Diagnosing Problems in Linear and Generalized Linear Models 6

Regression diagnostics are methods for determining whether a fitted regression model adequately represents the data. We use the term *regression* broadly in this chapter to include methods for both linear and generalized linear models, and many of the methods described here are also appropriate for other regression models. Because most of the methods for diagnosing problems in linear models extend naturally to generalized linear models, we deal at greater length with linear-model diagnostics, briefly introducing the extensions to GLMs.

Linear models fit by least squares make strong and sometimes unrealistic assumptions about the structure of the data. When these assumptions are violated, least-squares estimates can behave badly and may even completely misrepresent the data. Regression diagnostics can reveal such problems and often point the way toward solutions.

Section 6.1 describes various kinds of residuals in linear models, and Section 6.2 introduces basic scatterplots of residuals, along with related plots that are used to assess the fit of a model to data. The remaining sections are specialized to particular problems, describing methods for diagnosis and at least touching on possible remedies. Section 6.3 introduces methods for detecting unusual data, including outliers, high-leverage points, and influential observations. Section 6.4 returns to the topic of transformations of the response and predictors (discussed previously in Section 3.4) to correct problems such as nonnormally distributed errors and nonlinearity. Section 6.5 deals with nonconstant error variance. Section 6.6 describes the extension of diagnostic methods to GLMs such as logistic and Poisson regression. Finally, diagnosing collinearity in regression models is the subject of Section 6.7.

All the methods discussed in this chapter either are available in standard R functions or are implemented in the **car** package. A few functions that were once in earlier versions of the **car** package are now a standard part of R.

One goal of the **car** package is to make diagnostics for linear models and GLMs readily available in R. It is our experience that diagnostic methods are much more likely to be used when they are *convenient*. For example, added-variable plots (described in Section 6.3.3) are constructed by regressing a particular regressor and the response on all the other regressors, computing the residuals from these auxiliary regressions, and plotting one set of residuals against the other. This is not hard to do in R, although the steps are somewhat more complicated when there are factors, interactions, or polynomial or regression spline terms in the model. The `avPlots` function in the **car** package constructs all the added-variable plots for a linear model or GLM and adds enhancements such as a least-squares line and point identification.

6.1 Residuals

Residuals of one sort or another are the basis of most diagnostic methods. Suppose that we specify and fit a linear model assuming constant error variance σ^2. The *ordinary residuals* are given by the differences between the responses and the fitted values:

$$e_i = y_i - \widehat{y}_i,\ i = 1, \ldots, n \tag{6.1}$$

In OLS regression, the residual sum of squares is equal to $\sum e_i^2$. If the regression model includes an intercept, then $\sum e_i = 0$. The ordinary residuals are uncorrelated with the fitted values or indeed any linear combination of the regressors, and so patterns in the plots of ordinary residuals versus linear combinations of the regressors can occur only if one or more assumptions of the model are inappropriate.

If the regression model is correct, then the ordinary residuals are random variables with mean 0 and with variance given by

$$\mathrm{Var}(e_i) = \sigma^2(1 - h_i) \tag{6.2}$$

The quantity h_i is called a *leverage* or *hat-value*. In linear models with fixed predictors, h_i is a nonrandom value constrained to be between 0 and 1, depending on the location of the predictors for a particular observation relative to the other observations.[1] Large values of h_i correspond to observations with relatively unusual $\mathbf{x}_i$ values, whereas a small h_i value corresponds to observations close to the center of the regressor space (see Section 6.3.2).

Ordinary residuals for observations with large h_i have smaller variances. To correct for the nonconstant variance of the residuals, we can divide them by an estimate of their standard deviation. Letting $\widehat{\sigma}^2$ represent the estimate of σ^2, the *standardized residuals* are

$$e_{Si} = \frac{e_i}{\widehat{\sigma}\sqrt{1 - h_i}} \tag{6.3}$$

[1] In a model with an intercept, the minimum hat-value is $1/n$.

While the e_{Si} have constant variance, they are no longer uncorrelated with the fitted values or linear combinations of the regressors, so using standardized residuals in plots is not an obvious improvement.

Studentized residuals are given by

$$e_{Ti} = \frac{e_i}{\widehat{\sigma}_{(-i)}\sqrt{1 - h_i}} \tag{6.4}$$

where $\widehat{\sigma}^2_{(-i)}$ is the estimate of σ^2 computed from the regression without the ith observation. Like the standardized residuals, the Studentized residuals have constant variance. In addition, if the original errors are normally distributed, then e_{Ti} follows a t distribution with $n - k - 2$ df and can be used to test for outliers (see Section 6.3). One can show that

$$\widehat{\sigma}^2_{(-i)} = \frac{\widehat{\sigma}^2(n - k - 1 - e^2_{Si})}{n - k - 2} \tag{6.5}$$

and so computing the Studentized residuals doesn't really require refitting the regression without the ith observation.

If the model is fit by WLS regression with known positive weights w_i, then the ordinary residuals are replaced by the *Pearson residuals*:

$$e_{Pi} = \sqrt{w_i}e_i \tag{6.6}$$

In WLS estimation, the residual sum of squares is $\sum e^2_{Pi}$. If we construe OLS regression to have implicit weights of $w_i = 1$ for all i, then Equation 6.1 is simply a special case of Equation 6.6, and we will generally use the term *Pearson residuals* to cover both of these cases. The standardized and Studentized residuals are unaffected by weights because the weights cancel out in the numerator and denominator of their formulas.

The generic R function `residuals` can compute various kinds of residuals. The default for a linear model is to return the ordinary residuals even if weights are present. Setting the argument `type="pearson"` (with a lowercase `p`) returns the Pearson residuals, which produces correctly weighted residuals if weights are present and ordinary residuals if there are no weights. Pearson residuals are the default when `residuals` is used with a GLM. The functions `rstandard` and `rstudent` return the standardized and Studentized residuals, respectively. The function `hatvalues` returns the hat-values.

6.2 Basic Diagnostic Plots

The **car** package includes a number of functions that produce plots of residuals and related quantities. The variety of plots reflects the fact that no one diagnostic graph is appropriate for all purposes.

6.2.1 PLOTTING RESIDUALS

Plots of residuals versus fitted values and versus each of the predictors in turn are the most basic diagnostic graphs. If a linear model is correctly specified, then the Pearson residuals are independent of the fitted values and the predictors, and these graphs should be *null plots*, with no systematic features—in the sense that the conditional distribution of the residuals (on the vertical axis of the graph) should not change with the fitted values or with a predictor (on the horizontal axis). The presence of systematic features generally implies a failure of one or more assumptions of the model. Of interest in these plots are nonlinear trends, trends in variation across the graph, and isolated points.

Plotting residuals against fitted values and predictors is useful for revealing problems but less useful for determining the exact nature of the problem. Consequently, we will employ other diagnostic graphs to suggest improvements to a model.

Consider, for example, a modification of the model used in Section 4.2.2 for the Canadian occupational-prestige data:

```
> prestige.mod.2 <- lm(prestige ~ education + income + type,
+     data=Prestige)
```

In Section 3.4.7, we had suggested replacing `income` by its logarithm, and we followed that advice in Section 4.2.2. Here, we naively use `income` without transformation, in part to demonstrate what happens when a predictor needs transformation.[2]

The standard residual plots for this model are given by the `residualPlots` function in the **car** package:

```
> residualPlots(prestige.mod.2)

           Test stat Pr(>|t|)
education     -0.684    0.496
income        -2.886    0.005
type              NA       NA
Tukey test    -2.610    0.009
```

This command produces scatterplots of the Pearson residuals versus each of the predictors and versus the fitted values (Figure 6.1).

The most common diagnostic graph in linear regression is the plot of residuals versus the fitted values, shown at the bottom right of Figure 6.1. The plot has a curved general trend, suggesting that the model we fit is not adequate to describe the data. The plot of residuals versus `education` at the top left, however, resembles a null plot, in which no particular pattern is apparent. A null plot is consistent with an adequate model, but as is the case here, one null plot is insufficient to provide evidence of an adequate model, and indeed one nonnull plot is enough to suggest that the specified model does not match the data. The plot of residuals versus `income` at the top right is also curved, as

[2] Our experience in statistical consulting suggests that this kind of naivete is common.

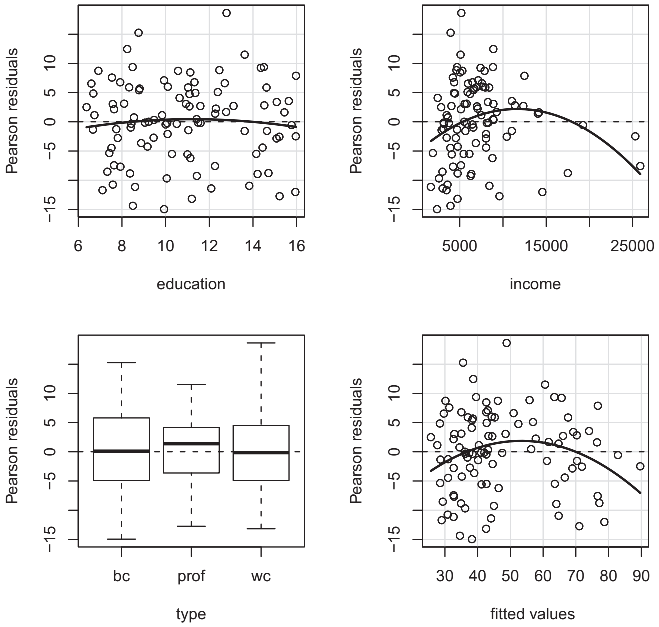

Figure 6.1 Basic residual plots for the regression of `prestige` on `education`, `income`, and `type` in the `Prestige` data set.

might have been anticipated in light of the results in Section 3.4.7. The residual plot for a factor such as `type`, at the bottom left, is a set of boxplots of the residuals at the various levels of the factor. In a null plot, the boxes should all have about the same center and spread, as is more or less the case here.

To help examine these residual plots, a *lack-of-fit test* is computed for each numeric predictor, and a curve is added to the graph. The lack-of-fit test for `education`, for example, is the *t* test for the regressor (`education`)^2 added to the model, for which the corresponding *p* value rounds to .50, indicating no lack-of-fit of this type. For `income`, the lack-of-fit test has the *p* value .005, clearly confirming the nonlinear pattern visible in the graph. The lines shown on the plot are the fitted quadratic regressions of the Pearson residuals on the numeric predictors.

For the plot of residuals versus fitted values, the test—called *Tukey's test for nonadditivity* (Tukey, 1949)—is obtained by adding the squares of the fitted values to the model and refitting. The significance level for Tukey's test is obtained by comparing the statistic with the standard-normal distribution. The test confirms the visible impression of curvature in the residual plot, further reinforcing the conclusion that the fitted model is not adequate.

The `residualPlots` function shares many arguments with other graphics functions in the **car** package; see `?residualPlots` for details. In `residualPlots`, all arguments other than the first are optional. The argument `id.n` could be set to a positive number to identify automatically the `id.n` most unusual cases, which by default are the cases with the largest (absolute) residuals (see Section 3.5). There are additional arguments to control the layout of the plots and the type of residual plotted. For example, setting `type="rstudent"` would plot Studentized residuals rather than Pearson residuals. Setting `smooth=TRUE, quadratic=FALSE` would display a lowess smooth rather than a quadratic curve on each plot, although the test statistics always correspond to the fitting quadratics.

If you want only the plot of residuals against fitted values, you can use

```
> residualPlots(prestige.mod.2, ~ 1, fitted=TRUE)
```

whereas the plot against `education` only can be obtained with

```
> residualPlots(prestige.mod.2, ~ education, fitted=FALSE)
```

The second argument to `residualPlots`—and to other functions in the **car** package that can produce graphs with several panels—is a *one-sided formula* that specifies the predictors against which to plot residuals. The formula `~ .` is the default, to plot against *all* the available predictors; `~ 1` plots against *none* of the predictors, and in the current context produces a plot against fitted values only; `~ . - income` plots against all predictors but `income`. Because the fitted values are not part of the formula that defined the model, there is a separate `fitted` argument, which is set to `TRUE` (the default) to include a plot of residuals against fitted values and `FALSE` to exclude it.

We could of course draw residual plots using `plot` or `scatterplot`, but we would have to be careful if there are missing data. In the current example, the value of `type` is missing for a few of the cases, and so the regression was computed using only the complete cases. Consequently, the vector of residuals is shorter than the vector of values for, say, `income`. We can circumvent this problem by setting `option(na.action=na.exclude)` (as explained in Section 4.8.5). Then the residual vector will include a value for each observation in the original data set, equal to `NA` for observations with missing values on one or more variables in the model.

