Chapter 3

Topics in Regression

The previous two chapters have provided a detailed introduction to the basic principles underlying simple linear regression. This chapter will cover some additional topics in regression, but not with the same detail as in the previous two chapters. Sometimes just a single example will illustrate a regression topic that deserves an entire chapter in a full-semester regression course. The topics considered in this chapter are forcing a regression line through the origin, diagnostics, remedial procedures, the matrix approach to simple linear regression, multiple linear regression, weighted least squares estimators, regression models with nonlinear terms, and logistic regression.

3.1 Regression Through the Origin

Applications occasionally arise in which it is of benefit to force a regression line to pass through the origin. To illustrate such applications, return to Examples 1.1 and 1.3 in which Bob and Cheryl each had the number of sales per week as an independent variable X. In both of the examples, $X = 0$ sales per week corresponds to $Y = 0$ commissions (for Bob) and $Y = 0$ revenue per week (from Cheryl's sales). In these settings it is sensible to force the regression line to pass through the origin; estimating a population intercept does not make sense. The resulting regression model does not contain the β_0 parameter. The simple linear regression model forced through the origin, sometimes abbreviated RTO for regression through the origin, is defined next.

Definition 3.1 A *simple linear regression model forced through the origin* is given by

$$Y = \beta_1 X + \varepsilon,$$

where

- X is the independent variable, assumed to be a fixed value observed without error,
- Y is the dependent variable, which is a continuous random variable,
- β_1 is the population slope of the regression line, which is an unknown constant, and
- ε is the error term, a random variable that accounts for the randomness in the relationship between X and Y, which has population mean zero and finite population variance σ^2.

The regression parameter β_1 can be estimated using least squares from the data pairs (X_i, Y_i) for $i = 1, 2, \ldots, n$.

Theorem 3.1 Let $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ be n data pairs satisfying $\sum_{i=1}^{n} X_i^2 > 0$. The *least squares estimator* of β_1,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i}{\sum_{i=1}^{n} X_i^2},$$

minimizes the sum of the squared deviations between Y_i and the associated fitted value $\hat{\beta}_1 X_i$ in the simple linear regression model forced through the origin.

Proof The sum of squared deviations between the observed values of the dependent variable and the associated fitted values is

$$S = \sum_{i=1}^{n} (Y_i - \beta_1 X_i)^2.$$

To minimize S with respect to β_1, take the derivative of S with respect to β_1:

$$\frac{dS}{d\beta_1} = -2 \sum_{i=1}^{n} X_i (Y_i - \beta_1 X_i) = 0$$

or

$$\sum_{i=1}^{n} X_i Y_i - \beta_1 \sum_{i=1}^{n} X_i^2 = 0.$$

This equation can be solved in closed-form for $\hat{\beta}_1$ as

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i}{\sum_{i=1}^{n} X_i^2}.$$

To show that the *least* squares estimator $\hat{\beta}_1$ minimizes S, take a second derivative of S:

$$\frac{d^2 S}{d\beta_1^2} = 2 \sum_{i=1}^{n} X_i^2.$$

Since $\sum_{i=1}^{n} X_i^2 > 0$, this second derivative, which is just twice a sum of squares, must be positive. Hence, S is minimized at $\hat{\beta}_1$. □

The next example conducts a hypothesis test to determine whether it is appropriate to drop the intercept term from the simple linear regression model based on the data pairs, and then proceeds to fit the reduced model.

Example 3.1 The R built-in data set `Formaldehyde` consists of the $n = 6$ data pairs given in Table 3.1. The independent variable `carb` is the carbohydrate level (ml) and the dependent variable `optden` is the optical density in a chemical experiment. Fit a simple linear regression to the model using the ordinary least squares estimates. If there is no statistically significant difference between the estimated intercept and zero, then fit a simple linear regression model forcing the regression line to pass through the origin to the data pairs.

carb	optden
0.1	0.086
0.3	0.269
0.5	0.446
0.6	0.538
0.7	0.626
0.9	0.782

Table 3.1: `Formaldehyde` data set from R.

The scatterplot given in Figure 3.1 shows a strong linear relationship between carbohydrates (measured in ml) and optical density (measured by the reading of the resulting purple color on a spectrophotometer) for the $n = 6$ data pairs. The nearly-perfect linear relationship provides overwhelming visual evidence that a simple linear regression model is appropriate for approximating the relationship between X and Y.

The R commands below fit the standard simple linear regression model (including an intercept) to the six data pairs.

```
fit = lm(optden ~ carb, data = Formaldehyde)
summary(fit)
```

The point estimates for the intercept and slope of the regression line are

$$\hat{\beta}_0 = 0.00509 \qquad \text{and} \qquad \hat{\beta}_1 = 0.876.$$

The call to `summary(fit)` indicates that there is no statistically significant difference between the point estimate for the intercept and 0. The p-value associated with the hypothesis test

$$H_0 : \beta_0 = 0$$

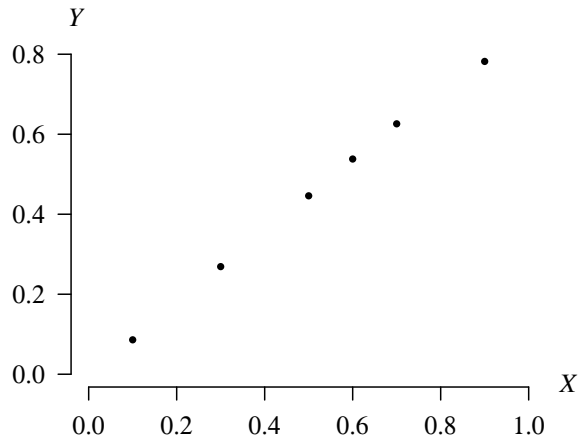

Figure 3.1: A scatterplot of the `Formaldehyde` data pairs.

versus

$$H_0 : \beta_0 \neq 0$$

is 0.55, which is statistical evidence that the intercept does not differ significantly from $\beta_0 = 0$. This p-value, perhaps along with some information about the chemical experiment itself, might cause the experimenter to consider the reduced model which is forced through the origin. This hypothesis test requires normally distributed error terms. The usual analysis of residuals to determine whether a simple linear regression model with normal error terms is appropriate in this setting will be abandoned here because of the small sample size. Histograms and statistical tests have diminished meaning with only $n = 6$ data pairs. The best we can do to assess the normality of the error terms is to use a graphical display such as a QQ plot.

Using Theorem 3.1, the least squares estimate for the slope of the regression line forced through the origin is

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i}{\sum_{i=1}^{n} X_i^2} = 0.884,$$

which can be calculated with the R statements given below.

```
x    = Formaldehyde$carb
y    = Formaldehyde$optden
beta = sum(x * y) / sum(x * x)
print(beta)
```

Not surprisingly, the slope of the regression line forced through the origin is very close to the slope of the regression line with the model that includes an intercept. The optical density increases by 0.884 for every unit increase in the carbohydrates. Figure 3.2 contains a scatterplot of the data pairs and the associated regression line forced through the origin. The model clearly provides an adequate approximation to the relationship between the independent variable X and the dependent variable Y over the scope of the model shown in Figure 3.2.

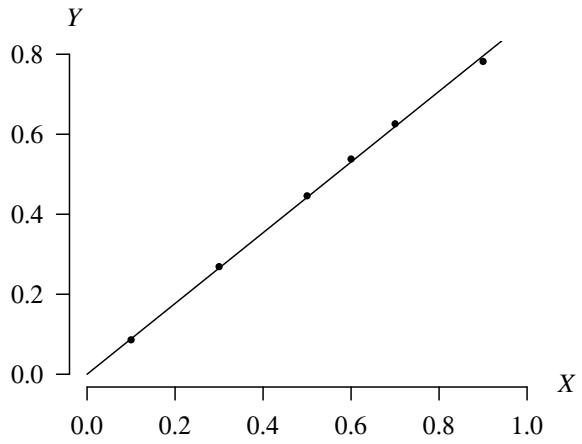

Figure 3.2: A scatterplot of the `Formaldehyde` data pairs with the fitted regression line.

These calculations can be performed in R by adding `-1` or `+0` to the `formula` argument in the `lm` function, which forces the regression line to pass through the origin.

```
fit2 = lm(optden ~ carb - 1, data = Formaldehyde)
fit2$coefficients
```

These R statements calculate the estimated slope of the regression line as $\hat{\beta}_1 = 0.884$.

Analogous theorems to those that were applied to simple linear regression with a population intercept parameter β_0 and a population slope parameter β_1 from Chapter 1 can also be derived associated with the simple linear regression model forced through the origin. In addition, the assumption of normal error terms from Chapter 2 can be added to the simple linear regression model forced through the origin, which allows for statistical inference (that is, constructing confidence intervals and performing hypothesis tests) concerning the population slope of the regression line β_1. For example, the *additional* R command

```
confint(fit2)
```

gives a very narrow 95% confidence interval for β_1 as

$$0.869 < \beta_1 < 0.899.$$

The narrowness of the confidence interval is a reflection of how close the points fall to the regression line in Figure 3.2.

The next example revisits the regression modeling of the stopping distance as a function of the speed of a car in the built-in `cars` data frame.

Example 3.2 Recall from Example 2.8 that X, the speed of a car in miles per hour, was used as an independent variable, and Y, the stopping distance in feet, was used as a dependent variable in a simple linear regression model. There are $n = 50$ data pairs in the `cars` data frame that is built into R. One critique of the simple linear regression model that was constructed for the data pairs in the built-in `cars` data frame from Example 2.8 was that the regression function did not pass through the origin (stationary cars require no stopping distance). Write R code to estimate the slope of the regression line through the origin and comment on the acceptability of this model.

The physics of the experiment indicates that stationary cars require no distance to stop, so forcing a regression line through the origin is appropriate in this setting. The R code below estimates the slope of the regression line that is forced to pass through the origin.

```
x   = cars$speed
y   = cars$dist
fit = lm(y ~ x - 1)
```

Figure 3.3 is a scatterplot of the data pairs (not jittered for ties) with the regression line superimposed. A car requires an additional distance of $\hat{\beta}_1 = 2.91$ feet to stop for every additional mile per hour in speed.

The *additional* R statements

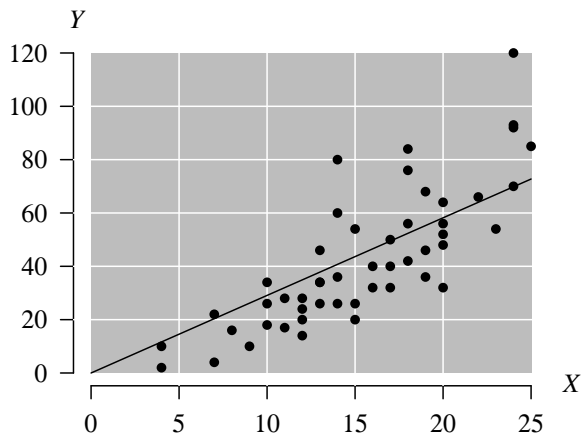

Figure 3.3: Fitted model $Y = \hat{\beta}_1 X$ of speed X and stopping distance Y for the `cars` data.

```
table(sign(fit$residuals))
sum(fit$residuals ^ 2)
```

reveal that 32 data pairs fall below the regression line and only 18 data pairs fall above the regression line. A plot of the standardized residuals can be generated with the R statements

```
res = lm(dist ~ speed - 1, data = cars)$residuals
plot(cars$speed, res / sqrt(sum(res ^ 2) / (length(cars$speed) - 2)))
```

and is given in Figure 3.4. The sum of squares increases from $SSE = 11{,}354$ as calculated in Example 2.8 for the full simple linear regression model to $SSE = 12{,}954$ by

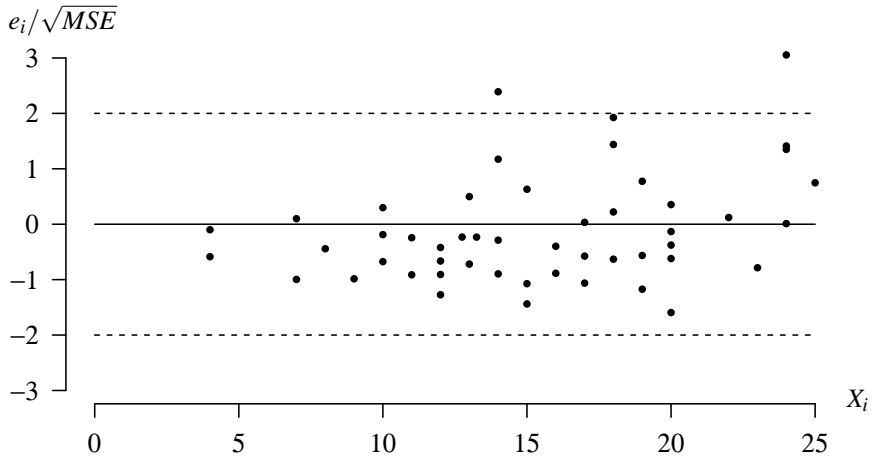

Figure 3.4: Standardized residuals for the `cars` data.

forcing the regression line through the origin. It is universally the case that *SSE* stays the same or increases by forcing the regression line to pass through the origin. Using the model as a subscript, this can be written symbolically as

$$SSE_{Y=\beta_0+\beta_1 X+\varepsilon} \leq SSE_{Y=\beta_1 X+\varepsilon}.$$

The nonsymmetry of the residuals in Figure 3.4 suggests that the fitted linear regression function might not be adequate. Perhaps a regression model with higher-order terms or a nonlinear model is worth investigating.

This ends the discussion of forcing the regression line through the origin. Occasions arise in regression modeling in which it is more appropriate to fit a statistical model with fewer parameters. Some of the results from the full simple linear regression model generalize to simple linear regression forced through the origin. The point estimate for β_1, for example, is unbiased. Three examples of results that do *not* generalize are (a) the residuals do not necessarily sum to zero, (b) the regression line does not necessarily pass through the point $\left(\bar{X}, \bar{Y}\right)$, and (c) it is possible that *SSE* can exceed the total sum of squares *SST*, which can result in a negative value of R^2.

3.2 Diagnostics

Diagnostic procedures are applied to fitted regression models to assess their conformity to the assumptions (for example, constant variance of the error terms) implicit in the simple linear regression model. We have already considered one such diagnostic procedure from the previous chapter, which is the examination of the residuals to assess their independence, constant variance, and normality. Two other diagnostic procedures will be examined here, which are the identification of data pairs known as *leverage points* and the identification of data pairs known as *influential points*. The subsequent section considers *remedial procedures*, which can be applied to a regression model that fails to satisfy one or more of the assumptions implicit in a regression model.

3.2.1 Leverage

Data pairs that have the ability to exert more influence on the regression line than other data pairs due to their independent variable values are known as *leverage points*. These data pairs should be given more scrutiny than the others because of the potential tug that they have on the regression line. More specifically, when the value of the independent variable is unusually far from $\bar{X}$ (either low or high), the data pair has the potential to exert more pull on the regression line than other points.

We begin developing the notion of leverage by expressing the predicted value of Y_i, denoted by $\hat{Y}_i$, as a function of Y_i. Using Theorems 1.1 and 1.3, the predicted value of Y_i is

$$\begin{aligned}
\hat{Y}_i &= \hat{\beta}_0 + \hat{\beta}_1 X_i \\
&= \bar{Y} - \hat{\beta}_1 \bar{X} + \hat{\beta}_1 X_i \\
&= \bar{Y} + \hat{\beta}_1 (X_i - \bar{X}) \\
&= \frac{1}{n} \sum_{j=1}^{n} Y_j + \sum_{j=1}^{n} a_j Y_j (X_i - \bar{X}) \\
&= \frac{1}{n} \sum_{j=1}^{n} Y_j + \sum_{j=1}^{n} \frac{X_j - \bar{X}}{S_{XX}} Y_j (X_i - \bar{X})
\end{aligned}$$

$$= \sum_{j=1}^{n} \left[\frac{1}{n} + \frac{(X_i - \bar{X})(X_j - \bar{X})}{S_{XX}} \right] Y_j$$
$$= \sum_{j=1}^{n} h_{ij} Y_j$$

for $i = 1, 2, \ldots, n$. The h_{ij} values form the elements of an $n \times n$ matrix $\mathbf{H}$, which is often referred to as the *hat matrix* or the *projection matrix*. The reason that this matrix is known as the projection matrix is that it provides a linear transformation from the observed values of the dependent variable to the associated fitted values. The diagonal elements of the hat matrix are known as the leverages of the data pairs, which are defined next.

Definition 3.2 The *leverage* of data pair (X_i, Y_i) in a simple linear regression model is

$$h_{ii} = \frac{1}{n} + \frac{(X_i - \bar{X})^2}{S_{XX}}$$

for $i = 1, 2, \ldots, n$.

The leverage is a measure of a data pair's potential to influence the regression line. Notice that the leverage is a function of the values of the independent variable $X_1, X_2, \ldots, X_n$ only; the heights of the data pairs do not play a role. Since the two denominators in the expression from Definition 3.2 are constants for a particular data set, only the numerator $(X_i - \bar{X})^2$ changes for each value of X_i. It reflects the distance between a particular X_i value and its associated sample mean. The leverage increases as the distance between X_i and $\bar{X}$ increases. There are several results concerning the leverages; one that concerns the average of the leverages is presented next.

Theorem 3.2 For data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ in a simple linear regression model, the sample mean of the leverages is $2/n$.

Proof The sample mean of the leverages is

$$\begin{aligned}
\frac{h_{11} + h_{22} + \cdots + h_{nn}}{n} &= \frac{1}{n} \left[\frac{1}{n} + \frac{(X_1 - \bar{X})^2}{S_{XX}} + \frac{1}{n} + \frac{(X_2 - \bar{X})^2}{S_{XX}} + \cdots + \frac{1}{n} + \frac{(X_n - \bar{X})^2}{S_{XX}} \right] \\
&= \frac{1}{n} \left[1 + \frac{(X_1 - \bar{X})^2 + (X_2 - \bar{X})^2 + \cdots + (X_n - \bar{X})^2}{S_{XX}} \right] \\
&= \frac{1}{n} \left[1 + \frac{S_{XX}}{S_{XX}} \right] \\
&= \frac{2}{n}.
\end{aligned}$$

□

To summarize what we know about the n leverages,

- the leverages are the diagonal elements of the hat matrix $\mathbf{H}$,
- all leverages are positive, with a minimum of $1/n$ (for $X_i = \bar{X}$) and a maximum of 1, and
- the sum of the leverages is 2, so the average of the leverages is $2/n$.

If all of the leverages are equal (this is always the case, for example, for $n = 2$ data pairs), then each leverage is $2/n$, which is the average from Theorem 3.2. We would like to establish a threshold at which a data pair has the ability to exert a significant influence over the regression line so that such points might be examined with additional scrutiny. Such data pairs are known as *leverage points*. Although not used universally, a common way to identify a leverage point is if the leverage h_{ii} is more than twice the average of the leverages. Symbolically, a point is designated a leverage point if

$$h_{ii} > \frac{4}{n}.$$

This threshold will be illustrated in the next example.

Example 3.3 To illustrate the identification of leverage points, we consider the first data set in Anscombe's quartet. For notational convenience, the $n = 11$ data pairs have been ordered by their independent variable values in Table 3.2. We will investigate the leverages associated with this data set and two other data sets with an extra data pair appended.

X_i	Y_i
4.0	4.26
5.0	5.68
6.0	7.24
7.0	4.82
8.0	6.95
9.0	8.81
10.0	8.04
11.0	8.33
12.0	10.84
13.0	7.58
14.0	9.96

Table 3.2: Data set I (sorted by X_i) in Anscombe's quartet.

The R code below calculates the $n = 11$ leverages using the formula from Definition 3.2.

```
x          = 4:14
xbar       = mean(x)
sxx        = sum((x - xbar) ^ 2)
n          = length(x)
leverages = 1 / n + (x - xbar) ^ 2 / sxx
```

Notice that the values of $Y_1, Y_2, \ldots, Y_{11}$ are not needed to compute the leverages. The leverages are displayed in Table 3.3. Not surprisingly, the leverages are symmetric about $\bar{X} = 9$ because the values of the independent variable are equally spaced. The leverage for $X_6 = 9$ is just $1/n = 1/11 \cong 0.09$, which is the first term in h_{ii} in Definition 3.2. None of the leverages exceeds the threshold value $4/n = 4/11 \cong 0.36$, so this data set does not contain any leverage points.

Calculating leverages is so common in regression analysis that R has two built-in functions that calculate leverages. The `hat` function calculates the leverages for Anscombe's first data set with the single statement

i	1	2	3	4	5	6	7	8	9	10	11
X_i	4	5	6	7	8	9	10	11	12	13	14
h_{ii}	0.32	0.24	0.17	0.13	0.10	0.09	0.10	0.13	0.17	0.24	0.32

Table 3.3: Leverages for data set I in Anscombe's quartet.

```
hat(4:14)
```

Alternatively, the `hatvalues` function with the fitted model as an argument can be used to calculate the leverages.

```
x = 4:14
y = c(4.26, 5.68, 7.24, 4.82, 6.95, 8.81, 8.04, 8.33, 10.84, 7.58, 9.96)
fit = lm(y ~ x)
hatvalues(fit)
```

The top graph in Figure 3.5 is a scatterplot of the data pairs and the associated regression line. From a cursory visual assessment, using a simple linear regression model to describe the relationship between X and Y seems reasonable for these data pairs. The leverages for the first three data pairs are identified on the graph. All three graphs in Figure 3.5 have the same horizontal and vertical scales for easier comparison.

The middle graph in Figure 3.5 includes all of the data values from the Anscombe's first data set, but adds the additional data pair $(19, 12.5)$, which was gleaned from Anscombe's fourth data set. The leverages are given in Table 3.4, with the leverage for the data pair $(19, 12.5)$ set in boldface because it has a leverage that exceeds $4/n = 4/12 \cong 0.33$. This data pair is a leverage point that warrants particular scrutiny. Although the data pair has the *ability* to exert unusual effect on the regression line, it is clear that the data point does not alter the regression line from where it was in the top graph. So although the new data pair is a leverage point (and is therefore circled in the middle graph), it does not contradict the existing trend from the other 11 points. In this sense, the leverage point provides some (scant) evidence that the scope of the model can be extended from $4 \leq X \leq 14$ to $4 \leq X \leq 19$.

The bottom graph in Figure 3.5 includes all of the data values from the Anscombe's first data set, but adds the additional data pair $(19, 4)$. Since the values of the independent variable have not changed, the leverages match those from Table 3.4. The leverage point $(19, 4)$ is circled on the graph. This leverage point exerts a significant downward tug on the right side of the regression line relative to the pattern established by the first 11 data pairs. A simple linear regression model is not appropriate in this case. There are several potential explanations for the deleterious effects of this leverage point.

i	1	2	3	4	5	6	7	8	9	10	11	12
X_i	4	5	6	7	8	9	10	11	12	13	14	19
h_{ii}	0.25	0.20	0.16	0.12	0.10	0.09	0.08	0.09	0.11	0.13	0.17	**0.50**

Table 3.4: Leverages for data set I in Anscombe's quartet with appended $X_{12} = 19$.

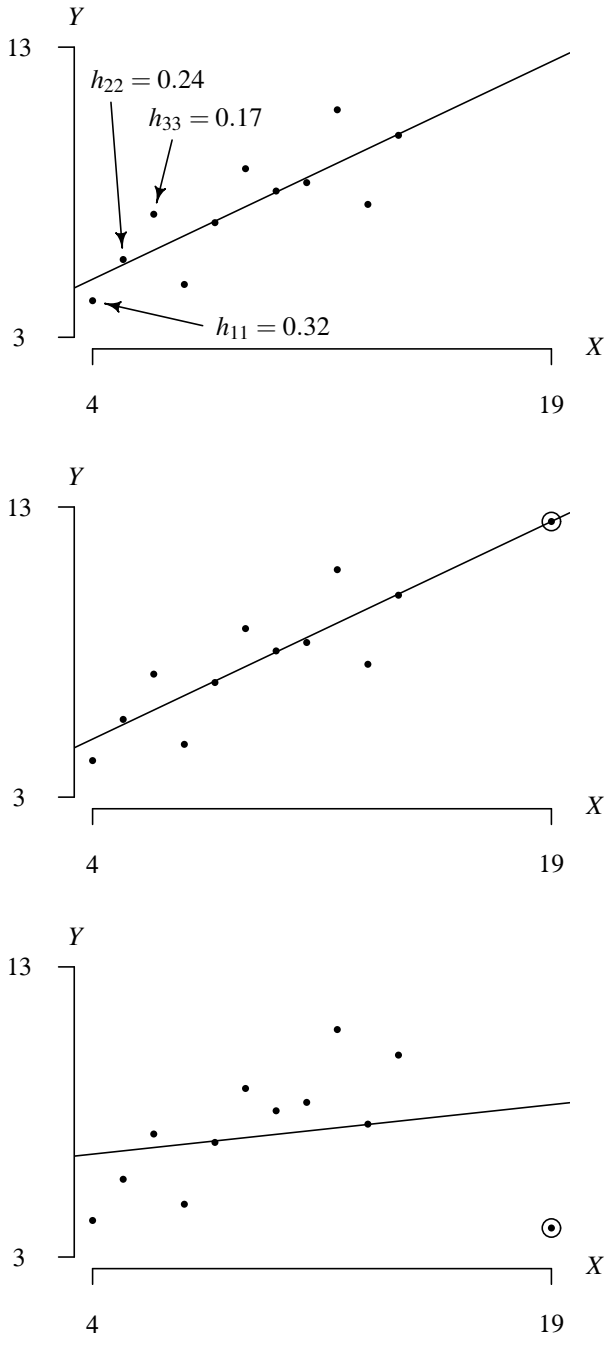

Figure 3.5: Fitted regression models and leverage points.

- The leverage point might have been incorrectly recorded.
- The leverage point might be fundamentally different than the others and does not belong in the data set.
- The leverage point might indicate that a nonlinear regression model is appropriate.
- The leverage point might signal that the scope of the model should be restricted to $4 \leq X \leq 14$, where a simple linear regression appears to be appropriate.
- The leverage point is legitimate and not fundamentally different than the others. It might just happen to be an extreme value. The linear model still might be appropriate, but more data pairs need to be collected to show that this is the case.

The previous example has indicated a fitted simple linear regression model is likely to pass close to a leverage point. Leverage points exert more tug on the regression line than those points whose independent variable value is closer to $\bar{X}$. The next illustration of identifying leverage points revisits the heights of couples from Example 2.7.

Example 3.4 Identify the leverage points for the $n = 96$ pairs of couples heights from Example 2.7.

The following R statements load the `PBImisc` package, set `x` to the heights of the wives, set `y` to the associated heights of the husbands, calculate the leverages using the `hat` function, store the indexes of those points whose leverage exceeds $4/n$ in the vector `i`, plot the data pairs using the `plot` function, plot the regression line using the `abline` function, and circle the leverage points using the `symbols` function.

```
library(PBImisc)
x         = heights$Wife
y         = heights$Husband
n         = length(x)
leverages = hat(x)
i         = leverages > 4 / n
m         = sum(i)
fit       = lm(y ~ x)
plot(x, y, pch = 16)
abline(fit$coefficients)
symbols(x[i], y[i], circles = rep(0.7, m), inches = FALSE, add = TRUE)
```

The resulting graph is displayed in Figure 3.6. There are a total of ten leverage points—seven on the left end of the scope of the model and three on the right end of the scope of the model. Examining each of the ten leverage points carefully, nine of the ten do not seem out of step with the rest of the data values. The leverage point $(147, 178)$, however, which corresponds to an unusually short wife marrying and fairly tall husband, is clearly a point that exerts a significant upward tug on the left side of the regression line. Assuming that the X and Y values were recorded correctly, there is no reason to remove this point from the data set. The impact of this point on the slope of the regression line is minimized by the large sample size.

