variables.

**Algebraic details*

The algebraic details are given for completeness. While not essential for understanding the account just given, they may help reinforce understanding.

The regression equations can be written:

1. $y = a_{y|x} + b_{y|x}\, x + e_{y|x}$.
   Coefficients are $a_{y|x}$ = -7.1047; $b_{y|x}$ = 0.9021 [Type `coef(yONx.lm)`]
2. $z = a_{z|x} + b_{z|x}\, x + e_{z|x}$, i.e., $e_{z|x} = z - a_{z|x} - b_{z|x}\, x$
   Coefficients are $a_{z|x}$ = -3.146; $b_{z|x}$ = 0.6404 [Type `coef(zONx.lm)`]
3. $e_{y|x} = b_{y|xz}\, e_{z|x} + e_{y|xz}$

The coefficient is $b_{y|xz} = 0.6814$. There is no intercept term because both sets of residuals have mean equal to 0. [Type `coef(e_yxONe_zx.lm)`]

Thus:

$$\begin{aligned} y &= a_{y|x} + b_{y|x}x + e_{y|x} \quad \text{(from 1 above)} \\ &= a_{y|x} + b_{y|x}x + b_{y|xz}e_{z|x} + e_{y|xz} \quad \text{(substituting from 3 above)} \\ &= a_{y|x} + b_{y|x}x + b_{y|xz}\,(z - a_{z|x} - b_{z|x}\,x) + e_{y|xz} \quad \text{(substituting from 2 above)} \\ &= a_{y|x} - b_{y|xz}\,a_{z|x} + (b_{y|x} - b_{z|x})\,x + b_{y|xz}\,z + e_{y|xz} \end{aligned}$$

Then the terms in the fitted equation $\widehat{y} = a + b_1 x + b_2 z$ are:

$a = a_{y|x} - b_{y|xz}\,a_{z|x}$ = -7.1047 - 0.6814 × (-3.146) = -4.9611
$b_1 = b_{y|x} - b_{y|xz}\,b_{z|x}$ = 0.9021 - 0.6814 × 0.6404 = 0.4658
$b_2 = b_{y|xz}$ = 0.6814

Compare the above coefficients with:

```
coef(lm(logtime ~ logclimb + logdist, data=lognihr))
```

```
(Intercept)    logclimb     logdist
    -4.9611      0.4658      0.6814
```

As noted above, wherever x appears, any number of explanatory variables can be fitted. The argument is essentially unchanged.

3.8.4 Nonlinear methods – an alternative to transformation?*

Aside from the brief discussion that follows, we have not found room for this important topic in the present text. We will investigate the use of the R `nls()` function (*stats* package) to shed light on the loglinear model that Subsection 3.2.1 used for the hill race data.

The analysis of Subsection 3.2.1 assumed additive errors on the transformed logarithmic scale. This implies, on the untransformed scale, multiplicative errors. We noted that the assumption of homogeneity of errors was in doubt.

One might alternatively assume that the noise term is additive on the untransformed scale, leading to the nonlinear model

$$y = x_1^{\alpha}\, x_2^{\beta} + \varepsilon$$

where $y =$ `time`, $x_1 =$ `dist`, and $x_2 =$ `climb`. We will use the `nls()` nonlinear least squares function to estimate α and β. The iterative procedure used requires starting values for α and β. Here, the estimates from the earlier loglinear regression will be used for this purpose. Because we could be taking a square or cube of the `climb` term, we prefer to work with the variable `climb.mi` that is obtained by dividing `climb` by 5280, so that numbers are of modest size:

```
nihr$climb.mi <- nihr$climb/5280
nihr.nls0 <- nls(time ~ (dist^alpha)*(climb.mi^beta), start =
                   c(alpha = 0.68, beta = 0.465), data = nihr)
## plot(residuals(nihr.nls0) ~ log(predict(nihr.nls0)))
```

The parameter estimates are:

```
signif(coef(summary(nihr.nls0)),3)
```

```
      Estimate Std. Error t value Pr(>|t|)
alpha    0.315    0.00806    39.1 4.25e-21
beta     0.814    0.02950    27.6 5.46e-18
```

These parameter estimates differ substantially from those obtained under the assumption of multiplicative errors. This is not an unusual occurrence; the nonlinear least squares problem has an error structure that is different from the linearized least-squares problem solved earlier. Residuals suggest a nonlinear pattern.