6.2.2 MARGINAL MODEL PLOTS

A variation on the basic residual plot is the *marginal model plot*, proposed by Cook and Weisberg (1997):

```
> marginalModelPlots(prestige.mod.2)
```

These plots (shown in Figure 6.2) all have the response variable, in this case `prestige`, on the vertical axis, while the horizontal axis is given in turn by each of the numeric predictors in the model and the fitted values. The plots of the response versus individual predictors display the conditional distribution of the response given each predictor, ignoring the other predictors; these are

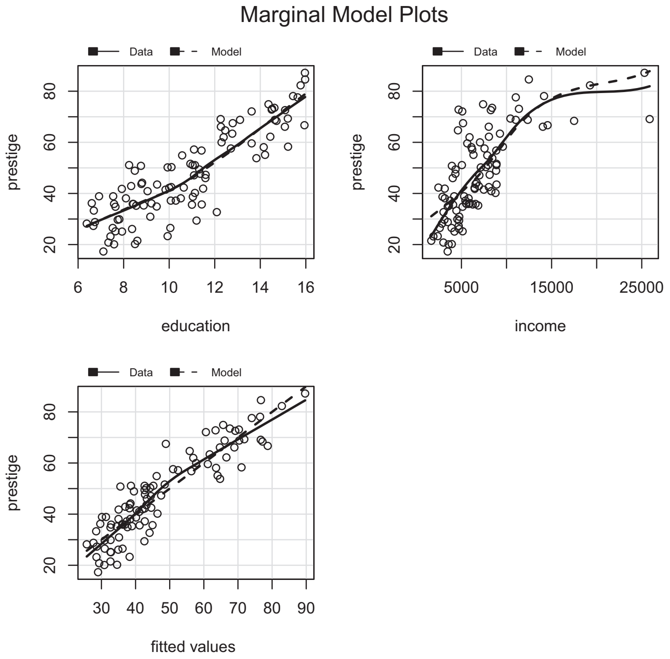

Figure 6.2 Marginal-model plots for the regression of `prestige` on `education`, `income`, and `type` in the `Prestige` data set.

marginal plots in the sense that they show the marginal relationship between the response and each predictor. The plot versus fitted values is a little different in that it displays the conditional distribution of the response given the fit of the model.

We can estimate a regression function for each of the marginal plots by fitting a smoother to the points in the plot. The `marginalModelPlots` function uses a `lowess` smooth, as shown by the solid line on the plot.

Now imagine a second graph that replaces the vertical axis with the fitted values from the model. If the model is appropriate for the data, then, under fairly mild conditions, the smooth fit to this second plot should also estimate the conditional expectation of the response given the predictor on the horizontal axis. The second smooth is also drawn on the marginal model plot, as a dashed line. If the model fits the data well, then the two smooths should match on each of the marginal model plots; if any pair of smooths fails to match, then we have evidence that the model does not fit the data well.

An interesting feature of the marginal model plots in Figure 6.2 is that even though the model that we fit to the `Prestige` data specifies linear *partial* relationships between `prestige` and each of `education` and `income`, it is able to reproduce nonlinear *marginal* relationships for these two predictors. Indeed, the model, as represented by the dashed lines, does a fairly good job of matching the marginal relationships represented by the solid lines, although the systematic failures discovered in the residual plots are discernable here as well.

Marginal model plots can be used with any fitting or modeling method that produces fitted values, and so they can be applied to some problems where the definition of residuals is unclear. In particular, marginal model plots generalize nicely to GLMs.

The `marginalModelPlots` function has an `SD` argument, which if set to `TRUE` adds estimated standard deviation lines to the graph. The plots can therefore be used to check both the regression function, as illustrated here, and the assumptions about variance. Other arguments to the `marginalModelPlots` function are similar to those for `residualPlots`.

6.2.3 ADDED-VARIABLE PLOTS

The marginal model plots of the previous section display the *marginal* relationships—both directly observed and implied by the model—between the response and each regressor *ignoring* the other regressors in the model. In contrast, *added-variable plots*, also called *partial-regression plots*, display the *partial* relationship between the response and a regressor, *adjusted for* all the other regressors.

Suppose that we have a regression problem with response y and regressors $x_1, \ldots, x_k$.[3] To draw the added-variable plot for one of the regressors—say the first, x_1—we must conduct the following two auxiliary regressions:

1. Regress y on all the regressors excluding x_1. The residuals from this regression are *the part of y that is not explained by all the regressors except* x_1.

2. Regress x_1 on the other regressors and again obtain the residuals. These residuals represent the *part of* x_1 *that is not explained by the other regressors*; put another way, the residuals are the part of x_1 that remains when we condition on the other regressors.

The added-variable plot for x_1 is simply a scatterplot with the residuals from Step 1 on the vertical axis and the residuals from Step 2 on the horizontal axis.

The `avPlots` function in the **car** package works both for linear models and GLMs. It has arguments for controlling which plots are drawn, point labeling, and plot layout, and these arguments are the same as for the `residualPlots` function (described in Section 6.2.1).

Added-variable plots for the Canadian occupational-prestige regression (in Figure 6.3) are produced by the following command:

```
> avPlots(prestige.mod.2, id.n=2, id.cex=0.6)
```

[3] Although it is not usually of interest, when there is an intercept in the model, it is also possible to construct an added-variable plot for the constant regressor, x_0, which is equal to 1 for every observation.

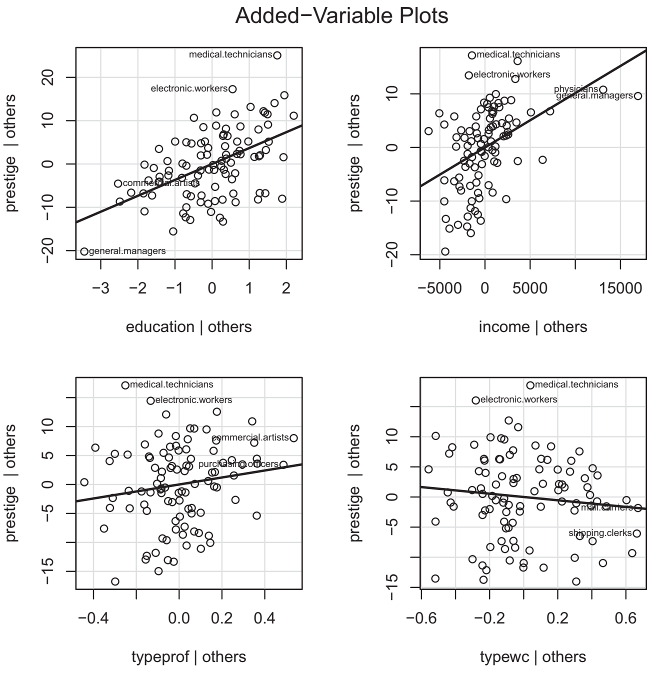

Figure 6.3 Added-variable plots for the regression of `prestige` on `education`, `income`, and `type` in the `Prestige` data set.

The argument `id.n=2` will result in identifying up to four points in each graph, the two that are farthest from the mean on the horizontal axis and the two with the largest absolute residuals from the fitted line. Because the case labels in the `Prestige` data set are very long, we used `id.cex=0.6` to reduce the printed labels to 60% of their default size.

The added-variable plot has several interesting and useful properties:

- The least-squares line on the added-variable plot for the regressor x_j has slope b_j, equal to the partial slope for x_j in the full regression. Thus, for example, the slope in the added-variable plot for `education` is $b_1 = 3.67$, and the slope in the added-variable plot for `income` is $b_2 = 0.00101$. (The income slope is small because the unit of income—\$1 of annual income—is small.)
- The residuals from the least-squares line in the added-variable plot are the same as the residuals e_i from the regression of the response on *all* of the regressors.
- Because the positions on the horizontal axis of the added-variable plot show values of x_j conditional on the other regressors, points far to the left or right represent observations for which the value of x_j is unusual given the values of the other regressors. Likewise, the variation of the variable on the horizontal axis is the conditional variation of x_j, and the

added-variable plot therefore allows us to visualize the precision of the estimation of b_j.

- For factors, an added-variable plot is produced for each of the contrasts that are used to define the factor, and thus, if we change the way the contrasts are coded for a factor, the corresponding added-variable plots will change as well.

The added-variable plot allows us to visualize the effect of each regressor after adjusting for all the other regressors in the model. In Figure 6.3, the plot for `income` has a positive slope, but the slope appears to be influenced by two high-income occupations (physicians and general managers), which pull down the regression line at the right. There don't seem to be any particularly noteworthy points in the added-variable plots for the other regressors.

Although added-variable plots are useful for studying the impact of observations on regression coefficients (see Section 6.3.3), they can prove misleading when diagnosing other sorts of problems, such as nonlinearity. A further disadvantage of the added-variable plot is that the variables on both axes are sets of residuals, and so neither the response nor the regressors are displayed directly.

Sall (1990) and Cook and Weisberg (1991) generalize added-variable plots to terms with more than 1 *df*, such as a factor or polynomial regressors. Following Sall, we call these graphs *leverage plots*. For terms with 1 *df*, the leverage plots are very similar to added-variable plots, except that the slope in the plot is always equal to 1, not to the corresponding regression coefficient. Although leverage plots can be misleading in certain circumstances,[4] they can be useful for locating groups of cases that are jointly high-leverage or influential. Leverage, influence, and related ideas are explored in the next section. There is a `leveragePlots` function in the **car** package, which works only for linear models.

6.3 Unusual Data

Unusual data can wreak havoc with least-squares estimates but may prove interesting in their own right. Unusual data in regression include outliers, high-leverage points, and influential observations.

6.3.1 OUTLIERS AND STUDENTIZED RESIDUALS

Regression outliers are y values that are unusual *conditional on* the values of the predictors. An illuminating way to search for outliers is via the *mean-shift outlier model*:

$$y_i = \alpha + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \gamma d_i + \varepsilon_i$$

[4] For example, if a particular observation causes one dummy-regressor coefficient to get larger and another smaller, these changes can cancel each other out in the leverage plot for the corresponding factor, even though a different pattern of results for the factor would be produced by removing the observation.

where d_i is a dummy regressor coded 1 for observation i and 0 for all other observations. If $\gamma \neq 0$, then the conditional expectation of the ith observation has the same dependence on $x_1, \ldots, x_k$ as the other observations, but its intercept is shifted from α to $\alpha + \gamma$. The t statistic for testing the null hypothesis H_0: $\gamma = 0$ against a two-sided alternative has $n - k - 2$ *df* if the errors are normally distributed and is the appropriate test for a single mean-shift outlier at observation i. Remarkably, this t statistic turns out to be identical to the ith Studentized residual, e_{Ti} (Equation 6.4, p. 287), and so we can get the test statistics for the n different null hypotheses, H_{0i}: case i is not a mean-shift outlier, $i = 1, \ldots, n$, at minimal computational cost.

Our attention is generally drawn to the largest absolute Studentized residual, and this presents a problem: Even if the Studentized residuals were independent, which they are not, there would be an issue of simultaneous inference entailed by picking the largest of n test statistics. The dependence of the Studentized residuals complicates the issue. We can deal with this problem (a) by a *Bonferroni adjustment* of the p value for the largest absolute Studentized residual, multiplying the usual two-tail p by the sample size, n, or (b) by constructing a quantile-comparison plot of the Studentized residuals with a confidence envelope that takes their dependence into account.

We reconsider Duncan's occupational-prestige data (introduced in Section 1.2), regressing `prestige` on occupational `income` and `education` levels:

```
> mod.duncan <- lm(prestige ~ income + education, data=Duncan)
```

The generic `qqPlot` function in the **car** package has a method for linear models, plotting Studentized residuals against the corresponding quantiles of $t(n-k-2)$. By default, `qqPlot` generates a 95% pointwise confidence envelope for the Studentized residuals, using a parametric version of the bootstrap, as suggested by Atkinson (1985):[5]

```
> qqPlot(mod.duncan, id.n=3)

[1] "minister"   "reporter"   "contractor"
```

The resulting plot is shown in Figure 6.4. Setting the argument `id.n=3`, the `qqPlot` function returns the names of the three observations with the largest absolute Studentized residuals (see Section 3.5 on point identification); in this case, only one observation, `minister`, strays slightly outside of the confidence envelope. If you repeat this command, your plot may look a little different from ours because the envelope is computed by simulation. The distribution of the Studentized residuals looks heavy-tailed compared to the reference t distribution: Perhaps a method of robust regression would be more appropriate for these data.[6]

[5] Bootstrap methods in R are described in the online appendix to the book.