Identifying leverage points is helpful for knowing which points to more carefully scrutinize. It is not appropriate to simply delete a leverage point because it falls far from the regression line. Leverage points can be helpful in highlighting an aspect of the model that was not originally considered

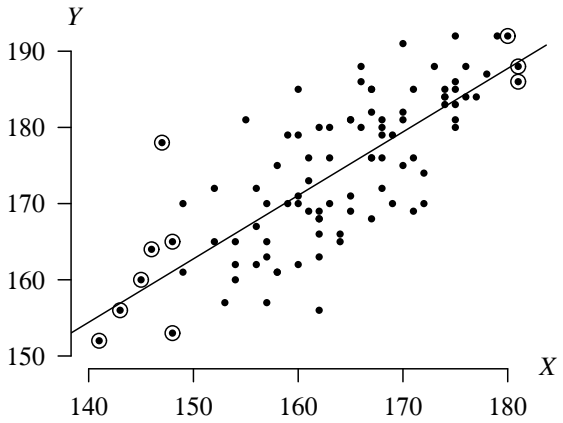

Figure 3.6: Fitted regression model and leverage points for the $n = 96$ data pairs.

relevant. The next subsection considers how to determine if a leverage point (or any other point) does produce a significant impact on $\hat{\beta}_0$ and $\hat{\beta}_1$.

3.2.2 Influential Points

Leverage points have the *potential* to produce large changes in the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ when they are deleted. How can we determine whether a leverage point (or any other point) *does* have significant impact on the regression line? American statistician R. Dennis Cook suggested a quantity that measures the influence of each data pair on the regression line.

Definition 3.3 For a simple linear regression model, Cook's distances $D_1, D_2, \ldots, D_n$ associated with the n data pairs have the following three equivalent definitions.

- $D_i = \dfrac{\sum_{j=1}^{n}\left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{2 \cdot MSE}$,
- $D_i = \dfrac{n\left(\hat{\beta}_{0(i)} - \hat{\beta}_0\right)^2 + 2\left(\hat{\beta}_{0(i)} - \hat{\beta}_0\right)\left(\hat{\beta}_{1(i)} - \hat{\beta}_1\right)\sum_{i=1}^{n} X_i + \left(\hat{\beta}_{1(i)} - \hat{\beta}_1\right)^2 \sum_{i=1}^{n} X_i^2}{2 \cdot MSE}$,
- $D_i = \dfrac{e_i^2 h_{ii}}{2 \cdot MSE\,(1 - h_{ii})^2}$,

where MSE is the mean square error (see Theorem 1.8), $\hat{Y}_{j(i)}$ is the fitted value of data pair j with data pair i removed, $\beta_{0(i)}$ is the estimated intercept of the regression line for the simple linear regression model with data pair i removed, $\beta_{1(i)}$ is the estimated slope of the regression line for the simple linear regression model with data pair i removed, and h_{ii} is the leverage of data pair i (see Definition 3.2), for $i = 1, 2, \ldots, n$.

The equivalence between the three very diverse formulas in Definition 3.3 is left as an exercise. The data pairs must not be collinear because MSE appears in the denominator of each formula. Each

of the three formulas is helpful in developing intuition about Cook's distance, so each is illustrated in the following three examples.

Example 3.5 Use the first formula from Definition 3.3 to calculate the Cook's distances for the $n = 11$ data pairs in the Anscombe's first data set (sorted by the values of the independent variable), appended with the point $(X_{12}, Y_{12}) = (19, 4)$. This was the last data set encountered in Example 3.3.

The bottom graph in Figure 3.5 shows that the first 11 data pairs are consistent with an underlying linear model, but the 12th data pair is not consistent with this model. The first formula from Definition 3.3 is

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{2 \cdot MSE}$$

for $i = 1, 2, \ldots, n$. Since the term $\hat{Y}_j - \hat{Y}_{j(i)}$ is a measure of the effect of dropping data pair i from the data set on the fitted value, larger values for D_i indicate that data pair i is more influential. Squaring $\hat{Y}_j - \hat{Y}_{j(i)}$ assures that the direction of the fitted value when data pair i is dropped makes a positive contribution to D_i. The R code below loops through the data points, excluding the data pairs one-by-one. Hence there will in general be a total of $n+1$ simple linear regression models fitted when using the first formula for computing Cook's distance—one regression model for all data pairs included and n other regression models for dropping each data pair once.

```
x      = c(4:14, 19)
y      = c(4.26, 5.68, 7.24, 4.82, 6.95, 8.81, 8.04, 8.33, 10.84,
           7.58, 9.96, 4)
n      = length(x)
fit    = lm(y ~ x)
mse    = sum(fit$residuals ^ 2) / (n - 2)
fitted = fit$fitted.values
cooks  = numeric(n)
for (i in 1:n) {
  fit.exclude    = lm(y[-i] ~ x[-i])
  beta0          = fit.exclude$coefficients[1]
  beta1          = fit.exclude$coefficients[2]
  fitted.exclude = beta0 + beta1 * x
  cooks[i]       = sum((fitted - fitted.exclude) ^ 2) / (2 * mse)
}
print(cooks)
```

Several of the Cook's distances are given in Table 3.5. Consistent with the bottom graph in Figure 3.5, the 12th Cook's distance $D_{12} = 3.621$ is substantially larger than

i	1	2	3	4	$\cdots$	11	12
D_i	0.236	0.029	0.005	0.069	$\cdots$	0.128	3.621

Table 3.5: Cook's distances for Anscombe's data set I with $(X_{12}, Y_{12}) = (19, 4)$ appended.

the second-largest Cook's distance $D_1 = 0.236$. So the 12th data pair, (X_{12}, Y_{12}), is the most influential point. The first data pair, (X_1, Y_1), is the second most influential point. Notice that these are the two points with the highest leverage (see Table 3.4).

To show some of the geometry associated with the calculation of $D_1, D_2, \ldots, D_n$, Figure 3.7 shows the regression line

$$Y = \hat{\beta}_0 + \hat{\beta}_1 X = 6.09 + 0.114X$$

fitted to all $n = 12$ data pairs, which are indicated by solid points ($\bullet$). This regression line corresponds to the fitted value at $X_{12} = 19$ of

$$\hat{Y}_{12} = 6.09 + (0.114)(19) = 8.25.$$

The other regression line,

$$Y = \hat{\beta}_{0(12)} + \hat{\beta}_{1(12)} X = 3.00 + 0.500X,$$

is the regression that excludes the influential 12th data pair $(X_{12}, Y_{12}) = (19, 4)$. This regression line corresponds to the fitted value at $X_{12} = 19$ of

$$\hat{Y}_{12(12)} = 3.00 + (0.500)(19) = 12.50.$$

The two fitted values are indicated by open points ($\circ$). So when calculating D_{12} using the first formula in Definition 3.3, one of the terms in the numerator is

$$\left(\hat{Y}_{12} - \hat{Y}_{12(12)}\right)^2 = (8.25 - 12.50)^2 = (-4.25)^2 = 18.07,$$

which makes a huge contribution to the numerator of D_{12}.

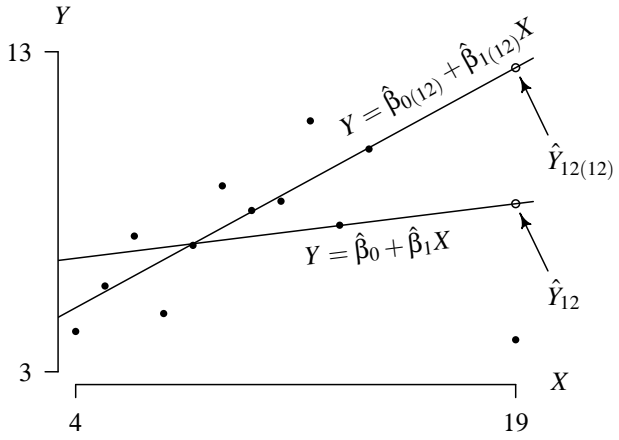

Figure 3.7: Calculating Cook's distances using fitted values.

The previous example has indicated that Cook's distance is a measure of the influence of each data pair based on the effect of removing each data pair sequentially, and measuring the associated impact on the fitted values. If the fitted values are not substantially altered by removing data pair i, then D_i will be small; if the fitted values are substantially altered by removing data pair i, then D_i will be large. This, however, does not explain why the denominator $2 \cdot MSE$ is in all four formulas in Definition 3.3. That will be addressed in the next example.

Example 3.6 Use the second formula from Definition 3.3 to calculate the Cook's distances for the $n = 11$ data pairs in the Anscombe's first data set (sorted by the values of the independent variable), appended with the point $(X_{12}, Y_{12}) = (19, 4)$.

The second formula for computing Cook's distance for data pair i from Definition 3.3 is

$$D_i = \frac{n\left(\hat{\beta}_{0(i)} - \hat{\beta}_0\right)^2 + 2\left(\hat{\beta}_{0(i)} - \hat{\beta}_0\right)\left(\hat{\beta}_{1(i)} - \hat{\beta}_1\right)\sum_{i=1}^n X_i + \left(\hat{\beta}_{1(i)} - \hat{\beta}_1\right)^2 \sum_{i=1}^n X_i^2}{2 \cdot MSE}$$

for $i = 1, 2, \ldots, n$. This formula emphasizes the change in the regression coefficients when data pair i is dropped. Figure 3.8 shows (a) the estimators $\left(\hat{\beta}_0, \hat{\beta}_1\right)$ for all $n = 12$ data pairs as a $+$, (b) the associated confidence regions for β_0 and β_1 at levels 0.25, 0.5, and 0.75, and (c) twelve points indicated by solid circles ($\bullet$) giving the values of the slope and intercept when data pair i is dropped, for $i = 1, 2, \ldots, n$. Not surprisingly, the estimated slope and intercept when the 12th data point, $(X_{12}, Y_{12}) = (19, 4)$, is dropped, strays the furthest from $\left(\hat{\beta}_0, \hat{\beta}_1\right)$. The other 11 estimated slope and intercept pairs all fall within the 0.25 confidence region.

The connection with the confidence region for β_0 and β_1 in this case illuminates why the $2 \cdot MSE$ appears in the denominator of all of the formulas for D_i in Definition 3.3.

Compare the right-hand side of the second formula in Definition 3.3 with the expression in Theorem 2.16. They are identical except that β_0 is replaced by $\beta_{0(i)}$ and β_1 is replaced by $\beta_{1(i)}$. So under the assumption that the data pairs are drawn from a simple linear regression model, one would expect that D_i is approximately $F(2, n-2)$. Some suggest using the median of a $F(2, n-2)$ distribution as a threshold for classifying a data pair as an influential point. Another approach is to observe that the population mean and variance of an $F(2, n-2)$ random variable are

$$E[D_i] = \frac{n-2}{n-4} \quad (\text{for } n > 4) \qquad \text{and} \qquad V[D_i] = \frac{(n-2)^3}{(n-4)^2(n-6)} \quad (\text{for } n > 6).$$

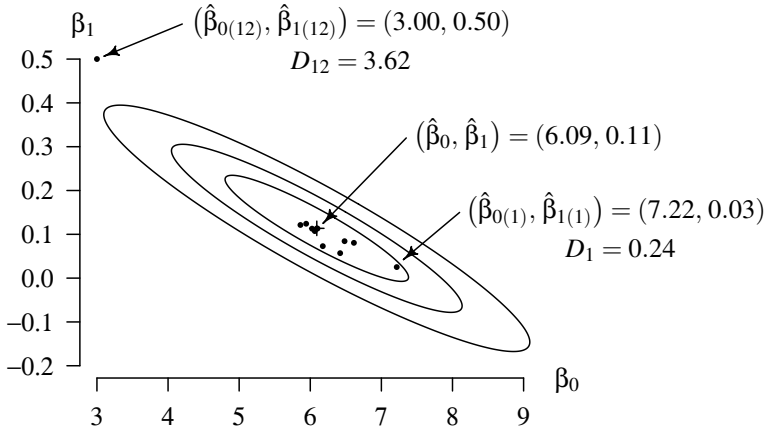

Figure 3.8: Calculating Cook's distances using the parameter estimates.

So in the limit as the number of data pairs increases,

$$\lim_{n\to\infty} E[D_i] = 1 \qquad \text{and} \qquad \lim_{n\to\infty} V[D_i] = 1.$$

It is for this reason that a threshold of 1 is used as a simple threshold for classifying a data point as influential based on Cook's distance. Regardless of whether the median of an $F(2,10)$ random variable (which is 0.743) or 1 is used as a threshold, the first 11 points are not deemed to be influential points, and the 12th point, $(19, 4)$, is deemed to be an influential point.

One weakness associated with the first two formulas for computing the Cook's distances in Definition 3.3 involves computation time. There are $n+1$ regression lines to estimate (one for all of the data pairs and then another n associated with dropping each of the data pairs). For large values of n, this can require significant computation time. The third formula is much faster, as illustrated next.

Example 3.7 Use the third formula from Definition 3.3 to calculate the Cook's distances for the $n = 96$ data pairs in the data set of heights of wives and husbands from Example 2.7.

The third formula for computing Cook's distance for data pair i from Definition 3.3 is

$$D_i = \frac{e_i^2 h_{ii}}{2 \cdot MSE\,(1-h_{ii})^2}$$

for $i = 1, 2, \ldots, n$. The advantage to using this formula over the other two formulas is that it only requires one regression line to be calculated, rather than $n+1$ regression lines in the other two formulas. This is a substantial time savings for large values of n. The R code below calculates Cook's distances for the heights data.

```
library(PBImisc)
x     = heights$Wife
y     = heights$Husband
n     = length(x)
fit   = lm(y ~ x)
mse   = sum(fit$residuals ^ 2) / (n - 2)
lev   = hat(x)
cooks = fit$residuals ^ 2 * lev / (2 * mse * (1 - lev) ^ 2)
plot(cooks)
```

The $n = 96$ Cook's distances are plotted in Figure 3.9. The 12th data pair, which is $(X_{12}, Y_{12}) = (147, 178)$, has a spectacular Cook's distance of $D_{12} = 0.192$. Since this does not exceed the first threshold (which is the median of an F random variable with 2 and 94 degrees of freedom: 0.698) or the second threshold (which is 1 using the asymptotic result), we conclude that there are no influential points. Cook's distances are calculated so frequently in regression analysis that R includes a function named `cooks.distance` that calculates the Cook's distances, as illustrated below.

```
library(PBImisc)
x   = heights$Wife
```

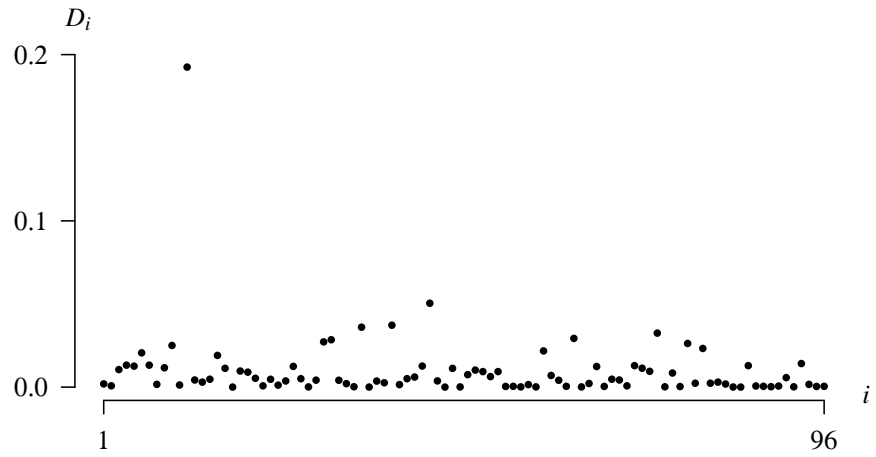

Figure 3.9: Cook's distances for the `heights` data.

```
y   = heights$Husband
fit = lm(y ~ x)
cooks.distance(fit)
```

Cook's distances are effective for identifying influential points. Once an influential point in a simple linear regression model has been identified, there are several possible next steps.

- The influential point might have been recorded or coded improperly; a typographical error has occurred. In most situations, this is easily remedied.
- The influential point has some unusual characteristic that is not present with the other data points that might account for it being deemed influential. Depending on the setting, the influential point can be removed and the regression model can be refitted without the influential point.
- The influential point might provide some evidence that an alternative regression model is appropriate. This might be a nonlinear regression model or a linear regression model with additional independent variables.
- The influential point might be at one of the extremes of the scope of the model. This might indicate that the scope of the model is too wide; narrowing the scope should be considered. It is often the case that a simple linear regression model is valid only over a rather limited scope. This might result in eliminating all data points outside of the narrowed scope and refitting the simple linear regression model.
- The high-leverage point is indeed within the scope of the model and was recorded correctly, but its extreme influence on the regression line is resulting in poor diagnostic measures. One approach here is to collect more data values, particularly at the extreme values of the independent variable within the scope of the model in order to mitigate the effect of the influential point.

3.3 Remedial Procedures

The diagnostic procedures presented in the previous section are designed to *identify* assumptions associated with the simple linear regression model that are not satisfied for a particular set of n data pairs. But these diagnostic procedures do not suggest *remedies* when model assumptions are not satisfied. This section considers remedial procedures.

Reasons that simple linear regression model with normal error terms can fail to satisfy the assumptions given in Definition 2.1 include

- the regression function is not linear,
- the regression model has not included an important independent variable,
- the error terms have a variance that varies with X,
- the error terms are not independent,
- the error terms are not normally distributed,
- the scope of the regression model is too wide,
- the scope of the regression model is too narrow, and
- an influential point has an unusually strong effect on the regression line.

Two common approaches to handling a regression model which violates one or more of the assumptions are (a) formulate and fit a regression model with nonlinear terms, and (b) transform the X-values or the Y-values (or both) in a fashion so that the simple linear regression assumptions are satisfied. Regression models with nonlinear terms will be considered in a subsequent section in this chapter; transformations will be considered here. Transformations will be illustrated in a single (long) example.

Example 3.8 A simple linear regression model with normal error terms for the speed of a car X (in miles per hour) versus the stopping distance Y (in feet) for the built-in R `cars` data set was abandoned in Example 2.8 for several reasons. A scatterplot (without jittering) with the associated regression line is displayed in Figure 3.10. The purpose of this example is to see whether a transformation can overcome the problems associated with

- the relationship between X and Y appears to be slightly nonlinear,
- the variance of the error terms appears to be increasing in X, and
- the residuals do not appear to be normally distributed.

Rather than providing a complete inventory of all possible patterns and associated potential helpful transformations, four transformations will be illustrated here. This trial-and-error approach is not what is typically relied on in practice. There are some patterns associated with data pairs that tend to give clues as to which transformations will be effective.

The first transformation is $X' = X^2$. The R code below implements the transformation, generates a scatterplot of the transformed data pairs, and plots the associated regression line.

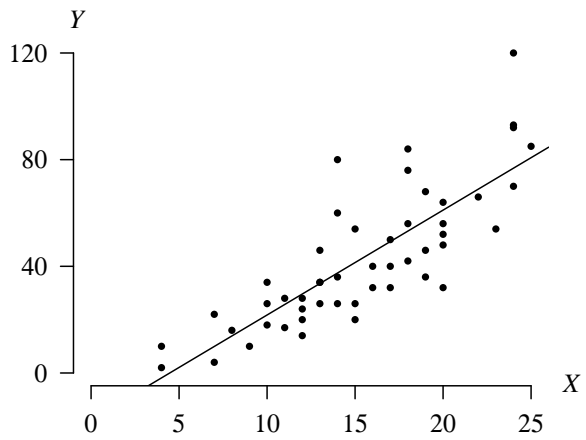

Figure 3.10: Scatterplot and regression line of speed X and stopping distance Y.

```
x = cars$speed ^ 2
y = cars$dist
plot(x, y)
abline(lm(y ~ x)$coefficients)
```

This scatterplot appears in the upper-left graph in Figure 3.11. Tick mark labels have been suppressed on these graphs because the interest is in gazing at the data pairs in order to determine whether the transformed data pairs conform to the simple linear regression model with normal error terms. For the transformation $X' = X^2$, little progress is made on the constant variance issue. The first 19 data pairs, which are associated with speeds from 4 to 13 miles per hour, seem to have a smaller variance in their stopping distances than the faster speeds. This transformation is deemed ineffective.

The second transformation is $Y' = \ln Y$. The R code below implements the transformation, generates a scatterplot of the transformed data pairs, and plots the associated regression line.

```
x = cars$speed
y = log(cars$dist)
plot(x, y)
abline(lm(y ~ x)$coefficients)
```

This scatterplot appears in the upper-right graph in Figure 3.11. The transformation $Y' = \ln Y$ also results in a nonconstant variance in the error terms; this time the variance in the stopping distances is greater for the slower speeds. So this transformation is also abandoned for lack of constant variance of the error terms.

The third transformation is $Y' = \sqrt{Y}$. The R code below implements the transformation, generates a scatterplot of the transformed data pairs, and plots the associated regression line.

```
x = cars$speed
```

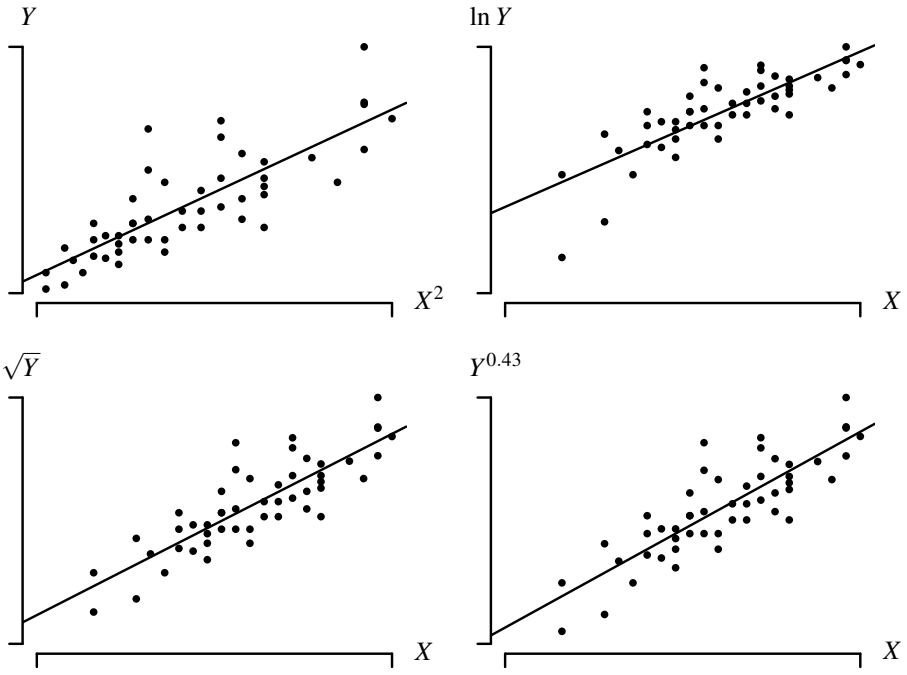

Figure 3.11: Scatterplots and estimated regression lines for transformed `cars` data.

```
y = sqrt(cars$dist)
plot(x, y)
abline(lm(y ~ x)$coefficients)
```

This scatterplot appears in the lower-left graph in Figure 3.11. The transformation $Y' = \sqrt{Y}$ is the first to show some promise for the use of the simple linear regression model with normal error terms. The variance of the error terms appears to be constant over the scope of the model. There is nothing magical, however, about the $1/2$ power in the transformation $Y' = \sqrt{Y} = Y^{1/2}$. Might the cube root be a superior transformation to the square root? This prompts a fourth transformation, which is $Y' = Y^{\lambda}$, and is known as the Box–Cox transformation, named after British statisticians George Box and David Cox. They suggested a similar transformation in 1964, which is

$$Y' = \frac{Y^{\lambda} - 1}{\lambda},$$

and the fitting of the λ parameter by maximum likelihood estimation can be performed by the `boxcox` function in the `MASS` package in R.

So the fourth transformation is $Y' = (Y^{\lambda} - 1)/\lambda$. The R code below calculates the maximum likelihood estimator of λ, implements the transformation, generates a scatterplot of the transformed data pairs, and plots the associated regression line. The `boxcox` function generates the log likelihood function for estimating λ, and the `which.max` function extracts the maximum likelihood estimator.

```
library(MASS)
x      = cars$speed
y      = cars$dist
bc     = boxcox(y ~ x, plotit = FALSE, lambda = seq(0, 1, by = 0.01))
lambda = bc$x[which.max(bc$y)]
y      = (y ^ lambda - 1) / lambda
plot(x, y)
abline(lm(y ~ x)$coefficients)
```

The log likelihood function and an associated 95% confidence interval for λ is generated by setting the `plotit` argument to `FALSE` in the call to `boxcox`. This confidence interval includes $\lambda = 1/2$. The maximum likelihood estimator $\hat{\lambda} = 0.43$ falls between a square root and cube root transformation. This scatterplot appears in the lower-right graph in Figure 3.11, and is very similar to the square root transformation; either would work fine for this data set. Since the last two scatterplots and associated regression lines are nearly identical, we move forward with the transformation $Y' = \sqrt{Y}$. So the tentative fitted model is

$$E\left[\sqrt{Y}\right] = 1.28 + 0.322X$$

where the regression coefficients $\beta_0' = 1.28$ and $\beta_1' = 0.322$ are calculated with the R statement

```
lm(sqrt(cars$dist) ~ cars$speed)$coefficients
```

The next step is to assess the aptness of the model by examining the residuals. The four graphs (read row-wise) in Figure 3.12 are (*a*) the residuals associated with the transformed model $\sqrt{Y} = 1.28 + 0.322X$ plotted against their index, (*b*) the standardized residuals $e_i/\sqrt{MSE}$ associated with the transformed model plotted against the value of the independent variable X_i, (*c*) a histogram of the standardized residuals $e_i/\sqrt{MSE}$ for the transformed model, and (*d*) a QQ plot of the standardized residuals $e_i/\sqrt{MSE}$ for the transformed model with theoretical quantiles on the horizontal axis and sample quantiles on the vertical axis. Although there is some nonsymmetry in the histogram of the residuals (which might be due to the binning of the 50 data pairs), the residual plots and the QQ plot make the simple linear regression model with normal error terms for the transformed data pairs seem plausible. A roughly mound-shaped histogram is typically adequate for the normality assumption. Moving from the visual assessment to statistical tests, the R code

```
x = cars$speed
y = sqrt(cars$dist)
fit = lm(y ~ x)
shapiro.test(fit$residuals)
max(cooks.distance(fit))
```

gives a p-value for the Shapiro–Wilk test of $p = 0.314$. This is a big improvement over the p-value obtained in Example 2.8, which rejected normality with $p = 0.0215$. The transformation is effective. The largest Cook's distance is 0.134, which occurs at the 49th observation $(X_{49}, Y_{49}) = (24, 120)$. Returning to the 49th observation in

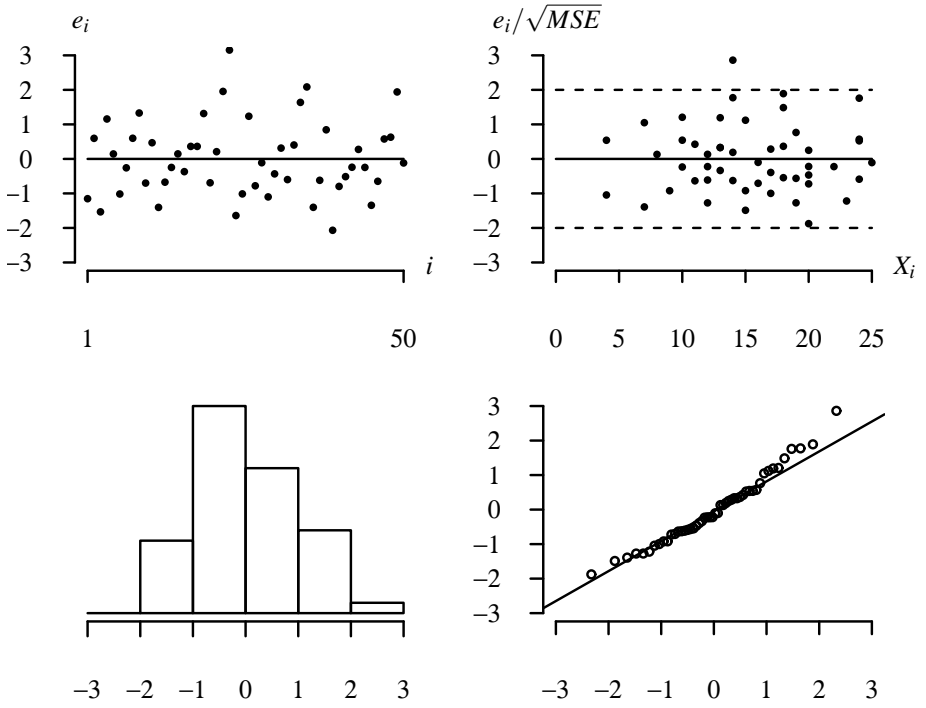

Figure 3.12: Visual assessment of the residuals of the transformed model.