Another possibility, that allows `time` to increase nonlinearly with `climb.mi`, is

$$y = \alpha + \beta x_1 + \gamma x_2^{\delta} + \varepsilon.$$

We then fit the model, using an arbitrary starting guess:

```
nihr.nls <- nls(time ~ gamma + delta1*dist^alpha + delta2*climb.mi^beta,
start=c(gamma = .045, delta1 = .09, alpha = 1,
delta2=.9, beta = 1.65), data=nihr)
## plot(residuals(nihr.nls) ~ log(predict(nihr.nls)))
```

The starting values were obtained from an initial model in which `alpha` was constrained to equal 1. The result is:

```
signif(coef(summary(nihr.nls)),3)
```

```
       Estimate Std. Error t value     Pr(>|t|)
gamma    0.1520     0.0714    2.13 0.0471000000
delta1   0.0399     0.0284    1.41 0.1770000000
alpha    1.3100     0.2710    4.84 0.0001330000
delta2   0.8660     0.0922    9.39 0.0000000234
beta     1.5100     0.1810    8.32 0.0000001380
```

There are no obvious outliers in the residual plot. In addition, there is no indication of an increase in the variance as the fitted values increase. Thus, a variance-stabilizing transformation or the use of weighted least squares is unnecessary.

3.9 Recap

A coefficient in a multiple regression equation predicts the effect of a variable when other variables are held constant. Coefficients can thus be different, sometimes dramatically, for a different choice of explanatory variables.

Regression equation predictions are for the data used to derive the equation, and reflect any sampling biases that affect that data. Biases that arise because data were not randomly sampled from the population can lead to predictions that, for the population as a whole, are seriously astray.

Plots that can be useful for checking regression assumptions and/or for checking whether individual data points may unduly influence results include: scatterplot matrices of the variables in the regression equation, plots of residuals against fitted values, partial residual plots such as are provided by the `termplot()` function (these

are a better guide than plots of residuals against individual explanatory variables), normal quantile-quantile plots of residuals, scale-location plots, and Cook's distance and related plots.

Watch for variables whose measurements may be affected by large inaccuracies. Where the interest is in the regression coefficients, their effects will be attenuated. They have the potential to generate spurious effects from other explanatory variables.

Robust methods downweight points that may be outliers. Methods that are resistant to outliers aim, as far as possible, to completely remove the contribution of outliers to the regression fit.

3.10 Further reading

Faraway (2014), and the more detailed and wide-ranging account Gelman, Hill, and Vehtari (2020), are both highly recommended. Both address many important practical issues, with Chapters 20 and 21 of the Gelman et al book a good starting point for considering issues of causal inference.

Cook and Weisberg (1999) rely heavily on graphical explorations to uncover regression relationships. Tu and Gilthorpe (2011) explore in detail a wide range of issues that arise for the interpretation of regression coefficients. Its insightful commentary has wide eneral relevance. Venables and Ripley (2002) is a basic reference for using R for regression calculations. See also Fox and Weisberg (2018). Harrell (2015) is wide-ranging and has extensive practical advice, but does makes strong technical demands on the reader. Hastie et al. (2009) explore beyond the models and modeling approaches that we have described.

Cunningham (2021) weaves a wide-ranging review of the causal inference literature into an engaging story, replete with examples that highlight important issues. Code is provided for R, Stata, and Python. While directed in the first place at economics students, the methodology and examples have wide social science, public health, and other relevance. Hernán and Robins (2020) is a more conventional style of text that is similarly wide-ranging and practically oriented, with R, SAS, Stata, and Python code. Rosenbaum (2002) is another good starting point for engaging with the issues that observational studies raise. See further Section 9.7 and additional references noted in Section 9.7. Results from observational studies can rarely be interpreted with the same confidence as for a carefully designed experimental study.