[6] R functions for robust and resistant regression are described in Section 4.3.7 and in the online appendix to the text.

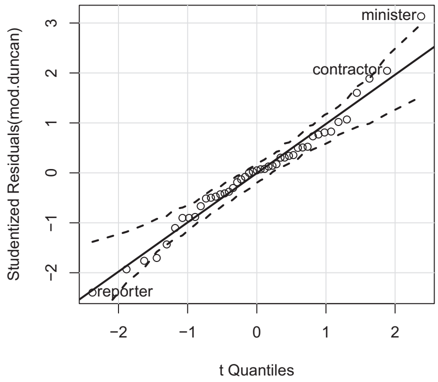

Figure 6.4 Quantile-comparison plot of Studentized residuals from Duncan's occupational-prestige regression, showing the pointwise 95% simulated confidence envelope.

The `outlierTest` function in the **car** package locates the largest Studentized residual in absolute value and computes the Bonferroni-corrected *t* test:

```
> outlierTest(mod.duncan)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
          rstudent unadjusted p-value Bonferonni p
minister    3.135           0.003177       0.1430
```

The Bonferroni-adjusted *p* value is not statistically significant, and so it isn't surprising that the largest Studentized residual in a sample of size $n = 45$ is as large as 3.135.

6.3.2 LEVERAGE: HAT-VALUES

Observations that are relatively far from the center of the regressor space, taking account of the correlational pattern among the regressors, have potentially greater influence on the least-squares regression coefficients; such points are said to have *high leverage*. The most common measures of leverage are the h_i, or *hat-values*.[7] The h_i are bounded between 0 and 1 (in models with an intercept, they are bounded between $1/n$ and 1), and their sum, $\sum h_i$, is always equal to the number of coefficients in the model, including the intercept. Problems in which there are a few very large h_i can be troublesome: In particular, large-sample normality of some linear combinations of the regressors is likely to fail, and high-leverage observations may exert undue influence on the results (see below).

[7] * The name *hat-values* comes from the relationship between the observed vector of responses and the fitted values. The vector of fitted values is given by $\widehat{\mathbf{y}} = \mathbf{Xb} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{Hy}$, where $\mathbf{H} = \{h_{ij}\} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, called the *hat-matrix*, projects $\mathbf{y}$ into the subspace spanned by the columns of the model matrix $\mathbf{X}$. Because $\mathbf{H} = \mathbf{H}'\mathbf{H}$, the hat-values h_i are simply the diagonal entries of the hat-matrix.

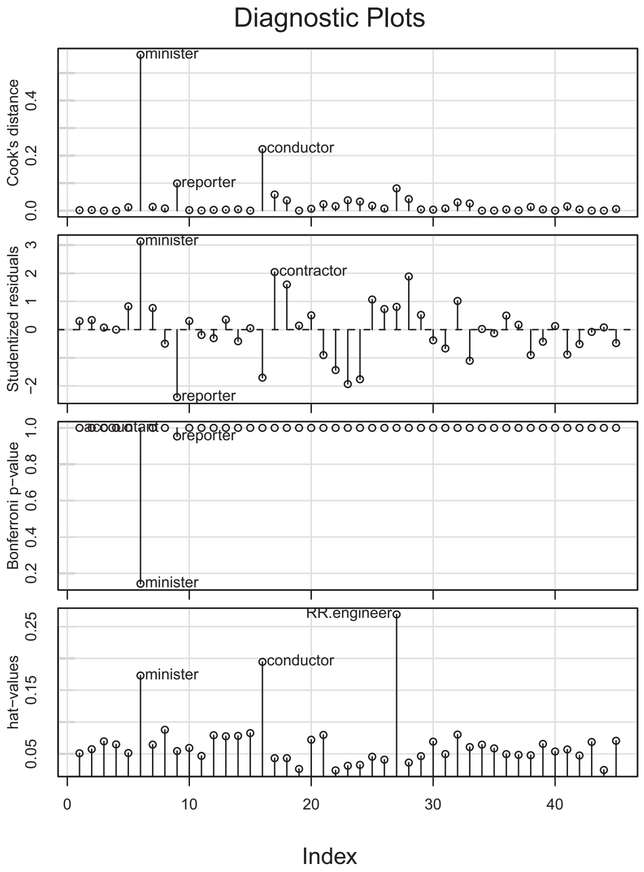

Figure 6.5 Index plots of diagnostic statistics for Duncan's occupational-prestige regression.

The `hatvalues` function works for both linear models and GLMs. One way of examining the hat-values and other individual-observation diagnostic statistics is to construct *index plots*, graphing the statistics against the corresponding observation indices.

For example, the following command uses the **car** function `influenceIndexPlot` to produce Figure 6.5, which includes index plots of Studentized residuals, the corresponding Bonferroni *p* values for outlier testing, the hat-values, and Cook's distances (discussed in the next section) for Duncan's occupational-prestige regression:

```
> influenceIndexPlot(mod.duncan, id.n=3)
```

The occupations railroad engineer (`RR.engineer`), `conductor`, and `minister` stand out from the rest in the plot of hat-values, indicating that their regressor values are unusual relative to the other occupations. In the plot of *p* values for the outlier tests, cases for which the Bonferroni bound is bigger than 1 are set equal to 1, and here only one case (`minister`) has a Bonferroni p value much less than 1.

6.3.3 INFLUENCE MEASURES

An observation that is both outlying and has high leverage exerts *influence* on the regression coefficients, in the sense that if the observation is removed, the coefficients change considerably. As usual, let **b** be the estimated value of the coefficient vector $\boldsymbol{\beta}$, and as new notation, define $\mathbf{b}_{(-i)}$ to be the estimate of $\boldsymbol{\beta}$ but now computed without the ith case.[8] Then the difference $\mathbf{b}_{(-i)} - \mathbf{b}$ directly measures the influence of the ith observation on the estimate of $\boldsymbol{\beta}$. If this difference is small, then the influence of observation i is small, whereas if the difference is large, then its influence is large.

COOK'S DISTANCE

It is convenient to summarize the size of the difference $\mathbf{b}_{(-i)} - \mathbf{b}$ by a single number, and this can be done in several ways. The most common summary measure of influence is *Cook's distance* (Cook, 1977), D_i, which is just a weighted sum of squares of the differences between the individual elements of the coefficient vectors.[9] Interestingly, Cook's distance can be computed from diagnostic statistics that we have already encountered:

$$D_i = \frac{e_{Si}^2}{k+1} \times \frac{h_i}{1-h_i}$$

where e_{Si}^2 is the squared standardized residual (Equation 6.3, p. 286) and h_i is the hat-value for observation i. The first factor may be thought of as a measure of outlyingness and the second as a measure of leverage. Observations for which D_i is large are potentially influential cases. If any noteworthy D_i are apparent, then it is prudent to remove the corresponding cases temporarily from the data, refit the regression, and see how the results change. Because an influential observation can affect the fit of the model at other observations, it is best to remove observations one at a time, refitting the model at each step and reexamining the resulting Cook's distances.

The generic function `cooks.distance` has methods for linear models and GLMs. Cook's distances are also plotted, along with Studentized residuals and hat-values, by the `influenceIndexPlot` function, as illustrated for Duncan's regression in Figure 6.5. The occupation `minister` is the most influential according to Cook's distance, and we therefore see what happens when we delete this case and refit the model:

[8] If vector notation is unfamiliar, simply think of **b** as the collection of estimated regression coefficients, $b_0, b_1, \ldots, b_k$.

[9] * In matrix notation,

$$D_i = \frac{\left(\mathbf{b}_{(-i)} - \mathbf{b}\right)' \mathbf{X}'\mathbf{X} \left(\mathbf{b}_{(-i)} - \mathbf{b}\right)}{(k+1)\widehat{\sigma}^2}.$$

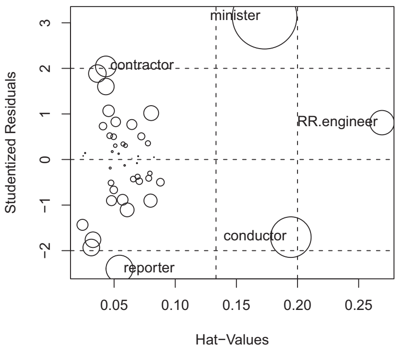

Figure 6.6 Plot of hat-values, Studentized residuals, and Cook's distances for Duncan's occupational-prestige regression. The size of the circles is proportional to Cook's D_i.

```
> mod.duncan.2 <- update(mod.duncan,
+     subset= rownames(Duncan) != "minister")
> compareCoefs(mod.duncan, mod.duncan.2)

Call:
1:lm(formula = prestige ~ income + education, data = Duncan)
2:lm(formula = prestige ~ income + education, data = Duncan,
     subset = rownames(Duncan) != "minister")
             Est. 1    SE 1  Est. 2    SE 2
(Intercept) -6.0647  4.2719 -6.6275  3.8875
income       0.5987  0.1197  0.7316  0.1167
education    0.5458  0.0983  0.4330  0.0963
```

The `compareCoefs` function displays the estimates from one or more fitted models in a compact table. Removing `minister` increases the coefficient for `income` by about 20% and decreases the coefficient for `education` by about the same amount. Standard errors are much less affected. In other problems, removing an observation can change significant results to insignificant ones, and vice-versa.

The `influencePlot` function in the **car** package provides an alternative to index plots of diagnostic statistics:

```
> influencePlot(mod.duncan, id.n=3)

             StudRes     Hat  CookD
minister       3.135 0.17306 0.7526
reporter      -2.397 0.05439 0.3146
conductor     -1.704 0.19454 0.4729
contractor     2.044 0.04326 0.2419
RR.engineer    0.809 0.26909 0.2845
```

This command produces a *bubble-plot*, shown in Figure 6.6, which combines the display of Studentized residuals, hat-values, and Cook's distances, with the

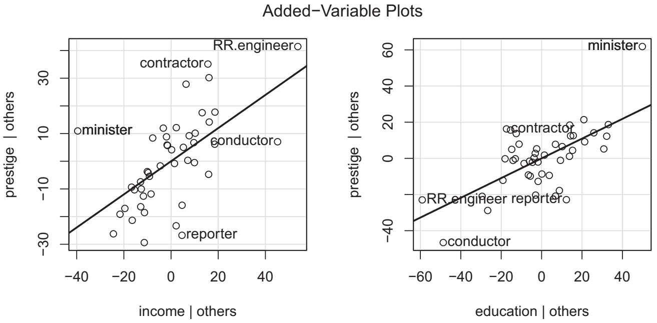

Figure 6.7 Added-variable plots for Duncan's occupational-prestige regression.

areas of the circles proportional to Cook's D_i.[10] As usual the id.n argument is used to label points. In this case, the id.n points with the largest hat-values, Cook's distances, or absolute Studentized residuals will be flagged, so more than id.n points in all may be labeled.

We invite the reader to continue the analysis by examining the influence diagnostics for Duncan's regression after the observation minister has been removed.

ADDED-VARIABLE PLOTS AS INFLUENCE DIAGNOSTICS

Added-variable plots (Section 6.2.3) are a useful diagnostic for finding potentially jointly influential points, which will correspond to sets of points that are out of line with the rest of the data and are at the extreme left or right of the horizontal axis. Figure 6.7, for example, shows the added-variable plots for income and education in Duncan's regression:

```
> avPlots(mod.duncan, id.n=3)
```

The observations minister, conductor, and RR.engineer (railroad engineer) have high leverage on both coefficients. The cases minister and conductor also work together to decrease the income slope and increase the education slope; RR.engineer, on the other hand, is more in line with the rest of the data. Removing *both* minister and conductor changes the regression coefficients dramatically—much more so than deleting minister alone:

```
> mod.duncan.3 <- update(mod.duncan,
+     subset = !(rownames(Duncan) %in% c("minister", "conductor")))
> compareCoefs(mod.duncan, mod.duncan.2, mod.duncan.3, se=FALSE)
```

[10] In Chapter 8, we describe how to write a similar function as a preliminary example of programming in R.

```
Call:
1:lm(formula = prestige ~ income + education, data = Duncan)
2:lm(formula = prestige ~ income + education, data = Duncan,
     subset = rownames(Duncan) != "minister")
3:lm(formula = prestige ~ income + education, data = Duncan,
     subset = !(rownames(Duncan) %in% c("minister", "conductor")))
            Est. 1 Est. 2 Est. 3
(Intercept) -6.065 -6.628 -6.409
income       0.599  0.732  0.867
education    0.546  0.433  0.332
```

INFLUENCE SEPARATELY FOR EACH COEFFICIENT

Rather than summarizing influence by looking at all coefficients simultaneously, we could create $k + 1$ measures of influence by looking at individual differences:

$$\text{dfbeta}_{ij} = b_{(-i)j} - b_j \text{ for } j = 0, \ldots, k$$

where b_j is the coefficient computed using all the data and $b_{(-i)j}$ is the same coefficient computed with case i omitted. As with D_i, computation of dfbeta$_{ij}$ can be accomplished efficiently without having to refit the model. The dfbeta$_{ij}$ are expressed in the metric (units of measurement) of the coefficient b_j. A standardized version, dfbetas$_{ij}$, divides dfbeta$_{ij}$ by an estimate of the standard error of b_j computed with observation i removed.