Figure 3.12, we see that it achieves the largest Cook's distance because of its leverage, but does not appear to be inconsistent with the transformed model.

So the visual assessment and statistical tests lead us to believe that a simple linear regression model with normal error terms for the transformed data is appropriate. The fitted regression model is

$$E\left[\sqrt{Y}\right] = 1.28 + 0.322X.$$

All of the statistical inference techniques can now be applied to the transformed data. For example, confidence intervals for the β_0' and β_1' (the intercept and slope of the regression line for the transformed data) can be calculated with the R statements

```
x   = cars$speed
y   = sqrt(cars$dist)
fit = lm(y ~ x)
confint(fit)
```

which give the 95% confidence intervals

$$0.303 < \beta_0' < 2.25 \qquad \text{and} \qquad 0.263 < \beta_1' < 0.382.$$

Figure 3.13 displays all of the exact two-sided 95% confidence intervals for $E[\sqrt{Y_h}]$ and all of the exact two-sided 95% prediction intervals for $\sqrt{Y_h^*}$ for all values of X_h in

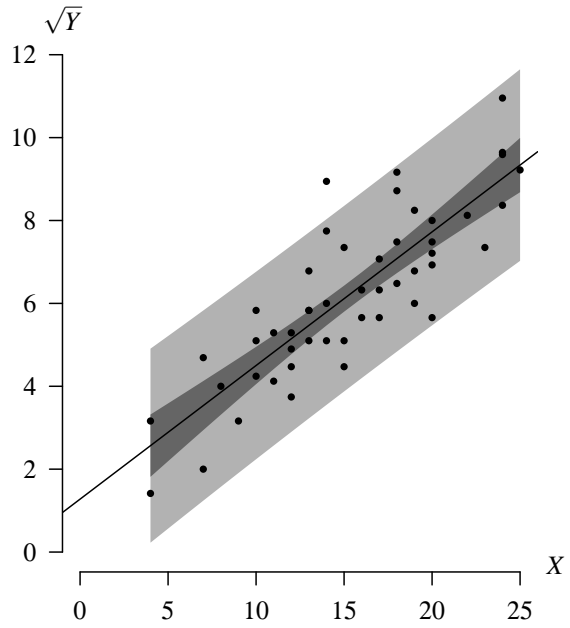

Figure 3.13: Transformed `cars` model 95% confidence and prediction intervals.

the scope of the regression model. For $X_h = 21$ miles per hour, for example, an exact two-sided 95% prediction interval for $\sqrt{Y_h^*}$ is

$$5.78 < \sqrt{Y_h^*} < 10.3,$$

which can be calculated with the R commands

```
x   = cars$speed
y   = sqrt(cars$dist)
fit = lm(y ~ x)
predict(fit, data.frame(x = 21), interval = "prediction")
```

So to translate this back to the original units, for a 51st car going $X_h = 21$ miles per hour, the expected stopping distance using the transformed model is

$$\hat{Y}_h = \left(1.28 + 0.322 \cdot 21\right)^2 = 64.8$$

feet, and an exact two-sided 95% prediction interval for the associated stopping distance is

$$33.5 < Y_h^* < 106.$$

The previous example took a trial-and-error approach to determining an appropriate transformation to apply to the raw data pairs in order to satisfy the assumptions implicit in a simple linear regression model with normal error terms. There are templates that can give a more systematic approach to determining these transformations.

There is a nice synergy between matrix algebra and regression, which will be presented in the next section.

3.4 Matrix Approach to Simple Linear Regression

So far, a purely algebraic approach has been taken to simple linear regression modeling. This section considers a matrix-based approach. There are (at least) four reasons to take this approach. First, the mathematical expressions are in many cases much more compact; summations from the algebraic approach are often equivalent to matrix multiplications. Second, matrix algebra can easily be implemented on a computer. Third, the matrix approach generalizes very easily to the multiple regression case in which there are several independent variables. Fourth, the matrix approach generalizes very easily to weighted least squares, which will be introduced in the next section.

We begin the matrix approach by defining certain critical matrices, which will be set in boldface. Let $\mathbf{X}$ be an $n \times 2$ matrix whose first column is all ones and whose second column contains the observed values of the independent variable, $\mathbf{Y}$ be an $n \times 1$ vector which holds the observed values of the dependent variable, $\boldsymbol{\beta}$ be a 2×1 vector which holds the population intercept and slope, and $\boldsymbol{\varepsilon}$ be an $n \times 1$ vector which holds the error terms:

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

The $\mathbf{X}$ matrix is known as the *design matrix*.

As before, the values of the independent variable (the second column of $\mathbf{X}$) are assumed to be fixed constants observed without error with at least two distinct values, the values of the dependent variable contained in $\mathbf{Y}$ are assumed to be continuous random responses, and the elements of the vector $\boldsymbol{\varepsilon}$ are assumed to be mutually independent random variables, each with population mean 0 and finite positive population variance σ^2. Stated another way, the expected value of $\boldsymbol{\varepsilon}$ is the zero vector and the variance–covariance matrix of $\boldsymbol{\varepsilon}$ is

$$\begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix}.$$

The simple linear regression model

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

for $i = 1, 2, \ldots, n$, can be written more explicitly in terms of each observed data pair as

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 X_1 + \varepsilon_1 \\ Y_2 &= \beta_0 + \beta_1 X_2 + \varepsilon_2 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 X_n + \varepsilon_n \end{aligned}$$

which, in matrix form, is

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or simply

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

This explains why the artificial column of ones appears as the first column of the $\mathbf{X}$ matrix; it is to account for the intercept term. To force a regression line through the origin, simply omit the column of ones in the $\mathbf{X}$ matrix. Taking the expected value of both sides of this equation results in

$$E[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$$

because $E[\varepsilon_i] = 0$, for $i = 1, 2, \ldots, n$, (that is, $E[\boldsymbol{\varepsilon}] = \mathbf{0}$). The left-hand side of this equation, $E[\mathbf{Y}]$, is an n-element column vector with elements $E[Y_1], E[Y_2], \ldots, E[Y_n]$. The sum of squares which is to be minimized to find the least squares estimators is

$$S = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

With this notation established, the algebraic results concerning the simple linear regression model can be restated more compactly in terms of these matrices. The results have already been proved, so there is no need to prove them again when stated in matrix form. The $'$ superscript denotes transpose. It is a good exercise to perform the algebra necessary to see that the algebraic and matrix versions of these definitions and theorems match. The dimensions of the matrices should be checked for conformity.

- **Definition 1.1.** The *simple linear regression model* is

  $$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

  where $E[\boldsymbol{\varepsilon}] = \mathbf{0}$, $V[\boldsymbol{\varepsilon}] = \sigma^2\mathbf{I}$, and $\mathbf{I}$ is the $n \times n$ identity matrix.

- **Theorem 1.1.** The least squares estimators of $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}} = \left(\hat{\beta}_0, \hat{\beta}_1\right)'$, solve the normal equations

  $$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y}.$$

  The $\mathbf{X}$ matrix has rank 2 because there are at least two distinct X_i values. So $\mathbf{X}'\mathbf{X}$ is invertible and the normal equations have the unique solution

  $$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y},$$

  by premultiplying both sides of the normal equations by $\left(\mathbf{X}'\mathbf{X}\right)^{-1}$.

- **Theorem 1.2.** The least squares estimator of $\boldsymbol{\beta}$ in a simple linear regression model is an unbiased estimator of $\boldsymbol{\beta}$ because

  $$E\left[\hat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}.$$

- **Theorem 1.3.** The least squares estimators of $\boldsymbol{\beta}$ in the simple linear regression model can be written as linear combinations of the dependent variables:

  $$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y},$$

  where the coefficients in the linear combinations are given by $\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$.

- **Theorem 1.4.** The variance–covariance matrix of the least squares estimators of $\boldsymbol{\beta}$ is

  $$\sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}.$$

- **Theorem 1.5** (Gauss–Markov theorem). The least squares estimators of $\boldsymbol{\beta}$ in a simple linear regression model, $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, have the smallest population variance amongst all linear unbiased estimators of $\boldsymbol{\beta}$.

- **Definition 1.2.** The vector of *fitted values* in a simple linear regression model is the $n \times 1$ column vector

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y},$$

  which is a linear combination of the dependent variables. The vector of *residuals* is the $n \times 1$ column vector

$$\begin{aligned}\mathbf{e} &= \mathbf{Y} - \hat{\mathbf{Y}} \\ &= \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}} \\ &= \mathbf{Y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= \left(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\mathbf{Y},\end{aligned}$$

  which is also a linear combination of the dependent variables. The matrix $\mathbf{I}$ is the $n \times n$ identity matrix.

- **Theorem 1.6.** For the simple linear regression model with fitted values $\hat{\mathbf{Y}}$ and residuals $\mathbf{e}$,

  - $\mathbf{e}'\mathbf{1} = 0$,
  - $\mathbf{Y}'\mathbf{1} = \hat{\mathbf{Y}}'\mathbf{1}$
  - $\hat{\mathbf{Y}}'\mathbf{e} = 0$,

  where $\mathbf{1}$ is an n-element column vector of ones.

- **Theorem 1.7.** An unbiased estimator of σ^2 in a simple linear regression model is

$$\hat{\sigma}^2 = MSE = \frac{\mathbf{e}'\mathbf{e}}{n-2}.$$

- **Theorem 1.8.** The sums of squares can be partitioned in a simple linear regression model as $SST = SSR + SSE$ or

$$\left(\mathbf{Y} - \bar{\mathbf{Y}}\right)'\left(\mathbf{Y} - \bar{\mathbf{Y}}\right) = \left(\hat{\mathbf{Y}} - \bar{\mathbf{Y}}\right)'\left(\hat{\mathbf{Y}} - \bar{\mathbf{Y}}\right) + \left(\mathbf{Y} - \hat{\mathbf{Y}}\right)'\left(\mathbf{Y} - \hat{\mathbf{Y}}\right),$$

  where $\bar{\mathbf{Y}}$ is an n-element column vector with identical elements which are each the sample mean of the values of the dependent variable.

- **Definition 1.3.** The *coefficient of determination* in a simple linear regression model is

$$R^2 = \frac{SSR}{SST} = \frac{\left(\hat{\mathbf{Y}} - \bar{\mathbf{Y}}\right)'\left(\hat{\mathbf{Y}} - \bar{\mathbf{Y}}\right)}{\left(\mathbf{Y} - \bar{\mathbf{Y}}\right)'\left(\mathbf{Y} - \bar{\mathbf{Y}}\right)},$$

  when $\left(\mathbf{Y} - \bar{\mathbf{Y}}\right)'\left(\mathbf{Y} - \bar{\mathbf{Y}}\right) \neq 0$. The *coefficient of correlation* is

$$r = \pm\sqrt{R^2},$$

  where the sign associated with r is positive when $\hat{\beta}_1 \geq 0$ and negative when $\hat{\beta}_1 < 0$.

- **Definition 2.1.** The *simple linear regression model with normal error terms* is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim N\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$.

- **Theorem 2.1.** For the simple linear regression model with normal error terms, the maximum likelihood estimators of $\boldsymbol{\beta}$ are

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

and the maximum likelihood estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}\left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)'\left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right).$$

Since the vector of error terms $\boldsymbol{\varepsilon}$ consists of independent and identically distributed normal random variables, $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is a vector of independent and identically distributed normal random variables, and the linear transformation $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ has normally distributed elements.

- **Theorem 2.2.** For the simple linear regression model with normal error terms,

$$\frac{\mathbf{e}'\mathbf{e}}{\sigma^2} \sim \chi^2(n-2),$$

and is independent of $\hat{\boldsymbol{\beta}}$.

- **Theorem 2.3.** For the simple linear regression model with normal error terms, an exact two-sided $100(1-\alpha)\%$ confidence interval for σ^2 is

$$\frac{\mathbf{e}'\mathbf{e}}{\chi^2_{n-2,\alpha/2}} < \sigma^2 < \frac{\mathbf{e}'\mathbf{e}}{\chi^2_{n-2,1-\alpha/2}}.$$

- **Theorems 2.4 and 2.7.** For the simple linear regression model with normal error terms,

$$\hat{\boldsymbol{\beta}} \sim N\left(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\right).$$

- **Theorem 2.12.** For the simple linear regression model with normal error terms, an exact two-sided $100(1-\alpha)\%$ confidence interval for $E[Y_h]$ for a given value of the independent variable X_h is

$$\mathbf{X}_h'\hat{\boldsymbol{\beta}} - t_{n-2,\alpha/2}\sqrt{\hat{\sigma}^2\mathbf{X}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h} < E[Y_h] < \mathbf{X}_h'\hat{\boldsymbol{\beta}} + t_{n-2,\alpha/2}\sqrt{\hat{\sigma}^2\mathbf{X}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h},$$

where $\mathbf{X}_h = (1, X_h)'$ and $\hat{\sigma}^2 = MSE$.

- **Theorem 2.15.** For the simple linear regression model with normal error terms, an exact two-sided $100(1-\alpha)\%$ prediction interval for $Y_h^\star$ for a given value of the independent variable X_h is

$$\mathbf{X}_h'\hat{\boldsymbol{\beta}} - t_{n-2,\alpha/2}\sqrt{\hat{\sigma}^2\left(1 + \mathbf{X}_h'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}_h\right)} < Y_h^\star < \mathbf{X}_h'\hat{\boldsymbol{\beta}} + t_{n-2,\alpha/2}\sqrt{\hat{\sigma}^2\left(1 + \mathbf{X}_h'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}_h\right)},$$

where $\mathbf{X}_h = (1, X_h)'$ and $\hat{\sigma}^2 = MSE$.

- **Theorem 2.16.** Under the simple linear regression model with normal error terms and parameters estimated from the data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$,

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})' \mathbf{X}' \mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{2 \cdot MSE} \sim F(2, n-2).$$

- **Theorem 2.17.** Under the simple linear regression model with normal error terms and parameters estimated from the data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, the values of β_0 and β_1 satisfying

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})' \mathbf{X}' \mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{2 \cdot MSE} \leq F_{2, n-2, \alpha}$$

  form an exact joint $100(1-\alpha)\%$ confidence region for β_0 and β_1.

- **Definition 3.2.** Under the simple linear regression model, the *hat matrix* is

$$\mathbf{H} = \mathbf{X} \left(\mathbf{X}' \mathbf{X} \right)^{-1} \mathbf{X}'.$$

  The diagonal elements of the hat matrix are the *leverages*. The matrix equation

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$$

  indicates that $\mathbf{H}$ transforms $\mathbf{Y}$ to $\hat{\mathbf{Y}}$. The hat matrix is symmetric (that is, $\mathbf{H} = \mathbf{H}'$) and idempotent (that is, $\mathbf{HH} = \mathbf{H}$).

The matrix approach applied to a simple linear regression model is illustrated for a small sample size next.

Example 3.9 Consider again the sales data set from Example 1.3. Let the independent variable X be the *number* of sales per week that Cheryl completes. Each sale results in a random amount of revenue to the company that can be attributed to Cheryl. Let the dependent random variable Y be the associated total revenue to the company from the sales attributed to Cheryl for that week, in thousands of dollars. The data pairs for the past $n = 3$ weeks are

$$(X_1, Y_1) = (6, 2), \qquad (X_2, Y_2) = (8, 9), \qquad \text{and} \qquad (X_3, Y_3) = (2, 2).$$

Use the matrix approach to simple linear regression to define the matrices $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\beta}$, and $\boldsymbol{\varepsilon}$. Calculate the least squares estimates of the population intercept β_0 and population slope β_1, the fitted values, the hat matrix, the residuals, the unbiased estimate of the variance of the error terms, SST, SSR, SSE, R^2, r, an exact 95% confidence interval for $E[Y_h]$ when $X_h = 5$ weekly sales, and an exact 95% prediction interval for $Y_h^\star$ when $X_h = 5$ weekly sales using the matrix approach to simple linear regression.

The $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\beta}$, and $\boldsymbol{\varepsilon}$ matrices associated with the $n = 3$ data pairs are

$$\mathbf{X} = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 1 & 2 \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} 2 \\ 9 \\ 2 \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}.$$

The R code below uses the matrix approach to simple linear regression to calculate the estimate of the intercept $\hat{\beta}_0$, the estimate of the slope $\hat{\beta}_1$, the fitted values $\hat{\mathbf{Y}}$, the hat

matrix **H**, the residuals **e**, and the estimate of the population variance of the error terms $\hat{\sigma}^2$. *SST*, *SSR*, *SSE*, R^2, r, an exact 95% confidence interval for $E[Y_h]$ when $X_h = 5$, and an exact 95% prediction interval for $Y_h^\star$ when $X_h = 5$ using the matrix approach to simple linear regression. The `t` function computes a matrix transpose, the `diag` function creates an identity matrix, and the `solve` function computes the inverse of $X'X$. The matrix multiplication operator is `%*%`.

```
x      = c(6, 8, 2)
y      = c(2, 9, 2)
x      = cbind(1, x)
beta   = solve(t(x) %*% x) %*% t(x) %*% y
yhat   = x %*% beta
H      = x %*% solve(t(x) %*% x) %*% t(x)
n      = length(y)
e      = (diag(n) - H) %*% y
sighat = (t(e) %*% e) / (n - 2)
ybar   = rep(mean(y), n)
sst    = t(y - ybar) %*% (y - ybar)
ssr    = t(yhat - ybar) %*% (yhat - ybar)
sse    = t(y - yhat) %*% (y - yhat)
R2     = ssr / sst
r      = sign(beta[2]) * sqrt(R2)
alpha  = 0.05
conf1  = c(sum(e ^ 2) / qchisq(1 - alpha / 2, n - 2),
           sum(e ^ 2) / qchisq(alpha / 2, n - 2))
xh     = matrix(c(1, 5), 2, 1)
half2  = qt(1 - alpha / 2, n - 2) *
         sqrt(sse / (n - 2) * t(xh) %*% solve(t(x) %*% x) %*% xh)
conf2  = c(t(xh) %*% beta - half2, t(xh) %*% beta + half2)
half3  = qt(1 - alpha / 2, n - 2) *
         sqrt(sse / (n - 2) * (1 + t(xh) %*% solve(t(x) %*% x) %*% xh))
conf3  = c(t(xh) %*% beta - half3, t(xh) %*% beta + half3)
```

The output of this code is given in the equations that follow. The least squares estimators of the intercept and slope of the regression line are

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y} = \begin{bmatrix} 3 & 16 \\ 16 & 104 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 \\ 6 & 8 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 9 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

The fitted values are

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 1 \end{bmatrix}.$$

The 3×3 hat matrix **H** is

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}' = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 & 16 \\ 16 & 104 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 \\ 6 & 8 & 2 \end{bmatrix} = \begin{bmatrix} 5/14 & 3/7 & 3/14 \\ 3/7 & 5/7 & -1/7 \\ 3/14 & -1/7 & 13/14 \end{bmatrix}.$$

The diagonal elements of the hat matrix are the leverages h_{11}, h_{22}, h_{33}. The vector of residuals is

$$\mathbf{e} = (\mathbf{I}-\mathbf{H})\mathbf{Y} = \begin{bmatrix} 9/14 & -3/7 & -3/14 \\ -3/7 & 2/7 & 1/7 \\ -3/14 & 1/7 & 1/14 \end{bmatrix} \begin{bmatrix} 2 \\ 9 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 2 \\ 1 \end{bmatrix}.$$

The fitted values and the residuals computed here are consistent with the geometry shown in Figure 1.15 from Example 1.8. The unbiased estimate of the population variance of the error terms is

$$\hat{\sigma}^2 = MSE = \frac{\mathbf{e}'\mathbf{e}}{n-2} = \frac{1}{3-2} \begin{bmatrix} -3 & 2 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 2 \\ 1 \end{bmatrix} = 14.$$

The sums of squares can be partitioned as $SST = SSR + SSE$ using

$$\left(\mathbf{Y}-\bar{\mathbf{Y}}\right)'\left(\mathbf{Y}-\bar{\mathbf{Y}}\right) = \left(\hat{\mathbf{Y}}-\bar{\mathbf{Y}}\right)'\left(\hat{\mathbf{Y}}-\bar{\mathbf{Y}}\right) + \left(\mathbf{Y}-\hat{\mathbf{Y}}\right)'\left(\mathbf{Y}-\hat{\mathbf{Y}}\right),$$

where $\bar{\mathbf{Y}}$ is an n-element column vector with identical elements which are each the sample mean of the values of the dependent variable. For the $n = 3$ data pairs, this becomes

$$\left(-\frac{7}{3}\right)^2 + \left(\frac{14}{3}\right)^2 + \left(-\frac{7}{3}\right)^2 = \left(\frac{2}{3}\right)^2 + \left(\frac{8}{3}\right)^2 + \left(-\frac{10}{3}\right)^2 + (-3)^2 + 2^2 + 1^2$$

or

$$\frac{98}{3} = \frac{56}{3} + 14.$$

Figure 3.14 show the geometry associated with $SST = SSR + SSE$ for the three data pairs. The sum of the areas of the three squares in the top graph is SST; the sum of the areas of the three squares in the middle graph is SSR; the sum of the areas of the three squares in the bottom graph is SSE.

The coefficient of determination and the correlation coefficient in a simple linear regression model are

$$R^2 = \frac{SSR}{SST} = \frac{\left(\hat{\mathbf{Y}}-\bar{\mathbf{Y}}\right)'\left(\hat{\mathbf{Y}}-\bar{\mathbf{Y}}\right)}{\left(\mathbf{Y}-\bar{\mathbf{Y}}\right)'\left(\mathbf{Y}-\bar{\mathbf{Y}}\right)} = \frac{56/3}{98/3} = \frac{4}{7} = 0.57 \qquad \text{and} \qquad r = 0.76.$$

The three intervals are

$$2.8 < \sigma^2 < 14000,$$

$$-24 < E[Y_h] < 32,$$

and

$$-51 < Y_h^\star < 59.$$

The intervals are unusually wide because there are only $n = 3$ data pairs which have significant deviation from the regression line. Notice that these results match those obtained earlier by algebraic methods and by using the `lm` (linear model) function as given in Examples 1.3, 1.7, 1.8, and 1.10.

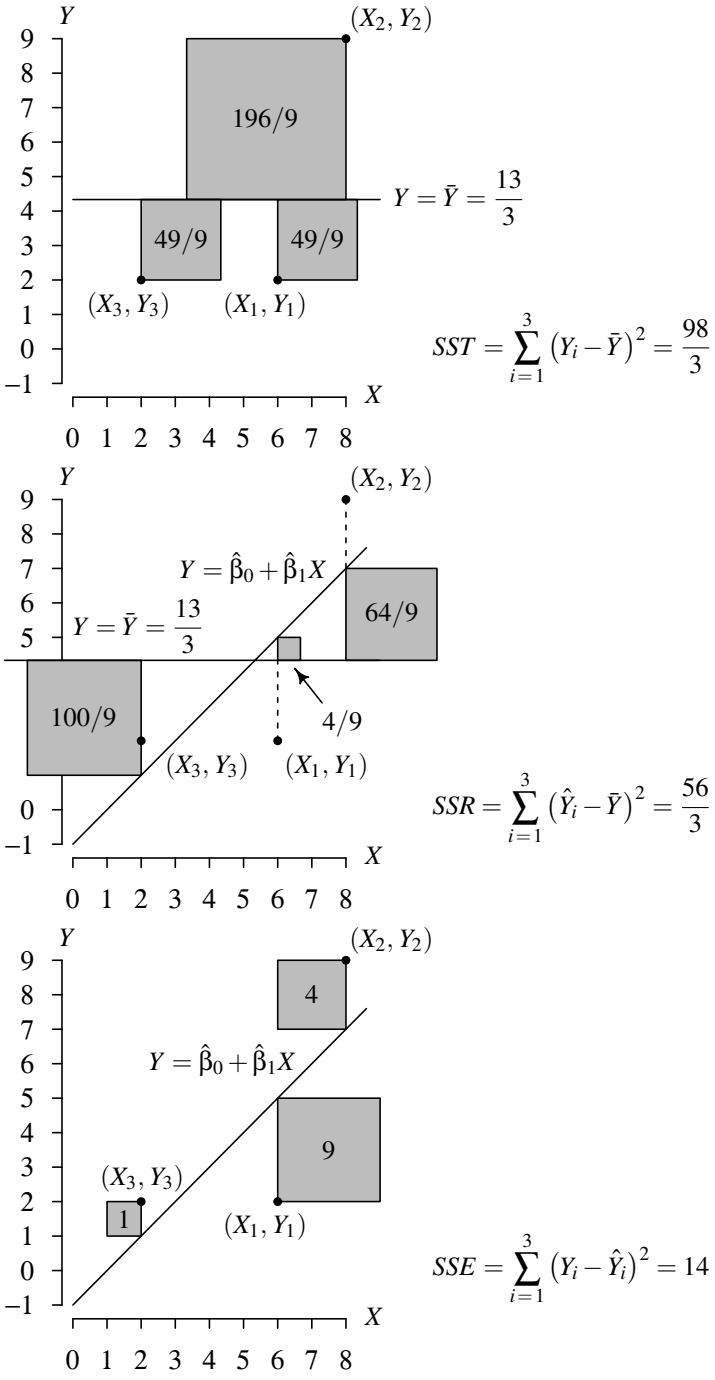

Figure 3.14: Geometry associated with $SST = SSR + SSE$ for the sales data.

Theorem 2.2 stated that under the simple linear regression model with normal errors,

$$\frac{SSE}{\sigma^2} \sim \chi^2(n-2).$$

An outline of the proof of Theorem 2.2 was given in Chapter 2 in purely algebraic terms. An outline of the proof to the result using the matrix approach to simple linear regression is given here to contrast the difference between the two approaches.