Several of the studies that are discussed in Leavitt and Dubner (2005), some with major public policy relevance, relied to a greater or lesser extent on regression methods. References in the notes at the end of their book allow interested readers to pursue technical details of the statistical and other methodology. The conflation of multiple sources of insight and evidence is invariably necessary, if conclusions are to carry conviction. Emphasizing the difficulty of reaching watertight conclusions, Leavitt's claim that legalizing abortion reduced the US crime rate has generated extensive controversy in the literature. See Shoesmith (2017) and Donohue and Levitt (2019).

Especially hazardous is the use of analyses where there are multiple potential confounding variables, i.e., variables which must be available and whose effects (including possible interaction effects) must be properly modeled if coefficients for other variables are to be genuinely suggestive of a causal link. These issues get further consideration in Section 9.7.3.

On variable selection, which warrants more attention than we have given it, see Bolker (2008), Harrell (2015), Hastie et al. (2009), and Venables (1998). Bolker's account extends (p.215) to approaches that weight and average models.

On errors in variables, see Carroll et al. (2006). Linear models are a special case of nonlinear models!

Structural equation models allow, in addition to explanatory variables and response variables, intermediate variables that are response with respect to one or more of the explanatory variables, and explanatory with respect to one or more of the response variables. Cox and Wermuth (1996) and Edwards (2000) describe approaches that use regression methods to elucidate the relationships. Cox and Wermuth is useful for the large number of examples and for its illuminating comments on practical issues, while Edwards has a more up to date account of the methodology.

Bates and Watts (1988) discuss nonlinear models in detail. A more elementary presentation is given in one of the chapters of Myers (1990).

3.11 Exercises

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

```
## Set up factor that identifies the `have' cities
cities <- DAAG::cities
cities$have <- with(cities, factor(REGION %in% c("ON","WEST"),
                                   labels=c("Have-not","Have")))
```

   Plot the 1996 population against that for 1992, distinguishing the two categories of city, both using the raw data and using the log transformed values, thus:

```
lattice::xyplot(POP1996~POP1992, groups=have, data=cities,
                auto.key=list(columns=2))
lattice::xyplot(log(POP1996)~log(POP1992), groups=have, data=cities,
                auto.key=list(columns=2))
```

   Which of these plots is preferable? Explain.

   Now carry out the regressions

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

   Examine diagnostic plots. Which model seems preferable? Interpret the results.

