Multiple Linear Regression

G. Jay Kerns

September 9, 2011

1 Multiple Linear Regression

We know a lot about simple linear regression models, and a next step is to study multiple regression models that have more than one independent (explanatory) variable. In the discussion that follows we will assume that we have p explanatory variables, where $p > 1$.

The language is phrased in matrix terms – for two reasons. First, it is quicker to write and (arguably) more pleasant to read. Second, the matrix approach will be required for later study of the subject; the reader might as well be introduced to it now.

Most of the results are stated without proof or with only a cursory justification. Those yearning for more should consult an advanced text in linear regression for details, such as *Applied Linear Regression Models* [**?**]or *Linear Models: Least Squares and Alternatives* [**?**].

What do I want them to know?

- the basic MLR model, and how it relates to the SLR
- how to estimate the parameters and use those estimates to make predictions
- basic strategies to determine whether or not the model is doing a good job
- a few thoughts about selected applications of the MLR, such as polynomial, interaction, and dummy variable models
- some of the uses of residuals to diagnose problems
- hints about what will be coming later

1.1 The Multiple Linear Regression Model

The first thing to do is get some better notation. We will write

$$\mathbf{Y}_{\mathrm{n}\times 1} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \text{and} \quad \mathbf{X}_{\mathrm{n}\times(\mathrm{p}+1)} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{p1} \\ 1 & x_{12} & x_{22} & \cdots & x_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{pn} \end{bmatrix}. \tag{1.1.1}$$

The vector $\mathbf{Y}$ is called the *response vector* and the matrix $\mathbf{X}$ is called the *model matrix* . As in Chapter **??**, the most general assumption that relates $\mathbf{Y}$ to $\mathbf{X}$ is

$$\mathbf{Y} = \mu(\mathbf{X}) + \epsilon, \tag{1.1.2}$$

where μ is some function (the *signal*) and ϵ is the *noise* (everything else). We usually impose some structure on μ and ϵ. In particular, the standard multiple linear regression model assumes

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \tag{1.1.3}$$

where the parameter vector β looks like

$$\beta_{(p+1)\times 1} = \begin{bmatrix}\beta_0 & \beta_1 & \cdots & \beta_p\end{bmatrix}^{\mathrm{T}}, \tag{1.1.4}$$

and the random vector $\epsilon_{n\times 1} = \begin{bmatrix}\epsilon_1 & \epsilon_2 & \cdots & \epsilon_n\end{bmatrix}^{\mathrm{T}}$ is assumed to be distributed

$$\epsilon \sim \mathsf{mvnorm}\left(\mathtt{mean} = \mathbf{0}_{n\times 1},\ \mathtt{sigma} = \sigma^2\mathbf{I}_{n\times n}\right). \tag{1.1.5}$$

The assumption on ϵ is equivalent to the assumption that $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$ are i.i.d.˜norm(mean = 0, sd = σ). It is a linear model because the quantity $\mu(\mathbf{X}) = \mathbf{X}\beta$ is linear in the parameters $\beta_0, \beta_1, \ldots, \beta_p$. It may be helpful to see the model in expanded form; the above matrix formulation is equivalent to the more lengthy

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \epsilon_i, \quad i = 1, 2, \ldots, n. \tag{1.1.6}$$

Example 1.1. Girth, Height, and Volume for Black Cherry trees. Measurements were made of the girth, height, and volume of timber in 31 felled black cherry trees. Note that girth is the diameter of the tree (in inches) measured at 4 ft 6 in above the ground. The variables are

1. `Girth`: tree diameter in inches (denoted x_1)
2. `Height`: tree height in feet (x_2).
3. `Volume`: volume of the tree in cubic feet. (y)

The data are in the `datasets` package and are already on the search path; they can be viewed with

```
head(trees)
```

```
  Girth Height Volume
1   8.3     70   10.3
2   8.6     65   10.3
3   8.8     63   10.2
4  10.5     72   16.4
5  10.7     81   18.8
6  10.8     83   19.7
```

Let us take a look at a visual display of the data. For multiple variables, instead of a simple scatterplot we use a scatterplot matrix which is made with the `splom` function in the `lattice` package [**?**] as shown below. The plot is shown in Figure **??**.

```
library(lattice)
splom(trees)
```

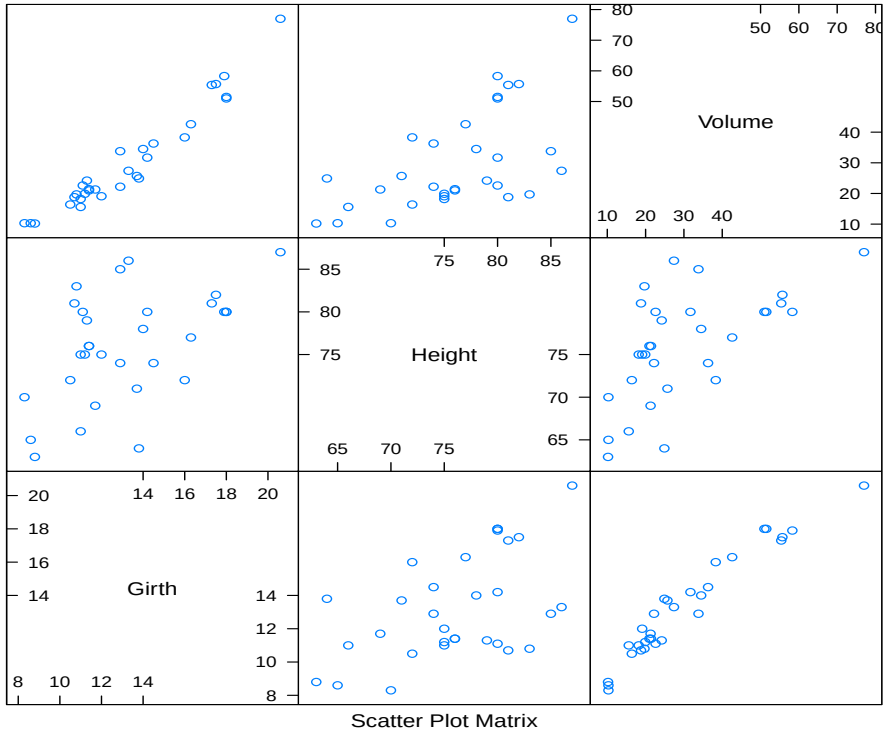

Figure 1.1.1: A scatterplot matrix of `trees` data.

The dependent (response) variable `Volume` is listed in the first row of the scatterplot matrix. Moving from left to right, we see an approximately linear relationship between

Volume and the independent (explanatory) variables Height and Girth. A first guess at a model for these data might be

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon, \tag{1.1.7}$$

in which case the quantity $\mu(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ would represent the mean value of Y at the point (x_1, x_2).

1.1.1 What does it mean?

The interpretation is simple. The intercept β_0 represents the mean Volume when all other independent variables are zero. The parameter β_i represents the change in mean Volume when there is a unit increase in x_i, while the other independent variable is held constant. For the trees data, β_1 represents the change in average Volume as Girth increases by one unit when the Height is held constant, and β_2 represents the change in average Volume as Height increases by one unit when the Girth is held constant.

In simple linear regression, we had one independent variable and our linear regression surface was 1D, simply a line. In multiple regression there are many independent variables and so our linear regression surface will be many-D... in general, a hyperplane. But when there are only two explanatory variables the hyperplane is just an ordinary plane and we can look at it with a 3D scatterplot.

One way to do this is with the R Commander in the Rcmdr package [**?**]. It has a 3D scatterplot option under the Graphs menu. It is especially great because the resulting graph is dynamic; it can be moved around with the mouse, zoomed, *etc*. But that particular display does not translate well to a printed book.

Another way to do it is with the scatterplot3d function in the scatterplot3d package. The code follows, and the result is shown in Figure **??**.

```
library(scatterplot3d)
s3d <- with(trees, scatterplot3d(Girth, Height, Volume,
                                 pch = 16, highlight.3d = TRUE,
                                 angle = 60))
fit <- lm(Volume ~ Girth + Height, data = trees)
```

Looking at the graph we see that the data points fall close to a plane in three dimensional space. (The plot looks remarkably good. In the author's experience it is rare to see points fit so well to the plane without some additional work.)