The `dfbeta` function in R takes a linear-model or GLM object as its argument and returns all values of dfbeta$_{ij}$; similarly `dfbetas` computes the dfbetas$_{ij}$, as in the following example for Duncan's regression:

```
> dfbs.duncan <- dfbetas(mod.duncan)
> head(dfbs.duncan)  # first few rows

           (Intercept)     income  education
accountant  -0.0225344  6.662e-04  0.0359439
pilot       -0.0254350  5.088e-02 -0.0081183
architect   -0.0091867  6.484e-03  0.0056193
author      -0.0000472 -6.018e-05  0.0001398
chemist     -0.0658168  1.700e-02  0.0867771
minister     0.1449367 -1.221e+00  1.2630190
```

We could examine each column of the dfbetas matrix separately (e.g., via an index plot), but because we are not really interested here in influence on the regression intercept and because there are just two slope coefficients, we instead plot influence on the `income` coefficient against influence on the `education` coefficient (Figure 6.8):

```
> plot(dfbs.duncan[ , c("income", "education")])  # for b1 and b2
> identify(dfbs.duncan[ , "income"], dfbs.duncan[ , "education"],
+     rownames(Duncan))

[1] "minister"    "conductor"   "RR.engineer"
```

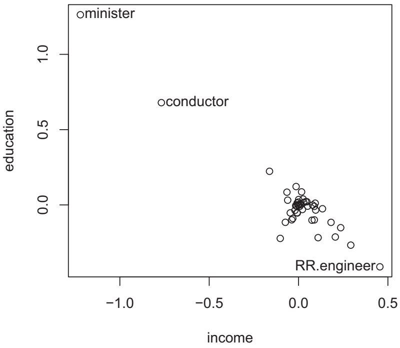

Figure 6.8 dfbetas$_{ij}$ values for the `income` and `education` coefficients in Duncan's occupational-prestige regression. Three points were identified interactively.

The negative relationship between the dfbetas$_{ij}$ values for the two regressors reflects the *positive* correlation of the regressors themselves. Two pairs of values stand out: Consistent with our earlier remarks, observations `minister` and `conductor` make the `income` coefficient smaller and the `education` coefficient larger. We also identified the occupation `RR.engineer` in the plot.

6.4 Transformations After Fitting a Regression Model

Suspected outliers and possibly cases with high leverage should be studied individually to decide whether or not they should be included in an analysis. Influential cases can cause changes in the conclusions of an analysis and also require special treatment. Other systematic features in residual plots—for example, curvature or apparent nonconstant variance—require action on the part of the analyst to modify the structure of the model to match the data more closely. Apparently distinct problems can also interact: For example, if the errors have a skewed distribution, then apparent outliers may be produced in the direction of the skew. Transforming the response to make the errors less skewed can solve this problem. Similarly, properly modeling a nonlinear relationship may bring apparently outlying observations in line with the rest of the data.

Transformations were introduced in Section 3.4 in the context of examining data and with the understanding that regression modeling is often easier and more effective when the predictors behave as if they were normal random variables. Transformations can also be used *after* fitting a model, to improve a model that does not adequately represent the data. The methodology in these two contexts is very similar.

6.4.1 TRANSFORMING THE RESPONSE

NONNORMAL ERRORS

Departures from the assumption of normally distributed errors are probably the most difficult problem to diagnose. The only data available for studying the error distribution are the residuals. Even for an otherwise correctly specified model, the residuals can have substantially different variances, can be strongly correlated, and tend to behave more like a normal sample than do the original errors, a property that has been called *supernormality* (Gnanadesikan, 1977).

A quantile-comparison plot of Studentized residuals against the t distribution (as described in Section 6.3.1) is useful in drawing our attention to the tail behavior of the residuals, possibly revealing heavy-tailed or skewed distributions. A nonparametric density estimate, however, does a better job of conveying a general sense of the shape of the residual distribution.

In Section 5.5, we fit a Poisson regression to Ornstein's data on interlocking directorates among Canadian corporations, regressing the number of interlocks maintained by each firm on the firm's assets, nation of control, and sector of operation. Because number of interlocks is a count, the Poisson model is a natural starting point, but the original source used a least-squares regression similar to the following:

```
> mod.ornstein <- lm(interlocks + 1 ~ log(assets) + nation + sector,
+     data=Ornstein)
```

We put `interlocks + 1` on the left-hand side of the model formula because there are some 0 values in `interlocks` and we will shortly consider power transformations of the response variable.

Quantile-comparison and density plots of the Studentized residuals for Ornstein's regression are produced by the following R commands (Figure 6.9):

```
> par(mfrow=c(1,2))
> qqPlot(mod.ornstein, id.n=0)
> plot(density(rstudent(mod.ornstein)))
```

Both tails of the distribution of Studentized residuals are heavier than they should be, but the upper tail is even heavier than the lower one, and consequently, the distribution is positively skewed. A positive skew in the distribution of the residuals can often be corrected by transforming y down the ladder of powers. The next section describes a systematic method for selecting a normalizing transformation of y.

BOX-COX TRANSFORMATIONS

The goal of fitting a model that exhibits linearity, constant variance, and normality can in principle require three different response transformations, but experience suggests that one transformation is often effective for all of these tasks. The most common method for selecting a transformation of the response in regression was introduced by Box and Cox (1964). If the response

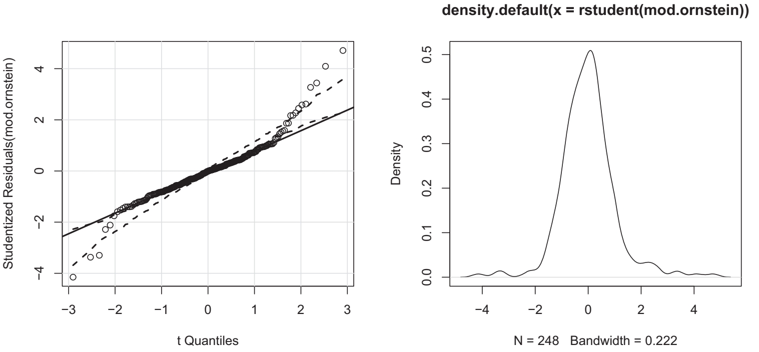

Figure 6.9 Quantile-comparison plot and nonparametric density estimate for the distribution of the Studentized residuals from Ornstein's interlocking-directorate regression.

y is a strictly positive variable, then the Box-Cox power transformations (introduced in Section 3.4.2), implemented in the `bcPower` function in the **car** package, are often effective:

$$T_{\mathrm{BC}}(y,\lambda) = y^{(\lambda)} = \begin{cases} \dfrac{y^{\lambda}-1}{\lambda} & \text{when } \lambda \neq 0 \\ \log_e y & \text{when } \lambda = 0 \end{cases} \tag{6.7}$$

If y is not strictly positive, then the Yeo-Johnson family, computed by the `yjPower` function, can be used in place of the Box-Cox family; alternatively, we can add a start to y to make all the values positive (as explained in Section 3.4.2).

Box and Cox proposed selecting the value of λ by analogy to the method of maximum likelihood, so that the residuals from the linear regression of $T_{\mathrm{BC}}(y,\lambda)$ on the predictors are as close to normally distributed as possible.[11] The **car** package provides two functions for estimating λ. The first, `boxCox`, is a slight generalization of the `boxcox` function in the **MASS** package (Venables and Ripley, 2002).[12] The second is the `powerTransform` function introduced in a related context in Section 3.4.7.

For Ornstein's least-squares regression, for example,

```
> boxCox(mod.ornstein, lambda = seq(0, 0.6, by=0.1))
```

This command produces the graph of the *profile log-likelihood* function for λ in Figure 6.10. The best estimate of λ is the value that maximizes the profile

[11] * If $T_{\mathrm{BC}}(y,\lambda_0)\,|\mathbf{x}$ is normally distributed, then $T_{\mathrm{BC}}(y,\lambda_1)\,|\mathbf{x}$ cannot be normally distributed for $\lambda_1 \neq \lambda_0$, and so the distribution changes for every value of λ. The method Box and Cox proposed ignores this fact to get a maximum-likelihood-like estimate that turns out to have properties similar to those of maximum-likelihood estimates.

[12] `boxCox` adds the argument `family`. If set to the default `family="bcPower"`, then the function is identical to the original `boxcox`. If set to `family="yjPower"`, then the Yeo-Johnson power transformations are used.

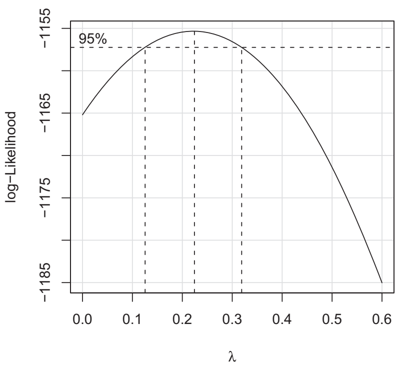

Figure 6.10 Profile log-likelihood for the transformation parameter λ in the Box-Cox model applied to Ornstein's interlocking-directorate regression.

likelihood, which in this example is $\lambda \approx 0.2$. An approximate 95% confidence interval for λ is the set of all λs for which the value of the profile log-likelihood is within 1.92 of the maximum—from about 0.1 to 0.3.[13] It is usual to round the estimate of λ to a familiar value, such as $-1, -1/2, 0, 1/3, 1/2, 1$, or 2. In this case, we would round to the cube-root transformation, $\lambda = 1/3$. Because the response variable `interlocks` is a count, however, we might prefer the log transformation ($\lambda = 0$) or the square-root transformation ($\lambda = 1/2$).

In the call to `boxCox`, we used only the linear-model object `mod.ornstein` and the optional argument `lambda`, setting the range of powers to be searched to λ in $[0, 0.6]$. We did this to provide more detail in the plot, and the default of `lambda = seq(-2, 2, by=0.1)` is usually recommended for an initial profile log-likelihood plot. If the maximum-likelihood estimate of λ turns out to lie outside this range, then the range can always be extended, although transformations outside $[-2, 2]$ are rarely helpful.

The function `powerTransform` in the **car** package performs calculations that are similar to those of the `boxCox` function when applied to an `lm` object, but it produces numeric rather than graphical output:

```
> summary(p1 <- powerTransform(mod.ornstein))

bcPower Transformation to Normality

   Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
Y1    0.2227   0.0493            0.126           0.3193

Likelihood ratio tests about transformation parameters
                         LRT df      pval
LR test, lambda = (0)  19.76  1 8.794e-06
LR test, lambda = (1) 243.40  1 0.000e+00
```

The maximum-likelihood estimate of the transformation parameter is $\widehat{\lambda} = 0.22$, with the 95% confidence interval for λ running from 0.13 to 0.32—quite

[13] The value 1.92 is $\frac{1}{2}\chi^2_{.95}(1) = \frac{1}{2}1.96^2$.

a sharp estimate that even excludes the cube-root transformation, $\lambda = 1/3$. The significance levels for the tests that $\lambda = 0$ and for $\lambda = 1$ are very small, suggesting that neither of these transformations is appropriate for the data.

The result returned by `powerTransform`, which we stored in `p1`, can be used to add the transformed values to the data frame:

```
> Ornstein1 <- transform(Ornstein,
+    y1=bcPower(interlocks + 1, coef(p1)),
+    y1round=bcPower(interlocks + 1, coef(p1, round=TRUE)))
> mod.ornstein.trans <- update(mod.ornstein, y1round ~ .,
+    data=Ornstein1)
```

This command saves the transformed values with λ rounded to the convenient value in the confidence interval that is closest to the point estimate. If none of the convenient values are in the interval, then no rounding is done.