Proof (Outline only; matrix approach) As given in the matrix version of Definition 1.2, the vector of *fitted values* in a simple linear regression model is the $n \times 1$ column vector

$$\hat{\mathbf{Y}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}.$$

The sum of squares for error in matrix form is

$$\begin{aligned} SSE &= \left(\mathbf{Y}-\hat{\mathbf{Y}}\right)'\left(\mathbf{Y}-\hat{\mathbf{Y}}\right) \\ &= \left[\mathbf{Y}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}\right]'\left[\mathbf{Y}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}\right] \\ &= \left[\mathbf{Y}'-\mathbf{Y}'\mathbf{X}''\left(\left(\mathbf{X}'\mathbf{X}\right)'\right)^{-1}\mathbf{X}'\right]\left[\mathbf{Y}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}\right] \\ &= \mathbf{Y}'\left[\mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right]\left[\mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right]\mathbf{Y}. \end{aligned}$$

Let $\mathbf{R} = \mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$, where I is the $n \times n$ identity matrix. This matrix plays a critical role in the proof. The matrix $\mathbf{R}$ is symmetric because

$$\mathbf{R}' = \left[\mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right]' = \mathbf{I}'-\mathbf{X}''\left(\left(\mathbf{X}'\mathbf{X}\right)'\right)^{-1}\mathbf{X}' = \mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1} = \mathbf{R}.$$

The matrix $\mathbf{R}$ is idempotent because

$$\begin{aligned} \mathbf{R}^2 &= \left[\mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right]\left[\mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right] \\ &= \mathbf{I}^2-2\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'+\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}' \\ &= \mathbf{I}-\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}' \\ &= \mathbf{R}. \end{aligned}$$

Since $\mathbf{R}$ is a symmetric idempotent matrix, it is a projection matrix. This has two implications. First, the rank of $\mathbf{R}$ equals the trace of $\mathbf{R}$, which in this case is $n-2$. Second, all eigenvalues of $\mathbf{R}$ are either zero or one, and in this setting, there are $n-2$ ones and 2 zeros. The rest of the proof proceeds as follows. Since $\mathbf{R}$ is symmetric matrix it can be orthogonally diagonalized as $\mathbf{R} = \mathbf{U}\mathbf{D}\mathbf{U}'$, where $\mathbf{U}$ is an orthogonal matrix and $\mathbf{D}$ is a diagonal matrix with $n-2$ ones and 2 zeros on the diagonal. The assumed normality of the error terms in the model results in normally distributed residuals, which can be simplified to yield $SSE/\sigma^2 \sim \chi^2(n-2)$. □

The matrix approach gives an alternative way of computing measures of interest in a simple linear regression. Using matrices also allows the following two helpful extensions to simple linear regression.

- Removing the first column of the $\mathbf{X}$ matrix that consists entirely of ones corresponds to forcing a regression line through the origin.
- Adding additional columns to the $\mathbf{X}$ matrix corresponds to including additional independent variables to the regression model, which is known as *multiple linear regression*. This is the topic of the next section.

3.5 Multiple Linear Regression

Multiple linear regression can often be applied when there are several independent variables (or predictors) $X_1, X_2, \ldots, X_p$ which can be used to explain a continuous dependent (or response) variable Y. Three examples are listed below.

- The asking price of a home Y is a function of
  - the number of square feet in the home,
  - the number of bedrooms, and
  - acreage of the land associated with the home.
- The annual amount of money a person donates to charity Y is a function of
  - the nationality of the person,
  - the annual income of the person,
  - the net worth of the person,
  - the religious affiliation of the person,
  - the age of the person, and
  - the gender of the person.
- The stopping distance of a car Y is a function of
  - the speed of the car,
  - the weight of the car, and
  - the type of brakes installed on the car.

One way to formulate a multiple linear regression model is to treat the left-hand side of the model as an expected value:

$$E[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

Since $E[Y]$ denotes a *conditional expectation* of Y given the values of the p independent variables $X_1, X_2, \ldots, X_n$, a more careful way to write this model is

$$E[Y \,|\, X_1, X_2, \ldots, X_n] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

So far, there has been no consideration of the probability distribution of the error terms, and that is addressed in the formal definition of a multiple linear regression model given next.

Definition 3.4 A *multiple linear regression model* is given by

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon,$$

where

- $X_1, X_2, \ldots, X_p$ are the independent variables, assumed to be a fixed values observed without error,
- Y is the dependent variable, which is a continuous random variable,
- β_0 is the population intercept of the regression plane, an unknown constant parameter,
- $\beta_1, \beta_2, \ldots, \beta_p$ are unknown constant parameters which control the inclination of the regression plane, and
- ε is the error term, a continuous random variable with population mean zero and positive, finite population variance σ^2 that accounts for the randomness in the relationship between $X_1, X_2, \ldots, X_p$ and Y.

To estimate the parameters in a multiple linear regression model, we collect n observations which each consist of the p independent variables and the associated dependent variable. In most applications, $p > n$. Occasions arise (often in biostatistical applications) in which $p < n$. The formulation of the simple linear regression model with notation included for the n observations is

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \varepsilon_i$$

for $i = 1, 2, \ldots, n$. So X_{ij} denotes the value of the jth independent variable collected on the ith observational unit. In the real estate example given at the beginning of this section, X_{83} is the value of the third independent variable (acreage) collected on the 8th home collected by the analyst. The associated asking price of the 8th home is Y_8.

Figure 3.15 shows a portion of the *population* regression plane $E[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ for a multiple linear regression model with $p = 2$ independent variables X_1 and X_2. The plane extends outward from the portion shown in Figure 3.15. The regression parameters β_0, β_1, and β_2 are fixed constants. The intercept β_0 is positive in Figure 3.15 because the plane strikes the Y-axis above the origin. Based on the inclination of the population regression plane relative to the X_1- and X_2-axes it is clear that $\beta_1 < 0$ and $\beta_2 > 0$. To avoid clutter and highlight the geometry and notation, only the ith data triple (X_{i1}, X_{i2}, Y_i) and the associated error term ε_i are shown in the figure.

Figure 3.16 shows a portion of the *estimated* regression plane $Y = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ for a multiple linear regression model with $p = 2$ independent variables X_1 and X_2. The estimated regression parameters $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ are random variables which are estimated from n data triples $(X_{11}, X_{12}, Y_1), (X_{21}, X_{22}, Y_2), \ldots, (X_{n1}, X_{n2}, Y_n)$. The estimated regression parameters are random variables because the dependent variable values $Y_1, Y_2, \ldots, Y_n$ are random variables. The estimated intercept $\hat{\beta}_0$ is positive in Figure 3.16 because the plane strikes the Y-axis above the origin. Based on the inclination of the estimated regression plane relative to the X_1- and X_2-axes it is clear that $\hat{\beta}_1 < 0$ and $\hat{\beta}_2 > 0$. To avoid clutter and highlight the geometry and notation, just the ith data triple (X_{i1}, X_{i2}, Y_i), the associated fitted value $(X_{i1}, X_{i2}, \hat{Y}_i)$, and the associated residual e_i are shown in the figure.

When there are $p > 2$ independent variables, the estimated regression model is a *hyperplane* in $\mathcal{R}^{p+1}$. Residual i is the distance $e_i = Y_i - \hat{Y}_i$, for $i = 1, 2, \ldots, n$.

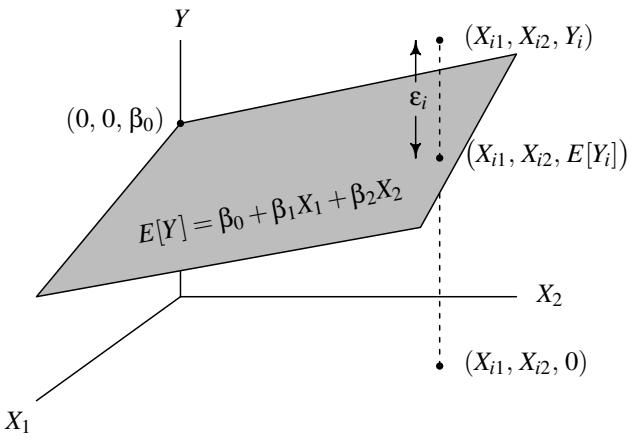

Figure 3.15: Population regression plane and a sample point.

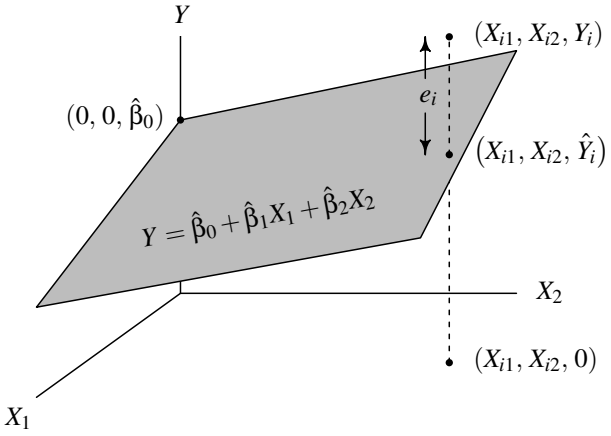

Figure 3.16: Estimated regression plane and a sample point.

When the error terms are assumed to be normally distributed, this model is known as the *multiple linear regression model with normal error terms*. This additional assumption allows for statistical inference concerning parameters and predicted values in a similar manner to that described in Chapter 2.

The multiple linear regression model can also be expressed in terms of matrices. Relative to the simple linear regression model, additional columns are appended to the $\mathbf{X}$ matrix, and the $\boldsymbol{\beta}$ vector is expanded to include the parameters associated with the additional parameters:

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1p} \\ 1 & X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

The vectors $\mathbf{Y}$ and $\boldsymbol{\varepsilon}$ remain unchanged from the simple linear regression formulation. The first row of $\mathbf{X}$ corresponds to the values of the independent variables collected on the first observational unit, the second row of $\mathbf{X}$ corresponds to the values of the independent variables collected on the second observational unit, etc. As was the case in simple linear regression, $\mathbf{X}$ is known as the *design matrix*.

The good news about the matrix approach to multiple linear regression is that the definitions and results from simple linear regression only require some minor tweaking in order to generalize to multiple regression. Several of these definitions and results are given below. In many cases, it is just a matter of replacing the word "simple" with the word "multiple" or updating the degrees of freedom to account for the p independent variables. It is assumed that the $\mathbf{X}$ matrix has rank $p+1$ (that is, a *full rank* matrix), which means that the columns of $\mathbf{X}$ are linearly independent.

- The *multiple linear regression model* is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

  where $E[\boldsymbol{\varepsilon}] = \mathbf{0}$, $V[\boldsymbol{\varepsilon}] = \sigma^2\mathbf{I}$, and $\mathbf{I}$ is the $n \times n$ identity matrix.

- The least squares estimators of $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}} = \left(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p\right)'$, solve the normal equations

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y}.$$

  Since $\mathbf{X}$ has full rank, $\mathbf{X}'\mathbf{X}$ is invertible and the normal equations have the unique solution

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y},$$

  by premultiplying both sides of the normal equations by $\left(\mathbf{X}'\mathbf{X}\right)^{-1}$.

- The least squares estimator of $\boldsymbol{\beta}$ in a multiple linear regression model is an unbiased estimator of $\boldsymbol{\beta}$ because

$$E\left[\hat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}.$$

- The least squares estimators of $\boldsymbol{\beta}$ in the multiple linear regression model can be written as linear combinations of the dependent variables:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y},$$

  where the coefficients in the linear combinations are given by $\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$.

- The variance–covariance matrix of the least squares estimators of $\boldsymbol{\beta}$ is

$$\sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}.$$

- (Gauss–Markov theorem) The least squares estimators of $\boldsymbol{\beta}$ in a multiple linear regression model, $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, have the smallest population variance amongst all linear unbiased estimators of $\boldsymbol{\beta}$.

- The vector of *fitted values* in a multiple linear regression model is the $n \times 1$ column vector

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y},$$

which is a linear combination of the dependent variables. The vector of *residuals* is the $n \times 1$ column vector

$$\begin{aligned}\mathbf{e} &= \mathbf{Y} - \hat{\mathbf{Y}} \\ &= \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}} \\ &= \mathbf{Y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= \left(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\mathbf{Y},\end{aligned}$$

which is also a linear combination of the dependent variables. The matrix $\mathbf{I}$ is the $n \times n$ identity matrix.

- The *multiple linear regression model with normal error terms* is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim N\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$.

- For the multiple linear regression model with normal error terms, the maximum likelihood estimators of $\boldsymbol{\beta}$ are

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

and the maximum likelihood estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}\left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)'\left(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right).$$

Since the vector of error terms $\boldsymbol{\varepsilon}$ consists of independent and identically distributed normal random variables, $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is a vector of independent and identically distributed normal random variables. Since $\hat{\boldsymbol{\beta}}$ is a linear transformation of Y, $\hat{\boldsymbol{\beta}} \sim N\left(\boldsymbol{\beta}, \sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right)$.

- Under the multiple linear regression model, the $n \times n$ *hat matrix* is

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'.$$

The diagonal elements of the hat matrix are the *leverages*. The matrix equation

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$$

indicates that $\mathbf{H}$ transforms $\mathbf{Y}$ to $\hat{\mathbf{Y}}$. The hat matrix is symmetric (that is, $\mathbf{H} = \mathbf{H}'$) and idempotent (that is, $\mathbf{H}\mathbf{H} = \mathbf{H}$). The trace of the hat matrix is $\sum_{i=1}^{n} h_{ii} = p + 1$.

The example of multiple linear regression that follows considers $p = 2$ predictors of the sales price of a home.

Example 3.10 In Example 2.9, the sales price, Y, of homes sold in Ames, Iowa between 2006 and 2010 with between 2500 and 3500 square feet were fitted to a simple linear regression model with the square footage as an independent variable X. There were $n = 120$ homes in the data frame that fit this criteria. In that analysis, the value of the land was estimated to be \$21,233 (although this was outside of the scope of the simple linear regression model), and the price of the home increased by an average of \$112 with each additional square foot of indoor space. Fit a multiple linear regression

model with normal error terms to the same data set using two independent variables, X_1, the square footage of indoor space, and X_2, the square footage of the lot. The dependent variable is again the sales price Y.

The multiple regression model in this setting is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon,$$

where $\varepsilon \sim N\left(\mu, \sigma_Z^2\right)$. The R code below estimates the regression parameters β_0, β_1, and β_2. The regression function

$$E[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2,$$

is a plane in $\mathcal{R}^3$. The values of β_1 and β_2 control the tilt of the regression plane, and the value of β_0 is the intercept of the regression plane with the $E[Y]$ axis. The regression plane will be fitted in two fashions in R: the matrix approach to multiple linear regression and the built-in `lm` function. The R code below defines the $\mathbf{X}$ and $\mathbf{Y}$ matrices, and then uses the formula

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}$$

to calculate the estimates of the regression coefficients.

```
library(modeldata)
i       = ames$Gr_Liv_Area >= 2500 & ames$Gr_Liv_Area <= 3500
sqft    = ames$Gr_Liv_Area[i]
lotarea = ames$Lot_Area[i]
X       = cbind(1, sqft, lotarea)
Y       = ames$Sale_Price[i]
beta    = solve(t(X) %*% X) %*% t(X) %*% Y
```

These R statements return the least squares regression parameter estimates $\hat{\beta}_0 = 26{,}515$, $\hat{\beta}_1 = 96.88$, and $\hat{\beta}_2 = 2.65$. The intercept is not meaningful in this setting because it is associated with a home with 0 square feet and no land. This situation does not make sense nor does it fall in the scope of the model. The naive interpretation of the other regression coefficients in the fitted model are (*a*) the sales price of a home increases by an average of $96.88 for each additional square foot in the home, and (*b*) the sales price of the home increases by $2.65 for each additional square foot in the lot size. The interpretation of the estimated regression coefficients is more nuanced in the case of multiple independent variables because those independent variables are often correlated. So reporting that "the value of $\hat{\beta}_1 = 96.88$ means that the sales price of the house increases by an average of $96.88 for each additional square foot of interior space with the lot size fixed" is not quite accurate because the interior space and lot size might be correlated. Larger homes might be built on larger lots, for example. Regression analysts acknowledge possible correlations between the independent variables by just stating "the sales price increases by an average of $96.88 for each additional square foot of interior space, adjusted for lot size" when interpreting $\hat{\beta}_1$. Likewise, "the sales price increases by an average of $2.65 for each additional square foot of lot size, adjusted for interior square footage" when interpreting $\hat{\beta}_2$.

A second way to calculate the estimated regression coefficients is to use R's built-in `lm` function.

```
library(modeldata)
i       = ames$Gr_Liv_Area >= 2500 & ames$Gr_Liv_Area <= 3500
sqft    = ames$Gr_Liv_Area[i]
lotarea = ames$Lot_Area[i]
price   = ames$Sale_Price[i]
fit     = lm(price ~ sqft + lotarea)
summary(fit)
```

The call to the `summary` function prints the following output concerning the fitted multiple linear regression model.

```
Call:
lm(formula = price ~ sqft + lotarea)

Residuals:
    Min      1Q  Median      3Q     Max
-226718  -61645   -5756   62774  288215

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.652e+04  1.087e+05   0.244   0.8077
sqft        9.688e+01  4.043e+01   2.396   0.0181 *
lotarea     2.645e+00  1.660e+00   1.593   0.1138
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 99890 on 117 degrees of freedom
Multiple R-squared:  0.08339,   Adjusted R-squared:  0.06772
F-statistic: 5.322 on 2 and 117 DF,  p-value: 0.006134
```

The estimated regression coefficients match those that were calculated using the matrix approach to multiple linear regression. The right-hand column of p-values tells us that the size of a home is a statistically significant predictor of the sales price of a home, but the lot size is not a statistically significant predictor of the sales price of a home.

A multiple linear regression model can easily be adapted to include nonlinear terms. A multiple regression model with two independent variables X_1 and X_2, for example, with a linear relationship between X_1 and Y and a quadratic relationship between X_2 and Y which includes an intercept term is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_2^2 + \varepsilon.$$

Using the R `lm` function to estimate the coefficients will be illustrated in Section 3.7.

Multiple linear regression has many more modeling issues that arise than simple linear regression. The subsections that follow consider the following topics within multiple regression: (*a*) handling categorical independent variables which fall in categories rather than quantitative values, (*b*) handling the case in which independent variables have interactive effects, (*c*) extending the ANOVA table to multiple independent variables, (*d*) calculation of the coefficient of determination for multiple linear regression, and an adjustment that can be made to reduce its bias, (*e*) the effect of multicollinearity among the independent variables, and (*f*) algorithms for model selection.

3.5.1 Categorical Independent Variables

Some regression models include independent variables which are not naturally quantitative, but are rather categorical. These categorical independent variables require some special treatment in order to be included in a multiple linear regression model. The cases in which a categorical independent variable falls in one of two categories will be considered separately from the case in which a categorical independent variable falls in one of more than two categories.

Categorical independent variable which falls in one of two categories. Consider a multiple linear regression model with $p = 2$ independent variables, X_1, which is age, and X_2, which is gender. The dependent variable is the annual salary Y. So the multiple linear regression model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon.$$

Regression models assume that the independent variables are quantitative rather than categorical like gender. One solution to this problem is to code the gender as 0 for female and 1 for male. The independent variable X_2 in this case is known as a *dummy variable* or an *indicator variable*. As a particular instance, consider $n = 6$ data points consisting of three women (ages 26, 71, and 34) and three men (ages 44, 65, and 21). In this case the design matrix is

$$\mathbf{X} = \begin{bmatrix} 1 & 26 & 0 \\ 1 & 71 & 0 \\ 1 & 34 & 0 \\ 1 & 44 & 1 \\ 1 & 65 & 1 \\ 1 & 21 & 1 \end{bmatrix}.$$

The elements of the six-element column vector $\mathbf{Y}$ are the associated salaries. The value of $\hat{\beta}_0$ is not meaningful here. Not only is it outside of the scope of the model, its interpretation as the annual salary of a newborn baby girl doesn't fit with societal norms. Newborn baby girls seldom earn annual salaries. The value of $\hat{\beta}_1$ indicates the increase in annual salary for each additional year in age, adjusted for gender. Since salaries tend to rise over time, we anticipate that $\hat{\beta}_1$ will be positive. The value of $\hat{\beta}_2$ indicates the change in salary associated being male rather than female, adjusted for age. If $\hat{\beta}_2$ is significantly greater than zero, then men's salaries are significantly higher than women's salaries, adjusted for age; if $\hat{\beta}_2$ is significantly less than zero, then women's salaries are significantly higher than men's salaries, adjusted for age. The choice of using an indicator of 0 for women and 1 for men was arbitrary. See if you can predict what would happen if instead we used 0 for men and 1 for women.

Categorical independent variable which falls in one of more than two categories. Let's extend the regression model to predict the annual salary to include another categorical variable: political affiliation. This categorical variable will have three levels: Republican, Democrat, and Independent. The third category includes anyone who is not affiliated with the two main political parties in the United States. Although it might be tempting to just let $X_3 = 1$ denote a Republican, $X_3 = 2$ denote a Democrat, and $X_3 = 3$ denote an Independent, this will likely produce erroneous results for two reasons. First, using the ordering $X_3 = 1$, $X_3 = 2$, and $X_3 = 3$ implies an ordering of the salaries associated with individuals from the three different political affiliations for $\beta_3 > 0$, or the opposite ordering of the salaries associated with individuals from the three political affiliations for $\beta_3 < 0$. This ordering might not be the correct ordering. Second, leaving a gap of 1 between each of the values of X_3 indicates that there is a known and equal salary gap between individuals from the ordered different political affiliations. The usual way to account for a categorical independent

variable which can take on c values is to define $c-1$ independent indicator variables. In the case of political affiliation, the independent variables X_3 and X_4 can be defined as

$$X_3 = \begin{cases} 0 & \text{not a Republican} \\ 1 & \text{Republican} \end{cases}$$

and

$$X_4 = \begin{cases} 0 & \text{not a Democrat} \\ 1 & \text{Democrat.} \end{cases}$$

So now the multiple linear regression model with $p = 4$ independent variables is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon.$$

In this fashion, the expected value of an Independent's salary is given by

$$E[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2,$$

the expected value of an Republican's salary is given by

$$E[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3,$$

and the expected value of a Democrat's salary is given by

$$E[Y] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_4 X_4.$$

With this arrangement of the levels of the categorical variable representing the political affiliation, there is no predicted ordering of salaries by the three political affiliations nor are the gaps between the affiliations necessarily equal.

As a particular instance, consider $n = 6$ data points with three women (a 26-year-old Independent, a 71-year-old Democrat, and a 34-year-old Republican) and three men (a 44-year-old Independent, a 65-year-old Democrat, and a 21-year-old Republican) in the study. The appropriate design matrix is

$$\mathbf{X} = \begin{bmatrix} 1 & 26 & 0 & 0 & 0 \\ 1 & 71 & 0 & 0 & 1 \\ 1 & 34 & 0 & 1 & 0 \\ 1 & 44 & 1 & 0 & 0 \\ 1 & 65 & 1 & 0 & 1 \\ 1 & 21 & 1 & 1 & 0 \end{bmatrix}.$$

The value of $\hat{\beta}_3$ is the estimated difference between the mean annual salary of an Independent and a Republican, adjusted for age and gender. The value of $\hat{\beta}_3$ is the estimated difference between the mean annual salary of an Independent and a Democrat, adjusted for age and gender. This example has been for illustrative purposes only. Estimating five parameters $\beta_0, \beta_1, \ldots, \beta_4$ from just six data values will almost certainly not provide strong statistical evidence concerning the effect of age, gender, and political affiliation on salary. Furthermore, many other important factors, such as years of education, years on the job, and type of work, have not been included in this regression model.

3.5.2 Interaction Terms

The multiple linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

assumes a linear relationship between each independent variable and Y and the slope associated with an independent variable is identical at all values of the other independent variables within the scope of the multiple linear regression model. This relationship is illustrated for some selected data points of smaller homes from the Ames, Iowa housing data set from Examples 2.9 and 3.10. In this case, X_1 is the interior square footage, X_2 is an indicator variable reflecting the lot size,

$$X_2 = \begin{cases} 0 & \text{lot size is less than or equal to 10,000 square feet} \\ 1 & \text{lot size is greater than 10,000 square feet,} \end{cases}$$

and Y is the sales price. The multiple linear regression model with the $p = 2$ independent variables is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon.$$

Figure 3.17 shows a scatterplot of the interior square footage and sales price of homes on smaller lots ($X_2 = 0$ as open points) and larger lots ($X_2 = 1$ as solid points). The values of $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ are indicated on the graph. The estimated intercept $\hat{\beta}_0 = 21{,}473$, although slightly outside of the scope of the model, gives the estimated sales price of a small lot containing no dwelling as \$21,473. The estimated regression coefficient $\hat{\beta}_1 = 31.33$ indicates that the sales price of a home increases by an estimated \$31.33 for each additional interior square foot, adjusted for lot size. The estimated regression coefficient $\hat{\beta}_2 = 35{,}693$ indicates that homes on larger lots cost \$35,693 more, on average, than homes on smaller lots, adjusted for interior square feet. Notice that this formulation of the multiple linear regression model forces the slopes of the two lines in Figure 3.17 to be identical, regardless of the value of X_2.

But is the assumption of equal slopes of the two lines in Figure 3.17 justified? Separate simple linear regression models are fitted to the homes built on smaller and larger lots, and the results are plotted in Figure 3.18. The lines do not appear to be parallel in this case, indicating that a more complex regression model is warranted. There appears, in this case, to be an *interaction effect*

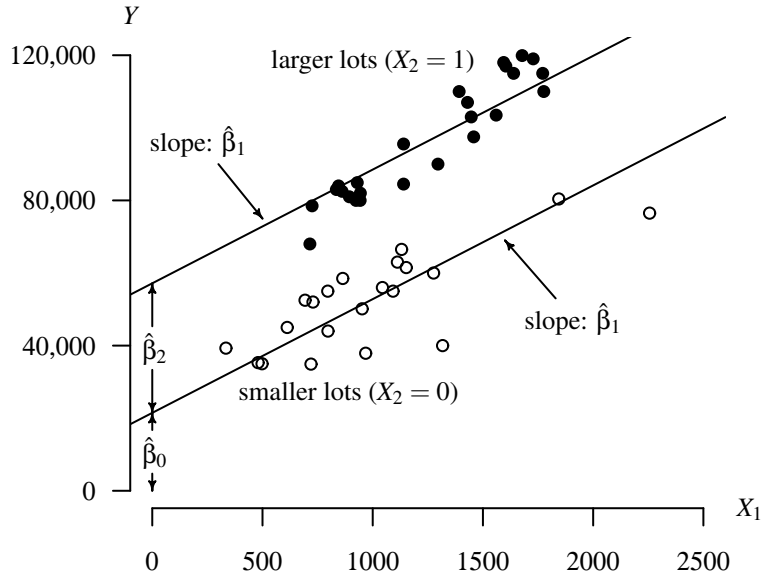

Figure 3.17: Fitted multiple linear regression model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$.

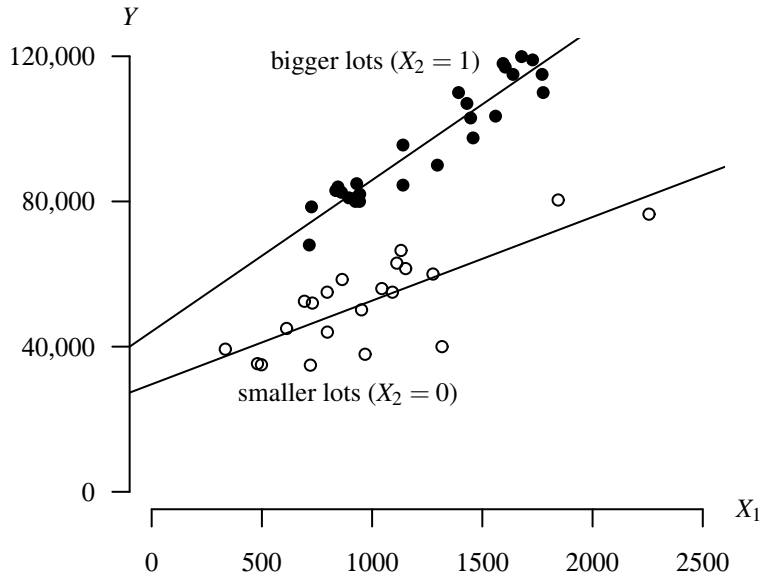

Figure 3.18: Fitted simple linear regression models $Y = \beta_0 + \beta_1 X_1 + \varepsilon$.

between X_1 and X_2. This means that the effect of one independent variable (X_1, for example, the interior size) on Y is altered based on the value of another independent variable (X_2, the lot size indicator).