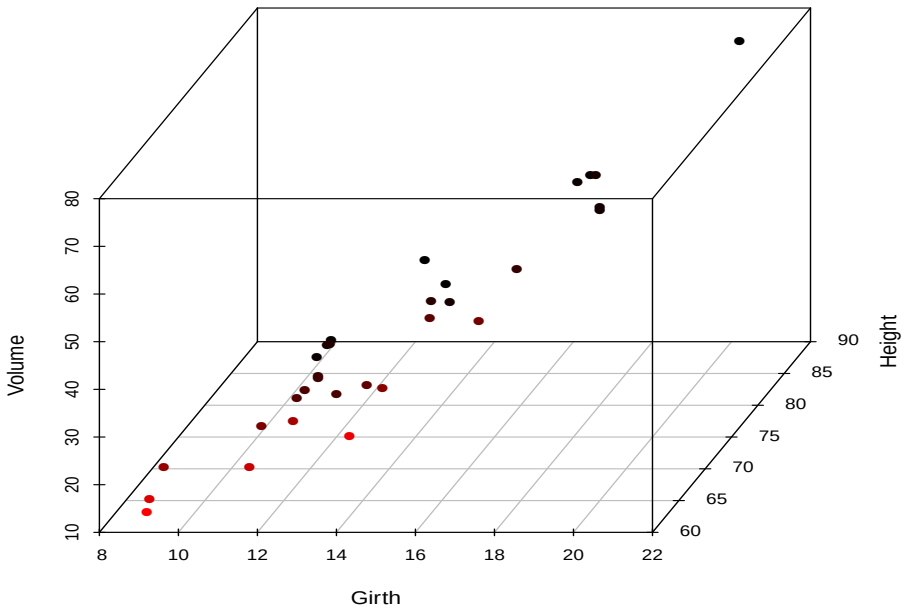

Figure 1.1.2: A 3D scatterplot with regression plane for the `trees` data.

1.2 Estimation and Prediction

1.2.1 Parameter estimates

We will proceed exactly like we did in Section **??**. We know

$$\epsilon \sim \mathsf{mvnorm}\left(\mathtt{mean} = \mathbf{0}_{n\times 1},\ \mathtt{sigma} = \sigma^2 \mathbf{I}_{n\times n}\right), \tag{1.2.1}$$

which means that $\mathbf{Y} = \mathbf{X}\beta + \epsilon$ has an $\mathsf{mvnorm}\left(\mathtt{mean} = \mathbf{X}\beta,\ \mathtt{sigma} = \sigma^2 \mathbf{I}_{n\times n}\right)$ distribution. Therefore, the likelihood function is

$$L(\beta, \sigma) = \frac{1}{2\pi^{n/2}\sigma} \exp\left\{-\frac{1}{2\sigma^2} (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta)\right\}. \tag{1.2.2}$$

To *maximize* the likelihood in β, we need to *minimize* the quantity $g(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta)$. We do this by differentiating g with respect to β. (It may be a good idea to brush up on the material in Appendices **??** and **??**.) First we will rewrite g:

$$g(\beta) = \mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\beta - \beta^T\mathbf{X}^T\mathbf{Y} + \beta^T\mathbf{X}^T\mathbf{X}\beta, \tag{1.2.3}$$

which can be further simplified to $g(\beta) = \mathbf{Y}^T\mathbf{Y} - 2\beta^T\mathbf{X}^T\mathbf{Y} + \beta^T\mathbf{X}^T\mathbf{X}\beta$ since $\beta^T\mathbf{X}^T\mathbf{Y}$ is 1×1 and thus equal to its transpose. Now we differentiate to get

$$\frac{\partial g}{\partial \beta} = \mathbf{0} - 2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\beta, \tag{1.2.4}$$

since $\mathbf{X}^T\mathbf{X}$ is symmetric. Setting the derivative equal to the zero vector yields the so called "normal equations"

$$\mathbf{X}^T\mathbf{X}\beta = \mathbf{X}^T\mathbf{Y}. \tag{1.2.5}$$

In the case that $\mathbf{X}^T\mathbf{X}$ is invertible [1], we may solve the equation for β to get the maximum likelihood estimator of β which we denote by $\mathbf{b}$:

$$\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}. \tag{1.2.6}$$

Remark 1.2. The formula in Equation **??** is convenient for mathematical study but is inconvenient for numerical computation. Researchers have devised much more efficient algorithms for the actual calculation of the parameter estimates, and we do not explore them here.

Remark 1.3. We have only found a critical value, and have not actually shown that the critical value is a minimum. We omit the details and refer the interested reader to [**?**].

How to do it with R We do all of the above just as we would in simple linear regression. The powerhouse is the `lm` function. Everything else is based on it. We separate explanatory variables in the model formula by a plus sign.

```
trees.lm <- lm(Volume ~ Girth + Height, data = trees)
trees.lm
```

```

Call:
lm(formula = Volume ~ Girth + Height, data = trees)

Coefficients:
(Intercept)        Girth       Height
   -57.9877       4.7082       0.3393
```

We see from the output that for the `trees` data our parameter estimates are

$$\mathbf{b} = \begin{bmatrix} -58.0 & 4.7 & 0.3 \end{bmatrix},$$

[1] We can find solutions of the normal equations even when $\mathbf{X}^T\mathbf{X}$ is not of full rank, but the topic falls outside the scope of this book. The interested reader can consult an advanced text such as Rao [**?**].

and consequently our estimate of the mean response is $\hat{\mu}$ given by

$$\hat{\mu}(x_1, x_2) = b_0 + b_1 x_1 + b_2 x_2, \tag{1.2.7}$$

$$\approx -58.0 + 4.7 x_1 + 0.3 x_2. \tag{1.2.8}$$

We could see the entire model matrix $\mathbf{X}$ with the `model.matrix` function, but in the interest of brevity we only show the first few rows.

```
head(model.matrix(trees.lm))
```

```
  (Intercept) Girth Height
1           1   8.3     70
2           1   8.6     65
3           1   8.8     63
4           1  10.5     72
5           1  10.7     81
6           1  10.8     83
```

1.2.2 Point Estimates of the Regression Surface

The parameter estimates $\mathbf{b}$ make it easy to find the fitted values, $\hat{\mathbf{Y}}$. We write them individually as $\hat{Y}_i$, $i = 1, 2, \ldots, n$, and recall that they are defined by

$$\hat{Y}_i = \hat{\mu}(x_{1i}, x_{2i}), \tag{1.2.9}$$

$$= b_0 + b_1 x_{1i} + b_2 x_{2i}, \quad i = 1, 2, \ldots, n. \tag{1.2.10}$$

They are expressed more compactly by the matrix equation

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{b}. \tag{1.2.11}$$

From Equation **??** we know that $\mathbf{b} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$, so we can rewrite

$$\hat{\mathbf{Y}} = \mathbf{X}\left[\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}\right], \tag{1.2.12}$$

$$= \mathbf{H}\mathbf{Y}, \tag{1.2.13}$$

where $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}$ is appropriately named *the hat matrix* because it "puts the hat on $\mathbf{Y}$". The hat matrix is very important in later courses. Some facts about $\mathbf{H}$ are

- $\mathbf{H}$ is a symmetric square matrix, of dimension n × n.
- The diagonal entries h_{ii} satisfy $0 \leq h_{ii} \leq 1$ (compare to Equation **??**).
- The trace is $\mathrm{tr}(\mathbf{H}) = p$.

- $\mathbf{H}$ is *idempotent* (also known as a *projection matrix*) which means that $\mathbf{H}^2 = \mathbf{H}$. The same is true of $\mathbf{I} - \mathbf{H}$.

Now let us write a column vector $\mathbf{x}_0 = (x_{10}, x_{20})^{\mathrm{T}}$ to denote given values of the explanatory variables `Girth` $= x_{10}$ and `Height` $= x_{20}$. These values may match those of the collected data, or they may be completely new values not observed in the original data set. We may use the parameter estimates to find $\hat{Y}(\mathbf{x}_0)$, which will give us

1. an estimate of $\mu(\mathbf{x}_0)$, the mean value of a future observation at $\mathbf{x}_0$, and
2. a prediction for $Y(\mathbf{x}_0)$, the actual value of a future observation at $\mathbf{x}_0$.