2. Calculate volumes (`volume`) and page areas (`area`) for the books on which information is given in the data frame `DAAG::oddbooks`.
   a. Plot `log(weight)` against `log(volume)`, and fit a regression line.
   b. Plot `log(weight)` against `log(area)`, and again fit a regression line.
   c. Which of the lines (a) and (b) gives the better fit?
   d. Repeat (a) and (b), now with `log(density)` in place of `log(weight)` as the dependent variable. Comment on how results from these regressions may help explain the results obtained in (a) and (b).
3. The `MASS::cpus` data frame gives information on 8 aspects for each of 209 different types of computers. Read the help page for more information.
   a. Construct a scatterplot matrix for these data, as in Figure 3.3 in Subsection 3.2.1. Should any of the variables be transformed before further analysis is conducted?
   b. How well does estimated performance (`estperf`) predict performance (`perf`)? Study this question by constructing a scatterplot of these two variables, after taking logarithms. Do the plotted points scatter about a straight line or is there an indication of nonlinearity? Is variability in performance the same at each level of performance?
4. In the dataset `MASS::cement`, examine the dependence of `y` (amount of heat produced) on `x1`, `x2`, `x3` and `x4` (which are proportions of four constituents). Begin by examining the scatterplot matrix. As the explanatory variables are proportions, do they require transformation, perhaps by taking $\log(x/(100-x))$? What alternative strategies might help identify an equation for predicting heat?
5. Use the model fitted to the data in `nihills` to give predicted values for the data in `hills2000`. Plot these against predicted values from the model fitted to `hills2000`, and use differences from observed values of `logtime` to estimate a prediction variance that is relevant when Northern Irish data are used to predict Scottish times. Would you expect this variance to be larger or smaller than the estimated error variance from the `hills2000` model fit? Is this what you see?
6. The data frame `DAAG::hills2000` updates the 1984 information in the dataset `hills`. Fit regression models, for men and women separately, based on the data in `hills`. Check whether they fit satisfactorily over the whole range of race times. Compare the two equations.
7. Section 3.1 used `lm()` to analyze the allbacks data that are presented in Figure 3.1. Repeat the analysis using (1) the function `MASS::rlm()`, and (2) the function `MASS::lqs()`. Compare the two sets of results with the results in Section 3.1.
8. The following investigates the consequences of not using a logarithmic transformation for the `nihills` data analysis. The second differs from the first in having a `dist` × `climb` interaction term, additional to linear terms in `dist` and `climb`.
   a. Fit and compare the two models:

```
nihills.lm  <- lm(time ~ dist+climb, data=DAAG::nihills)
nihillsX.lm <- lm(time ~ dist+climb+dist:climb, data=DAAG::nihills)
```

```
anova( nihills.lm , nihillsX.lm )    # Use `anova()` to make the comparison
coef(summary(nihillsX.lm))           # Check coefficient for interaction term
drop1(nihillsX.lm)
```

b. Using the F-test result, make a tentative choice of model, and proceed to examine diagnostic plots. Are there any problematic observations? What happens if these points are removed? Re-fit both of the above models, and check the diagnostics again.

9. Fit the model `brainwt ~ bodywt + lsize` to the `litters` dataset, then checking the variance inflation factors for `bodywt` and for `lsize`. Comment.
10. Apply the `lm.ridge()` function to the `litters` data, using the generalized cross-validation (GCV) criterion to choose the tuning parameter. (GCV is an approximation to cross-validation.)

a. In particular, estimate the coefficients of the model relating `brainwt` to `bodywt` and `lsize` and compare with the results obtained using `lm()`.
b. Using both ridge and ordinary regression, estimate the mean brain weight when litter size is 10 and body weight is 7. Use the bootstrap, with case-resampling, to compute approximate 95% percentile confidence intervals using each method. Compare with the interval obtained using `predict.lm()`.

11. *Compare the ranges of `dist` and `climb` in the data frames `nihills` and `hills2000`. In which case would you expect it to be harder to find a well-fitting model? For each of these data frames, fit the models based on each of the formulae:

```
log(time) ~ log(dist) + log(climb)     ## lm model
time ~ alpha*dist + beta*I(climb^2)    ## nls model
```

Is there one model that gives the best fit in both cases?

12. The data frame `MPV::table.b3` has data on gas mileage and eleven other variables for a sample of 32 automobiles.

a. Construct a scatterplot of `y` (mpg) versus `x1` (displacement). Is the relationship between these variables nonlinear?
b. Use the `xyplot()` function, and `x11` (type of transmission) as a `group` variable. Is a linear model reasonable for these data?
c. Fit the model, relating `y` to `x1` and `x11`, which gives two lines having possibly different gradients and intercepts. Check the diagnostics. Are there any influential observations? Are there any influential outliers?
d. Plot the residuals against the variable `x7` (number of transmission speeds), again using `x11` as a `group` variable. Comment on anything unusual about this plot?