CONSTRUCTED-VARIABLE PLOT FOR THE BOX-COX TRANSFORMATION

Atkinson (1985) suggests an approximate score test and diagnostic plot for the Box-Cox transformation of y, based on the *constructed variable*

$$g_i = y_i \left[\log_e \left(\frac{y_i}{\widetilde{y}} \right) - 1 \right]$$

where $\widetilde{y}$ is the geometric mean of y; that is,

$$\widetilde{y} = (y_1 \times y_2 \times \cdots \times y_n)^{1/n} = \exp\left(\frac{1}{n} \sum \log_e y_i \right)$$

The constructed variable is added as a regressor, and the t statistic for this variable is the approximate score statistic for the transformation. Although the score test isn't terribly interesting in light of the ready availability of likelihood ratio tests for the transformation parameter, an added-variable plot for the constructed variable in the auxiliary regression—called a *constructed-variable plot*—shows leverage and influence on the decision to transform y.

The `boxCoxVariable` function in the **car** package facilitates the computation of the constructed variable. Thus, for Ornstein's regression:

```
> mod.ornstein.cv <- update(mod.ornstein,
+     . ~ . + boxCoxVariable(interlocks + 1))
> summary(
+   mod.ornstein.cv)$coef["boxCoxVariable(interlocks + 1)", ,
+                          drop=FALSE]

                               Estimate Std. Error t value
boxCoxVariable(interlocks + 1)   0.6161    0.02421   25.45
                                Pr(>|t|)
boxCoxVariable(interlocks + 1) 3.176e-69

> avPlots(mod.ornstein.cv, "boxCoxVariable(interlocks + 1)")
```

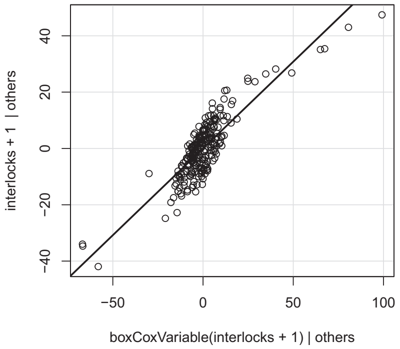

Figure 6.11 Constructed-variable plot for the Box-Cox transformation of y in Ornstein's interlocking-directorate regression.

We are only interested in the t test and added-variable plot for the constructed variable, so we printed only the row of the coefficient table for that variable. The argument `drop=FALSE` told R to print the result as a matrix, keeping the labels, rather than as a vector (see Section 2.3.4). The constructed-variable plot is obtained using `avPlots`, with the second argument specifying the constructed variable. The resulting constructed-variable plot is shown in Figure 6.11. The t statistic for the constructed variable demonstrates that there is very strong evidence of the need to transform y, agreeing with the preferred likelihood ratio test. The constructed-variable plot suggests that this evidence is spread through the data rather than being dependent on a small fraction of the observations.

INVERSE RESPONSE PLOTS

An alternative—or, better, a complement—to the Box-Cox method for transforming the response is the *inverse response plot*, proposed by Cook and Weisberg (1994). While this method produces a transformation toward linearity rather than normality, the results are often similar in cases where the Box-Cox method can be applied. The inverse response plot provides both a numeric estimate and a useful graphical summary; moreover, the inverse response plot can be used even if transformations outside a power family are needed.

The inverse response plot is a special case of the inverse transformation plots introduced in Section 3.4.6. In the current context, we plot the response on the horizontal axis and the fitted values on the vertical axis. To illustrate, we introduce an example that is of historical interest, because it was first used by Box and Cox (1964). Box and Cox's data are from an industrial experiment to study the strength of wool yarn under various conditions. Three predictors were varied in the experiment: `len`, the length of each sample of yarn in millimeters; `amp`, the amplitude of the loading cycle in minutes; and `load`, the amount of weight used in grams. The response, `cycles`, was the number

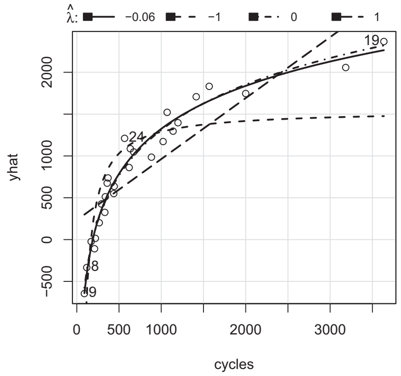

Figure 6.12 Inverse response plot for an additive-regression model fit to Box and Cox's `Wool` data.

of cycles until the sample failed. Data were collected using a 3 × 3 × 3 design, with each of the predictors at three levels. We fit a linear model with main effects only:

```
> (wool.mod <- lm(cycles ~ len + amp + load, data=Wool))

Call:
lm(formula = cycles ~ len + amp + load, data = Wool)

Coefficients:
(Intercept)          len          amp         load
     4521.4         13.2       -535.8        -62.2
```

The inverse response plot for the model is drawn by the following command (and appears in Figure 6.12):

```
> inverseResponsePlot(wool.mod, id.n=4)

    lambda     RSS
1 -0.06052  503066
2 -1.00000 3457493
3  0.00000  518855
4  1.00000 3995722
```

Four lines are shown on the inverse response plot, each of which is from the nonlinear regression of $\widehat{y}$ on $T_{\mathrm{BC}}(y, \lambda)$, for $\lambda = -1, 0, 1$ and for the value of λ that best fits the points in the plot. A linearizing transformation of the response would correspond to a value of λ that matches the data well. In the example, the linearizing transformation producing the smallest residual sum of squares, $\lambda = -0.06$, is essentially the log-transform. As can be seen on the graph, the optimal transformation and log-transform produce essentially the same fitted line, while the other default choices are quite a bit worse. The printed output from the function gives the residual sums of squares for the four fitted lines.

As an alternative approach, the Box-Cox method can be used to find a normalizing transformation, as in the original analysis of these data by Box and Cox (1964):

```
> summary(powerTransform(wool.mod))

bcPower Transformation to Normality

   Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
Y1   -0.0592   0.0611          -0.1789           0.0606

Likelihood ratio tests about transformation parameters
                          LRT df   pval
LR test, lambda = (0)  0.9213  1 0.3371
LR test, lambda = (1) 84.0757  1 0.0000
```

Both methods therefore suggest that the log-transform is appropriate here. The reader is invited to explore these data further. Without transformation, inclusion of higher-order terms in the predictors is required, but in the log-transformed scale, there is a very simple model that closely matches the data.

One advantage of the inverse response plot is that we can visualize the leverage and influence of individual observations on the choice of a transformation; separated points tend to be influential. In Figure 6.12, we marked the four points with the largest residuals from the line for $\lambda = 1$. All these points are very well fit by the log-transformed curve and are in the same pattern as the rest of the data; there are no observations that appear to be overly influential in determining the transformation.

For the `Ornstein` data described earlier in this section, the inverse response plot (not shown) is not successful in selecting a transformation of the response. For these data, the problem is lack of normality, and the inverse response plots transform for linearity, not directly for normality.

6.4.2 PREDICTOR TRANSFORMATIONS

In some instances, predictor transformations resolve problems with a fitted model. Often, these transformations can, and should, be done *before* fitting models to the data (as outlined in Section 3.4). Even well-behaved predictors, however, aren't necessarily linearly related to the response, and graphical diagnostic methods are available that can help select a transformation *after* fitting a model. Moreover, some kinds of nonlinearity can't be fixed by transforming a predictor, and other strategies, such as polynomial regression or regression splines, may be entertained.

COMPONENT-PLUS-RESIDUAL AND CERES PLOTS

Component-plus-residual plots, also called *partial-residual plots*, are a simple graphical device that can be effective in detecting the need to transform a predictor, say x_j, to a new variable $T(x_j)$, for some transformation T. The plot has x_{ij} on the horizontal axis and the *partial residuals*, $e_{\text{Partial},ij} = e_i + b_j x_{ij}$, on

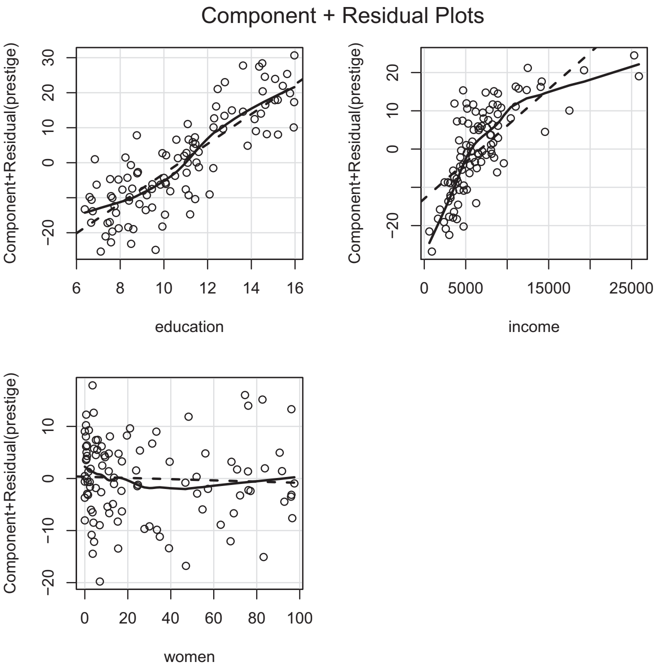

Figure 6.13 Component-plus-residual plots of `order=2` for the Canadian occupational-prestige regression.

the vertical axis. Cook (1993) shows that if the regressions of x_j on the other xs are approximately linear, then the regression function in the component-plus-residual plot provides a visualization of T. Alternatively, if the regressions of x_j on the other xs resemble polynomials, then a modification of the component-plus-residual plot due to Mallows (1986) can be used.

The `crPlots` function in the **car** package constructs component-plus-residual plots for linear models and GLMs. By way of example, we return to the Canadian occupational-prestige regression (from Section 6.2.1). A scatterplot matrix of the three predictors, `education`, `income`, and `women` (Figure 3.13, p. 126), suggests that the predictors are not all linearly related to each other, but no more complicated than quadratic regressions should provide reasonable approximations. Consequently, we draw the component-plus-residual plots specifying `order=2`, permitting quadratic relationships among the predictors:

```
> prestige.mod.3 <- update(prestige.mod.2, ~ . - type + women)
> crPlots(prestige.mod.3, order=2)
```

The component-plus-residual plots for the three predictors appear in Figure 6.13. The broken line on each panel is the *partial fit*, $b_j x_j$, assuming linearity in the partial relationship between y and x_j. The solid line is a `lowess` smooth, and it should suggest a transformation if one is appropriate, for example, via the bulging rule (see Section 3.4.6). Alternatively, the smooth might suggest a quadratic or cubic partial regression or, in more complex cases, the use of a regression spline.

For the Canadian occupational-prestige regression, the component-plus-residual plot for `income` is the most clearly curved, and transforming this variable first and refitting the model is therefore appropriate. In contrast, the component-plus-residual plot for `education` is only slightly nonlinear, and the partial relationship is not simple (in the sense of Section 3.4.6). Finally, the component-plus-residual plot for `women` looks mildly quadratic (although the lack-of-fit test computed by the `residualPlots` command does not suggest a significant quadratic effect), with `prestige` first declining and then rising as `women` increases.

Trial-and-error experimentation moving `income` down the ladder of powers and roots suggests that a log transformation of this predictor produces a reasonable fit to the data:

```
> prestige.mod.4 <- update(prestige.mod.3,
+     . ~ . + log2(income) - income)
```

which is the model we fit in Section 4.2.2. The component-plus-residual plot for `women` in the revised model (not shown) is broadly similar to the plot for `women` in Figure 6.13 (and the lack-of-fit test computed in `residualPlots` has a *p* value of .025) and suggests a quadratic regression:

```
> prestige.mod.5 <- update(prestige.mod.4,
+     . ~ . - women + poly(women, 2))
> summary(prestige.mod.5)$coef

                Estimate Std. Error t value  Pr(>|t|)
(Intercept)      -110.60    13.9817  -7.910 4.160e-12
education           3.77     0.3475  10.850 1.985e-18
log2(income)        9.36     1.2992   7.204 1.262e-10
poly(women, 2)1    15.09     9.3357   1.616 1.093e-01
poly(women, 2)2    15.87     6.9704   2.277 2.499e-02
```

The quadratic term for `women` is statistically significant but not overwhelmingly so.

If the regressions among the predictors are strongly nonlinear and not well described by polynomials, then the component-plus-residual plots may not be effective in recovering nonlinear partial relationships between the response and the predictors. For this situation, Cook (1993) provides another generalization of component-plus-residual plots, called *CERES plots* (for *C*ombining conditional *E*xpectations and *RES*iduals). CERES plots use nonparametric-regression smoothers rather than polynomial regressions to adjust for nonlinear relationships among the predictors. The `ceresPlots` function in the **car** package implements Cook's approach.