We can represent $\hat{Y}(\mathbf{x}_0)$ by the matrix equation

$$\hat{Y}(\mathbf{x}_0) = \mathbf{x}_0^{\mathrm{T}}\mathbf{b}, \tag{1.2.14}$$

which is just a fancy way to write

$$\hat{Y}(x_{10}, x_{20}) = b_0 + b_1 x_{10} + b_2 x_{20}. \tag{1.2.15}$$

Example 1.4. If we wanted to predict the average volume of black cherry trees that have `Girth` $=$ `15` in and are `Height` $=$ `77` ft tall then we would use the estimate

$$\begin{aligned}\hat{\mu}(15,\ 77) &= -\ 58 + 4.7(15) + 0.3(77),\\ &\approx 35.6\ \text{ft}^3.\end{aligned}$$

We would use the same estimate $\hat{Y} = 35.6$ to predict the measured `Volume` of another black cherry tree – yet to be observed – that has `Girth` $=$ `15` in and is `Height` $=$ `77` ft tall.

How to do it with R The fitted values are stored inside `trees.lm` and may be accessed with the `fitted` function. We only show the first five fitted values.

```
fitted(trees.lm)[1:5]
```

```
        1         2         3         4         5
4.837660  4.553852  4.816981 15.874115 19.869008
```

The syntax for general prediction does not change much from simple linear regression. The computations are done with the `predict` function as described below.

The only difference from SLR is in the way we tell R the values of the explanatory variables for which we want predictions. In SLR we had only one independent variable but in MLR we have many (for the `trees` data we have two). We will store values for the independent variables in the data frame `new`, which has two columns (one for each independent variable) and three rows (we shall make predictions at three different locations).

```
new <- data.frame(Girth = c(9.1, 11.6, 12.5), Height = c(69, 74, 87))
```

We can view the locations at which we will predict:

```
new
```

```
   Girth Height
1   9.1     69
2  11.6     74
3  12.5     87
```

We continue just like we would have done in SLR.

```
predict(trees.lm, newdata = new)
```

```
        1         2         3
 8.264937 21.731594 30.379205
```

Example 1.5. Using the `trees` data,

1. Report a point estimate of the mean `Volume` of a tree of `Girth` 9.1 in and `Height` 69 ft.

   The fitted value for $x_1 = 9.1$ and $x_2 = 69$ is 8.3, so a point estimate would be 8.3 cubic feet.

2. Report a point prediction for and a 95% prediction interval for the `Volume` of a hypothetical tree of `Girth` 12.5 in and `Height` 87 ft.

   The fitted value for $x_1 = 12.5$ and $x_2 = 87$ is 30.4, so a point prediction for the `Volume` is 30.4 cubic feet.

1.2.3 Mean Square Error and Standard Error

The residuals are given by

$$\mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{H}\mathbf{Y} = (\mathbf{I} - \mathbf{H})\mathbf{Y}. \tag{1.2.16}$$

Now we can use Theorem **??** to see that the residuals are distributed

$$\mathbf{E} \sim \mathsf{mvnorm}(\mathtt{mean} = \mathbf{0}, \mathtt{sigma} = \sigma^2(\mathbf{I} - \mathbf{H})), \tag{1.2.17}$$

since $(\mathbf{I} - \mathbf{H})\mathbf{X}\beta = \mathbf{X}\beta - \mathbf{X}\beta = \mathbf{0}$ and $(\mathbf{I} - \mathbf{H})\,(\sigma^2\mathbf{I})\,(\mathbf{I} - \mathbf{H})^{\mathrm{T}} = \sigma^2(\mathbf{I} - \mathbf{H})^2 = \sigma^2(\mathbf{I} - \mathbf{H})$. The sum of squared errors SSE is just

$$SSE = \mathbf{E}^{\mathrm{T}}\mathbf{E} = \mathbf{Y}^{\mathrm{T}}(\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})\mathbf{Y} = \mathbf{Y}^{\mathrm{T}}(\mathbf{I} - \mathbf{H})\mathbf{Y}. \tag{1.2.18}$$

Recall that in SLR we had two parameters (β_0 and β_1) in our regression model and we estimated σ^2 with $s^2 = SSE/(n-2)$. In MLR, we have $p+1$ parameters in our regression model and we might guess that to estimate σ^2 we would use the *mean square error* S^2 defined by

$$S^2 = \frac{SSE}{n-(p+1)}. \tag{1.2.19}$$

That would be a good guess. The *residual standard error* is $S = \sqrt{S^2}$.

How to do it with R The residuals are also stored with `trees.lm` and may be accessed with the `residuals` function. We only show the first five residuals.

```
residuals(trees.lm)[1:5]
```

```
        1         2         3         4         5
5.4623403 5.7461484 5.3830187 0.5258848 -1.0690084
```

The `summary` function output (shown later) lists the `Residual Standard Error` which is just $S = \sqrt{S^2}$. It is stored in the `sigma` component of the `summary` object.

```
treesumry <- summary(trees.lm)
treesumry$sigma
```

```
[1] 3.881832
```

For the `trees` data we find $s \approx 3.882$.

1.2.4 Interval Estimates of the Parameters

We showed in Section **??** that $\mathbf{b} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$, which is really just a big matrix – namely $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}$ – multiplied by $\mathbf{Y}$. It stands to reason that the sampling distribution of $\mathbf{b}$ would be intimately related to the distribution of $\mathbf{Y}$, which we assumed to be

$$\mathbf{Y} \sim \mathsf{mvnorm}\left(\mathtt{mean} = \mathbf{X}\beta,\ \mathtt{sigma} = \sigma^2\mathbf{I}\right). \tag{1.2.20}$$

Now recall Theorem **??** that we said we were going to need eventually (the time is now). That proposition guarantees that

$$\mathbf{b} \sim \mathsf{mvnorm}\left(\mathtt{mean} = \beta,\ \mathtt{sigma} = \sigma^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right), \tag{1.2.21}$$

since

$$\mathbb{E}\mathbf{b} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\beta) = \beta, \tag{1.2.22}$$

and

$$\text{Var}(\mathbf{b}) = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T(\sigma^2\mathbf{I})\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1} = \sigma^2\left(\mathbf{X}^T\mathbf{X}\right)^{-1}, \tag{1.2.23}$$

the first equality following because the matrix $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ is symmetric.

There is a lot that we can glean from Equation **??**. First, it follows that the estimator **b** is unbiased (see Section **??**). Second, the variances of $b_0, b_1, \ldots, b_n$ are exactly the diagonal elements of $\sigma^2\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$, which is completely known except for that pesky parameter σ^2. Third, we can estimate the standard error of b_i (denoted S_{b_i}) with the mean square error S (defined in the previous section) multiplied by the corresponding diagonal element of $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$. Finally, given estimates of the standard errors we may construct confidence intervals for β_i with an interval that looks like

$$b_i \pm \mathrm{t}_{\alpha/2}(\mathtt{df} = n - p - 1)S_{b_i}. \tag{1.2.24}$$

The degrees of freedom for the Student's t distribution [2] are the same as the denominator of S^2.

How to do it with R To get confidence intervals for the parameters we need only use `confint`:

```
confint(trees.lm)
```

```
                  2.5 %      97.5 %
(Intercept) -75.68226247 -40.2930554
Girth         4.16683899   5.2494820
Height        0.07264863   0.6058538
```

For example, using the calculations above we say that for the regression model `Volume ~ Girth + Height` we are 95% confident that the parameter β_1 lies somewhere in the interval [4.2, 5.2].

1.2.5 Confidence and Prediction Intervals

We saw in Section **??** how to make point estimates of the mean value of additional observations and predict values of future observations, but how good are our estimates? We need confidence and prediction intervals to gauge their accuracy, and lucky for us the formulas look similar to the ones we saw in SLR.

[2] We are taking great leaps over the mathematical details. In particular, we have yet to show that s^2 has a chi-square distribution and we have not even come close to showing that b_i and s_{b_i} are independent. But these are entirely outside the scope of the present book and the reader may rest assured that the proofs await in later classes. See C.R. Rao for more.

In Equation **??** we wrote $\hat{Y}(\mathbf{x}_0) = \mathbf{x}_0^{\mathrm{T}}\mathbf{b}$, and in Equation **??** we saw that

$$\mathbf{b} \sim \mathsf{mvnorm}\left(\mathtt{mean} = \beta,\ \mathtt{sigma} = \sigma^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right). \tag{1.2.25}$$

The following is therefore immediate from Theorem **??**:

$$\hat{Y}(\mathbf{x}_0) \sim \mathsf{mvnorm}\left(\mathtt{mean} = \mathbf{x}_0^{\mathrm{T}}\beta,\ \mathtt{sigma} = \sigma^2\mathbf{x}_0^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}_0\right). \tag{1.2.26}$$

It should be no surprise that confidence intervals for the mean value of a future observation at the location $\mathbf{x}_0 = \begin{bmatrix} x_{10} & x_{20} & \dots & x_{p0} \end{bmatrix}^{\mathrm{T}}$ are given by

$$\hat{Y}(\mathbf{x}_0) \pm \mathsf{t}_{\alpha/2}(\mathtt{df} = n - p - 1)\, S\, \sqrt{\mathbf{x}_0^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}_0}. \tag{1.2.27}$$

Intuitively, $\mathbf{x}_0^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}_0$ measures the distance of $\mathbf{x}_0$ from the center of the data. The degrees of freedom in the Student's t critical value are $n - (p + 1)$ because we need to estimate $p + 1$ parameters.