13. The following code is designed to explore effects that can result from the omission of explanatory variables:

```
x1 <- runif(10)              # predictor which will be missing
x2 <- rbinom(10, 1, 1-x1)
  ## observed predictor, depends on missing predictor
y <- 5*x1 + x2 + rnorm(10,sd=.1)  # simulated model; coef of x2 is positive
```

```
y.lm <- lm(y ~ factor(x2)) # model fitted to observed data
coef(y.lm)
```

```
(Intercept) factor(x2)1
      3.971      -0.827
```

```
y.lm2 <- lm(y ~ x1 + factor(x2))  # correct model
coef(y.lm2)
```

```
(Intercept)          x1 factor(x2)1
     0.3744      4.5778      0.8724
```

What happens if `x2` is generated according to `x2 <- rbinom(10, 1, x1)`? Repeat with `x2 <- rbinom(10, 1, .5)`.

14. Fit the model investigated in Subsection 3.8.4, omitting the parameter α. Investigate and comment on changes in the fitted coefficients, standard errors and fitted values.
15. Figure 3.23 used the function `errorsINx()`, with the argument `gpdiff=1.5`, to simulate data in which the regression relationship $y = 15 + 1.5x$ is the same in each of two groups (called `ctl` and `trt`). The left panel identifies the two fitted lines when the explanatory variable is measured without error. These are, to within statistical error, identical. The right panel shows the fitted regression lines when random error of the same order of magnitude as the within groups variation in x is added to x, giving the column of values $zWITHerr$.
    a. Run the function for several different values of `gpdiff` in the interval (0,1.5), and plot the estimate of the treatment effect against `gpdiff`.
    b. Run the function for several different values of `timesSDz` in the interval (0,1.5) and plot the estimate of the treatment effect against `gpdiff`.
    c. Run the function with `beta = c(-1.5,0)`. How does the estimate of the treatment effect change, as compared with `b = c(1.5,0)`? Explain the change.
16. Fit the following two resistant regressions, in each case plotting the residuals against `Year`.

```
bomData <- DAAG::bomregions2021
nraw.lqs <- MASS::lqs(northRain ~ SOI + CO2, data=bomData)
north.lqs <- MASS::lqs(I(northRain^(1/3)) ~ SOI + CO2, data=bomData)
plot(residuals(nraw.lqs) ~ Year, data=bomData)
plot(residuals(north.lqs) ~ Year, data=bomData)
```

Compare, also, normal quantile-quantile plots for the two sets of residuals.

    a. Repeat the calculations several times. Comment on the extent of variation, from one run to the next, in the regression coefficients.
    b. Based on examination of the residuals, which regression model seems more acceptable: `nraw.lqs` or `north.lqs`?
    c. Compare the two sets of regression coefficients. Can you explain why they are so very different?

(More careful modeling will take into account the temporal sequence in the observations. See Section 6.2 for an analysis that does this.)

17. Consider the National Football League (NFL) 1976 team performance data in `MPV::table.b1`. The data frame has 28 observations on 10 variables, including `y`, the number of games won.
    a. Fit a model relating `y` to the main effects of all other variables, assigning the `lm` object to `full.lm`.
    b. Fit a model relating `y` to `x2` and `x8` (opponents' rushing yards) together, assigning the `lm` object to `reduced2.lm`.
    c. Fit a model relating `y` to `x2` (passing yards) only, assigning the `lm` object to `reduced1.lm`.
    d. Use the `anova()` function with `test="Cp"` to decide which of the three models should be preferred.
    e. Compare the AIC statistics. Do they tell a similar story?
    f. Use the following code to create a plot that compares estimated effect sizes for the replicate with those for the original, for the social psychology studies in the dataset `DAAG::repPsych`, discussed in 1.10.2, that relates to the paper Open Science Collaboration (2015).

```
socpsych <- subset(DAAG::repPsych, Discipline=='Social')
with(socpsych, scatter.smooth(T_r.R~T_r.O))
abline(v=.5)
```

    Now fit a robust regression to the points for which the original estimated effect size was 0.5 or less:

```
soc.rlm <- MASS::rlm(T_r.R~T_r.O, data=subset(socpsych, T_r.O<=0.5))
## Look at summary plots
termplot(soc.rlm,   partial.resid =T, se=T)
```

```
plot(soc.rlm)
```

    Which points appear to differ from the fitted line by an amount that is greater than comfortably explained as statistical variation?