Experience suggests that nonlinear relationships among the predictors create problems for component-plus-residual plots only when these relationships are very strong. In such cases, a component-plus-residual plot can appear nonlinear even when the true partial regression is linear—a phenomenon termed *leakage*. For the Canadian occupational-prestige regression, higher-order component-plus-residual plots (in Figure 6.13) and CERES plots are nearly identical to the standard component-plus-residual plots, as the reader may verify.

THE BOX-TIDWELL METHOD FOR CHOOSING PREDICTOR TRANSFORMATIONS

As in transforming the response, transformations of the predictors in regression can be estimated by maximum likelihood. This possibility was suggested by Box and Tidwell (1962), who introduced the model for strictly positive predictors,

$$y = \beta_0 + \beta_1 T_{\mathrm{BC}}(x_1, \gamma_1) + \cdots + \beta_k T_{\mathrm{BC}}(x_k, \gamma_k) + \varepsilon$$

where $T_{\mathrm{BC}}(x_j, \gamma_j)$ is a Box-Cox power transformation (Equation 6.7, p. 304) and the errors ε_i are assumed to be independent and normally distributed with common variance σ^2. Of course, we do not necessarily want to transform *all* the predictors, and in some contexts—such as when dummy regressors are present in the model—it does not even make sense to do so.

The Box-Tidwell regression model is a nonlinear model, which in principle can be fit by nonlinear least-squares.[14] Box and Tidwell describe an approximate computational approach, implemented in the `boxTidwell` function in the **car** package. We apply this function to the Canadian occupational-prestige regression, estimating power transformation parameters for `income` and `education`[15] but specifying a quadratic partial regression for `women`:

```
> boxTidwell(prestige ~ income + education,
+     other.x = ~ poly(women, 2), data=Prestige)

          Score Statistic p-value MLE of lambda
income             -5.301  0.0000       -0.0378
education           2.406  0.0161        2.1928

iterations =  12
```

The one-sided formula for the argument `other.x` indicates the terms in the model that are *not* to be transformed—here the quadratic in `women`. The score tests for the power transformations of `income` and `education` suggest that both predictors need to be transformed; the maximum-likelihood estimates of the transformation parameters are $\widehat{\gamma}_1 = -0.04$ for `income` (effectively, the log transformation of `income`) and $\widehat{\gamma}_2 = 2.2$ for `education` (effectively, the square of `education`).

Constructed variables for the Box-Tidwell transformations of the predictors are given by $x_j \log_e x_j$. These can be easily computed and added to the regression model to produce approximate score tests and constructed-variable plots.

[14] Nonlinear least squares is taken up in the online appendix to this *Companion*.

[15] The component-plus-residual plot for `education` in the preceding section reveals that the curvature of the partial relationship of `prestige` to `education`, which is in any event small, appears to change direction—that is, though monotone is not simple—and so a power transformation is not altogether appropriate here.

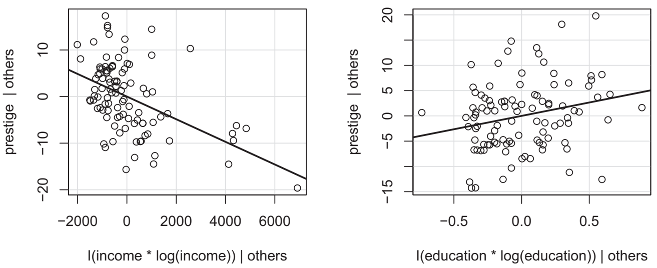

Figure 6.14 Constructed-variable plots for the Box-Tidwell transformation of `income` and `education` in the Canadian occupational-prestige regression.

Indeed, these constructed variables are the basis for Box and Tidwell's computational approach to fitting the model and yield the score statistics printed by the `boxTidwell` function.

To obtain constructed-variable plots (Figure 6.14) for `income` and `education` in the Canadian occupational-prestige regression:[16]

```
> mod.prestige.cv <- lm(prestige ~ income + education
+     + poly(women, 2)
+     + I(income * log(income)) + I(education * log(education)),
+     data=Prestige)
> summary(
+     mod.prestige.cv)$coef["I(income * log(income))", ,
+                           drop=FALSE]

                        Estimate Std. Error t value  Pr(>|t|)
I(income * log(income)) -0.00243  0.0004584  -5.301 7.459e-07

> summary(
+     mod.prestige.cv)$coef["I(education * log(education))", ,
+                           drop=FALSE]

                              Estimate Std. Error t value
I(education * log(education))    5.298      2.202   2.406
                              Pr(>|t|)
I(education * log(education))  0.01808
```

The *identity function* `I()` was used to protect the multiplication operator (`*`), which would otherwise be interpreted specially within a model formula, inappropriately generating main effects and an interaction (see Section 4.8).

The constructed-variable plot for `income` reveals some high-leverage points in determining the transformation of this predictor, but even when these points are removed, there is still substantial evidence for the transformation in the rest of the data.

[16] The observant reader will notice that the *t* values for the constructed-value regression are the same as the score statistics reported by `boxTidwell` but that there are small differences in the *p* values. These differences occur because `boxTidwell` uses the standard-normal distribution for the score test, while the standard summary for a linear model uses the *t* distribution.

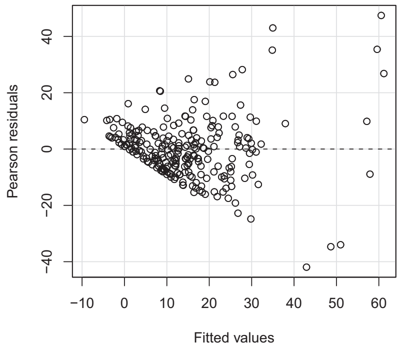

Figure 6.15 Plot of Pearson residuals against fitted values for Ornstein's interlocking-directorate regression.

6.5 Nonconstant Error Variance

One of the assumptions of the standard linear model is that the error variance is fully known apart from an unknown constant, σ^2. It is, however, possible that the error variance depends on one or more of the predictors, on the magnitude of the response, or systematically on some other variable.

To detect nonconstant variance as a function of a variable z, we would like to plot the Pearson residuals, or perhaps their absolute values, versus z. Nonconstant variance would be diagnosed if the variability of the residuals in the graph increased from left to right, decreased from left or right, or displayed another systematic pattern, such as large variation in the middle of the range of z and smaller variation at the edges.

In multiple regression, there are many potential plotting directions. Because obtaining a two-dimensional graph entails projecting the predictors from many dimensions onto one horizontal axis, however, we can never be sure if a plot showing nonconstant spread really reflects nonconstant error variance or some other problem, such as unmodeled nonlinearity (see Cook, 1998, sec. 1.2.1).

For Ornstein's interlocking-directorate regression, for example, we can obtain a plot of residuals against fitted values from the `residualPlots` function in the **car** package (introduced in Section 6.2.1), producing Figure 6.15:

```
> residualPlots(mod.ornstein, ~ 1, fitted=TRUE, id.n=0,
+     quadratic=FALSE, tests=FALSE)
```

The obvious fan-shaped array of points in this plot indicates that residual variance appears to increase as a function of the fitted values—that is, with the estimated magnitude of the response. In Section 5.5, we modeled these data using Poisson regression, for which the variance does increase with the mean, and so reproducing that pattern here is unsurprising. A less desirable

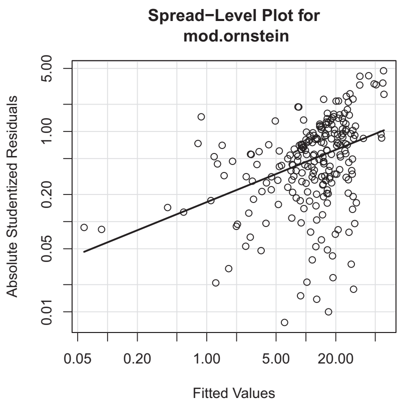

Figure 6.16 Spread-level plot of Studentized residuals against fitted values, for Ornstein's interlocking-directorate regression.

alternative to a regression model that is specifically designed for count data is to try to stabilize the error variance in Ornstein's least-squares regression by transforming the response, as described in the next section.

6.5.1 SPREAD-LEVEL PLOTS

Another diagnostic for nonconstant error variance uses an adaptation of the spread-level plots (Tukey, 1977; introduced in Section 3.4.5), graphing the log of the absolute Studentized residuals against the log of the fitted values. This approach also produces a suggested spread-stabilizing power transformation of y. The `spreadLevelPlot` function in the **car** package has a method for linear models:

```
> spreadLevelPlot(mod.ornstein)

Suggested power transformation:  0.554

Warning message:
In spreadLevelPlot.lm(mod.ornstein) :
    16 negative fitted values removed
```

The linear-regression model fit to Ornstein's data doesn't constrain the fitted values to be positive, even though the response variable `interlocks + 1` is positive. The `spreadLevelPlot` function removes negative fitted values, as indicated in the warning message, before computing logs. The spread-level plot, shown in Figure 6.16, has an obvious tilt to it. The suggested transformation, $\lambda = 0.55$, is not quite as strong as the normalizing transformation estimated by the Box-Cox method, $\widehat{\lambda} = 0.22$ (Section 6.4.1).

6.5.2 SCORE TESTS FOR NONCONSTANT ERROR VARIANCE

Breusch and Pagan (1979) and Cook and Weisberg (1983) suggest a score test for nonconstant error variance in a linear model. The idea is that either the variance is constant or it depends on the mean,

$$\text{Var}(\varepsilon_i) = \sigma^2 g[\text{E}(y|\mathbf{x})]$$

or on a linear combination of regressors $z_1, \ldots, z_p$,

$$\text{Var}(\varepsilon_i) = \sigma^2 g(\gamma_1 z_{i1} + \cdots + \gamma_p z_{ip})$$

In typical applications, the zs are the same as the regressors in the linear model (i.e., the xs), but other choices of zs are possible. For example, in an industrial experiment, variability might differ among the factories that produce a product, and a set of dummy regressors for factory would be candidates for the zs.

The `ncvTest` function in the **car** package implements this score test. We apply `ncvTest` to test for the dependence of spread on level (the default) in Ornstein's regression and for a more general dependence of spread on the predictors in the regression, given in a one-sided formula as the optional second argument to `ncvTest`:

```
> ncvTest(mod.ornstein)

Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 205.9    Df = 1     p = 1.070e-46

> ncvTest(mod.ornstein,~log(assets)+nation+sector, data=Ornstein)

Non-constant Variance Score Test
Variance formula: ~ log(assets) + nation + sector
Chisquare = 290.9    Df = 13     p = 1.953e-54
```

Both tests are highly statistically significant, and the difference between the two suggests that the relationship of spread to level does not entirely account for the pattern of nonconstant error variance in these data. It was necessary to supply the `data` argument in the second command because the `ncvTest` function does not assume that the predictors of the error variance are included in the linear-model object.

6.5.3 OTHER APPROACHES TO NONCONSTANT ERROR VARIANCE

We have suggested transformation as a strategy for stabilizing error variance, but other approaches are available. In particular, if the error variance is proportional to a variable z, then we can fit the model using WLS, with the weights given by $1/z$. In Ornstein's regression, for example, we might take the error variance to increase with the log(`assets`) of the firm, in which case the correct weights would be the inverses of these values:

```
> mod.ornstein.wts <- update(mod.ornstein, weights = 1/log(assets))
```

Still another approach is to rely on the unbiasedness of the least-squares regression coefficients, even when the error variance is misspecified, and then to use a sandwich estimate of the coefficient variances (see Section 4.3.6) to correct the standard errors of the estimated coefficients. These corrections may also be used in the `linearHypothesis`, `deltaMethod`, and `Anova` functions in the **car** package.

6.6 Diagnostics for Generalized Linear Models

Most of the diagnostics of the preceding sections extend straightforwardly to GLMs. These extensions typically take advantage of the computation of maximum-likelihood estimates for GLMs by IRLS (see Section 5.12), which in effect approximates the true log-likelihood by a WLS problem. At the convergence of the IWLS algorithm, diagnostics are formed as if the WLS problem were the problem of interest, and so the exact diagnostics for the WLS fit are approximate diagnostics for the original GLM. Seminal work on the extension of linear least-squares diagnostics to GLMs was done by Pregibon (1981), Landwehr et al. (1980), Wang (1985, 1987), and Williams (1987).

The following functions, some in standard R and some in the **car** package, have methods for GLMs: `rstudent`, `hatvalues`, `cooks.distance`, `dfbeta`, `dfbetas`, `outlierTest`, `avPlots`, `residualPlots`, `marginalModelPlots`, `crPlots`, and `ceresPlots`. We will illustrate the use of these functions selectively, rather than exhaustively repeating all the topics covered for linear models in the previous sections of the chapter.

