Example: predicting the yields of mesquite bushes

We illustrate some ideas of model checking with a real-data example that is nonetheless somewhat artificial in being presented in isolation from its applied context. Partly because this example is not a "success story" and our results are inconclusive, it represents the sort of analysis a student might perform in exploring a new dataset.

Data were collected in order to develop a method of estimating the total production (biomass) of mesquite leaves using easily measured parameters of the plant, before actual harvesting takes place. Two separate sets of measurements were taken, one on a group of 26 mesquite bushes and the other on a different group of 20 mesquite bushes measured at a different time of year. All the data were obtained in the same geographical location (ranch), but neither constituted a strictly random sample.

The outcome variable is the total weight (in grams) of photosynthetic material as derived from actual harvesting of the bush. The input variables are:

diam1:	diameter of the canopy (the leafy area of the bush) in meters, measured along the longer axis of the bush
diam2:	canopy diameter measured along the shorter axis
canopy.height:	height of the canopy
total.height:	total height of the bush
density:	plant unit density (# of primary stems per plant unit)
group:	group of measurements (0 for the first group, 1 for the second group)

It is reasonable to predict the leaf weight using some sort of regression model. Many formulations are possible. The simplest approach is to regress `weight` on all of the predictors, yielding the estimates:

R output

```
lm(formula = weight ~ diam1 + diam2 + canopy.height + total.height +
    density + group, data = mesquite)
                coef.est coef.se
(Intercept)       -729     147
diam1              190     113
diam2              371     124
canopy.height      356     210
total.height      -102     186
density            131      34
group             -363     100
  n = 46, k = 7
  residual sd = 269, R-Squared = 0.85
```

To get a sense of the importance of each predictor, it is useful to know the range of each variable:

R output

```
                 min   q25 median   q75    max   IQR
diam1            0.8   1.4    2.0   2.5    5.2   1.1
diam2            0.4   1.0    1.5   1.9    4.0   0.9
canopy.height    0.5   0.9    1.1   1.3    2.5   0.4
total.height     0.6   1.2    1.5   1.7    3.0   0.5
density          1.0   1.0    1.0   2.0    9.0   1.0
group            0.0   0.0    0.0   1.0    1.0   1.0

weight            60   220    360   690   4050   470
```

"IQR" in the last column refers to the *interquartile range*—the difference between the 75^{th} and 25^{th} percentile points of each variable.

But perhaps it is more reasonable to fit on the logarithmic scale, so that effects are multiplicative rather than additive:

R output

```
lm(formula = log(weight) ~ log(diam1) + log(diam2) + log(canopy.height) +
    log(total.height) + log(density) + group, data = mesquite)
                  coef.est coef.se   IQR of predictor
(Intercept)           5.35    0.17    --
log(diam1)            0.39    0.28    0.6
log(diam2)            1.15    0.21    0.6
log(canopy.height)    0.37    0.28    0.4
log(total.height)     0.39    0.31    0.4
log(density)          0.11    0.12    0.3
group                -0.58    0.13    1.0
  n = 46, k = 7
  residual sd = 0.33, R-Squared = 0.89
```

Instead of, "each meter difference in canopy height is associated with an additional 356 grams of leaf weight," we have, "a difference of x% in canopy height is associated with an (approximate) positive difference of $0.37x$% in leaf weight" (evaluated at the same levels of all other variables across comparisons).

So far we have been throwing all the predictors directly into the model. A more "minimalist" approach is to try to come up with a simple model that makes sense. Thinking geometrically, we can predict leaf weight from the volume of the leaf canopy, which we shall roughly approximate as

$$\text{canopy.volume} = \text{diam1} \cdot \text{diam2} \cdot \text{canopy.height}.$$

This model is an oversimplification: the leaves are mostly on the surface of a bush, not in its interior, and so some measure of surface area is perhaps more appropriate. We shall return to this point shortly.

It still makes sense to work on the logarithmic scale:

R output

```
lm(formula = log(weight) ~ log(canopy.volume))
                  coef.est coef.se
(Intercept)           5.17    0.08
log(canopy.volume)    0.72    0.05
  n = 46, k = 2
  residual sd = 0.41, R-Squared = 0.80
```

Thus, leaf weight is approximately proportional to `canopy.volume` to the 0.72 power. It is perhaps surprising that this power is not closer to 1. The usual explanation for this is that there is variation in `canopy.volume` that is unrelated to the weight of the leaves, and this tends to *attenuate* the regression coefficient—that is, to decrease its absolute value from the "natural" value of 1 to something lower. Similarly, regressions of "after" versus "before" typically have slopes of less than 1. (For another example, Section 7.3 has an example of forecasting congressional elections in which the vote in the previous election has a coefficient of only 0.58.)