Prediction intervals for a new observation at $\mathbf{x}_0$ are given by

$$\hat{Y}(\mathbf{x}_0) \pm \mathsf{t}_{\alpha/2}(\mathtt{df} = n - p - 1)\, S\, \sqrt{1 + \mathbf{x}_0^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{x}_0}. \tag{1.2.28}$$

The prediction intervals are wider than the confidence intervals, just as in Section **??**.

How to do it with R The syntax is identical to that used in SLR, with the proviso that we need to specify values of the independent variables in the data frame `new` as we did in Section **??** (which we repeat here for illustration).

```
new <- data.frame(Girth = c(9.1, 11.6, 12.5), Height = c(69, 74, 87))
```

Confidence intervals are given by

```
predict(trees.lm, newdata = new, interval = "confidence")
```

```
        fit      lwr      upr
1  8.264937  5.77240 10.75747
2 21.731594 20.11110 23.35208
3 30.379205 26.90964 33.84877
```

Prediction intervals are given by

```
predict(trees.lm, newdata = new, interval = "prediction")
```

```
        fit         lwr      upr
1  8.264937 -0.06814444 16.59802
2 21.731594 13.61657775 29.84661
3 30.379205 21.70364103 39.05477
```

As before, the interval type is decided by the `interval` argument and the default confidence level is 95% (which can be changed with the `level` argument).

Example 1.6. Using the `trees` data,

1. Report a 95% confidence interval for the mean `Volume` of a tree of `Girth` 9.1 in and `Height` 69 ft.

   The 95% CI is given by [5.8, 10.8], so with 95% confidence the mean `Volume` lies somewhere between 5.8 cubic feet and 10.8 cubic feet.

2. Report a 95% prediction interval for the `Volume` of a hypothetical tree of `Girth` 12.5 in and `Height` 87 ft.

   The 95% prediction interval is given by [26.9, 33.8], so with 95% confidence we may assert that the hypothetical `Volume` of a tree of `Girth` 12.5 in and `Height` 87 ft would lie somewhere between 26.9 cubic feet and 33.8 feet.

1.3 Model Utility and Inference

1.3.1 Multiple Coefficient of Determination

We saw in Section **??** that the error sum of squares SSE can be conveniently written in MLR as

$$SSE = \mathbf{Y}^{\mathrm{T}}(\mathbf{I} - \mathbf{H})\mathbf{Y}. \tag{1.3.1}$$

It turns out that there are equally convenient formulas for the total sum of squares $SSTO$ and the regression sum of squares SSR. They are:

$$SSTO = \mathbf{Y}^{\mathrm{T}}\left(\mathbf{I} - \frac{1}{n}\mathbf{J}\right)\mathbf{Y} \tag{1.3.2}$$

and

$$SSR = \mathbf{Y}^{\mathrm{T}}\left(\mathbf{H} - \frac{1}{n}\mathbf{J}\right)\mathbf{Y}. \tag{1.3.3}$$

(The matrix $\mathbf{J}$ is defined in Appendix **??**.) Immediately from Equations **??**, **??**, and **??** we get the *Anova Equality*

$$SSTO = SSE + SSR. \tag{1.3.4}$$

(See Exercise **??**.) We define the *multiple coefficient of determination* by the formula

$$R^2 = 1 - \frac{SSE}{SSTO}. \tag{1.3.5}$$

We interpret R^2 as the proportion of total variation that is explained by the multiple regression model. In MLR we must be careful, however, because the value of R^2 can be artificially inflated by the addition of explanatory variables to the model, regardless of whether or not the added variables are useful with respect to prediction of the response variable. In fact, it can be proved that the addition of a single explanatory variable to a regression model will increase the value of R^2, *no matter how worthless* the explanatory variable is. We could model the height of the ocean tides, then add a variable for the length of cheetah tongues on the Serengeti plain, and our R^2 would inevitably increase.

This is a problem, because as the philosopher, Occam, once said: "causes should not be multiplied beyond necessity". We address the problem by penalizing R^2 when parameters are added to the model. The result is an *adjusted* R^2 which we denote by $\overline{R}^2$.

$$\overline{R}^2 = \left(R^2 - \frac{p}{n-1}\right)\left(\frac{n-1}{n-p-1}\right). \tag{1.3.6}$$

It is good practice for the statistician to weigh both R^2 and $\overline{R}^2$ during assessment of model utility. In many cases their values will be very close to each other. If their values differ substantially, or if one changes dramatically when an explanatory variable is added, then (s)he should take a closer look at the explanatory variables in the model.

How to do it with R For the `trees` data, we can get R^2 and $\overline{R}^2$ from the `summary` output or access the values directly by name as shown (recall that we stored the `summary` object in `treesumry`).

```
treesumry$r.squared
```

```
[1] 0.94795
```

```
treesumry$adj.r.squared
```

```
[1] 0.9442322
```

High values of R^2 and $\overline{R}^2$ such as these indicate that the model fits very well, which agrees with what we saw in Figure **??**.

1.3.2 Overall *F*-Test

Another way to assess the model's utility is to to test the hypothesis

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0 \text{ versus } H_1 : \text{ at least one } \beta_i \neq 0.$$

The idea is that if all β_i's were zero, then the explanatory variables $X_1, \ldots, X_p$ would be worthless predictors for the response variable Y. We can test the above hypothesis with the overall F statistic, which in MLR is defined by

$$F = \frac{SSR/p}{SSE/(n-p-1)}. \tag{1.3.7}$$

When the regression assumptions hold and under H_0, it can be shown that $F \sim \mathsf{f}(\texttt{df1} = p, \texttt{df2} = n - p - 1)$. We reject H_0 when F is large, that is, when the explained variation is large relative to the unexplained variation.

How to do it with R The overall F statistic and its associated p-value is listed at the bottom of the `summary` output, or we can access it directly by name; it is stored in the `fstatistic` component of the `summary` object.

```
treesumry$fstatistic
```

```
    value     numdf     dendf
254.9723    2.0000   28.0000
```

For the `trees` data, we see that $F = 254.972337410669$ with a p-value $<$ `2.2e-16`. Consequently we reject H_0, that is, the data provide strong evidence that not all β_i's are zero.

1.3.3 Student's *t* Tests

We know that

$$\mathbf{b} \sim \mathsf{mvnorm}\left(\texttt{mean} = \beta, \texttt{sigma} = \sigma^2 \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right) \tag{1.3.8}$$

and we have seen how to test the hypothesis $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$, but let us now consider the test

$$H_0 : \beta_i = 0 \text{ versus } H_1 : \beta_i \neq 0, \tag{1.3.9}$$

where β_i is the coefficient for the i^{th} independent variable. We test the hypothesis by calculating a statistic, examining it's null distribution, and rejecting H_0 if the *p-value* is small. If H_0 is rejected, then we conclude that there is a significant relationship between Y and x_i *in the regression model* $Y \sim (x_1, \ldots, x_p)$. This last part of the sentence is very

important because the significance of the variable x_i sometimes depends on the presence of other independent variables in the model [3].

To test the hypothesis we go to find the sampling distribution of b_i, the estimator of the corresponding parameter β_i, when the null hypothesis is true. We saw in Section **??** that

$$T_i = \frac{b_i - \beta_i}{S_{b_i}} \tag{1.3.10}$$

has a Student's t distribution with $n - (p + 1)$ degrees of freedom. (Remember, we are estimating $p + 1$ parameters.) Consequently, under the null hypothesis $H_0 : \beta_i = 0$ the statistic $t_i = b_i / S_{b_i}$ has a t(df $= n - p - 1$) distribution.