Regression analysts account for this interaction by including cross-product terms in the regression model. In this Ames housing data set example, the regression model with an interaction term is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon.$$

If the regression parameter $\hat{\beta}_3$ differs statistically from 0, then the inclusion of the interaction term is warranted. Notice that when $X_2 = 0$ (smaller lots), the model reduces to

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon,$$

which is a simple linear regression model with intercept parameter β_0 and slope parameter β_1. On the other hand, when $X_2 = 1$ (larger lots), the model reduces to

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 + \beta_3 X_1 + \varepsilon$$

or

$$Y = \beta_0 + \beta_2 + (\beta_1 + \beta_3) X_1 + \varepsilon$$

which is a simple linear regression model with intercept parameter $\beta_0 + \beta_2$ and slope parameter $\beta_1 + \beta_3$. It is in this fashion that the two non-parallel lines depicted in Figure 3.18 can be estimated in a single regression model. Not surprisingly, it requires four parameters, β_0, β_1, β_2, and β_3, to do so. The multiple linear regression model with an interaction term can be fitted using the `lm` function in R by simply replacing the usual + in the formula with *. All four parameters are statistically significant at the 0.05 level in this case, so the inclusion of an interaction term is warranted.

3.5.3 The ANOVA Table

The degrees of freedom for the sums of squares in multiple linear regression are modified because of the additional parameters estimated relative to those given in the ANOVA table from Table 2.2 for simple linear regression. The ANOVA table for a multiple linear regression model with p independent variables and normal error terms is given in Table 3.6. Formulas for the sums of squares

Source	SS	df	MS	F
Regression	SSR	p	MSR	MSR/MSE
Error	SSE	$n-p-1$	MSE	
Total	SST	$n-1$		

Table 3.6: Basic ANOVA table for multiple linear regression.

using the matrix formulation for multiple linear regression are $SST = SSR + SSE$, which is

$$\left(\mathbf{Y}-\bar{\mathbf{Y}}\right)'\left(\mathbf{Y}-\bar{\mathbf{Y}}\right) = \left(\hat{\mathbf{Y}}-\bar{\mathbf{Y}}\right)'\left(\hat{\mathbf{Y}}-\bar{\mathbf{Y}}\right) + \left(\mathbf{Y}-\hat{\mathbf{Y}}\right)'\left(\mathbf{Y}-\hat{\mathbf{Y}}\right),$$

where $\bar{\mathbf{Y}}$ is an n-element column vector with identical elements which are each the sample mean of the values of the dependent variable. Equivalently,

$$SST = \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{J}\mathbf{Y}/n, \qquad SSR = \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{Y} - \mathbf{Y}'\mathbf{J}\mathbf{Y}/n, \qquad SSE = \mathbf{Y}'\mathbf{Y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{Y},$$

where $\mathbf{J}$ is an $n \times n$ matrix with all elements being equal to 1. The mean square error for regression is $MSR = SSR/p$, the mean square error is $MSE = SSE/(n-p-1)$, and the test statistic $F = MSR/MSE$ can be used for testing

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

versus

$$H_1 : \text{not all } \beta_1, \beta_2, \ldots, \beta_p \text{ equal } 0$$

where F has an $F(p, n-p-1)$ distribution under H_0. The `anova` function in R can be used to generate an ANOVA table associated with a multiple linear regression model fitted by the `lm` function. For the Ames, Iowa housing data from Example 3.10 which used $p = 2$ independent variables (interior square footage and lot size), the R `summary` function returns the test statistic $F = 5.322$, which is associated with a p-value of $p = 0.006$ based on the F distribution with $p = 2$ and $n-p-1 = 120-2-1 = 117$ degrees of freedom. There is strong statistical evidence that one or both of the coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ is statistically different from zero. One or both of the independent variables is effective in predicting the sales price.

3.5.4 Adjusted Coefficient of Determination

The coefficient of determination for a multiple linear regression model is defined as

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} = 1 - \frac{SSE}{SST},$$

and it measures the fraction of variation in $Y_1, Y_2, \ldots, Y_n$ about $\bar{Y}$ that is accounted for by the linear relationship between the independent variables $X_1, X_2, \ldots, X_p$ and Y. As before $0 \leq R^2 \leq 1$, and the

extreme cases are associated with $\hat{\beta}_1 = \hat{\beta}_2 = \cdots = \hat{\beta}_p = 0$ (for $R^2 = 0$) and all Y-values falling in the estimated regression hyperplane (for $R^2 = 1$).

Now consider a multiple linear regression model with p independent variables $X_1, X_2, \ldots, X_p$. What is the effect on *SST* and *SSE* of adding another independent variable, X_{p+1}, to the model? Adding another independent variable does not affect *SST* because it depends only on $Y_1, Y_2, \ldots, Y_n$. The value of *SSE* cannot increase with the addition of the new independent variable because either (a) *SSE* will remain the same if $\hat{\beta}_{p+1} = 0$, or (b) *SSE* will decrease if $\hat{\beta}_{p+1} \neq 0$. The impact on R^2 is that it must stay the same or increase for every additional independent variable that is added to the model.

It is for this reason that R^2 tends to be a biased estimator of the fraction of variation in $Y_1, Y_2, \ldots, Y_n$ accounted for by the independent variables. Some regression software (including R) calculate an *adjusted coefficient of variation* by dividing the sums of squares by their associated degrees of freedom

$$R^2_{\text{adj}} = 1 - \frac{SSE/(n-p-1)}{SST/(n-1)}.$$

Both values are reported in the call to the `summary` function with the Ames, Iowa housing data in Example 3.10 as

$$R^2 = 0.08339 \qquad \text{and} \qquad R^2_{\text{adj}} = 0.06772.$$

3.5.5 Multicollinearity

In many settings, the values of the independent variables are correlated. In the housing data set from Example 3.10, for example, the independent variables X_1 (interior square footage) and X_2 (lot size) are probably positively correlated. Intuition suggests that larger homes are built on larger lots, on average. In the extreme case, what if homes in Ames were required by some bizarre municipal code to all be single story homes with the square footage of the lot always exactly four times the square footage of the interior of the home? In this case, $X_2 = 4X_1$, so knowing the value of either X_1 or X_2 allows you to know the value of the other. Intuitively, one of the two independent variables is superfluous. When this is the case, the design matrix $\mathbf{X}$ has two columns which are multiples of one another, so these columns are linearly dependent and the matrix does not have full rank. This implies that the matrix $\mathbf{X}'\mathbf{X}$ (which is used in computing the estimates of the regression coefficients) is singular, so it does not have an inverse. In this case, the usual formula for the regression coefficients,

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y},$$

is undefined because the matrix $\mathbf{X}'\mathbf{X}$ does not have an inverse. In the case in which $X_2 = 4X_1$, all pairs of the independent variables fall on a line, so it is impossible to know the proper tilt of the fitted regression plane in $\mathcal{R}^3$. There are many planes that minimize the sum of squared errors.

Multicollinearity is the condition associated with independent variables that are highly correlated among themselves in a multiple regression model. More specifically, multicollinearity occurs when two or more of the independent variables have a high correlation. This can appear as an approximately linear relationship between two of the independent variables. Multicollinearity is a condition associated with the design matrix $\mathbf{X}$ rather than the values of the dependent variable $\mathbf{Y}$ or the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. In cases in which multicollinearity exists, the matrix $\mathbf{X}'\mathbf{X}$ has an inverse, but it is ill-conditioned and subject to slight variations in the data or is unstable because of large differences in the magnitudes of the various values of the independent variables. One of the key practical issues when multicollinearity is present is that an estimated regression coefficient for a particular

independent variable depends on whether the other independent variables are included or left out of the model.

So multicollinearity has been loosely defined as high correlation among the independent variables. There is redundancy to the information contained in the independent variables. The next paragraphs describe how to detect multicollinearity, its consequences, and some remedies.

Although the hypothetical perfect correlation between the interior space and the lot size of a home from Ames, Iowa described previously occurs seldom in practice, highly correlated independent variables can result in some unusual behavior of regression coefficients as a regression model is constructed. Some signs that multicollinearity might be present in a multiple linear regression model include the following.

- Large values of the estimated standard deviations of the regression coefficients.
- Including or not including an independent variable in the model results in large changes to the estimated regression coefficients.
- An estimated regression coefficient that is statistically significant when the associated independent variable is considered alone, but becomes insignificant when one or more other independent variables are added to the model.
- An estimated regression coefficient with a sign that is inconsistent with expected sign or inconsistent with previous similar data sets.
- The pairwise sample correlation among the independent variables is high. The `cor` function in R can be used to assess the correlation among independent variables. The R statement

  ```
  cor(swiss)
  ```

  for example, calculates the correlation matrix for the columns of the built-in data frame named `swiss`. The off-diagonal elements of this matrix range from -0.69 to 0.70, indicating that multicollinearity is present.

All of the criteria listed above are informal. A more formal way to determine whether multicollinearity is present is to introduce a statistic which reflects multicollinearity. The estimate of the variance of $\hat{\beta}_j$ can be written as

$$\hat{V}\left[\hat{\beta}_j\right] = \frac{1}{1-R_j^2}\left[\frac{MSE}{\sum_{i=1}^{n}(X_{ij}-\bar{X}_j)^2}\right],$$

where $\bar{X}_j = \sum_{i=1}^{n} X_{ij}$, $MSE = SSE/(n-p-1)$ for the full multiple regression model, and R_j^2 is the coefficient of determination obtained by conducting a multiple linear regression with X_j as the dependent variable and the other $p-1$ X-values as the independent variables, for $j = 1, 2, \ldots, p$. The coefficient on the right-hand side of this equation,

$$VIF_j = \frac{1}{1-R_j^2},$$

is known as a *variance inflation factor* for independent variable j, for $j = 1, 2, \ldots, p$. In the extreme case when $R_j^2 = 0$, the associated variance inflation factor is $VIF_j = 1$. This corresponds to the case in which X_j is not linearly related to the other independent variables. As R_j^2 increases, VIF_j also increases, corresponding to increased correlation between the independent variables. When the largest

of the VIF_j values exceeds the threshold value of 10, one can conclude that the multicollinearity is present among the independent variables.

The R code below calculates the variance inflation factors for the data values in the `swiss` data frame, where the independent variables

- X_1, the percentage of males involved in agriculture as an occupation,
- X_2, the percentage of draftees receiving the highest make on an army examination,
- X_3, the percentage of draftees with education beyond the primary school,
- X_4, the percentage of Catholics, and
- X_5, the percentage of live births who live less than one year,

are used to predict Y, a common standardized fertility measure, from the $n = 47$ French-speaking provinces of Switzerland in about the year 1888. The R code below computes the variance inflation factors for the $p = 5$ independent variables.

```
swiss = as.matrix(swiss)
p   = 5
y   = swiss[ , 1]
n   = length(y)
x   = cbind(1, swiss[ , 2:(p + 1)])
for (i in 2:(p + 1)) {
  yy     = x[ , i]
  xx     = x[ , -i]
  beta   = solve(t(xx) %*% xx) %*% t(xx) %*% yy
  fitted = xx %*% beta
  resid  = yy - fitted
  sse    = sum(resid ^ 2)
  m      = mean(yy)
  sst    = sum((yy - m) ^ 2)
  r2     = 1 - sse / sst
  vif    = 1 / (1 - r2)
  print(vif)
}
```

The variance inflation factors for the $p = 5$ independent variables are

$$VIF_1 = 2.28, VIF_2 = 3.68, VIF_3 = 2.77, VIF_4 = 1.94, VIF_5 = 1.11.$$

Since none of these five values exceeds 10, we can conclude that the multicollinearity that exists in the independent variables is not strong enough to cause concern. (Some regression analysts use 5 as a threshold rather than 10.) Some keystrokes can be saved by using the `vif` function from the `car` package on a multiple linear regression model fitted by the `lm` function.

```
library(car)
fit = lm(Fertility ~ Agriculture + Examination + Education +
         Catholic + Infant.Mortality, data = swiss)
summary(fit)
vif(fit)
```

One popular remedy for multicollinearity is known as *ridge regression*, which is a parameter estimation technique that abandons the requirement of unbiased parameter estimates. The approach taken with ridge regression is to choose estimates for the regression parameters that are biased, but have a smaller variance than the ordinary least squares estimates. The goal is to generate parameter estimates with tolerable bias but smaller variance. The typical approach used in statistics to overcome this bias/variability trade-off is to use the estimates that minimize the mean square errors. Assuming that the X and Y values have been centered, we can dispense with the need for an intercept term in the multiple regression model. Rather than minimizing the usual sum of squared errors

$$S = \sum_{i=1}^{n} (Y_i - \beta_1 X_{i1} - \beta_2 X_{i2} - \cdots - \beta_p X_{ip})^2,$$

ridge regression minimizes

$$S_R = \sum_{i=1}^{n} (Y_i - \beta_1 X_{i1} - \beta_2 X_{i2} - \cdots - \beta_p X_{ip})^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

There are now two terms in the modified sum of squares. The second term in S_R is known as the *penalty term*. The new parameter λ is known as the *penalty parameter*. When $\lambda = 0$, S_R reduces to the ordinary least squares case and achieves a value *SSE* at the ordinary least squares estimators. As λ increases, the estimators converge to $\hat{\beta}_1 = \hat{\beta}_2 = \cdots = \beta_p = 0$. We desire a λ value that introduces some bias into the parameter estimates, but also have a reduced variance.

The geometry associated with ridge regression for $p = 2$ independent variables X_1 and X_2 in a multiple linear regression model is illustrated in Figure 3.19. The ellipses are level surfaces of the first term in S_R. The center of the ellipses is the ordinary least squares estimators of $(\beta_1, \beta_2) = (\hat{\beta}_1, \hat{\beta}_2)$, which are the values that minimize the first term of S_R. The circles centered at the origin are level surfaces of the second term in S_R. The ridge regression estimators for β_1 and β_2 will occur at the intersection of one of elliptical and circular contours. In Figure 3.19 the two outermost level surfaces intersect at a point, which is a value of the ridge regression estimates of β_1 and β_2 which correspond to one particular value of the penalty parameter λ. The point at which this intersection

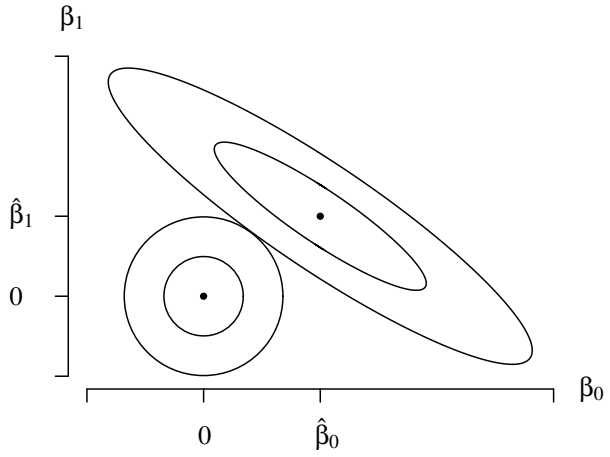

Figure 3.19: Ridge regression geometry for $p = 2$ independent variables.

occurs is a function of the penalty parameter λ. In higher dimensions, the circles become spheres and the ellipses become ellipsoids.

Determining the value of the penalty parameter is critical in ridge regression, but its choice depends on the regression model and associated data set. A common technique for determining an optimal value for λ is known as k-fold cross-validation. There are several functions in R which can perform ridge regression: the `lm.ridge` function from the `MASS` package, the `linearRidge` function from the `ridge` package, and the `glmnet` function from the `glmnet` package. Ridge regression is related to the lasso (least absolute shrinkage and selection operator) estimator and elastic net regularization, two other popular parameter estimation techniques that are often applied for large values of p.

Is there a way to completely avoid multicollinearity? In some settings, the answer is yes. When the values of the independent variables are chosen so that they are uncorrelated, the regression coefficients associated with a simple linear regression model of each independent variable separately match the regression coefficients of any model involving more independent variables. This fact provides a strong argument for a designed experiment which can result in uncorrelated independent variables whenever the setting of the regression problem make this possible.

3.5.6 Model Selection

It is common in regression modeling to have a large number of potential independent variables that might adequately predict the dependent variable Y that need to be sifted through in order to decide whether each should be included or excluded from the regression model. If there are p potential independent variables in the multiple linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

then there are 2^p possible regression models (always including an intercept term and not considering interaction terms or nonlinear terms) because each independent variable will either be included or not included in the regression model. Since the number of regression models to fit can be daunting, even for moderate values of p, we desire an algorithm for selecting the appropriate independent variables to include in the model. *Forward stepwise regression* is one such automatic search procedure used to select the independent variables to include in a multiple linear regression model. The procedure begins with the null model $Y = \beta_0 + \varepsilon$ and progressively adds independent variables to the model that are deemed to be statistically significant. In the initial step, p simple linear regression models are fit for each potential independent variable. The independent variable with the smallest p-value falling below a prescribed threshold (commonly, $\alpha = 0.05$) associated with the t-test described in Section 2.3.2 is added to the model. In the second step, $p-1$ multiple linear regression models with two independent variables are fitted using the previously selected independent variable and each of the other potential independent variables. The independent variable with the smallest p-value is added to the model. This process continues until no more independent variables meet the criteria. This is the multiple linear regression model selected by forward stepwise regression. Several other variants of forward stepwise regression and other model selection algorithms are outlined below.

- Foreward stepwise regression often includes a test to determine whether independent variables that have previously been added to the model have p-values that exceed the threshold and should consequently be removed from the model.

- Backward stepwise regression starts by including all p independent variables in the regression model and eliminates the independent variable with the largest p-value on each step.

Unfortunately, there is no guarantee that forward stepwise regression and backward stepwise regression will result in the same final regression model.

- Once this statistically significant independent variables have been identified, a similar stepwise procedure can be executed to test for statistically significant interaction terms.
- A similar stepwise procedure can be executed to test for the significance of nonlinear terms in the regression model.
- With increased computer speeds and a moderate value of p, the number of independent variables, it is possible to fit all 2^p possible regression models and compare them to determine an appropriate final regression model.
- Comparing potential regression models using p-values is not universal. The Akaike Information Criterion (AIC) is a measure which extracts a penalty for each additional parameter in a model in an effort to avoid overfitting.

In summary, selecting a multiple linear regression model is not easy. The skills required to select a model include the ability to (*a*) detect and remedy multicollinearity, (*b*) assess evidence of interaction effects between independent variables and include them in the model when appropriate, (*c*) assess evidence of nonlinear relationships between some or all of the independent variables and the dependent variable and include appropriate terms in the model, (*d*) execute the appropriate multidimensional diagnostic procedures (outlined in the simple linear regression case in Section 3.2) and execute the appropriate remedial procedures (outlined in the simple linear regression case in Section 3.3) when model assumptions are violated, and (*e*) assess the normality of the residuals.

3.6 Weighted Least Squares

The three approaches to estimating the parameters in a simple linear regression model that we have encountered thus far,

- the algebraic approach,
- the matrix approach,
- using the R `lm` (linear model) function,

all have the same assumptions regarding the independent variable, the dependent variable, and the model $Y = \beta_0 + \beta_1 X + \varepsilon$. In all three approaches, the error terms are assumed to be mutually independent random variables, each with population mean 0 and population variance–covariance matrix $V[\varepsilon] = \sigma_Z^2 I$, where I is the $n \times n$ identity matrix. This means that $V[\varepsilon_i] = \sigma_Z^2$, for $i = 1, 2, \ldots, n$. There is also an implicit assumption that each of the data pairs (X_i, Y_i) are each given equal weight in the regression.

Settings occasionally arise in which some data values should be given different weights. There might be evidence that some of the Y_i values have more precision than others. Weights can be placed on each of the data pairs to account for this difference in precision. This leads to a *weighted least squares* approach to estimating the coefficients in a regression model.

In the standard simple linear regression model, the assumption

$$V[\varepsilon_i] = \sigma_Z^2,$$

for $i = 1, 2, \ldots, n$, means that the variance of the dependent variable from the regression line is equal for all of the n data pairs, regardless of the value of the independent variable. In *weighted least squares* modeling, the positive weights $w_1, w_2, \ldots, w_n$ are determined so that

$$V[\varepsilon_i] = \sigma_Z^2 / w_i$$

for $i = 1, 2, \ldots, n$, which means that certain data pairs have more precision than other data pairs. The weights are fixed constants. There is no requirement that the weights sum to one. Data pairs with larger weights are assumed to have a lower variability to their error terms. This allows for a population variance that changes from one data pair to another.

As an illustration, the values of the dependent variable Y might be sample means at the various values of the independent variable X. Furthermore, if the sample sizes associated with the sample means are known and unequal, then we would like to assign higher weights to the data pairs associated with larger sample sizes. If n_i is the sample size for data pair i, for $i = 1, 2, \ldots, n$, then the appropriate weight for data pair i is $w_i = n_i$ so that

$$V[\varepsilon_i] = \sigma_Z^2 / n_i$$

for $i = 1, 2, \ldots, n$.

So rather than minimizing the sum of squares

$$S = \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i)^2$$

as was the case in the standard simple linear regression model, weighted least squares minimizes the weighted sum of squares

$$S = \sum_{i=1}^{n} w_i (Y_i - \beta_0 - \beta_1 X_i)^2.$$

Notice that this reduces to the ordinary sum of squares when $w_1 = w_2 = \cdots = w_n = 1$. As before, calculus can be used to minimize S with respect to β_0 and β_1 to arrive at the least squares estimators $\hat{\beta}_0$ and $\hat{\beta}_1$. The partial derivatives of S with respect to β_0 and β_1 are

$$\frac{\partial S}{\partial \beta_0} = -2 \sum_{i=1}^{n} w_i (Y_i - \beta_0 - \beta_1 X_i) = 0$$

and

$$\frac{\partial S}{\partial \beta_1} = -2 \sum_{i=1}^{n} w_i X_i (Y_i - \beta_0 - \beta_1 X_i) = 0.$$

These can be simplified to give the normal equations

$$\beta_0 \sum_{i=1}^{n} w_i + \beta_1 \sum_{i=1}^{n} w_i X_i = \sum_{i=1}^{n} w_i Y_i$$

and

$$\beta_0 \sum_{i=1}^{n} w_i X_i + \beta_1 w_i X_i^2 = \sum_{i=1}^{n} w_i X_i Y_i.$$

The normal equations are a system of two linear equations in the two unknowns β_0 and β_1, given the data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ and the weights $w_1, w_2, \ldots, w_n$. The normal equations

can be solved to yield the weighted least squares estimators. This derivation constitutes a proof of the following theorem.

Theorem 3.3 Let $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ be n data pairs with at least two distinct X_i values. Let $w_1, w_2, \ldots, w_n$ be the weights associated with the data pairs. The *weighted least squares estimators* of β_0 and β_1 in the simple linear regression model are the solution to the simultaneous *normal equations*

$$\beta_0 \sum_{i=1}^{n} w_i + \beta_1 \sum_{i=1}^{n} w_i X_i = \sum_{i=1}^{n} w_i Y_i$$
$$\beta_0 \sum_{i=1}^{n} w_i X_i + \beta_1 w_i X_i^2 = \sum_{i=1}^{n} w_i X_i Y_i.$$

and are given by

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} w_i (X_i - \bar{X}_w)(Y_i - \bar{Y}_w)}{\sum_{i=1}^{n} w_i (X_i - \bar{X}_w)^2}$$

and

$$\hat{\beta}_0 = \bar{Y}_w - \hat{\beta}_1 \bar{X}_w,$$

where $\bar{X}_w$ and $\bar{Y}_w$ are the weighted sample means

$$\bar{X}_w = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i} \qquad \text{and} \qquad \bar{Y}_w = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}.$$

The matrix approach can also be applied to weighted least squares. Define the $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$ matrices as in Section 3.4:

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

In addition, assume that the matrix $\mathbf{W}$ is a diagonal matrix with the weights $w_1, w_2, \ldots, w_n$ on the diagonal:

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}.$$

In this case, the normal equations can be written in matrix form as

$$\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{Y}.$$

Pre-multiplying both sides of this equation by $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ gives the least squares estimators for the regression parameters in matrix form as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.$$

As before, the fitted values can also be written in matrix form as

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

or

$$\hat{\mathbf{Y}} = \mathbf{X}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.$$

The residuals $e_i = Y_i - \hat{Y}_i$ for $i = 1, 2, \ldots, n$, can also be written in matrix form as

$$\begin{aligned}\mathbf{e} &= \mathbf{Y} - \hat{\mathbf{Y}} \\ &= \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}} \\ &= \mathbf{Y} - \mathbf{X}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y} \\ &= \left(\mathbf{I} - \mathbf{X}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\right)\mathbf{Y},\end{aligned}$$

where $\mathbf{e}$ is the column vector of residuals $\mathbf{e} = (e_1, e_2, \ldots, e_n)'$. These matrix results are summarized in the following theorem.

Theorem 3.4 Let $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\beta}$, and $\boldsymbol{\varepsilon}$ be the matrices associated with a simple linear regression model with weights $w_1, w_2, \ldots, w_n$ associated with the data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$. Let $\mathbf{W}$ be an $n \times n$ diagonal matrix with the weights on the diagonal elements. The least squares estimators of β_0 and β_1 are

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.$$

The fitted values are

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}.$$

The residuals are

$$\mathbf{e} = \left(\mathbf{I} - \mathbf{X}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\right)\mathbf{Y}.$$

The algebraic approach, matrix approach, and R approach to weighted least squares problem will be illustrated in the next example. Establishing the weights $w_1, w_2, \ldots, w_n$ can be a nontrivial problem, and differs depending on the setting in which the weighted regression model is employed.

Example 3.11 In reliability, *current status data* is generated by testing a randomly selected group of items with varying ages from a population at a particular fixed time in order to determine whether or not each item has failed or is operating at its particular age. Items were selected at ages 100, 200, 300, and 400 hours to see if they are operating. In this case, the independent variable X is the age, measured in hours, at which an item is tested. Each item tested is deemed to be either operating or failed. Table 3.7 contains the results of the test. Notice that 100 items were tested at ages $X_1 = 100$ and $X_2 = 200$, but only 10 items were tested at ages $X_3 = 300$ and $X_4 = 400$. The dependent variable in this setting is the fraction of items that survive to a particular age. The sample size at each testing age is denoted by n_i, $i = 1, 2, 3, 4$. So a total of $n_1 + n_2 + n_3 + n_4 = 220$ items were tested. The number of items that are operating at each testing age is denoted by S_i, $i = 1, 2, 3, 4$. The fraction of items that are operating at each testing age, which is the dependent variable in the regression, is denoted by Y_i, $i = 1, 2, 3, 4$. Notice that the fraction surviving is not necessarily decreasing from one time to the next because of random sampling variability. The small sample sizes at

Time (hours)	$X_1 = 100$	$X_2 = 200$	$X_3 = 300$	$X_4 = 400$
Sample size	$n_1 = 100$	$n_2 = 100$	$n_3 = 10$	$n_4 = 10$
Number surviving	$S_1 = 50$	$S_2 = 25$	$S_3 = 4$	$S_4 = 3$
Fraction surviving	$Y_1 = 0.5$	$Y_2 = 0.25$	$Y_3 = 0.4$	$Y_4 = 0.3$

Table 3.7: Current status data test results.

times $X_3 = 300$ and $X_4 = 400$ magnify this problem with the data set. The goal here is to establish a regression function that will adequately smooth the data values in order to estimate the survivor function for the items at any time.