6.6.1 RESIDUALS AND RESIDUAL PLOTS

One of the major philosophical, though not necessarily practical, differences between linear-model diagnostics and GLM diagnostics is in the definition of residuals. In linear models, the ordinary residual is the difference $\widehat{y} - y$, which is meant to mimic the statistical error $\varepsilon = \mathrm{E}(y|\eta) - y$. Apart from Gaussian or normal linear models, there is no additive error in the definition of a GLM, and so the idea of a residual has a much less firm footing.

Residuals for GLMs are generally defined in analogy to linear models. Here are the various types of GLM residuals that are available in R:

- *Response residuals* are simply the differences between the observed response and its estimated expected value: $y_i - \widehat{\mu}_i$. These differences correspond to the ordinary residuals in the linear model. Apart from the Gaussian or normal case, the response residuals are not used in diagnostics, however, because they ignore the nonconstant variance that is part of a GLM.

- *Pearson residuals* are casewise components of the Pearson goodness-of-fit statistic for the model:

$$e_{Pi} = \frac{y_i - \widehat{\mu}_i}{\sqrt{\widehat{\mathrm{Var}}(y_i|\mathbf{x})/\widehat{\phi}}}$$

Formulas for Var($y|\mathbf{x}$) are given in the last column of Table 5.2 (p. 231). This definition of e_{Pi} corresponds exactly to the Pearson residuals defined in Equation 6.6 (p. 287) for WLS regression. These are a basic set of residuals for use with a GLM because of their direct analogy to linear models. For a model named `m1`, the command `residuals(m1, type="pearson")` returns the Pearson residuals.

- *Standardized Pearson residuals* correct for conditional response variation and for the leverage of the observations:

$$e_{PSi} = \frac{y_i - \widehat{\mu}_i}{\sqrt{\widehat{\mathrm{Var}}(y_i|\mathbf{x})(1 - h_i)}}$$

To compute the e_{PSi}, we need to define the hat-values h_i for GLMs. The h_i are taken from the final iteration of the IWLS procedure for fitting the model and have the usual interpretation, except that, unlike in a linear model, the hat-values in a GLM depend on y as well as on the configuration of the xs.

- *Deviance residuals*, e_{Di}, are the square roots of the casewise components of the residual deviance, attaching the sign of $y_i - \widehat{\mu}_i$. In the linear model, the deviance residuals reduce to the Pearson residuals. The deviance residuals are often the preferred form of residual for GLMs, and are returned by the command `residuals(m1, type= "deviance")`.

- *Standardized deviance residuals* are

$$e_{DSi} = \frac{e_{Di}}{\sqrt{\widehat{\phi}(1 - h_i)}}$$

- The ith Studentized residual in linear models is the scaled difference between the response and the fitted value computed without case i. Because of the special structure of the linear model, these differences can be computed without actually refitting the model by removing case i, but this is not the case for GLMs. While computing n regressions to get the Studentized residuals is not impossible, it is not a desirable option when the sample size is large. An approximation proposed by Williams (1987) is therefore used instead:

$$e_{Ti} = \mathrm{sign}(y_i - \widehat{\mu}_i)\sqrt{(1 - h_i)\,e_{DSi}^2 + h_i e_{PSi}^2}$$

The approximate Studentized residuals are computed when the function `rstudent` is applied to a GLM. A Bonferroni outlier test using the standard-normal distribution may be based on the largest absolute Studentized residual.

As an example, we return to the Canadian women's labor-force participation data, described in Section 5.7. We define a binary rather than a polytomous response, with categories working or not working outside the home, and fit a logistic-regression model to the data:

```
> mod.working <- glm(partic != "not.work" ~ hincome + children,
+       family=binomial, data=Womenlf)
> summary(mod.working)

Call:
glm(formula = partic != "not.work" ~ hincome + children,
    family = binomial, data = Womenlf)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.677  -0.865  -0.777   0.929   1.997

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)       1.3358     0.3838    3.48   0.0005
hincome          -0.0423     0.0198   -2.14   0.0324
childrenpresent  -1.5756     0.2923   -5.39    7e-08

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 356.15  on 262  degrees of freedom
Residual deviance: 319.73  on 260  degrees of freedom
AIC: 325.7

Number of Fisher Scoring iterations: 4
```

The expression `partic != "not.work"` creates a logical vector, which serves as the binary-response variable in the model.

The `residualPlots` function provides the basic plots of residuals versus the predictors and versus the linear predictor:

```
> residualPlots(mod.working, layout=c(1, 3))

         Test stat Pr(>|t|)
hincome      1.226    0.268
children        NA       NA
```

We used the `layout` argument to reformat the graph to have one row and three columns. The function plots Pearson residuals versus each of the predictors in turn. Instead of plotting residuals against fitted values, however, `residualPlots` plots residuals against the estimated linear predictor, $\widehat{\eta}(\mathbf{x})$. Each panel in the graph by default includes a smooth fit rather than a quadratic fit; a lack-of-fit test is provided only for the numeric predictor `hincome` and not for the factor `children` or for the estimated linear predictor.

In binary regression, the plots of Pearson residuals or deviance residuals are strongly patterned—particularly the plot against the linear predictor, where the residuals can take on only two values, depending on whether the response is equal to 0 or 1. In the plot versus `hincome`, we have a little more variety in

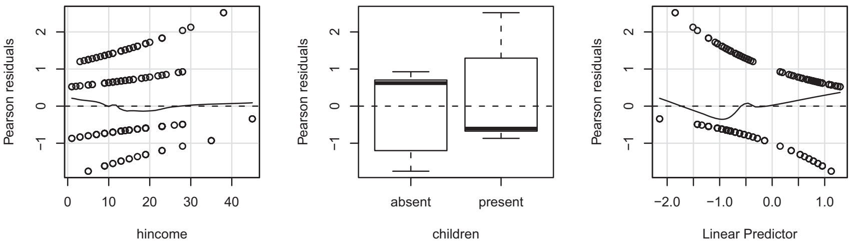

Figure 6.17 Residual plots for the binary logistic regression fit to the Canadian women's labor-force participation data.

the possible residuals: `children` can take on two values, and so the residuals can take on four values for each value of `hincome`. Even in this extreme case, however, a correct model requires that the conditional mean function in any residual plot be constant as we move across the plot. The fitted smooth helps us learn about the conditional mean function, and neither of the smooths shown is especially curved. The lack-of-fit test for `hincome` has a large significance level, confirming our view that this plot does not indicate lack of fit. The residuals for `children` are shown as a boxplot because `children` is a factor. The boxplots for `children` are difficult to interpret because of the discreteness in the distribution of the residuals.

6.6.2 INFLUENCE MEASURES

An approximation to Cook's distance for GLMs is

$$D_i = \frac{e_{PSi}^2}{k+1} \times \frac{h_i}{1-h_i}$$

These values are returned by the `cooks.distance` function. Approximate values of dfbeta$_{ij}$ and dfbetas$_{ij}$ may be obtained directly from the final iteration of the IWLS procedure.

Figure 6.18 shows index plots of Cook's distances and hat-values, produced by the following command:

```
> influenceIndexPlot(mod.working, vars=c("Cook", "hat"), id.n=3)
```

Setting `vars=c("Cook", "hat")` limited the graphs to these two diagnostics. Cases 76 and 77 have the largest Cook distances, although even these are quite small. We remove both Cases 76 and 77 as a check:

```
> compareCoefs(mod.working, update(mod.working, subset=-c(76, 77)))

Call:
1:glm(formula = partic != "not.work" ~ hincome + children,
     family = binomial, data = Womenlf)
2:glm(formula = partic != "not.work" ~ hincome + children,
     family = binomial, data = Womenlf, subset = -c(76, 77))
```

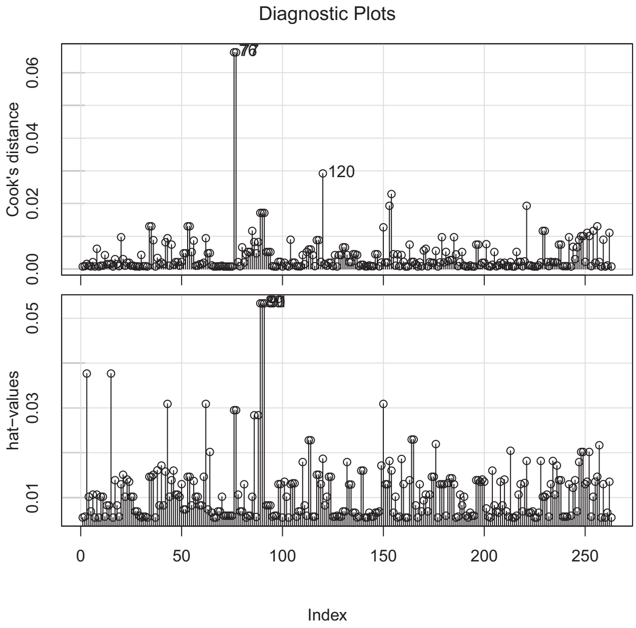

Figure 6.18 Index plots of diagnostic statistics for the logistic regression fit to the Canadian women's labor-force participation data.

```
                 Est. 1    SE 1  Est. 2    SE 2
(Intercept)       1.3358  0.3838  1.6090  0.4052
hincome          -0.0423  0.0198 -0.0603  0.0212
childrenpresent -1.5756  0.2923 -1.6476  0.2978
```

The reader can verify that removing just one of the two observations does not alter the results much, but removing *both* observations changes the coefficient of husband's income by more than 40%, about one standard error. Apparently, the two cases mask each other, and removing them both is required to produce a meaningful change in the coefficient for `hincome`. Cases 76 and 77 are women working outside the home even though both have children and high-income husbands.

6.6.3 GRAPHICAL METHODS: ADDED-VARIABLE AND COMPONENT-PLUS-RESIDUAL PLOTS

We are aware of two extensions of added-variable plots to GLMs. Suppose that the focal regressor is x_j. Wang (1985) proceeds by refitting the model with x_j removed, extracting the working residuals from this fit. Then x_j is regressed on the other xs by WLS, using the weights from the last IWLS step and obtaining residuals. Finally, the two sets of residuals are plotted against each other. The Arc regression software developed by Cook and Weisberg (1999) employs a similar procedure, except that weights are not used in the least-squares regression of x_j on the other xs. The `avPlots` function in

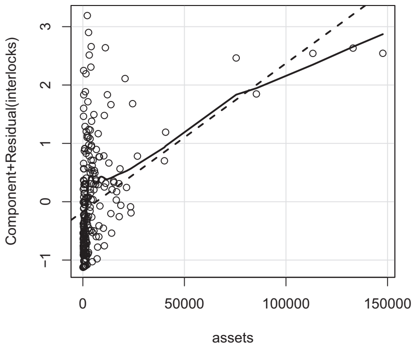

Figure 6.19 Component-plus-residual plot for `assets` in the Poisson regression fit to Ornstein's interlocking-directorate data.

the **car** package implements both approaches, with Wang's procedure as the default. Added-variable plots for binary-regression models can be uninformative because of the extreme discreteness of the response variable.

Component-plus-residual and CERES plots also extend straightforwardly to GLMs. Nonparametric smoothing of the resulting scatterplots can be important for interpretation, especially in models for binary responses, where the discreteness of the response makes the plots difficult to examine. Similar, if less striking, effects can occur for binomial and Poisson data.

For an illustrative component-plus-residual plot, we reconsider Ornstein's interlocking-directorate Poisson regression (from Section 5.5), but now we fit a model that uses `assets` as a predictor rather than the log of `assets`:

```
> mod.ornstein.pois <- glm(interlocks ~ assets + nation + sector,
+     family=poisson, data=Ornstein)
> crPlots(mod.ornstein.pois, "assets")
```

The component-plus-residual plot for `assets` is shown in Figure 6.19. This plot is difficult to interpret because of the extreme positive skew in `assets`, but it appears as if the `assets` slope is a good deal steeper at the left than at the right. The bulging rule, therefore, points toward transforming `assets` down the ladder of powers, and indeed the log-rule in Section 3.4.1 suggests replacing `assets` by its logarithm *before* fitting the regression in the first place (which, of course, is what we did originally):

```
> mod.ornstein.pois.2 <- update(mod.ornstein.pois,
+     . ~ log2(assets) + nation + sector)
> crPlots(mod.ornstein.pois.2, "log2(assets)")
```

The linearity of the follow-up component-plus-residual plot in Figure 6.20 confirms that the log-transform is a much better scale for `assets`.