How to do it with R The Student's t tests for significance of the individual explanatory variables are shown in the `summary` output.

```
treesumry


Call:
lm(formula = Volume ~ Girth + Height, data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4065 -2.6493 -0.2876  2.2003  8.4847

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
Girth         4.7082     0.2643  17.816  < 2e-16 ***
Height        0.3393     0.1302   2.607   0.0145 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.882 on 28 degrees of freedom
Multiple R-squared: 0.948,Adjusted R-squared: 0.9442
F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

We see from the *p-values* that there is a significant linear relationship between `Volume` and `Girth` and between `Volume` and `Height` in the regression model `Volume ~ Girth + Height`. Further, it appears that the `Intercept` is significant in the aforementioned model.

[3] In other words, a variable might be highly significant one moment but then fail to be significant when another variable is added to the model. When this happens it often indicates a problem with the explanatory variables, such as /multicollinearity/. See Section **??**.

1.4 Polynomial Regression

1.4.1 Quadratic Regression Model

In each of the previous sections we assumed that μ was a linear function of the explanatory variables. For example, in SLR we assumed that $\mu(x) = \beta_0 + \beta_1 x$, and in our previous MLR examples we assumed $\mu(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$. In every case the scatterplots indicated that our assumption was reasonable. Sometimes, however, plots of the data suggest that the linear model is incomplete and should be modified.

```
qplot(Girth, Volume, data = trees)
```

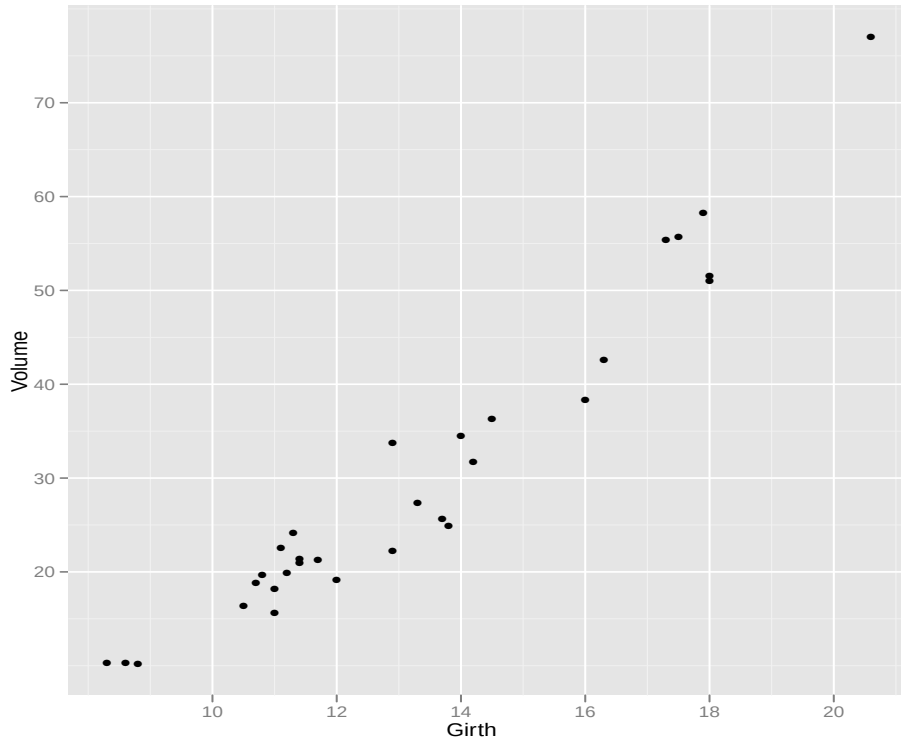

Figure 1.4.1: A scatterplot of `Volume` versus `Girth` for the `trees` data.

For example, let us examine a scatterplot of `Volume` versus `Girth` a little more closely. See Figure **??**. There might be a slight curvature to the data; the volume curves ever so slightly upward as the girth increases. After looking at the plot we might try to capture the curvature with a mean response such as

$$\mu(x_1) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2. \tag{1.4.1}$$

The model associated with this choice of μ is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \epsilon. \tag{1.4.2}$$

The regression assumptions are the same. Almost everything indeed is the same. In fact, it is still called a "linear regression model", since the mean response μ is linear *in the parameters* β_0, β_1, and β_2.

However, there is one important difference. When we introduce the squared variable in the model we inadvertently also introduce strong dependence between the terms which can cause significant numerical problems when it comes time to calculate the parameter estimates. Therefore, we should usually rescale the independent variable to have mean zero (and even variance one if we wish) **before** fitting the model. That is, we replace the x_i's with $x_i - \overline{x}$ (or $(x_i - \overline{x})/s$) before fitting the model [4].

How to do it with R There are multiple ways to fit a quadratic model to the variables `Volume` and `Girth` using R.

1. One way would be to square the values for `Girth` and save them in a vector `Girthsq`. Next, fit the linear model `Volume ~ Girth + Girthsq`.

2. A second way would be to use the *insulate* function in R, denoted by `I`:

```
Volume ~ Girth + I(Girth^2)
```

The second method is shorter than the first but the end result is the same. And once we calculate and store the fitted model (in, say, `treesquad.lm`) all of the previous comments regarding R apply.

1. A third and "right" way to do it is with orthogonal polynomials:

```
Volume ~ poly(Girth, degree = 2)
```

   See `?poly` and `?cars` for more information. Note that we can recover the approach in 2 with `poly(Girth, degree = 2, raw = TRUE)`.

Example 1.7. We will fit the quadratic model to the `trees` data and display the results with `summary`, being careful to rescale the data before fitting the model. We may rescale the `Girth` variable to have zero mean and unit variance on-the-fly with the `scale` function.

[4] Rescaling the data gets the job done but a better way to avoid the multicollinearity introduced by the higher order terms is with /orthogonal polynomials/, whose coefficients are chosen just right so that the polynomials are not correlated with each other. This is beginning to linger outside the scope of this book, however, so we will content ourselves with a brief mention and then stick with the rescaling approach in the discussion that follows. A nice example of orthogonal polynomials in action can be run with `example(cars)`.

```
treesquad.lm <- lm(Volume ~ scale(Girth) + I(scale(Girth)^2),
summary(treesquad.lm)


Call:
lm(formula = Volume ~ scale(Girth) + I(scale(Girth)^2), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4889 -2.4293 -0.3718  2.0764  7.6447

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        27.7452     0.8161  33.996  < 2e-16 ***
scale(Girth)       14.5995     0.6773  21.557  < 2e-16 ***
I(scale(Girth)^2)   2.5067     0.5729   4.376 0.000152 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.335 on 28 degrees of freedom
Multiple R-squared: 0.9616,Adjusted R-squared: 0.9588
F-statistic: 350.5 on 2 and 28 DF,  p-value: < 2.2e-16
```

We see that the F statistic indicates the overall model including `Girth` and `Girth^2` is significant. Further, there is strong evidence that both `Girth` and `Girth^2` are significantly related to `Volume`. We may examine a scatterplot together with the fitted quadratic function using the `lines` function, which adds a line to the plot tracing the estimated mean response.

```
a <- ggplot(trees, aes(scale(Girth), Volume))
a + stat_smooth(method = lm, formula = y ~ poly(x, 2)) + geom_point()
```

The plot is shown in Figure **??**. Pay attention to the scale on the x-axis: it is on the scale of the transformed `Girth` data and not on the original scale.

Remark 1.8. When a model includes a quadratic term for an independent variable, it is customary to also include the linear term in the model. The principle is called *parsimony*. More generally, if the researcher decides to include x^m as a term in the model, then (s)he should also include all lower order terms x, x^2, ...,x^{m-1} in the model.

We do estimation/prediction the same way that we did in Section **??**, except we do not need a `Height` column in the dataframe `new` since the variable is not included in the quadratic model.

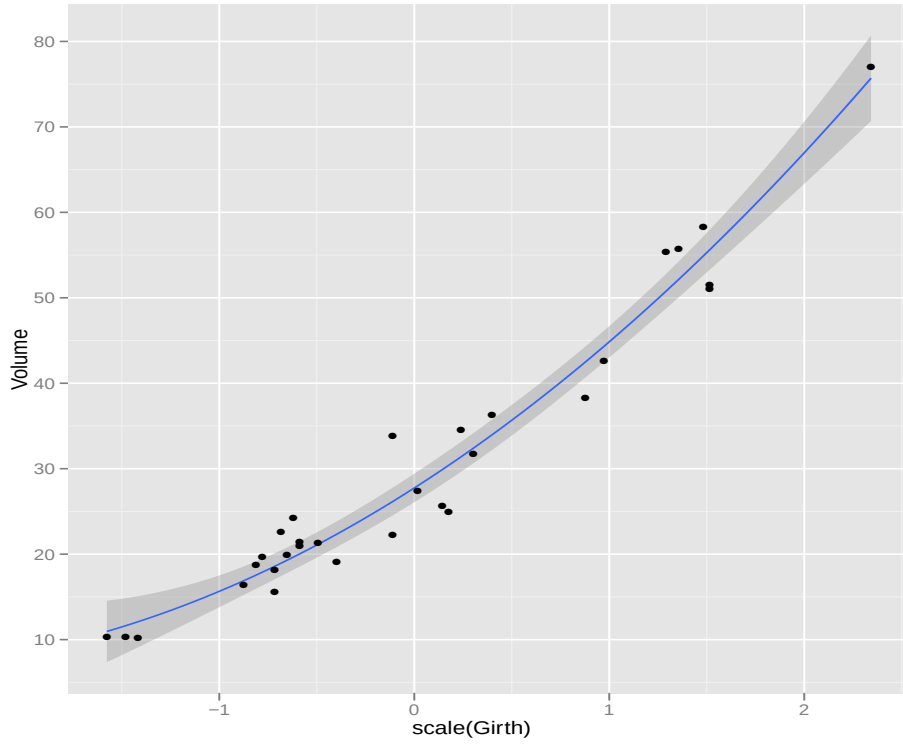

Figure 1.4.2: A quadratic model for the `trees` data.