The regression with only `canopy.volume` is satisfyingly simple, with an impressive R-squared of 80%. However, the predictions are still much worse than the model with all the predictors. Perhaps we should go back and put in the other predictors. We shall define:

$$\begin{aligned} \text{canopy.area} &= \text{diam1} \cdot \text{diam2} \\ \text{canopy.shape} &= \text{diam1}/\text{diam2}. \end{aligned}$$

The set (canopy.volume, canopy.area, canopy.shape) is then just a different parameterization of the three canopy dimensions. Including them all in the model yields:

R output
```
lm(formula = log(weight) ~ log(canopy.volume) + log(canopy.area) +
    log(canopy.shape) + log(total.height) + log(density) + group)
                   coef.est coef.se
(Intercept)            5.35    0.17
log(canopy.volume)     0.37    0.28
log(canopy.area)       0.40    0.29
log(canopy.shape)     -0.38    0.23
log(total.height)      0.39    0.31
log(density)           0.11    0.12
group                 -0.58    0.13
  n = 46, k = 7
  residual sd = 0.33, R-Squared = 0.89
```

This fit is identical to that of the earlier log-scale model (just a linear transformation of the predictors), but to us these coefficient estimates are more directly interpretable:

- Canopy volume and area are both positively associated with weight. Neither is statistically significant, but we keep them in because they both make sense: (1) a larger-volume canopy should have more leaves, and (2) conditional on volume, a canopy with larger cross-sectional area should have more exposure to the sun.
- The negative coefficient of `canopy.shape` implies that bushes that are more circular in cross section have more leaf weight (after controlling for volume and area). It is not clear whether we should "believe" this. The coefficient is not statistically significant; we could keep this predictor in the model or leave it out.
- Total height is positively associated with weight, which could make sense if the bushes are planted close together—taller bushes get more sun. The coefficient is not statistically significant, but it seems to make sense to "believe" it and leave it in.
- It is not clear how to interpret the coefficient for `density`. Since it is not statistically significant, maybe we can exclude it.
- For whatever reason, the coefficient for `group` is large and statistically significant, so we must keep it in. It would be a good idea to learn how the two groups differ so that a more relevant measurement could be included for which `group` is a proxy.

This leaves us with a model such as

R output
```
lm(formula = log(weight) ~ log(canopy.volume) + log(canopy.area) +
    group)
                   coef.est coef.se
(Intercept)            5.22    0.09
log(canopy.volume)     0.61    0.19
log(canopy.area)       0.29    0.24
group                 -0.53    0.12
  n = 46, k = 4
  residual sd = 0.34, R-Squared = 0.87
```

or

R output
```
lm(formula = log(weight) ~ log(canopy.volume) + log(canopy.area) +
    log(canopy.shape) + log(total.height) + group)
                   coef.est coef.se
(Intercept)            5.31    0.16
log(canopy.volume)     0.38    0.28
```

```
log(canopy.area)       0.41    0.29
log(canopy.shape)     -0.32    0.22
log(total.height)      0.42    0.31
group                 -0.54    0.12
  n = 46, k = 6
  residual sd = 0.33, R-Squared = 0.88
```

We want to include both volume and area in the model, since for geometrical reasons we expect both to be positively predictive of leaf volume. It would also make sense to look at some residual plots to look for any patterns in the data beyond what has been fitted by the model.

Finally, it would seem like a good idea to include interactions of `group` with the other predictors. Unfortunately, with only 46 data points, it turns out to be impossible to estimate these interactions accurately: none of them are statistically significant.

To conclude this example: we have had some success in transforming the outcome and input variables to obtain a reasonable predictive model. However, we do not have any clean way of choosing among the models (or combining them). We also do not have any easy way of choosing between the linear and log-transformation models, or bridging the gap between them. For this problem, the log model seems to make much more sense, but we would also like a data-based reason to prefer it, if it is indeed preferable.

4.7 Fitting a series of regressions

It is common to fit a regression model repeatedly, either for different datasets or to subsets of an existing dataset. For example, one could estimate the relation between height and earnings using surveys from several years, or from several countries, or within different regions or states within the United States.

As discussed in Part 2 of this book, multilevel modeling is a way to estimate a regression repeatedly, partially pooling information from the different fits. Here we consider the more informal procedure of estimating the regression separately—with no pooling between years or groups—and then displaying all these estimates together, which can be considered as an informal precursor to multilevel modeling.[4]

Predicting party identification

Political scientists have long been interested in party identification and its changes over time. We illustrate here with a series of cross-sectional regressions modeling party identification given political ideology and demographic variables.

We use the National Election Study, which asks about party identification on a 1–7 scale (1 = strong Democrat, 2 = Democrat, 3 = weak Democrat, 4 = independent, ..., 7 = strong Republican), which we treat as a continuous variable. We include the following predictors: political ideology (1 = strong liberal, 2 = liberal, ..., 7 = strong conservative), ethnicity (0 = white, 1 = black, 0.5 = other), age (as categories: 18–29, 30–44, 45–64, and 65+ years, with the lowest age category as a baseline), education (1 = no high school, 2 = high school graduate, 3 = some college, 4 =

[4] The method of repeated modeling, followed by time-series plots of estimates, is sometimes called the "secret weapon" because it is so easy and powerful but yet is rarely used as a data-analytic tool. We suspect that one reason for its rarity of use is that, once one acknowledges the time-series structure of a dataset, it is natural to want to take the next step and model that directly. In practice, however, there is a broad range of problems for which a cross-sectional analysis is informative, and for which a time-series display is appropriate to give a sense of trends.