Assume for now that the standard (non-weighted) least squares approach using the $n = 4$ data pairs

$$(100, 0.5), \qquad (200, 0.25), \qquad (300, 0.4), \qquad \text{and} \qquad (400, 0.3)$$

is taken to this problem. The R code below fits the simple linear regression model to the data.

```
x   = c(100, 200, 300, 400)
n   = c(100, 100,  10,  10)
s   = c( 50,  25,   4,   3)
y   = s / n
fit = lm(y ~ x)
fit$coefficients
```

The regression line in this case has intercept $\hat{\beta}_0 = 0.475$ and slope $\hat{\beta}_1 = -0.00045$. The survival probability of a brand-new item is estimated to be 0.475, and the survival probability decreases by 0.00045 for every hour that passes. The unimpressive survival probability of 0.475 for a new item is outside of the scope of the simple linear regression model, so its interpretation is not meaningful.

But using the standard simple linear regression approach is not appropriate here. The first two data pairs, both of which involved testing 100 items, should be weighted more heavily that the last two data pairs, which only involved testing 10 items. Determining the appropriate weights, however, is nontrivial.

Assume that the test results for each item are mutually independent Bernoulli trials. The number of items that survive a test at one particular time (that is, S_i using the notation from Table 3.7) is a binomial random variable with parameters n_i and p_i, where p_i is the population probability that item i is operating at time X_i. The population variance of the dependent variable $Y_i = S_i/n_i$ is

$$V[\hat{p}_i] = V[Y_i] = V\left[\frac{S_i}{n_i}\right] = \frac{1}{n_i^2} V[S_i] = \frac{n_i p_i (1-p_i)}{n_i^2} = \frac{p_i(1-p_i)}{n_i},$$

for $i = 1, 2, 3, 4$. Using the point estimate for p_i on the right-hand side of this expression results in the following estimated variances for the four dependent variables:

$$\hat{V}[Y_1] = \frac{\frac{50}{100}\left(1 - \frac{50}{100}\right)}{100} = \frac{1}{400}, \qquad \hat{V}[Y_2] = \frac{\frac{25}{100}\left(1 - \frac{25}{100}\right)}{100} = \frac{3}{1600},$$

$$\hat{V}[Y_3] = \frac{\frac{4}{10}\left(1-\frac{4}{10}\right)}{10} = \frac{24}{1000}, \qquad \hat{V}[Y_4] = \frac{\frac{3}{10}\left(1-\frac{3}{10}\right)}{10} = \frac{21}{1000}.$$

Not surprisingly, the first two variance estimates are about an order of magnitude smaller than the second two variance estimates because of the differences in the sample sizes. This approach will have problems if one of the testing times has all successes ($S_i = n_i$) or all failures ($S_i = 0$).

Since the weights w_i appear in the denominator of the expression $V[\varepsilon_i] = \sigma_Z^2/w_i$, the reciprocals of these variance estimates will be used as the weights in the weighted least squares regression:

$$w_1 = \frac{400}{1}, \qquad w_2 = \frac{1600}{3}, \qquad w_3 = \frac{1000}{24}, \qquad w_4 = \frac{1000}{21}.$$

The regression coefficients will be calculated in three ways, all of which yield identical results: the algebraic approach, the matrix approach, and using the `lm` function.

First, the algebraic approach for calculating the slope and intercept of the regression line using weighted least squares uses the following R statements. These are an implementation of Theorem 3.3.

```
x     = c(100, 200, 300, 400)
n     = c(100, 100,  10,  10)
s     = c( 50,  25,   4,   3)
y     = s / n
w     = n / (y * (1 - y))
meanx = sum(w * x) / sum(w)
meany = sum(w * y) / sum(w)
slope = sum(w * (x - meanx) * (y - meany)) / (sum(w * (x - meanx) ^ 2))
inter = meany - slope * meanx
print(c(inter, slope))
```

The weighted mean of the X values is

$$\bar{X}_w = \frac{\sum_{i=1}^n w_i X_i}{\sum_{i=1}^n w_i} = 174.2725.$$

Notice that this is slightly lower than the unweighted mean of the x values, which is $(100+200+300+400)/4 = 250$ hours. This is due to the larger sample sizes at testing times 100 and 200, resulting in larger weights for these values. The weighted mean of the Y values is

$$\bar{Y}_w = \frac{\sum_{i=1}^n w_i Y_i}{\sum_{i=1}^n w_i} = 0.3562.$$

The estimates for the slope and intercept of the regression line for weighted least squares is

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n w_i (X_i - \bar{X}_w)(Y_i - \bar{Y}_w)}{\sum_{i=1}^n w_i (X_i - \bar{X}_w)^2} = -0.001081$$

and

$$\hat{\beta}_0 = \bar{Y}_w - \hat{\beta}_1 \bar{X}_w = 0.5447.$$

The interpretation of these estimates is that the estimated probability of survival at time 0 is 0.5447 and the probability of survival decreases by 0.001081 with every hour that passes.

Second, using the matrix approach, the $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{W}$ matrices associated with this data set are

$$\mathbf{X} = \begin{bmatrix} 1 & 100 \\ 1 & 200 \\ 1 & 300 \\ 1 & 400 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 0.50 \\ 0.25 \\ 0.40 \\ 0.30 \end{bmatrix}, \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} \frac{400}{1} & 0 & 0 & 0 \\ 0 & \frac{1600}{3} & 0 & 0 \\ 0 & 0 & \frac{1000}{24} & 0 \\ 0 & 0 & 0 & \frac{1000}{21} \end{bmatrix}.$$

The R code below uses the matrix approach to simple linear regression with weights to calculate the estimated slope $\hat{\beta}_0$ and intercept $\hat{\beta}_1$, the fitted values $\hat{\mathbf{Y}}$, and the residuals $\mathbf{e}$ for the current status data set using Theorem 3.4. The R `solve` function is used to compute the inverse of $\mathbf{X}'\mathbf{X}$.

```
options(digits = 4)
x      = c(100, 200, 300, 400)
n      = c(100, 100,  10,  10)
s      = c( 50,  25,   4,   3)
y      = s / n
w      = n / (y * (1 - y))
w      = diag(w)
x      = cbind(1, x)
beta   = solve(t(x) %*% w %*% x) %*% t(x) %*% w %*% y
fitted = x %*% beta
e      = y - fitted
```

The results of these calculations are given below. The point estimators of the slope and intercept are

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y} = \begin{bmatrix} 0.5447 \\ -0.001081 \end{bmatrix}.$$

The fitted values are

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \begin{bmatrix} 0.4365 \\ 0.3284 \\ 0.2203 \\ 0.1121 \end{bmatrix}.$$

The residuals are

$$\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}} = \begin{bmatrix} 0.0635 \\ -0.0784 \\ 0.1797 \\ 0.1879 \end{bmatrix}.$$

Third, the built-in function `lm` can be used for weighted least squares by using the `weights` argument. The R code below calculates the estimates of the regression coefficients, the fitted values, and the residuals.

```
x    = c(100, 200, 300, 400)
n    = c(100, 100,  10,  10)
```

```
s    = c( 50,  25,   4,   3)
y    = s / n
w    = n / (y * (1 - y))
fitw = lm(y ~ x, weights = w)
print(fitw$coefficients)
print(fitw$fitted.values)
print(fitw$residuals)
print(weighted.residuals(fitw))
```

The three approaches all yield the same results. The regression line associated with ordinary least squares and weighted least squares can be compared graphically. The R code below plots the four data pairs and the associated ordinary least squares and weighted least squares regression lines.

```
x    = c(100, 200, 300, 400)
n    = c(100, 100,  10,  10)
s    = c( 50,  25,   4,   3)
y    = s / n
fit  = lm(y ~ x)
w    = n / (y * (1 - y))
fitw = lm(y ~ x, weights = w)
plot(x, y)
abline(fit$coefficients)
abline(fitw$coefficients)
```

Figure 3.20 contains the resulting plot, which shows the ordinary least squares line with equal weighting to the four data values and the weighted least squares line with much more weight to the first two data pairs and much less weight to the last two data pairs. Extra circles have been added to the two data pairs associated with the larger sample sizes with larger weights in Figure 3.20. The effect of the larger weights on the first

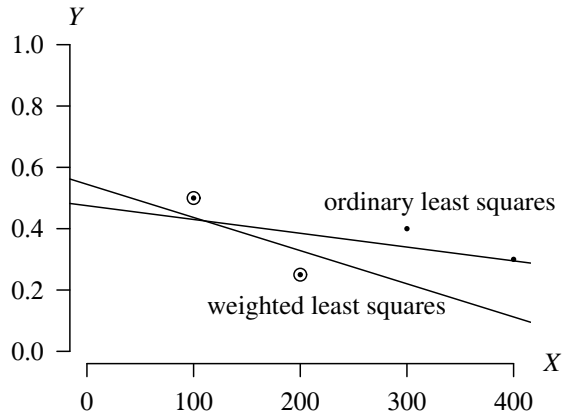

Figure 3.20: Current status data ordinary and weighted least squares fits.

two data pairs is apparent in the weighted least squares regression line. The rightmost two data pairs exert significantly less tug on the weighted least squares regression line because of their smaller weights.

Using simple linear regression in the previous example, either weighted or unweighted, might not be the best approach. The dependent variable Y is the probability that an item of age X is functioning. This dependent variable must lie between 0 and 1, but the regression line could potentially fall outside of that range within the scope of the model. Two potential remedies are given in the next two sections: using a regression model with nonlinear terms such as X^2 or X^3, or a survivor function of a lifetime model rather than a line, or a nonlinear model known as a *logistic regression model*, whose dependent variable necessarily lies between 0 and 1.

3.7 Regression Models with Nonlinear Terms

Regression models with nonlinear terms arise frequently in regression modeling. One simple example is polynomial regression. A quadratic regression model, for example, is

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon,$$

where β_0, β_1, and β_2 are the regression coefficients, and ε is a white noise term. This model is still linear in β_0, β_1, and β_2. One way to think about this model is to consider X and X^2 to be the $p = 2$ independent variables in a multiple regression model. The next example fits a quadratic model to the data pairs in which the independent variable X is the speed of an automobile and the dependent variable Y is its stopping distance.

Example 3.12 Consider the $n = 50$ data pairs from Example 2.8 which give the speed (in miles per hour) as X and the stopping distance (in feet) as Y. These data pairs are built into the base R language in the data frame named `cars`, where the `speed` column contains the values of X and the `dist` column contains the values of Y. Fit a quadratic regression model forced through the origin to the data pairs.

Since the quadratic regression model is being forced through the origin in order to account for the fact that stationary cars ($X = 0$) do not require any distance ($Y = 0$) to stop, the quadratic regression model is

$$Y = \beta_1 X + \beta_2 X^2 + \varepsilon,$$

where $\varepsilon \sim WN\left(0, \sigma_Z^2\right)$. R is capable of fitting nonlinear models to data. The `I` (inhibit interpretation) function allows the modeling of some function of a particular independent variable. For the data pairs in the `cars` data frame, a quadratic regression model that is forced through the origin can be fit with `lm` function.

```
fit = lm(dist ~ speed + I(speed ^ 2) - 1, data = cars)
```

The `-1` part of the formula forces the regression function to pass through the origin. The output generated by the `summary(fit)` statement is given below.

```
Call:
lm(formula = dist ~ speed + I(speed^2) - 1, data = cars)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-28.836  -9.071  -3.152   4.570  44.986

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
speed       1.23903    0.55997   2.213  0.03171 *
I(speed^2)  0.09014    0.02939   3.067  0.00355 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 15.02 on 48 degrees of freedom
Multiple R-squared:  0.9133,    Adjusted R-squared:  0.9097
F-statistic: 252.8 on 2 and 48 DF,  p-value: < 2.2e-16
```

The fitted quadratic regression model that is forced to pass through the origin is

$$Y = 1.24X + 0.0901X^2,$$

where X is speed and Y is stopping distance. Notice that $\hat{\beta}_2 = 0.0901 > 0$, which means that a graph of the fitted regression function—a parabola that passes through the origin—is concave up. Since the p-value associated with the linear term is $p = 0.032$ and the p-value associated with the quadratic term is $p = 0.0036$, both of the regression coefficients are statistically significant. The R commands

```
plot(cars, xlim = c(0, 25), pch = 16, las = 1)
fit = lm(dist ~ speed + I(speed ^ 2) - 1, data = cars)
x   = 0:25
y   = fit$coefficients[1] * x + fit$coefficients[2] * x ^ 2
lines(x, y)
```

plot the fitted model over the scatterplot. This graph appears in Figure 3.21.

How do we compare the simple linear regression model to the quadratic regression model forced through the origin? Both have two parameters, but which one of the models is a better approximation to the data pairs? One way to compare the two models is with the sum of squared residuals for each of the models, which are computed with the R commands

```
sum(lm(dist ~ speed, data = cars)$residuals ^ 2)
sum(lm(dist ~ speed + I(speed ^ 2) - 1, data = cars)$residuals ^ 2)
```

The simple linear regression model has a sum of squared residuals of $S = 11{,}353.52$, and the quadratic regression model forced through the origin has a sum of squared residuals of $S = 10{,}831.12$. Using the quadratic regression model forced through the origin reduces the sum of squared residuals by 522.4. Higher-order polynomials can be fit using the `lm` function in a similar manner. As was the case in multiple regression, adding more terms generally results in a reduction in the sum of squared residuals.

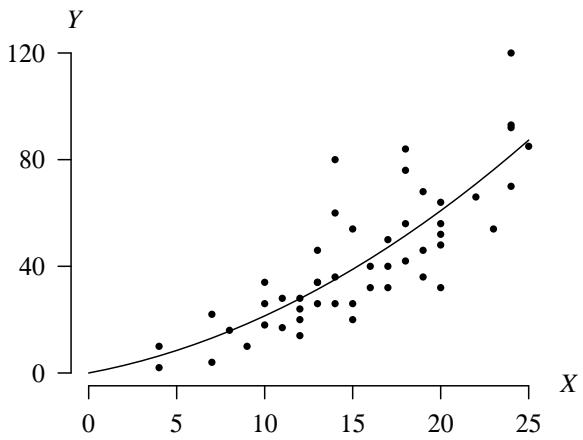

Figure 3.21: Scatterplot and quadratic fit of speed X and stopping distance Y.

Nonlinear regression modeling is not limited to just polynomial regression models. The next two examples fit the same data set concerning the national debt in the United States between 1970 and 2020 to a nonlinear regression model using two fundamentally different approaches. The first approach is to transform the nonlinear regression model to a linear regression model and then apply the standard techniques for parameter estimation to the transformed model. The second approach is to use numerical methods to minimize the sum of squares in the usual least squares fashion described previously.

Example 3.13 The national debt of the United States, in trillions of dollars, between 1970 and 2020 is given in Table 3.8. These values are not adjusted for inflation. Fit an exponential regression model to the national debt of the United States, where X is the year and Y is the debt, by transforming an exponential regression model to a linear model.

Year	Debt
1970	0.37
1975	0.53
1980	0.91
1985	1.82
1990	3.23
1995	4.97
2000	5.67
2005	7.93
2010	13.56
2015	18.15
2020	27.75

Table 3.8: United States national debt, 1970–2020.

The scatterplot in Figure 3.22 shows that a simple linear regression model is not appro-

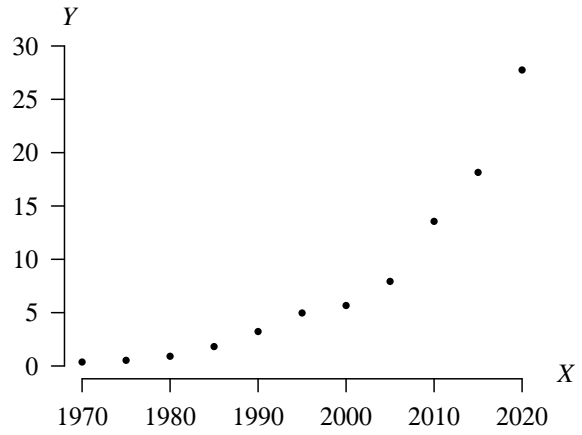

Figure 3.22: Scatterplot of the year X and the national debt Y.

priate for these data pairs. A regression model that reflects the exponential growth rate in the debt is warranted. Both savings and debt tend to grow exponentially, so an exponential regression model is a reasonable initial model to investigate. Consider fitting the regression model

$$Y = e^{\beta_0+\beta_1 X+\varepsilon}$$

to the data set, where X is the year, Y is the debt, ε is an error term, and β_0 and β_1 are unknown regression parameters to be estimated from the data pairs. This model can be transformed to a linear model by taking the natural logarithm of both sides of the model:

$$\ln Y = \beta_0 + \beta_1 X + \varepsilon.$$

This model is now in the form of a simple linear regression with independent variable X and dependent variable $\ln Y$. The intercept of the fitted model is β_0 and the slope of the fitted model is β_1. So a graph that contains X on the horizontal axis and $\ln Y$ on the vertical axis should be approximately linear if this transformation approach is appropriate. Such a graph is given in Figure 3.23, which is much closer to linear than the raw data points. It is apparent that some work on debt reduction occurred in the late 1990s, resulting in a slight bit of nonlinearity. We will proceed with fitting the transformed model. The R code below follows a similar pattern to the earlier examples, but this time the `formula` used in the call to the `lm` function is `log(debt) ~ year`. The `curve` function is used to add the fitted regression function to the scatterplot.

```
year = seq(1970, 2020, by = 5)
debt = c(.37, .53, .91, 1.82, 3.23, 4.97, 5.67, 7.93, 13.56, 18.15, 27.75)
fit  = lm(log(debt) ~ year)                    # fit an exponential model
b0   = coef(fit)[1]                            # estimated beta0 value
b1   = coef(fit)[2]                            # estimated beta1 value
plot(year, debt, las = 1, pch = 16)            # scatterplot of data pairs
curve(exp(b0 + b1 * x), add = TRUE)            # plot fitted model
```

The fitted model is displayed in Figure 3.24. The values of the estimated parameters are $\hat{\beta}_0 = -170.4$ and $\hat{\beta}_1 = 0.08606$.

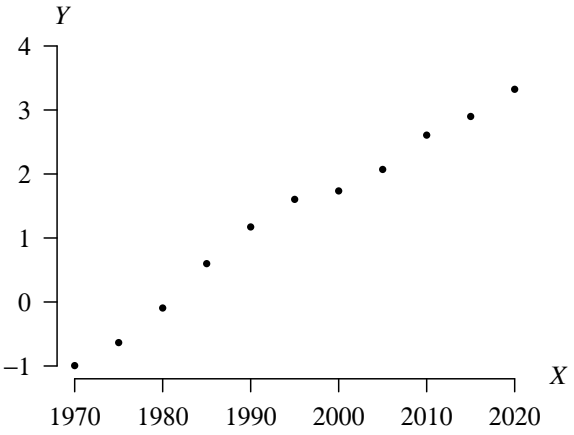

Figure 3.23: Scatterplot of the year X and the logarithm of the national debt Y.

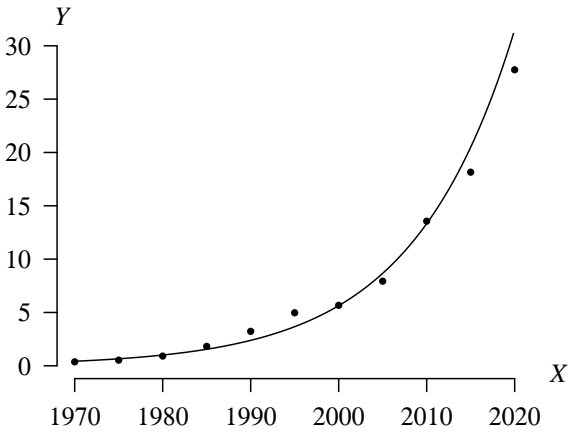

Figure 3.24: Scatterplot and exponential fit of year X and debt Y.

There is a second approach to fitting an exponential regression model to the national debt data pairs that follows the standard approach to least squares estimation, which is given next.

Example 3.14 Fit an additive exponential regression model to the United States national debt data pairs from Example 3.13.

The second approach to fitting an exponential regression model to the debt data pairs is to use the additive model

$$Y = e^{\beta_0+\beta_1 X} + \varepsilon.$$

Using the traditional least squares approach, the sum of squares

$$S = \sum_{i=1}^{n} \left(Y_i - e^{\beta_0+\beta_1 X_i} \right)^2$$

is minimized with respect to β_0 and β_1, yielding the associated least squares estimators $\hat{\beta}_0$ and $\hat{\beta}_1$. The estimators cannot be expressed in closed form, so numerical methods must be used to estimate β_0 and β_1. A nonlinear least squares R function named `nls` can be used to estimate the parameters. Here is a first attempt at fitting the model.

```
year = seq(1970, 2020, by = 5)
debt = c(.37, .53, .91, 1.82, 3.23, 4.97, 5.67, 7.93, 13.56, 18.15, 27.75)
fit  = nls(debt ~ exp(b0 + b1 * year))             # fit exponential model
```

This code returns an error message indicating that the `nls` function was unable to estimate the parameters. What went wrong? The way that the model has been formulated, the parameter e^{β_0} represents the United States national debt in the year 0. This is why we had the parameter estimate $e^{\hat{\beta}_0} = e^{-170.4} = 10^{-74}$ from the transformation approach in Example 3.13. The `nls` function attempts to do a search over all values of β_0 and β_1 to minimize the sum of squares. Finding the value of $\hat{\beta}_0$ is like finding a needle in a haystack. We need to give the `nls` function some help. We will give `nls` some starting values in a list named `start` to make the internal search performed by the `nls` function easier. The initial values for $\hat{\beta}_0$ and $\hat{\beta}_1$ will be the estimates for β_0 and β_1 from the transformation approach from the previous example.

```
year = seq(1970, 2020, by = 5)
debt = c(.37, .53, .91, 1.82, 3.23, 4.97, 5.67, 7.93, 13.56, 18.15, 27.75)
fit  = nls(debt ~ exp(b0 + b1 * year), start = list(b0 = -170, b1 = 0.09))
b0   = coef(fit)[1]                              # fitted beta0 value
b1   = coef(fit)[2]                              # fitted beta1 value
plot(year, debt, pch = 16)                       # scatterplot of data pairs
curve(exp(b0 + b1 * x), add = TRUE)              # plot fitted model
```

The estimated parameters are $\hat{\beta}_0 = -151.7$ and $\hat{\beta}_1 = 0.07676$. Thus, the fitted nonlinear regression model is

$$E[Y] = e^{\hat{\beta}_0 + \hat{\beta}_1 X}.$$

The fitted exponential regression model is displayed in Figure 3.25. The two different exponential regression models can be compared by computing the sums of squares for the two models, which can be computed by the *additional* R command

```
sum((debt - exp(b0 + b1 * year)) ^ 2)            # calculate sum of squares
```

The sum of squares for fitting the exponential regression model using the transformation technique is 22.7 and the sum of squares for the nonlinear least squares is 3.1. So consistent with Figures 3.24 and 3.25, the nonlinear least squares approach provides a better fit to the data pairs.

One drawback that emerged from the survival function estimation example from the previous section (involving current status data) is that fitting a regression line results in a survival probability that can be negative or greater than one when extrapolated outside of the range of the independent variable in the data pairs. In addition, the estimated probability of survival at time zero for both the ordinary simple linear regression model and the weighted simple linear regression model seemed low. Typically, a brand-new item is not defective. A nonlinear regression function is an attractive alternative model in this particular setting. The next example combines a nonlinear regression model and weighted least squares estimators to provide an improved regression model.

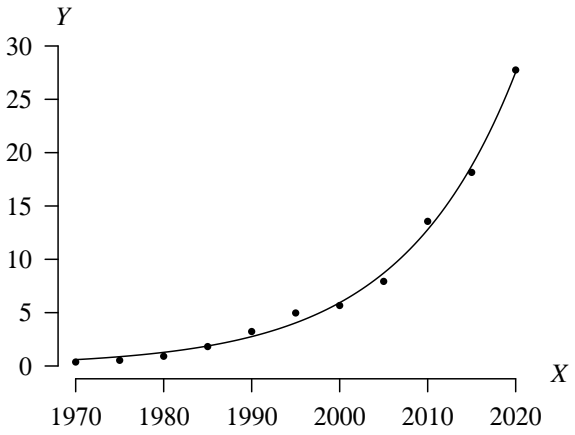

Figure 3.25: Scatterplot and exponential fit of year X and debt Y.

Example 3.15 Consider again the estimate of the probability of survival from the current status data given in Example 3.11. A simple nonlinear model that might be appropriate for the current status data set is to assume that the lifetime of the item under consideration follows the exponential(λ) distribution. The survivor function for an exponential random variable T with positive failure rate λ is

$$S(t) = P(T \geq t) = e^{-\lambda t} \qquad t > 0,$$

where t is the failure time in hours.

(a) Fit this nonlinear regression model using ordinary least squares.

(b) Fit this nonlinear regression model using weighted least squares.

(c) Compare the two fitted regression models.

(a) There are two ways to proceed with this regression problem. The first is to minimize the squared deviations

$$S = \sum_{i=1}^{n} \left(Y_i - e^{-\lambda X_i} \right)^2$$

with respect to λ to arrive at an appropriate regression parameter estimator. Equivalently, the least squares estimator of λ is

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \sum_{i=1}^{n} \left(Y_i - e^{-\lambda X_i} \right)^2.$$

This is the usual least squares approach. The second is to perform algebraic manipulations to the model in order to "linearize" the model so that the theory associated with the simple linear regression model can be implemented. The second approach is considered here. Treating this as a regression problem with X as time

and Y as the survival probability results in the multiplicative nonlinear regression model

$$Y = e^{-\lambda X}\varepsilon.$$

Taking the natural logarithm of both sides of this model results in

$$\ln Y = -\lambda X + \varepsilon$$

or

$$-\ln Y = \lambda X + \varepsilon.$$

(Notice that when the error distribution is symmetric, which is often the case, the last step is justified.) This can be thought of as a linear regression problem with X as the independent variable and $-\ln Y$ as the dependent variable. There is no intercept in this model, so it can be treated as forcing the regression line through the origin and the single regression parameter λ corresponds to the slope of the regression line.

The R code below uses unweighted least squares to estimate the slope λ using the algebraic approach that forces the regression line through the origin using the techniques from Section 3.1.

```
x      = c(100, 200, 300, 400)
n      = c(100, 100,  10,  10)
s      = c( 50,  25,   4,   3)
y      = s / n
logy   = -log(y)
lamhat = sum(x * logy) / sum(x * x)
```

The R code using the matrix approach is identical to the algebraic approach in this case. Likewise, the regression parameter λ can be estimated using the `lm` function with the `- 1` parameter to force the regression through the origin via the code below.

```
x    = c(100, 200, 300, 400)
n    = c(100, 100,  10,  10)
s    = c( 50,  25,   4,   3)
y    = s / n
logy = -log(y)
lm(logy ~ x - 1)$coefficients
```

Using any of these approaches to estimating λ, the estimate for the failure rate is

$$\hat{\lambda} = 0.003677$$

failures per hour.

(b) For the current status data set, it is sensible to incorporate the weights that are based on the various sample sizes into the regression model.