```
new <- data.frame(Girth = c(9.1, 11.6, 12.5))
predict(treesquad.lm, newdata = new, interval = "prediction")
```

```
        fit       lwr      upr
 1 11.56982  4.347426 18.79221
 2 20.30615 13.299050 27.31325
 3 25.92290 18.972934 32.87286
```

The predictions and intervals are slightly different from what they were previously. Notice that it was not necessary to rescale the `Girth` prediction data before input to the `predict` function; the model did the rescaling for us automatically.

Remark 1.9. We have mentioned on several occasions that it is important to rescale the explanatory variables for polynomial regression. Watch what happens if we ignore this advice:

```
summary(lm(Volume ~ Girth + I(Girth^2), data = trees))
```

```
Call:
lm(formula = Volume ~ Girth + I(Girth^2), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4889 -2.4293 -0.3718  2.0764  7.6447

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.78627   11.22282   0.961 0.344728
Girth       -2.09214    1.64734  -1.270 0.214534
I(Girth^2)   0.25454    0.05817   4.376 0.000152 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.335 on 28 degrees of freedom
Multiple R-squared: 0.9616,Adjusted R-squared: 0.9588
F-statistic: 350.5 on 2 and 28 DF,  p-value: < 2.2e-16
```

Now nothing is significant in the model except `Girth^2`. We could delete the `Intercept` and `Girth` from the model, but the model would no longer be *parsimonious*. A novice may see the output and be confused about how to proceed, while the seasoned statistician recognizes immediately that `Girth` and `Girth^2` are highly correlated (see Section **??**). The only remedy to this ailment is to rescale `Girth`, which we should have done in the first place.

In Example **??** of Section **??** we investigate this issue further.

1.5 Interaction

In our model for tree volume there have been two independent variables: `Girth` and `Height`. We may suspect that the independent variables are related, that is, values of one variable may tend to influence values of the other. It may be desirable to include an additional term in our model to try and capture the dependence between the variables. Interaction terms are formed by multiplying one (or more) explanatory variable(s) by another.

Example 1.10. Perhaps the `Girth` and `Height` of the tree interact to influence the its `Volume`; we would like to investigate whether the model (`Girth` $= x_1$ and `Height` $= x_2$)

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \tag{1.5.1}$$

would be significantly improved by the model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{1:2} x_1 x_2 + \epsilon, \tag{1.5.2}$$

where the subscript $1:2$ denotes that $\beta_{1:2}$ is a coefficient of an interaction term between x_1 and x_2.

What does it mean? Consider the mean response $\mu(x_1, x_2)$ as a function of x_2:

$$\mu(x_2) = (\beta_0 + \beta_1 x_1) + \beta_2 x_2. \tag{1.5.3}$$

This is a linear function of x_2 with slope β_2. As x_1 changes, the y-intercept of the mean response in x_2 changes, but the slope remains the same. Therefore, the mean response in x_2 is represented by a collection of parallel lines all with common slope β_2.

Now think about what happens when the interaction term $\beta_{1:2} x_1 x_2$ is included. The mean response in x_2 now looks like

$$\mu(x_2) = (\beta_0 + \beta_1 x_1) + (\beta_2 + \beta_{1:2} x_1) x_2. \tag{1.5.4}$$

In this case we see that not only the y-intercept changes when x_1 varies, but the slope also changes in x_1. Thus, the interaction term allows the slope of the mean response in x_2 to increase and decrease as x_1 varies.

How to do it with R There are several ways to introduce an interaction term into the model.

1. Make a new variable `prod <- Girth * Height`, then include `prod` in the model formula `Volume ~ Girth + Height + prod`. This method is perhaps the most transparent, but it also reserves memory space unnecessarily.

2. Once can construct an interaction term directly in R with a colon ":=". For this example, the model formula would look like

```
Volume ~ Girth + Height + Girth:Height
```

For the `trees` data, we fit the model with the interaction using method two and see if it is significant:

```
treesint.lm <- lm(Volume ~ Girth + Height + Girth:Height, data = trees)
summary(treesint.lm)
```

```
Call:
lm(formula = Volume ~ Girth + Height + Girth:Height, data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5821 -1.0673  0.3026  1.5641  4.6649

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  69.39632   23.83575   2.911  0.00713 **
Girth        -5.85585    1.92134  -3.048  0.00511 **
Height       -1.29708    0.30984  -4.186  0.00027 ***
Girth:Height  0.13465    0.02438   5.524 7.48e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.709 on 27 degrees of freedom
Multiple R-squared: 0.9756,Adjusted R-squared: 0.9728
F-statistic: 359.3 on 3 and 27 DF,  p-value: < 2.2e-16
```

We can see from the output that the interaction term is highly significant. Further, the estimate $b_{1:2}$ is positive. This means that the slope of $\mu(x_2)$ is steeper for bigger values of `Girth`. Keep in mind: the same interpretation holds for $\mu(x_1)$; that is, the slope of $\mu(x_1)$ is steeper for bigger values of `Height`.

For the sake of completeness we calculate confidence intervals for the parameters and do prediction as before.

```
confint(treesint.lm)
```

```
                   2.5 %      97.5 %
 (Intercept)  20.48938699 118.3032441
 Girth        -9.79810354  -1.9135923
 Height       -1.93282845  -0.6613383
 Girth:Height  0.08463628   0.1846725
```

```
new <- data.frame(Girth = c(9.1, 11.6, 12.5), Height = c(69, 74, 87))
predict(treesint.lm, newdata = new, interval = "prediction")
```

```
        fit       lwr      upr
 1 11.15884  5.236341 17.08134
 2 21.07164 15.394628 26.74866
 3 29.78862 23.721155 35.85608
```

Remark 1.11. There are two other ways to include interaction terms in model formulas. For example, we could have written `Girth * Height` or even `(Girth + Height)^2` and both would be the same as `Girth + Height + Girth:Height`.

These examples can be generalized to more than two independent variables, say three, four, or even more. We may be interested in seeing whether any pairwise interactions are significant. We do this with a model formula that looks something like `y ~ (x1 + x2 + x3 + x4)^2`.

1.6 Qualitative Explanatory Variables

We have so far been concerned with numerical independent variables taking values in a subset of real numbers. In this section, we extend our treatment to include the case in which one of the explanatory variables is qualitative, that is, a *factor*. Qualitative variables take values in a set of *levels*, which may or may not be ordered. See Section **??**.

Note. The `trees` data do not have any qualitative explanatory variables, so we will construct one for illustrative purposes [5]. We will leave the `Girth` variable alone, but we will replace the variable `Height` by a new variable `Tall` which indicates whether or not the cherry tree is taller than a certain threshold (which for the sake of argument will be the sample median height of 76 ft). That is, `Tall` will be defined by

$$\texttt{Tall} = \begin{cases} \texttt{yes}, & \text{if } \texttt{Height} > 76, \\ \texttt{no}, & \text{if } \texttt{Height} \leq 76. \end{cases} \tag{1.6.1}$$

We can construct `Tall` very quickly in R with the `cut` function:

```
trees$Tall <- cut(trees$Height, breaks = c(-Inf, 76, Inf),
                  labels = c("no","yes"))
trees$Tall[1:5]
```