The other diagnostics described in Section 6.4 for selecting a predictor transformation lead to the log-transform as well. For example, the Box-Tidwell

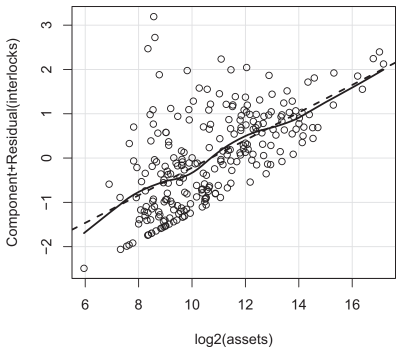

Figure 6.20 Component-plus-residual plot for the log of `assets` in the respecified Poisson regression for Ornstein's data.

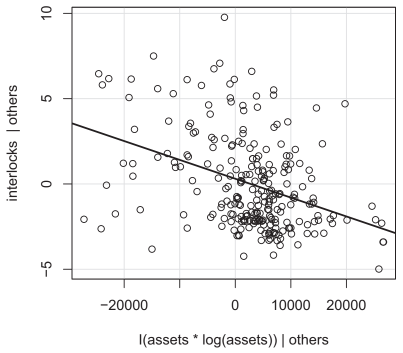

Figure 6.21 Constructed-variable plot for the power transformation of `assets` in Ornstein's interlocking-directorate Poisson regression.

constructed-variable plot for the power transformation of a predictor (introduced in Section 6.4.2) also extends directly to GLMs, augmenting the model with the constructed variable $x_j \log_e x_j$. We can use this method with Ornstein's Poisson regression:

```
> mod.ornstein.pois.cv <- update(mod.ornstein.pois,
+     . ~ . + I(assets*log(assets)))
> avPlots(mod.ornstein.pois.cv, "I(assets * log(assets))", id.n=0)
> summary(
+     mod.ornstein.pois.cv)$coef["I(assets * log(assets))", ,
+                                drop=FALSE]

                         Estimate Std. Error z value  Pr(>|z|)
I(assets * log(assets)) -2.177e-05  1.413e-06  -15.41 1.409e-53
```

Only the z test statistic for the constructed variable `I(assets * log(assets))` is of interest, and it leaves little doubt about the need for transforming `assets`. The constructed-variable plot in Figure 6.21 supports the transformation.

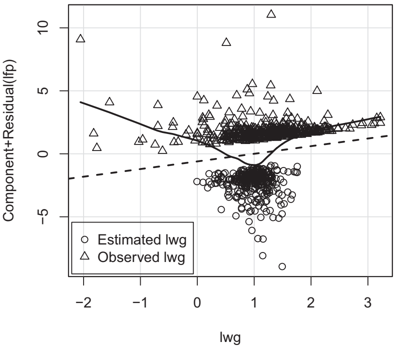

Figure 6.22 Component-plus-residual plot for `lwg` in the binary logistic regression for Mroz's women's labor force participation data.

An estimate of the transformation parameter can be obtained from the coefficient of `assets` in the *original* Poisson regression (2.09×10^{-5}) and the coefficient of the constructed variable (-2.18×10^{-5}):[17]

$$\widetilde{\lambda} = 1 + \frac{-2.18 \times 10^{-5}}{2.09 \times 10^{-5}} = -0.043$$

that is, essentially the log transformation, $\lambda = 0$.

We conclude with a reexamination of the binary logistic-regression model fit to Mroz's women's labor force participation data (introduced in Section 5.3). One of the predictors in this model—the log of the woman's expected wage rate (`lwg`)—has an odd definition: For women in the labor force, for whom the response `lfp` = `"yes"`, `lwg` is the log of the *actual* wage rate, while for women not in the labor force, for whom `lfp` = `"no"`, `lwg` is the log of the *predicted* wage rate from the regression of wages on the other predictors.

To obtain a component-plus-residual plot for `lwg` (Figure 6.22):

```
> mod.mroz <- glm(lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
+     family=binomial, data=Mroz)
> crPlots(mod.mroz, "lwg", pch=as.numeric(Mroz$lfp))
> legend("bottomleft",c("Estimated lwg", "Observed lwg"),
+     pch=1:2, inset=0.01)
```

The peculiar split in the plot reflects the binary-response variable, with the lower cluster of points corresponding to `lfp` = `"no"` and the upper cluster to `lfp` = `"yes"`. It is apparent that `lwg` is much less variable when `lfp` = `"no"`, inducing an artifactually curvilinear relationship between `lwg` and `lfp`: We expect fitted values (such as the values of `lwg` when `lfp` = `"no"`) to be more homogeneous than observed values, because fitted values lack a residual component of variation.

We leave it to the reader to construct component-plus-residual or CERES plots for the other predictors in the model.

[17] Essentially the same calculation is the basis of Box and Tidwell's iterative procedure for finding transformations in linear least-squares regression (Section 6.4.2).

6.7 Collinearity and Variance Inflation Factors

When there are strong linear relationships among the predictors in a regression analysis, the precision of the estimated regression coefficients in linear models declines compared with what it would have been were the predictors uncorrelated with each other. Other important aspects of regression analysis beyond coefficients, such as prediction, are much less affected by collinearity (as discussed in Weisberg, 2005, sec. 10.1).

The estimated sampling variance of the *j*th regression coefficient may be written as

$$\widehat{\mathrm{Var}}(b_j) = \frac{\widehat{\sigma}^2}{(n-1)s_j^2} \times \frac{1}{1-R_j^2}$$

where $\widehat{\sigma}^2$ is the estimated error variance, s_j^2 is the sample variance of x_j, and $1/(1-R_j^2)$, called the *variance inflation factor* (VIF_j) for b_j, is a function of the multiple correlation R_j from the regression of x_j on the other *x*s. The VIF is the simplest and most direct measure of the harm produced by collinearity: The square root of the VIF indicates how much the confidence interval for β_j is expanded relative to similar uncorrelated data, were it possible for such data to exist. If we wish to explicate the collinear relationships among the predictors, then we can examine the coefficients from the regression of each predictor with a large VIF on the other predictors.

The VIF is not applicable, however, to sets of related regressors for multiple-degree-of-freedom effects, such as polynomial regressors or contrasts constructed to represent a factor. Fox and Monette (1992) generalize the notion of variance inflation by considering the relative size of the joint confidence region for the coefficients associated with a related set of regressors. The resulting measure is called a *generalized variance inflation factor* (or GVIF).[18] If there are p regressors in a term, then $\mathrm{GVIF}^{1/2p}$ is a one-dimensional expression of the decrease in the precision of estimation due to collinearity—analogous to taking the square root of the usual VIF. When $p = 1$, the GVIF reduces to the usual VIF.

The `vif` function in the **car** package calculates VIFs for the terms in a linear model. When each term has one degree of freedom, the usual VIF is returned, otherwise the GVIF is calculated.

As a first example, consider the data on the 1980 U.S. Census undercount in the data frame `Ericksen` (Ericksen et al., 1989):

[18]* Let $\mathbf{R}_{11}$ represent the correlation matrix among the regressors in the set in question; $\mathbf{R}_{22}$, the correlation matrix among the other regressors in the model; and $\mathbf{R}$, the correlation matrix among all the regressors in the model. Fox and Monette show that the squared area, volume, or hyper-volume of the joint confidence region for the coefficients in either set is expanded by the GVIF,

$$\mathrm{GVIF} = \frac{\det \mathbf{R}_{11} \det \mathbf{R}_{22}}{\det \mathbf{R}}$$

relative to similar data in which the two sets of regressors are uncorrelated with each other. This measure is independent of the bases selected to span the subspaces of the two sets of regressors and so is independent, for example, of the contrast-coding scheme employed for a factor.

```
> head(Ericksen)

             minority crime poverty language highschool housing
Alabama          26.1    49    18.9      0.2       43.5     7.6
Alaska            5.7    62    10.7      1.7       17.5    23.6
Arizona          18.9    81    13.2      3.2       27.6     8.1
Arkansas         16.9    38    19.0      0.2       44.5     7.0
California.R     24.3    73    10.4      5.0       26.0    11.8
Colorado         15.2    73    10.1      1.2       21.4     9.2
              city conventional undercount
Alabama      state            0      -0.04
Alaska       state          100       3.35
Arizona      state           18       2.48
Arkansas     state            0      -0.74
California.R state            4       3.60
Colorado     state           19       1.34
```

These variables describe 66 areas of the United States, including 16 major cities, the 38 states without major cities, and the remainders of the 12 states that contain the 16 major cities. The following variables are included:

- `minority`: Percentage of residents who are black or Hispanic.
- `crime`: Number of serious crimes per 1,000 residents.
- `poverty`: Percentage of residents who are poor.
- `language`: Percentage having difficulty speaking or writing English.
- `highschool`: Percentage of those 25 years of age or older who have *not* finished high school.
- `housing`: Percentage of dwellings in small, multi-unit buildings.
- `city`: A factor with levels `state` and `city`.
- `conventional`: Percentage of households counted by personal enumeration (rather than by a mail-back questionnaire with follow-up).
- `undercount`: The estimated percent undercount (with negative values indicating an estimated *over*count).

We regress the Census `undercount` on the other variables:

```
> mod.census <- lm(undercount ~ ., data=Ericksen)
> summary(mod.census)

Call:
lm(formula = undercount ~ ., data = Ericksen)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8356 -0.8033 -0.0553  0.7050  4.2467

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.61141    1.72084   -0.36  0.72368
minority      0.07983    0.02261    3.53  0.00083
crime         0.03012    0.01300    2.32  0.02412
poverty      -0.17837    0.08492   -2.10  0.04012
language      0.21512    0.09221    2.33  0.02320
highschool    0.06129    0.04477    1.37  0.17642
```

```
housing      -0.03496    0.02463   -1.42  0.16126
citystate    -1.15998    0.77064   -1.51  0.13779
conventional  0.03699    0.00925    4.00  0.00019

Residual standard error: 1.43 on 57 degrees of freedom
Multiple R-squared: 0.708,       Adjusted R-squared: 0.667
F-statistic: 17.2 on 8 and 57 DF,  p-value: 1.04e-12
```

We included the `data` argument to `lm`, so we may use a period (`.`) on the right-hand side of the model formula to represent all the variables in the data frame with the exception of the response—here, `undercount`.

Checking for collinearity, we see that three coefficients—for `minority`, `poverty`, and `highschool`—have VIFs exceeding 4, indicating that confidence intervals for these coefficients are more than twice as wide as they would be for uncorrelated predictors:

```
> vif(mod.census)

    minority        crime      poverty     language   highschool
       5.009        3.344        4.625        1.636        4.619
     housing         city conventional
       1.872        3.538        1.691
```

To illustrate the computation of GVIFs, we return to Ornstein's interlocking-directorate regression, where it turns out that collinearity is relatively slight:

```
> vif(mod.ornstein)

             GVIF Df GVIF^(1/(2*Df))
log(assets) 1.909  1           1.382
nation      1.443  3           1.063
sector      2.597  9           1.054
```

The `vif` function can also be applied to GLMs, such as the Poisson-regression model fit to Ornstein's data:[19]

```
> vif(mod.ornstein.pois.2)

              GVIF Df GVIF^(1/(2*Df))
log2(assets) 2.617  1           1.618
nation       1.620  3           1.084
sector       3.718  9           1.076
```

Other, more complex, approaches to collinearity include principal-components analysis of the predictors or standardized predictors and singular-value decomposition of the model matrix or the mean-centered model matrix. These, too, are simple to implement in R: See the `princomp`, `prcomp`, `svd`, and `eigen` functions (the last two of which are discussed in Section 8.2).

[19] Thanks to a contribution from Henric Nilsson.

6.8 Complementary Reading and References

- Residuals and residual plotting for linear models are discussed in Weisberg (2005, sec. 8.1–8.2). Marginal model plots, introduced in Section 6.2.2, are described in Weisberg (2005, sec. 8.4). Added-variable plots are discussed in Weisberg (2005, sec. 3.1). Outliers and influence are taken up in Weisberg (2005, chap. 9).
- Diagnostics for unusual and influential data are described in Fox (2008, chap. 11); for nonnormality, nonconstant error variance, and nonlinearity in Fox (2008, chap. 12); and for collinearity in Fox (2008, chap. 13). A general treatment of residuals in models without additive errors, which expands on the discussion in Section 6.6.1, is given by Cox and Snell (1968). Diagnostics for GLMs are taken up in Fox (2008, sec. 15.4).
- For further information on various aspects of regression diagnostics, see Cook and Weisberg (1982, 1994, 1997, 1999), Fox (1991), Cook (1998), and Atkinson (1985).