The algebraic and the matrix approach to the nonlinear weighted least squares model, which will be a regression model forced through the origin, have identical R code, which is given below.

```
x    = c(100, 200, 300, 400)
n    = c(100, 100,  10,  10)
s    = c( 50,  25,   4,   3)
y    = s / n
w    = n / (y * (1 - y))
logy = -log(y)
sum(x * w * logy) / sum(x * w * x)
```

The R code using the `lm` function to estimate the parameter λ is given below.

```
x      = c(100, 200, 300, 400)
n      = c(100, 100,  10,  10)
s      = c( 50,  25,   4,   3)
y      = s / n
w      = n / (y * (1 - y))
logy   = -log(y)
lamhat = lm(logy ~ x - 1, weights = w)$coefficients
```

Regardless of which approach is taken, the least squares estimate for the failure rate is

$$\hat{\lambda} = 0.005721$$

failures per hour, which is slightly higher than the estimated failure rate in the ordinary least squares approach.

(c) The two approaches (ordinary least squares and weighted least squares) for the nonlinear regression model can be compared graphically by plotting the two estimated survivor functions associated with the two fitted models. The R code below generates that plot. The estimated failure rate in the case of ordinary nonlinear least squares is stored in `lambda.ols`. The estimated failure rate in the case of weighted nonlinear least squares is stored in `lambda.wls`.

```
x          = c(100, 200, 300, 400)
n          = c(100, 100,  10,  10)
s          = c( 50,  25,   4,   3)
y          = s / n
logy       = -log(y)
lambda.ols = lm(logy ~ x - 1)$coefficients
w          = n / (y * (1 - y))
lambda.wls = lm(logy ~ x - 1, weights = w)$coefficients
xx         = 0:400
plot(x, y, xlim = c(0, 400), ylim = c(0, 1))
lines(xx, exp(-lambda.ols * xx))
lines(xx, exp(-lambda.wls * xx))
```

Figure 3.26 contains the graph. The ordinary least squares fit with $\hat{\lambda} = 0.003677$ gives equal weight to the four data pairs; the weighted least squares fit with $\hat{\lambda} = 0.005721$ gives significantly more weight to the first two data pairs. The two data pairs with the larger sample sizes are again circled in the figure. The weighted least squares model indicates that there is a higher estimated failure rate when increased weight is placed on the first two data pairs.

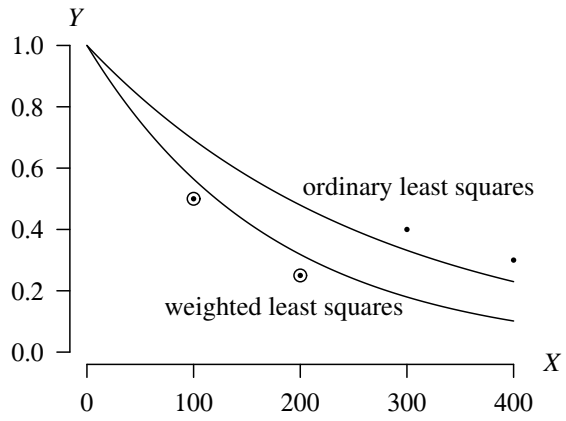

Figure 3.26: Current status data ordinary and weighted least squares fits for the exponential model.

3.8 Logistic Regression

Logistic regression is appropriate when the dependent variable Y can assume one of two values: zero and one. This is sometimes known as a *binary* or *dichotomous* response variable. For now, to keep the mathematics and interpretations simple, assume that there is a single predictor X. This is known as a *simple logistic regression model*, and is a special type of nonlinear regression model. Including multiple independent variables in a logistic regression model is a straightforward extension. For dichotomous data, instead of predicting 0 or 1, we predict the probability of getting a 1 [that is, $P(Y = 1)$]. So we need a regression model that predicts values of the interval $[0, 1]$.

The following example will be used throughout this section to motivate the need for a special model to accommodate a binary dependent variable, and to illustrate the techniques for the estimation of the model parameters.

> **Example 3.16** As an example to motivate the application of simple logistic regression, consider the $n = 948$ field goal attempts in the National Football League during the 2003 season. Let the independent variable X be the length of the field goal attempt (in yards) and the dependent variable Y be the outcome (0 for failure and 1 for success). The scatterplot (without jittering for ties) of the data values is shown in Figure 3.27, along with the associated least squares regression line with estimated intercept $\hat{\beta}_0 = 1.35$ and slope $\hat{\beta}_1 = -0.015$. The regression line is heading in the correct direction because longer field goals are less likely to be successful. Simple linear regression is clearly not an appropriate statistical model in this setting because it predicts probabilities outside of the interval $[0, 1]$. Even if predictions greater than 1 are set to 1 and negative predictions are set to zero, the model predicts that all 20-yard field goal attempts will be successful, and, at the other extreme, it predicts that the probability of kicking an 85-yard field goal is 0.06. This is inconsistent with the fact that the longest field goal ever in the NFL was a 66-yard field goal by Justin Tucker of the Baltimore Ravens on September 26, 2021. Obviously we can build a better regression model.

One of the initial considerations in developing a statistical model for the outcome of a field goal as a function of the length of the field goal attempt is to find a function that will only assume values

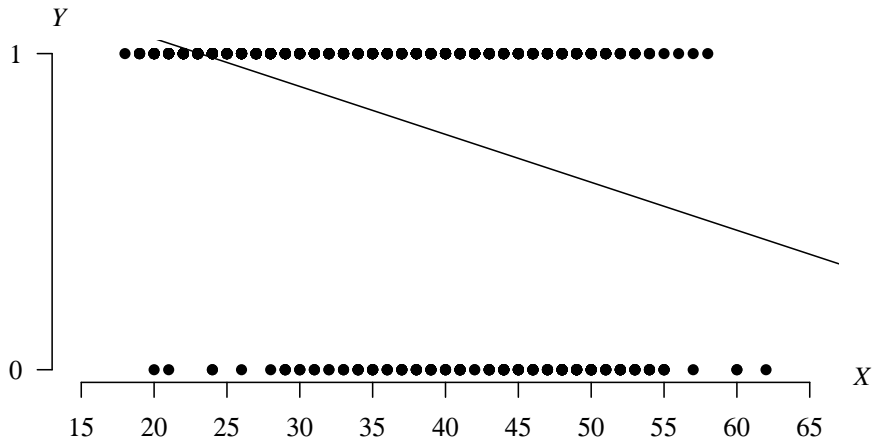

Figure 3.27: Scatterplot of field goal outcomes vs. yards with regression line.

between 0 and 1. A diagram that gives some guidance with regard to this function is to batch the data into 5-year increments. So the bins are all field goals that fall in the ranges 20 ± 2, 25 ± 2, ..., 60 ± 2. This window is long enough so that the random sampling variability associated with nearby attempts is damped considerably, and yet short enough so that outcome patterns as a function of yardage are still apparent. The R code below batches the independent variable into the 5-yard increments and plots the estimated probability of success for attempts in each batch at its midpoint. This estimated probability is just the fraction of successful field goals within a particular range. Furthermore, the area of each point plotted is proportional to the number of attempts in that particular bin. For example, there were 79 attempts in the first bin (18–22 yards) and only 4 attempts in the last bin (58–62 yards). The R code below reads a data set off of the web that contains the results of $n = 948$ NFL field goal attempts during 2003. The data consists of columns that give the length of the field goal attempt and the outcome, failure ($Y = 0$) or success ($Y = 1$). The R code rounds each length to the nearest 5 yards, and plots the midpoint of the rounded field goal lengths versus the estimated probability of success.

```
df = read.table("http://users.stat.ufl.edu/~winner/data/fieldgoal.dat")
yards = df[, 1]
outcome = df[, 2]
plot(NA, xlim = c(15, 65), ylim = c(0, 1))
yards = floor((yards + 2) / 5) * 5
for (i in 1:9) points(5 * i + 15, mean(outcome[yards / 5 - 3 == i]),
                      pch = 16, cex = 0.12 * sqrt(table(yards / 5 - 3)[i]))
```

While the performance of NFL field goal kickers varies from one kicker to the next, these points give us an idea of what we would like for a smooth regression function in this setting.

The results are shown in Figure 3.28. It is clear that the estimated probability of making a field goal decreases as the length of the field goal attempt increases, as one would expect. There is a strong relationship between the length of the field goal attempt and the probability of success. Our goal is to fit a nonlinear regression function to the raw data values that smooths the random sampling variability and can be used for the purpose of prediction.

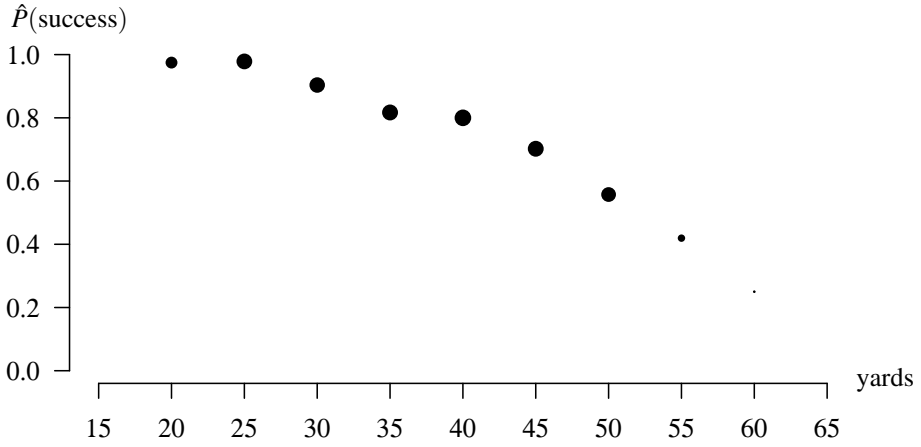

Figure 3.28: Field goal outcomes vs. yards in 5-yard increments.

When the dependent variable only takes on the values zero and one, the usual mean response function for the simple linear regression model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

is

$$E[Y] = \beta_0 + \beta_1 X,$$

where $E[Y]$ denotes the conditional expected value of Y given a particular setting of the independent variable X. This mean response function does not limit the values of Y to just zero and one. With normally distributed error terms, this model would allow for Y values which could be less than 0 or greater than 1.

In logistic regression, this type of curve, regardless of whether it begins near one and ends near zero or it begins near zero and ends near one, is known as a *sigmoidal* response function. A natural choice for the sigmoidal response function is a cumulative distribution function associated with a random variable, or its complement (the survivor function). Three popular probability distributions whose cumulative distribution functions are used in logistic regression are the standard logistic distribution (also commonly called the *logit* model), the standard normal distribution (also commonly called the *probit* model), and the standard extreme value distribution (also commonly called the complimentary *log-log* model). These are described in the next paragraph.

The standard logistic distribution has probability density function

$$f(x) = \frac{e^x}{(1+e^x)^2} \qquad -\infty < x < \infty$$

and cumulative distribution function

$$F(x) = \frac{e^x}{1+e^x} \qquad -\infty < x < \infty.$$

The probability density function is symmetric about the population mean $E[X] = 0$ and has population variance $V[X] = \pi^2/3$. The standard normal distribution has probability density function

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \qquad -\infty < x < \infty$$

and cumulative distribution function

$$F(x) = \int_{-\infty}^{x} f(w)\,dw \qquad\qquad -\infty < x < \infty.$$

The probability density function is also symmetric about the population mean $E[X] = 0$ and has population variance $V[X] = 1$. The probability density function for the standard logistic distribution is similar in shape (that is, bell-shape) to that for the standard normal distribution, but has heavier tails. The symmetry of the probability density functions for the standard logistic distribution and the standard normal distribution limits the shape of the associated cumulative distribution function. A nonsymmetric distribution often provides a better fit. This leads to a search for a probability distribution with a nonsymmetric probability density function. One such probability distribution is the extreme value distribution. The standard extreme value distribution has probability density function

$$f(x) = e^{x - e^x} \qquad\qquad -\infty < x < \infty$$

and cumulative distribution function

$$F(x) = 1 - e^{-e^x} \qquad\qquad -\infty < x < \infty.$$

The population mean and the population variance are not mathematically tractable, but the numeric values, to ten digits, are

$$E[X] = -0.5772156649 \qquad \text{and} \qquad V[X] = 1.644934067.$$

The probability density function is not symmetric about the mean.

The R code below plots these three probability density functions on the same set of axes. The standard normal probability density function is taken directly from the formulas in the previous paragraph. The probability density functions for the standard logistic distribution and the standard extreme value distribution have been standardized (by subtracting their population mean and dividing by the population standard deviation) so that all three probability density functions can be viewed on an equal footing. The plot emphasizes the shape of the various probability density functions.

```
x = seq(-3, 3, by = 0.01)
k = pi / sqrt(3)
y = k * exp(k * x) / (1 + exp(k * x)) ^ 2
plot(x, y, type = "l", xlim = c(-3, 3), ylim = c(0, 0.5))
lines(x, dnorm(x))
mu = -0.5772156649
sig = sqrt(1.644934067)
y = sig * exp(mu + sig * x - exp(mu + sig * x))
lines(x, y)
```

The results are displayed in Figure 3.29. All three probability distributions have support on the entire real number line $-\infty < x < \infty$, although the graph only includes the values within three standard deviation units from the population mean. As expected, the probability density functions for the standard normal distribution and the standardized version of the standard logistic distribution are symmetric and bell-shaped. The probability density function of the standardized version of the standard extreme value distribution is nonsymmetric. The R code below plots the cumulative distribution function associated with the standardized version of the standard logistic distribution.

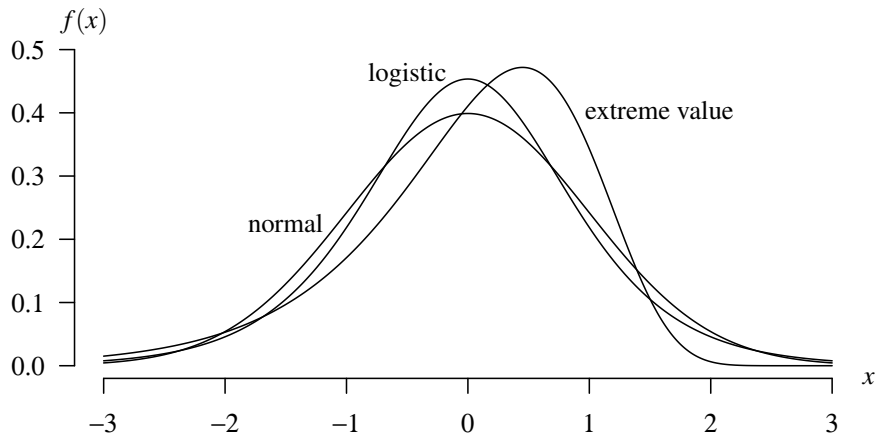

Figure 3.29: Standardized logistic, normal, and extreme value probability density functions.

```
x = seq(-3, 3, by = 0.01)
k = pi / sqrt(3)
y = exp(k * x) / (1 + exp(k * x))
plot(x, y, type = "l", xlim = c(-3, 3), ylim = c(0, 1))
```

The cumulative distribution function $F(x) = P(X \leq x)$ is graphed in Figure 3.30. This cumulative distribution function is monotone increasing and satisfies $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$. Notice that a plot of $F(-x)$ gives the complement of the cumulative distribution function. In other words, $S(x) = 1 - F(x) = P(X \geq x)$. This function is monotone decreasing and satisfies $\lim_{x \to -\infty} S(x) = 1$ and $\lim_{x \to \infty} S(x) = 0$. This function is known in survival analysis as the *survivor function*.

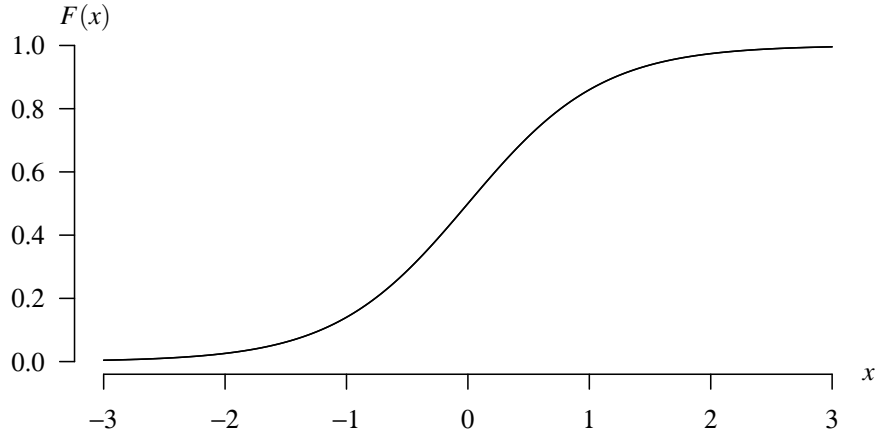

Figure 3.30: Standardized version of the standard logistic cumulative distribution function.

Now that cumulative distribution functions and their complements have been identified as a reasonable way to estimate the probability of success for the field goal data, we would like to establish a mechanism for incorporating the value of the predictor X into the probability model. The emphasis here will be on using the cumulative distribution function for the logistic distribution, since that seems to be the most commonly used in logistic regression.

The usual form of the mean response function for simple linear regression is

$$E[Y] = \beta_0 + \beta_1 X.$$

But in the case of a binary outcome, the constraint

$$0 \leq E[Y] \leq 1$$

must be imposed. This is done naturally using the cumulative distribution functions and their complements for the various probability distributions described earlier. Let $\pi(X)$ be the mean response function for a regression model with a binary response. Using the cumulative distribution function for the logistic distribution, the mean response function is

$$\pi(X) = E[Y] = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

Since the random variable Y can only assume the values 0 and 1 for a particular value of X, it is a Bernoulli random variable with probability of success $\pi(X)$. Since the expected value and the probability that a Bernoulli random variable assumes the value 1 are equal, the mean response function can also be expressed as

$$\pi(X) = P(Y=1) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}},$$

where $P(Y=1)$ is the probability that the dependent variable Y equals 1 for a particular fixed setting of the independent variable X. The parameters β_0 and β_1 assume the following roles.

- The sign of β_1 controls whether the mean response function is monotone increasing or decreasing. Table 3.9 shows the direction of the relationship associated with the sign of β_1. The statistical significance of the point estimator of β_1 depends on its magnitude.
- The magnitude of β_1 controls the steepness of the mean response function, with larger magnitudes corresponding to steeper mean response functions.
- The value of β_0 controls the location of the mean response function on the X-axis.

A graph that illustrates the effect of varying values of β_1 for the fixed value of $\beta_0 = 0$ on the mean response function $\pi(X)$ is given in Figure 3.31. As expected, the mean response function $\pi(X)$

Condition	$\lim_{X\to-\infty} \pi(X)$	$\lim_{X\to\infty} \pi(X)$
$\beta_1 < 0$	1	0
$\beta_1 > 0$	0	1
$\beta_1 = 0$	$e^{\beta_0}/\left(1+e^{\beta_0}\right)$	$e^{\beta_0}/\left(1+e^{\beta_0}\right)$

Table 3.9: Direction of monotonicity of $\pi(X)$.

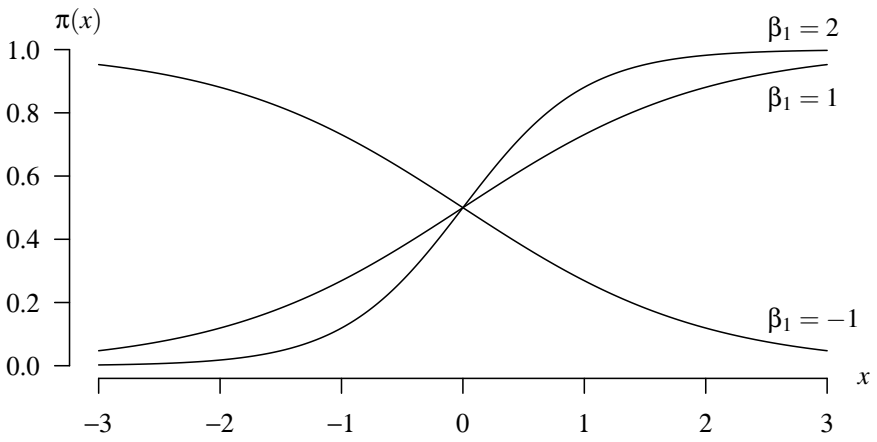

Figure 3.31: Mean response functions for $\beta_0 = 0$ and various β_1 values.

is monotone decreasing for $\beta_1 < 1$ and monotone increasing for $\beta_1 > 1$. The mean response function is steeper as the magnitude of β_1 increases.

A graph that illustrates the effect of varying values of β_0 for the fixed value of $\beta_1 = 1$ on the mean response function $\pi(X)$ is given in Figure 3.32. As expected, the mean response function $\pi(X)$ is monotone increasing in all cases because $\beta_1 > 1$. The effect of varying β_0 is to shift the mean response functions horizontally. The rationale behind the horizontal shift can be seen by writing the mean response function with $\beta_1 = 1$ as

$$\pi(X) = \frac{e^{\beta_0 + X}}{1 + e^{\beta_0 + X}}.$$

So the effect of increasing β_0 in this case is to shift the mean response function to the right (for

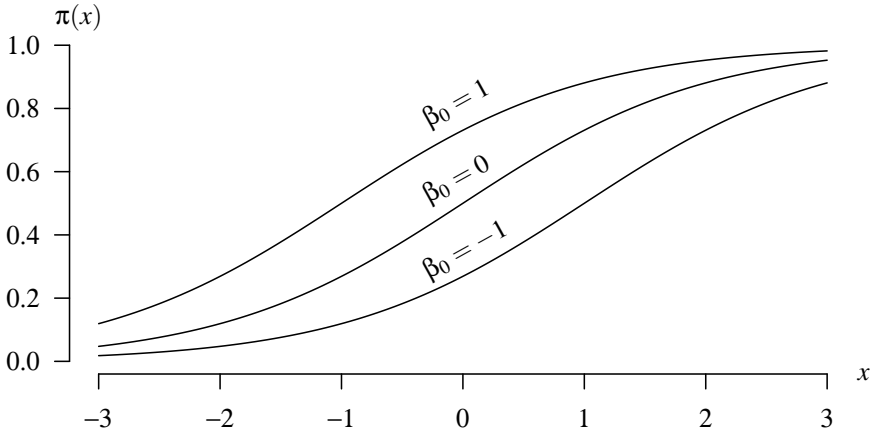

Figure 3.32: Mean response functions for $\beta_1 = 1$ and various β_0 values.

$\beta_1 < 1$) or to the left (for $\beta_1 > 1$) relative to the $\pi(X)$ curve associated with $\beta_0 = 0$.

To summarize, the sign of β_1 controls the direction of the monotonicity of $\pi(X)$, the magnitude of β_1 controls the steepness of $\pi(X)$, and β_0 controls the location of $\pi(X)$ along the X-axis.

We now consider the estimation of the parameters β_0 and β_1 from a data set consisting of the n data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$. The first components $X_1, X_2, \ldots, X_n$ are real numbers and the second components $Y_1, Y_2, \ldots, Y_n$ assume only the values 0 and 1. Since

$$P(Y = 1) = \pi(X) \qquad \text{and} \qquad P(Y = 0) = 1 - \pi(X)$$

the contribution to the likelihood function of the data pair (X_i, Y_i) is

$$\pi(X_i)^{Y_i} [1 - \pi(X_i)]^{1-Y_i}$$

for $i = 1, 2, \ldots, n$. When $Y_i = 0$, the contribution to the likelihood function is $1 - \pi(X_i)$, which is $P(Y_i = 0)$, where $P(Y_i = 0)$ is the probability that $Y_i = 0$ for the particular setting of the independent variable at X_i. When $Y_i = 1$, the contribution to the likelihood function is $\pi(X_i)$, which is $P(Y_i = 1)$. Since X_i is assumed to be observed without error, Y_i is a random binary response, and the responses are assumed to be mutually independent random variables, the likelihood function is

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} \pi(X_i)^{Y_i} [1 - \pi(X_i)]^{1-Y_i}.$$

The log likelihood function is

$$\ln L(\beta_0, \beta_1) = \sum_{i=1}^{n} Y_i \ln \big[\pi(X_i)\big] + (1 - Y_i) \ln \big[1 - \pi(X_i)\big].$$

This can be written in terms of β_0 and β_1 as

$$\ln L(\beta_0, \beta_1) = \sum_{i=1}^{n} Y_i \left[\beta_0 + \beta_1 X_i - \ln\left(1 + e^{\beta_0 + \beta_1 X_i}\right)\right] - (1 - Y_i) \ln\left(1 + e^{\beta_0 + \beta_1 X_i}\right)$$

or

$$\ln L(\beta_0, \beta_1) = \sum_{i=1}^{n} Y_i (\beta_0 + \beta_1 X_i) - \ln\left(1 + e^{\beta_0 + \beta_1 X_i}\right).$$

The likelihood function and the log likelihood function are maximized at the same values of β_0 and β_1 because the natural logarithm is a monotonic transformation. The score vector is comprised of the partial derivatives of the log likelihood function with respect to β_0 and β_1:

$$\frac{\partial \ln L(\beta_0, \beta_1)}{\partial \beta_0} = \sum_{i=1}^{n} \left(Y_i - \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}}\right)$$

and

$$\frac{\partial \ln L(\beta_0, \beta_1)}{\partial \beta_1} = \sum_{i=1}^{n} \left(X_i Y_i - \frac{X_i e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}}\right).$$

When these two equations are equated to zero, there is no closed form solution for $\hat{\beta}_0$ and $\hat{\beta}_1$, so numerical methods must be relied on to calculate these point estimates. The second derivatives of the log likelihood function after simplification are

$$\frac{\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_0^2} = -\sum_{i=1}^{n} \frac{e^{\beta_0 + \beta_1 X_i}}{\left(1 + e^{\beta_0 + \beta_1 X_i}\right)^2},$$

$$\frac{\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_0 \partial \beta_1} = -\sum_{i=1}^{n} \frac{X_i e^{\beta_0+\beta_1 X_i}}{\left(1+e^{\beta_0+\beta_1 X_i}\right)^2},$$

and

$$\frac{\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_1^2} = -\sum_{i=1}^{n} \frac{X_i^2 e^{\beta_0+\beta_1 X_i}}{\left(1+e^{\beta_0+\beta_1 X_i}\right)^2}.$$

The Fisher information matrix is the matrix of expected values of these partial derivatives:

$$I(\beta_0, \beta_1) = \begin{pmatrix} E\left[\dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_0^2}\right] & E\left[\dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_0 \beta_1}\right] \\ E\left[\dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_1 \beta_0}\right] & E\left[\dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_1^2}\right] \end{pmatrix}.$$

The expected values in this matrix can be determined because they do not contain any random variables. Their values cannot be calculated, however, because the values of the parameters β_0 and β_1 are unknown. The observed information matrix

$$O\left(\hat{\beta}_0, \hat{\beta}_1\right) = \left(\begin{array}{cc} \dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_0^2} & \dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_0 \beta_1} \\ \dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_1 \beta_0} & \dfrac{-\partial^2 \ln L(\beta_0, \beta_1)}{\partial \beta_1^2} \end{array} \right)_{\beta_0 = \hat{\beta}_0, \beta_1 = \hat{\beta}_1}$$

can be estimated from data values once the maximum likelihood estimators are computed. This matrix is the variance–covariance matrix of the score vector and its inverse is the asymptotic variance–covariance matrix of the maximum likelihood estimators. The square roots of the diagonal elements of this inverse matrix provide estimates of the standard errors of the maximum likelihood estimates.