```
 [1] no  no  no  no  yes
 Levels: no yes
```

Note that `Tall` is automatically generated to be a factor with the labels in the correct order. See `?cut` for more.

[5] This procedure of replacing a continuous variable by a discrete/qualitative one is called *binning*, and is almost *never* the right thing to do. We are in a bind at this point, however, because we have invested this chapter in the `trees` data and I do not want to switch mid-discussion. I am currently searching for a data set with pre-existing qualitative variables that also conveys the same points present in the trees data, and when I find it I will update this chapter accordingly.

Once we have `Tall`, we include it in the regression model just like we would any other variable. It is handled internally in a special way. Define a "dummy variable" `Tallyes` that takes values

$$\texttt{Tallyes} = \begin{cases} 1, & \text{if } \texttt{Tall} = \texttt{yes}, \\ 0, & \text{otherwise.} \end{cases} \tag{1.6.2}$$

That is, `Tallyes` is an *indicator variable* which indicates when a respective tree is tall. The model may now be written as

$$\texttt{Volume} = \beta_0 + \beta_1 \texttt{Girth} + \beta_2 \texttt{Tallyes} + \epsilon. \tag{1.6.3}$$

Let us take a look at what this definition does to the mean response. Trees with `Tall = yes` will have the mean response

$$\mu(\texttt{Girth}) = (\beta_0 + \beta_2) + \beta_1 \texttt{Girth}, \tag{1.6.4}$$

while trees with `Tall = no` will have the mean response

$$\mu(\texttt{Girth}) = \beta_0 + \beta_1 \texttt{Girth}. \tag{1.6.5}$$

In essence, we are fitting two regression lines: one for tall trees, and one for short trees. The regression lines have the same slope but they have different y intercepts (which are exactly $|\beta_2|$ far apart).

How to do it with R The important thing is to double check that the qualitative variable in question is stored as a factor. The way to check is with the `class` command. For example,

```
class(trees$Tall)
```

```
 [1] "factor"
```

If the qualitative variable is not yet stored as a factor then we may convert it to one with the `factor` command. See Section **??**. Other than this we perform MLR as we normally would.

```
treesdummy.lm <- lm(Volume ~ Girth + Tall, data = trees)
summary(treesdummy.lm)
```

```
Call:
lm(formula = Volume ~ Girth + Tall, data = trees)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-5.7788 -3.1710  0.4888  2.6737 10.0619

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -34.1652     3.2438  -10.53 3.02e-11 ***
Girth         4.6988     0.2652   17.72  < 2e-16 ***
Tall[T.yes]   4.3072     1.6380    2.63   0.0137 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.875 on 28 degrees of freedom
Multiple R-squared: 0.9481,Adjusted R-squared: 0.9444
F-statistic: 255.9 on 2 and 28 DF,  p-value: < 2.2e-16
```

From the output we see that all parameter estimates are statistically significant and we conclude that the mean response differs for trees with `Tall = yes` and trees with `Tall = no`.

Remark 1.12. We were somewhat disingenuous when we defined the dummy variable `Tallyes` because, in truth, R defines `Tallyes` automatically without input from the user [6]. Indeed, the author fit the model beforehand and wrote the discussion afterward with the knowledge of what R would do so that the output the reader saw would match what (s)he had previously read. The way that R handles factors internally is part of a much larger topic concerning *contrasts*, which falls outside the scope of this book. The interested reader should see Neter et al [**?**] or Fox [**?**] for more.

Remark 1.13. In general, if an explanatory variable `foo` is qualitative with n levels `bar1`, `bar2`, ..., `barn` then R will by default automatically define $n-1$ indicator variables in the following way:

$$\texttt{foobar2} = \begin{cases} 1, & \text{if } \texttt{foo} = \text{"}\texttt{bar2}\text{"}, \\ 0, & \text{otherwise.} \end{cases}, \ldots, \texttt{foobarn} = \begin{cases} 1, & \text{if } \texttt{foo} = \text{"}\texttt{barn}\text{"}, \\ 0, & \text{otherwise.} \end{cases}$$

The level `bar1` is represented by $\texttt{foobar2} = \cdots = \texttt{foobarn} = 0$. We just need to make sure that `foo` is stored as a factor and R will take care of the rest.

1.6.1 Graphing the Regression Lines

We can see a plot of the two regression lines with the following mouthful of code.

[6] That is, R by default handles contrasts according to its internal settings which may be customized by the user for fine control. Given that we will not investigate contrasts further in this book it does not serve the discussion to delve into those settings, either. The interested reader should check `?contrasts` for details.

```
treesTall <- split(trees, trees$Tall)
treesTall[["yes"]]$Fit <- predict(treesdummy.lm, treesTall[["yes"]])
treesTall[["no"]]$Fit <- predict(treesdummy.lm, treesTall[["no"]])
plot(Volume ~ Girth, data = trees, type = "n")
points(Volume ~ Girth, data = treesTall[["yes"]], pch = 1)
points(Volume ~ Girth, data = treesTall[["no"]], pch = 2)
lines(Fit ~ Girth, data = treesTall[["yes"]])
lines(Fit ~ Girth, data = treesTall[["no"]])
```

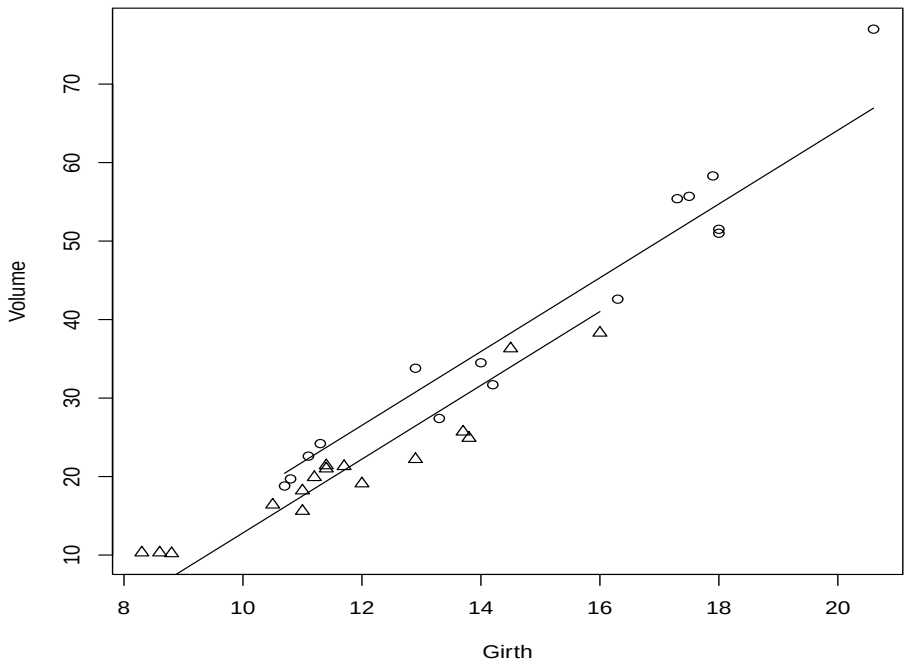

Figure 1.6.1: A dummy variable model for the `trees` data.

It may look intimidating but there is reason to the madness. First we `split` the `trees` data into two pieces, with groups determined by the `Tall` variable. Next we add the `Fit`ted values to each piece via `predict`. Then we set up a `plot` for the variables `Volume` versus `Girth`, but we do not plot anything yet (`type = n`) because we want to use different symbols for the two groups. Next we add `points` to the plot for the `Tall = yes` trees and use an open circle for a plot character (`pch = 1`), followed by `points` for the `Tall = no` trees with a triangle character (`pch = 2`). Finally, we add regression `lines` to the plot, one for each group.

There are other – shorter – ways to plot regression lines by groups, namely the `scatterplot` function in the `car` [**?**] package and the `xyplot` function in the `lattice` package. We elected to introduce the reader to the above approach since many advanced plots in R are done in a similar, consecutive fashion.

1.7 Partial *F* Statistic

We saw in Section **??** how to test $H_0 : \beta_0 = \beta_1 = \cdots = \beta_p = 0$ with the overall F statistic and we saw in Section **??** how to test $H_0 : \beta_i = 0$ that a particular coefficient β_i is zero. Sometimes, however, we would like to test whether a certain part of the model is significant. Consider the regression model

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_j x_j + \beta_{j+1} x_{j+1} + \cdots + \beta_p x_p + \epsilon, \tag{1.7.1}$$

where $j \geq 1$ and $p \geq 2$. Now we wish to test the hypothesis

$$H_0 : \beta_{j+1} = \beta_{j+2} = \cdots = \beta_p = 0 \tag{1.7.2}$$

versus the alternative

$$H_1 : \text{at least one of } \beta_{j+1},\ \beta_{j+2},\ , \ldots, \beta_p \neq 0. \tag{1.7.3}$$

The interpretation of H_0 is that none of the variables $x_{j+1}, \ldots, x_p$ is significantly related to Y and the interpretation of H_1 is that at least one of $x_{j+1}, \ldots, x_p$ is significantly related to Y. In essence, for this hypothesis test there are two competing models under consideration:

$$\text{the full model:} \quad y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon, \tag{1.7.4}$$

$$\text{the reduced model:} \quad y = \beta_0 + \beta_1 x_1 + \cdots + \beta_j x_j + \epsilon, \tag{1.7.5}$$

Of course, the full model will always explain the data *better* than the reduced model, but does the full model explain the data *significantly better* than the reduced model? This question is exactly what the partial F statistic is designed to answer.

We first calculate SSE_f, the unexplained variation in the full model, and SSE_r, the unexplained variation in the reduced model. We base our test on the difference $SSE_r - SSE_f$ which measures the reduction in unexplained variation attributable to the variables $x_{j+1}, \ldots, x_p$. In the full model there are $p+1$ parameters and in the reduced model there are $j+1$ parameters, which gives a difference of $p - j$ parameters (hence degrees of freedom). The partial F statistic is

$$F = \frac{(SSE_r - SSE_f)/(p - j)}{SSE_f/(n - p - 1)}. \tag{1.7.6}$$

It can be shown when the regression assumptions hold under H_0 that the partial F statistic has an $\mathsf{f}(\texttt{df1} = p-j, \texttt{df2} = n-p-1)$ distribution. We calculate the p-value of the observed partial F statistic and reject H_0 if the p-value is small.

How to do it with R The key ingredient above is that the two competing models are *nested* in the sense that the reduced model is entirely contained within the complete model. The way to test whether the improvement is significant is to compute `lm` objects both for the complete model and the reduced model then compare the answers with the `anova` function.

Example 1.14. For the `trees` data, let us fit a polynomial regression model and for the sake of argument we will ignore our own good advice and fail to rescale the explanatory variables.