The NFL field goal data set has a large sample size ($n = 948$) and a strong statistical relationship between the length of the field goal attempt and the probability of success. The R code below again uses the `optim` function to calculate the parameter estimates. The first argument to `optim` are initial parameter estimates. The second argument to `optim` is the function to be *minimized*, so the negative of the log likelihood function is given as the second argument. Once the maximum likelihood estimates are calculated, the observed information matrix, standard errors, z-statistics, and associated p-values are calculated.

```
df = read.table("http://users.stat.ufl.edu/~winner/data/fieldgoal.dat")
yards = df[, 1]
outcome = df[, 2]
logl = function(parameters) {
  beta0 = parameters[1]
  beta1 = parameters[2]
  sum(-outcome * (beta0 + beta1 * yards) + log(1 + exp(beta0 + beta1 * yards)))
}
fit         = optim(c(0, -1), logl)
beta0hat    = fit$par[1]
beta1hat    = fit$par[2]
oim         = matrix(0, 2, 2)
oim[1, 1]   = sum(exp(beta0hat + beta1hat * yards) /
                  (1 + exp(beta0hat + beta1hat * yards)) ^ 2)
```

```
oim[1, 2]   = sum(yards * exp(beta0hat + beta1hat * yards) /
                  (1 + exp(beta0hat + beta1hat * yards)) ^ 2)
oim[2, 1]   = oim[1, 2]
oim[2, 2]   = sum(yards * yards * exp(beta0hat + beta1hat * yards) /
                  (1 + exp(beta0hat + beta1hat * yards)) ^ 2)
print(oim)
se.beta0hat = sqrt(solve(oim)[1, 1])
se.beta1hat = sqrt(solve(oim)[2, 2])
z0          = beta0hat / se.beta0hat
z1          = beta1hat / se.beta1hat
p0          = 2 * (1 - pnorm(abs(z0)))
p1          = 2 * (1 - pnorm(abs(z1)))
print(c(beta0hat, se.beta0hat, z0, p0))
print(c(beta1hat, se.beta1hat, z1, p1))
```

The results of the code are summarized in Table 3.10. The values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are both statistically significant with p-values near zero. The observed information matrix for the NFL field goal data set

i	$\hat{\beta}_i$	$\hat{\sigma}_{\hat{\beta}_i}$	z	p
0	5.69	0.451	12.6	0.00
1	-0.110	0.0106	-10.4	0.00

Table 3.10: Summary statistics for NFL field goal data.

is

$$O\left(\hat{\beta}_0, \hat{\beta}_1\right) = \begin{pmatrix} 130.83 & 5470.26 \\ 5470.26 & 237,653.57 \end{pmatrix}.$$

These values can be compared to the values obtained using the `glm` (generalized linear model) function:

```
df = read.table("http://users.stat.ufl.edu/~winner/data/fieldgoal.dat")
yards = df[, 1]
outcome = df[, 2]
fit = glm(outcome ~ yards, family = binomial(link = logit))
summary(fit)
```

The results match those given in Table 3.10. When the `link` parameter within the `binomial` family is set to `logit`, the cumulative distribution function (or its complement) for the standard logistic distribution is employed. When the `link` parameter is set to `probit`, the cumulative distribution function (or its complement) for the standard normal distribution is employed. The logit and probit choices force the sigmoidal function to be symmetric, so that it approaches 0 and 1 at the same rate. When the `link` parameter is set to `cloglog`, the cumulative distribution function (or its complement) for the standard extreme value distribution is employed. It approaches 0 and 1 at the different rates.

When the following R statements are added to the code that generated Figure 3.28, the fitted mean response function $\hat{\pi}(X)$ is added to the graph.

```
x = seq(15, 65, by = 0.1)
beta0hat = 5.6942693
```

```
beta1hat = -0.1098488
y = exp(beta0hat + beta1hat * x) / (1 + exp(beta0hat + beta1hat * x))
lines(x, y)
```

The graph is shown in Figure 3.33. The estimated mean response function is monotone decreasing because $\hat{\beta}_1 < 0$. Furthermore, the mean response curve does an adequate job of modeling the probability of success as the points lie very close to the estimated mean response function. The estimated

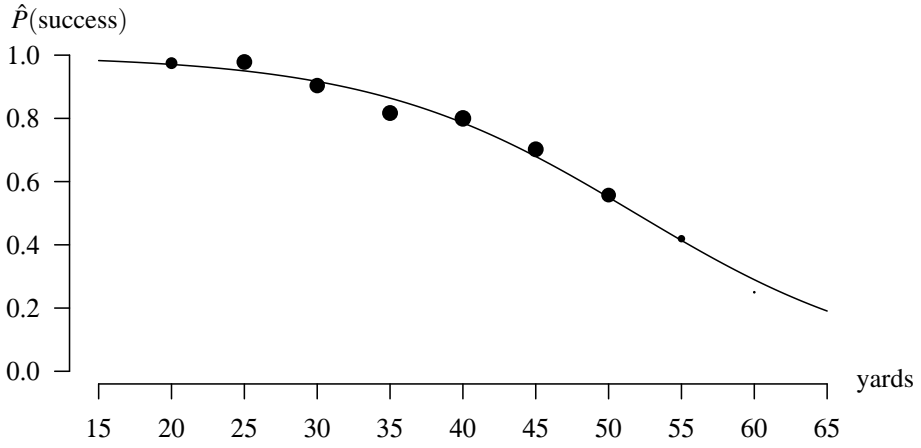

Figure 3.33: Field goal outcomes and estimated mean response function.

mean response function can be used for prediction. The estimated probability that a 38-yard field goal attempt is successful is

$$\hat{\pi}(38) = \frac{e^{5.6942693 - 0.1098488(38)}}{1 + e^{5.6942693 - 0.1098488(38)}} = 0.82.$$

This value can be generated with the `predict` function in R with the *additional* statements

```
linear = predict(fit, newdata = data.frame(yards = 38))
exp(linear) / (1 + exp(linear))
```

Some keystrokes can be saved by using the `type = "response"` argument in the call to `predict`.

```
predict(fit, newdata = data.frame(yards = 38), type = "response")
```

The limitations of a symmetric mean response function also become apparent in this case. The estimated probability that a 71-yard field goal attempt is successful is

$$\hat{\pi}(71) = \frac{e^{5.6942693 - 0.1098488(71)}}{1 + e^{5.6942693 - 0.1098488(71)}} = 0.11,$$

even though the NFL field goal record from 2021 is 66 yards. This is clearly a case of extrapolating beyond the range of the data, which is discouraged. The meaningful range of $\hat{\pi}(X)$ is over the scope of the model $18 \leq X \leq 62$, whose endpoints are the shortest and longest field goal attempt during the 2003 season. The symmetric nature of the logistic distribution makes the $\hat{\pi}(X)$ values associated with X-values greater than 62 yards higher than are meaningful.

Confidence intervals for the parameters in a logistic regression model can be calculated with the `confint` and `confint.default` functions. These confidence intervals give a measure of the precision of the point estimates. The R code below calculates the 95% confidence intervals for the parameters using the `confint` and `confint.default` functions for the NFL field goal data.

```
df = read.table("http://users.stat.ufl.edu/~winner/data/fieldgoal.dat")
yards = df[, 1]
outcome = df[, 2]
fit = glm(outcome ~ yards, family = binomial(link = logit))
confint(fit)
confint.default(fit)
```

The first set of confidence intervals that are returned via `confint` use the profiled log likelihood function to return the confidence intervals given in the output below. The default is a 95% confidence interval.

```
                2.5 %      97.5 %
(Intercept)  4.8435441  6.61425072
yards       -0.1312492 -0.08970744
```

To three significant digits, these 95% confidence intervals are

$$4.84 < \beta_0 < 6.61 \qquad \text{and} \qquad -0.131 < \beta_1 < -0.0897.$$

The second set of confidence intervals that are returned via `confint.default` are based on the asymptotic normality of the maximum likelihood estimators. The call to `confint.default` returns the confidence intervals given in the output below.

```
                2.5 %      97.5 %
(Intercept)  4.8137433  6.58201706
yards       -0.1306527 -0.08916745
```

To three significant digits, these 95% confidence intervals are

$$4.82 < \beta_0 < 6.58 \qquad \text{and} \qquad -0.131 < \beta_1 < -0.0892.$$

Alternatively, the 95% confidence interval for β_1 can be calculated by using the `qnorm` function to calculate the appropriate quantile from the standard normal distribution.

```
df = read.table("http://users.stat.ufl.edu/~winner/data/fieldgoal.dat")
yards = df[, 1]
outcome = df[, 2]
fit = glm(outcome ~ yards, family = binomial(logit))
coef(fit)[2] + c(-1, 1) * qnorm(0.975) * summary(fit)$coefficients[2, 2]
```

The 95% confidence interval for β_1 that is returned matches that returned by `confint.default`. The confidence intervals based on the asymptotic normality of the maximum likelihood estimator from `confint.default` will be symmetric about the maximum likelihood estimators, but the confidence interval based on the profiled log likelihood function from `confint` will not be symmetric about the maximum likelihood estimators. The confidence intervals given here are somewhat narrow because of the large sample size of $n = 948$ for the NFL field goal data.

The last topic is the interpretation of the point estimators for the coefficients. This interpretation is much more difficult than the interpretation of the coefficients in a standard simple linear regression model. The next paragraph defines the odds and the log odds. The subsequent paragraph relates the log odds to the logistic regression model.

Consider an event which occurs with probability 0.9. The probability that the event will not occur is 0.1. The odds are defined as the ratio of the probability that the event will occur to the probability that the event will not occur. In this case that ratio is 9, so the odds are often referred to as 9 to 1. Table 3.11 gives several probability values and associated odds for several probability values.

Probability	Odds
0.2	0.25
0.5	1
0.6	1.5
0.75	3
0.8	4
0.9	9
0.99	99

Table 3.11: Probability and odds.

The R code below generates a graph of the odds on the vertical axis versus the probability on the horizontal axis.

```
prob = seq(0, 0.9, by = 0.01)
odds = prob / (1 - prob)
plot(prob, odds, type = "l", xlim = c(0, 1), ylim = c(0, 9))
```

Figure 3.34 shows the transformation from probability to odds, which reveals a monotone increasing function. Probabilities fall on the interval $[0, 1]$; odds fall on the interval $[0, \infty)$. The natural loga-

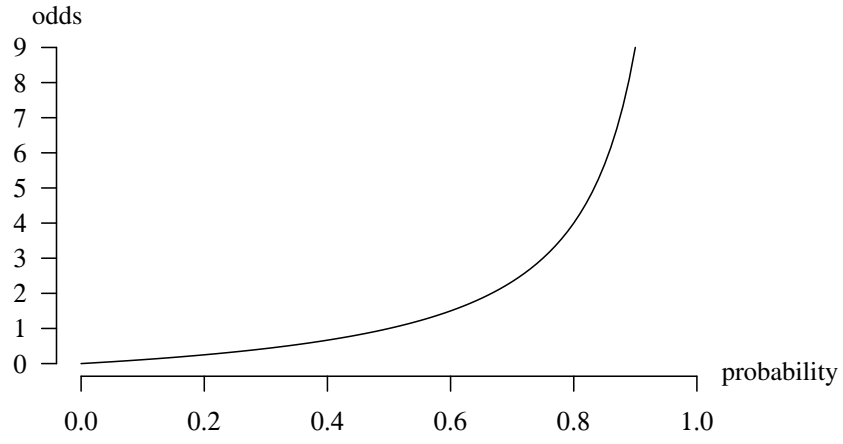

Figure 3.34: Odds versus probability.

rithm of the odds is the function, known as the log odds, which is a transformation of the probability p in the following fashion:

$$\ln\left(\frac{p}{1-p}\right).$$

This is a transformation from $[0, 1]$ to $(-\infty, \infty)$. Table 3.12 extends the previous table by including a column for the log odds. Notice that a probability of $1/2$ corresponds to a log odds of 0 and the symmetry of the log odds associated with the probabilities 0.2 and 0.8. The R code below graphs

Probability	Odds	Log Odds
0.2	0.25	−1.3863
0.5	1	0
0.6	1.5	0.4055
0.75	3	1.0986
0.8	4	1.3863
0.9	9	2.1972
0.99	99	4.5951

Table 3.12: Probability, odds, and log odds.

the log odds versus the probability.

```
prob    = seq(0.045, 0.955, by = 0.001)
odds    = prob / (1 - prob)
logodds = log(odds)
plot(prob, logodds, type = "l", xlim = c(0, 1), ylim = c(-3, 3))
```

The associated graph is shown in Figure 3.35. The shape of the log odds is a transformed version of the mean response functions seen earlier. The purpose of defining the log odds is to convert from probability, which has a restricted range between 0 and 1, and the log odds, which has an unrestricted range.

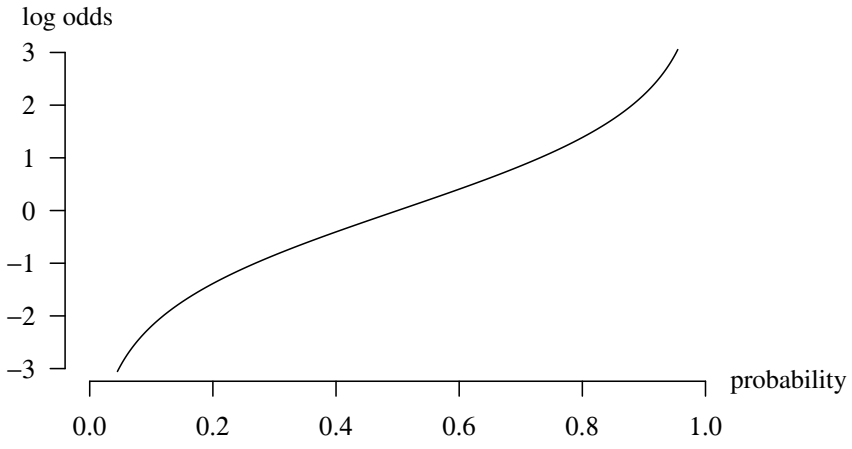

Figure 3.35: Log odds versus probability.

Now back to logistic regression and the interpretation of the estimated coefficients. Recall that for a simple logistic regression problem, the mean response function is

$$\pi(x) = E[Y \mid X = x] = P(Y = 1 \mid X = x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}},$$

where x is the independent variable and Y is the response variable. The *logit transformation* of $\pi(x)$ is

$$\ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \ln\left[e^{\beta_0+\beta_1 x}\right] = \beta_0 + \beta_1 x.$$

Since $\pi(x)$ is a probability, the expression on the left-hand side of this equation is a log odds.

Now consider the NFL data. From the earlier work, the estimated intercept provided by the R `glm` function is $\hat{\beta}_0 = 5.6979$ and the estimated coefficient associated with the length of the field goal attempt in yards is $\hat{\beta}_1 = -0.1099$. The estimated intercept is the log odds of a kicker making a field goal from a (theoretical) zero yards, which has no meaningful interpretation in this setting. The value of $\hat{\beta}_1 = -0.1099$ is the change in the log odds for a one-yard change in the length of the field goal attempt. Additionally, the quantity

$$e^{\hat{\beta}_1} = e^{-0.1099} = 0.8959$$

is the multiplier that gives the change in the odds for a one-unit change in the independent variable. We expect to see a 10.4% decrease in the odds associated with the probability of success for a field goal attempt for every additional yard added to the field goal attempt. This value and an associated 95% confidence interval can be generated with the *additional* R statement

```
exp(cbind(oddsratio = coef(fit), confint(fit)))
```

The analysis of the NFL data given here is a composite of all kickers in the NFL during 2003. Individual kickers within the NFL will have their own logistic regression curve.

With this background concerning simple logistic regression in place, it is straightforward to extend this to more complicated modeling situations. Additional topics in logistic regression include constructing a confidence interval for a predicted value, the calculation of deviance residuals, including multiple independent variables in a logistic regression model, model assessment, and interpreting estimated coefficients for interaction terms.

3.9 Exercises

3.1 Write a paragraph that describes why the sum of squares for error associated with the simple linear regression model $Y = \beta_0 + \beta_1 X + \varepsilon$ will always be less than or equal to the sum of squares for error associated with the simple linear regression model forced through the origin $Y = \beta_1 X + \varepsilon$ for the same data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$.

3.2 Under what condition(s) does the regression line forced through the origin have the same sum of squares for error as the simple linear regression for the full model $Y = \beta_0 + \beta_1 X + \varepsilon$ for the same data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$.

3.3 Consider the simple linear regression model forced through the origin

$$Y = \beta_1 X + \varepsilon.$$

Show that the least squares estimator $\hat{\beta}_1$ is an unbiased estimator of β_1.

3.4 Consider the simple linear regression model forced through the origin

$$Y = \beta_1 X + \varepsilon.$$

Find $V\left[\hat{\beta}_1\right]$.

3.5 Consider the simple linear regression model forced through the origin with normal error terms,

$$Y = \beta_1 X + \varepsilon,$$

where $\varepsilon \sim N\left(0, \sigma^2\right)$.

(a) Find the maximum likelihood estimators of β_1 and σ^2.

(b) Show that the maximum likelihood estimators maximize the log likelihood function.

3.6 Give an example of $n = 2$ data pairs corresponding to the case in which a simple linear regression line forced through the origin contains the point $(\bar{X}, \bar{Y})$.

3.7 Give an example of $n = 2$ data pairs corresponding to the case in which a simple linear regression line forced through the origin does not contain the point $(\bar{X}, \bar{Y})$.

3.8 Consider the simple linear regression model forced through the origin with normal error terms

$$Y = \beta_1 X + \varepsilon,$$

with *known* parameters β_1 and σ^2. Find an exact two-sided $100(1-\alpha)\%$ confidence interval for β_1 from n data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$.

3.9 Consider the simple linear regression model forced through the origin with normal error terms,

$$Y = \beta_1 X + \varepsilon,$$

with unknown parameters β_1 and σ^2. Show that the R statement

```
confint(lm(Formaldehyde$optden ~ Formaldehyde$carb - 1))
```

uses the formula

$$\hat{\beta}_1 - t_{n-1,\alpha/2}\sqrt{\frac{SSE}{(n-1)\sum_{i=1}^n X_i^2}} < \beta_1 < \hat{\beta}_1 + t_{n-1,\alpha/2}\sqrt{\frac{SSE}{(n-1)\sum_{i=1}^n X_i^2}}$$

to calculate the 95% two-sided confidence interval for β_1 for the data pairs in the built-in R data frame `Formaldehyde`. Notice that the degrees of freedom are one more than the associated degrees of freedom for the full simple linear regression model.

3.10 Consider the simple linear regression model forced through the origin with normal error terms,

$$Y = \beta_1 X + \varepsilon,$$

with unknown parameters β_1 and σ^2. Conduct a Monte Carlo simulation experiment to provide convincing numerical evidence that the two-sided $100(1-\alpha)\%$ confidence interval

$$\hat{\beta}_1 - t_{n-1,\alpha/2}\sqrt{\frac{SSE}{(n-1)\sum_{i=1}^n X_i^2}} < \beta_1 < \hat{\beta}_1 + t_{n-1,\alpha/2}\sqrt{\frac{SSE}{(n-1)\sum_{i=1}^n X_i^2}}$$

is an *exact* confidence interval for β_1 for the following parameter settings: $n = 3$, $\alpha = 0.05$, $\beta_1 = 2$, $X_1 = 1$, $X_2 = 2$, $X_3 = 3$, and $\sigma^2 = 1$.

3.11 The Brown–Forsythe test can be used to determine whether the error terms have constant variance. In particular, it tests for equality of the variances of the error terms in two subsets of the data values. The test is analogous to a t-test. The test is robust with respect to departures from normality of the error terms. The data pairs are partitioned by a threshold value of X which is not one of the $X_1, X_2, \ldots, X_n$ values. Let n_1 be the number of data pairs with X-values less than the threshold value and n_2 be the number of data pairs with X-values greater than the threshold value so that $n = n_1 + n_2$. In addition, let

- e_{i1} be residual i for group 1,
- e_{i2} be residual i for group 2,
- $\tilde{e}_1$ be the sample median of the group 1 residuals,
- $\tilde{e}_2$ be the sample median of the group 2 residuals,
- $d_{i1} = |e_{i1} - \tilde{e}_1|$,
- $d_{i2} = |e_{i2} - \tilde{e}_2|$,
- $\bar{d}_1 = (1/n_1)\sum_{i=1}^{n_1} d_{i1}$, and
- $\bar{d}_2 = (1/n_2)\sum_{i=1}^{n_2} d_{i2}$.

The test statistic for the Brown–Forsythe test is

$$t = \frac{\bar{d}_1 - \bar{d}_2}{s\sqrt{1/n_1 + 1/n_2}},$$

where s^2 is the pooled sample variance

$$s^2 = \frac{\sum_{i=1}^{n_1}\left(d_{i1} - \bar{d}_1\right)^2 + \sum_{i=1}^{n_2}\left(d_{i2} - \bar{d}_2\right)^2}{n-2}.$$

The test statistic is approximately $t(n-2)$ when the population variances of the error terms in the two groups are equal n_1 and n_2 are large enough so that the dependency between the residuals is not too large. Write R code to compute the p-value for the Brown–Forsythe test for the `cars` data set using `speed` as the independent variable and `dist` as the dependent variable with a threshold value of 13.5 miles per hour.

3.12 Find the leverages for $n = 2$ data pairs in a simple linear regression model.

3.13 For a simple linear regression model with $X_i = i$, for $i = 1, 2, \ldots, n$, derive a formula for the leverage of the ith data pair.

3.14 Write R functions named `cooks.distance1`, `cooks.distance2`, and `cooks.distance3`, which calculate the Cook's distances for each of the n data pairs associated with the simple linear regression model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

using the three formulas from Definition 3.3. The arguments for these three functions are the vector `x`, which contains the n values of the independent variable, and the vector `y`, which contains the n values of the dependent variable. Test your functions on the `Formaldehyde` data set which is built into R, with `carb` as the independent variable and `optden` as the dependent variable.

3.15 Make a scatterplot (with associated regression line) of the $n = 11$ data pairs in the third data set in Anscombe's quartet with the R commands

```
x = anscombe[ , 3]
y = anscombe[ , 7]
plot(x, y, xlim = c(4, 19), ylim = c(3, 13), pch = 16)
abline(lm(y ~ x))
```

Without doing any calculations,

(a) circle the point(s) with the largest leverage, and

(b) circle the point(s) with the largest Cook's distance.

3.16 What is the smallest and largest possible leverage?

3.17 Show that leverage is scale invariant. In other words, show that the leverages remain unchanged when the scale of the independent variable changes (for example, from centimeters to meters).

3.18 Use Monte Carlo simulation to estimate the probability that all of the Cook's distances are less than 1 for a simple linear regression model with normal error terms and the following parameter settings: $\beta_0 = 1$, $\beta_1 = 1/2$, $\sigma = 1$, $n = 10$, and $X_i = i$ for $i = 1, 2, \ldots, n$. Is this probability affected by changes is σ or n?

3.19 Use Monte Carlo simulation to draw empirical cumulative distribution functions of Cook's distances D_1, D_2, D_3, D_4, and D_5 for a simple linear regression model with the following parameter settings: $\beta_0 = 1$, $\beta_1 = 1/2$, $\sigma = 1$, $n = 10$, and $X_i = i$ for $i = 1, 2, \ldots, n$.

3.20 Consider a simple linear regression model with the independent variable X and the dependent variable Y having the same units (for example, centimeters). If the same linear transformation is applied to both X and Y so as to change their units (for example, from centimeters to meters), show that the Cook's distances remain unchanged.

3.21 Show that the row sums of the hat matrix are all equal to 1 for data pairs (X_1, Y_1), (X_2, Y_2), $\ldots$, (X_n, Y_n) in a simple linear regression model.

3.22 Perform a Monte Carlo simulation to provide convincing numerical evidence that

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})' \mathbf{X}'\mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{2 \cdot \text{MSE}} \sim F(2, n-2)$$

for a simple linear regression model with normal error terms of your choice. This result is used to establish a $100(1-\alpha)\%$ confidence region for β_0 and β_1.

3.23 Show that the residuals $e_i = Y_i - \hat{Y}_i$ for $i = 1, 2, \ldots, n$, can be written in terms of the hat matrix $\mathbf{H}$ as

$$\mathbf{e} = (\mathbf{I} - \mathbf{H})\mathbf{Y}.$$

3.24 For the simple linear regression model with normal error terms, the variance–covariance matrix of $\hat{\boldsymbol{\beta}}$ is

$$\sigma^2 \left(\mathbf{X}'\mathbf{X}\right)^{-1}.$$

For data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, give an estimator for this matrix.

3.25 For the simple linear regression model, show that

$$\mathbf{X}_h' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{X}_h = \frac{1}{n} + \frac{(X_h - \bar{X})^2}{S_{XX}}.$$

3.26 For a simple linear regression model, show that the matrix equation

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y},$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \qquad \text{and} \qquad \hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix},$$

corresponds to the normal equations given in Theorem 1.1 as

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} X_i = \sum_{i=1}^{n} Y_i$$
$$\hat{\beta}_0 \sum_{i=1}^{n} X_i + \hat{\beta}_1 \sum_{i=1}^{n} X_i^2 = \sum_{i=1}^{n} X_i Y_i.$$

3.27 A multiple linear regression model is used to determine the relationship between the sales price of a home Y as a function of the two predictor variables: X_1, the number of square feet in the home, and X_2, the distance from downtown in miles. The fitted model is

$$Y = 170,024 + 133X_1 - 14,123X_2.$$

One home sells for \$314,159. Find the predicted sales price for a second home, which is the same size as the first but is ten miles further away from downtown that the first home.

3.28 The R built-in data frame named `swiss` contains a standardized fertility measure and five socio-economic indicators for 47 French-speaking provinces in Switzerland from about 1888.

(a) Using a forward stepwise regression with threshold $\alpha = 0.05$, determine a multiple linear regression model with a dependent variable Y, the standardized fertility measure, and the five associated potential independent variables.

(b) Using a backward stepwise regression with threshold $\alpha = 0.05$, determine a multiple linear regression model with a dependent variable Y, the standardized fertility measure, and the five associated potential independent variables.

(c) For one of the two final multiple linear regression models determined in parts (a) and (b), test the statistical significance of all possible interaction terms.

3.29 Show that when the independent variables X_1 and X_2 in a multiple linear regression model are uncorrelated, the estimator for $\hat{\beta}_1$ is the same for both the simple linear regression model involving just X_1 and Y and the multiple linear regression model involving X_1, X_2, and Y.

3.30 Consider a simple linear regression model that uses the weighted least squares estimation. When all of the weights $w_1, w_2, \ldots, w_n$ are equal, show that the weighted least squares normal equations reduce to the associated unweighted least squares normal equations.

3.31 "I first believed I was dreaming ... but it is absolutely certain and exact that the ratio which exists between the period times of any two planets is precisely the ratio of the 3/2th power of the mean distance" was the reaction of Johannes Kepler upon discovering the relationship

$$y = \beta x^{3/2}$$

as translated from *Harmonies of the World* by Kepler in 1619, where x is the distance between a planet and the sun and y is the period. Using the data from the Wikipedia webpage titled *Kepler's Laws of Planetary Motion*, the data values for the $n = 8$ planets are given below.

Planet	Semi-major axis (AU) x	Period (days) y
Mercury	0.38710	87.9693
Venus	0.72333	224.7008
Earth	1	365.2564
Mars	1.52366	686.9796
Jupiter	5.20336	4332.8201
Saturn	9.53707	10,775.599
Uranus	19.1913	30,687.153
Neptune	30.0690	60,190.03

The semi-major axes values are measured in Astronomical Units (AU).

(a) Make an appropriate scatterplot to visually assess whether a regression model is appropriate.

(b) Find the least squares point estimate for β.

(c) Perhaps fit a least squares model in another fashion.

(d) Interpret the value for $\hat{\beta}$.

(e) Find a 95% confidence interval for β.

3.32 Fit the quadratic regression function forced through the origin

$$Y = \beta_1 X^2 + \varepsilon,$$

to the data pairs in the `cars` data frame in R, where X is the speed of the car in miles per hour and Y is the stopping distance in feet.

3.33 Using an extreme value distribution as a link function, fit a regression function to the 2003 NFL field goal data from Section 3.8 and use the fitted model to predict that probability of success on a 38-yard field goal attempt.