```
treesfull.lm <- lm(Volume ~ Girth + I(Girth^2) + Height +
                     I(Height^2), data = trees)
summary(treesfull.lm)


Call:
lm(formula = Volume ~ Girth + I(Girth^2) + Height + I(Height^2),
    data = trees)

Residuals:
   Min     1Q Median     3Q    Max
-4.368 -1.670 -0.158  1.792  4.358

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.955101  63.013630  -0.015    0.988
Girth       -2.796569   1.468677  -1.904    0.068 .
I(Girth^2)   0.265446   0.051689   5.135 2.35e-05 ***
Height       0.119372   1.784588   0.067    0.947
I(Height^2)  0.001717   0.011905   0.144    0.886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.674 on 26 degrees of freedom
Multiple R-squared: 0.9771,Adjusted R-squared: 0.9735
F-statistic:   277 on 4 and 26 DF,  p-value: < 2.2e-16
```

In this ill-formed model nothing is significant except `Girth` and `Girth^2`. Let us continue down this path and suppose that we would like to try a reduced model which contains nothing but `Girth` and `Girth^2` (not even an `Intercept`). Our two models are now

$$\text{the full model:} \quad Y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2 + \beta_4 x_2^2 + \epsilon,$$
$$\text{the reduced model:} \quad Y = \beta_1 x_1 + \beta_2 x_1^2 + \epsilon,$$

We fit the reduced model with `lm` and store the results:

```
treesreduced.lm <- lm(Volume ~ -1 + Girth + I(Girth^2), data = trees)
```

To delete the intercept from the model we used `-1` in the model formula. Next we compare the two models with the `anova` function. The convention is to list the models from smallest to largest.

```
anova(treesreduced.lm, treesfull.lm)
```

```
 Analysis of Variance Table

 Model 1: Volume ~ -1 + Girth + I(Girth^2)
 Model 2: Volume ~ Girth + I(Girth^2) + Height + I(Height^2)
   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
 1     29 321.65
 2     26 185.86  3    135.79 6.3319 0.002279 **
 ---
 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see from the output that the complete model is highly significant compared to the model that does not incorporate `Height` or the `Intercept`. We wonder (with our tongue in our cheek) if the `Height^2` term in the full model is causing all of the trouble. We will fit an alternative reduced model that only deletes `Height^2`.

```
treesreduced2.lm <- lm(Volume ~ Girth + I(Girth^2) + Height,
                       data = trees)
anova(treesreduced2.lm, treesfull.lm)
```

```
  Analysis of Variance Table

 Model 1: Volume ~ Girth + I(Girth^2) + Height
 Model 2: Volume ~ Girth + I(Girth^2) + Height + I(Height^2)
   Res.Df    RSS Df Sum of Sq      F Pr(>F)
 1     27 186.01
 2     26 185.86  1   0.14865 0.0208 0.8865
```

In this case, the improvement to the reduced model that is attributable to `Height^2` is not significant, so we can delete `Height^2` from the model with a clear conscience. We notice that the *p-value* for this latest partial F test is 0.8865, which seems to be remarkably close to the *p-value* we saw for the univariate t test of `Height^2` at the beginning of this example. In fact, the *p-values* are *exactly* the same. Perhaps now we gain some insight into the true meaning of the univariate tests.

1.8 Residual Analysis and Diagnostic Tools

We encountered many, many diagnostic measures for simple linear regression in Sections **??** and **??**. All of these are valid in multiple linear regression, too, but there are some slight changes that we need to make for the multivariate case. We list these below, and apply them to the trees example.

Shapiro-Wilk, Breusch-Pagan, Durbin-Watson: unchanged from SLR, but we are now equipped to talk about the Shapiro-Wilk test statistic for the residuals. It is defined by the formula

$$W = \frac{\mathbf{a}^{\mathrm{T}}\mathbf{E}^{*}}{\mathbf{E}^{\mathrm{T}}\mathbf{E}}, \tag{1.8.1}$$

where $\mathbf{E}^{*}$ is the sorted residuals and $\mathbf{a}_{1\times \mathrm{n}}$ is defined by

$$\mathbf{a} = \frac{\mathbf{m}^{\mathrm{T}}\mathbf{V}^{-1}}{\sqrt{\mathbf{m}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{V}^{-1}\mathbf{m}}}, \tag{1.8.2}$$

where $\mathbf{m}_{\mathrm{n}\times 1}$ and $\mathbf{V}_{\mathrm{n}\times \mathrm{n}}$ are the mean and covariance matrix, respectively, of the order statistics from an $\mathsf{mvnorm}(\mathtt{mean} = \mathbf{0}, \mathtt{sigma} = \mathbf{I})$ distribution.

Leverages: are defined to be the diagonal entries of the hat matrix $\mathbf{H}$ (which is why we called them h_{ii} in Section **??**). The sum of the leverages is $\mathrm{tr}(\mathbf{H}) = p + 1$. One rule of thumb considers a leverage extreme if it is larger than double the mean leverage value, which is $2(p+1)/n$, and another rule of thumb considers leverages bigger than 0.5 to indicate high leverage, while values between 0.3 and 0.5 indicate moderate leverage.

Standardized residuals: unchanged. Considered extreme if $|R_i| > 2$.

Studentized residuals: compared to a $\mathsf{t}(\mathtt{df} = n - p - 2)$ distribution.

DFBETAS: The formula is generalized to

$$(DFBETAS)_{j(i)} = \frac{b_j - b_{j(i)}}{S_{(i)}\sqrt{c_{jj}}}, \quad j = 0, \ldots p,\ i = 1, \ldots, n, \tag{1.8.3}$$

where c_{jj} is the j^{th} diagonal entry of $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$. Values larger than one for small data sets or $2/\sqrt{n}$ for large data sets should be investigated.

DFFITS: unchanged. Larger than one in absolute value is considered extreme.

Cook's D: compared to an $\mathsf{f}(\mathtt{df1} = p + 1, \mathtt{df2} = n - p - 1)$ distribution. Observations falling higher than the 50^{th} percentile are extreme.

Note that plugging the value $p = 1$ into the formulas will recover all of the ones we saw in Chapter **??**.

1.9 Additional Topics

1.9.1 Nonlinear Regression

We spent the entire chapter talking about the `trees` data, and all of our models looked like `Volume ~ Girth + Height` or a variant of this model. But let us think again: we know from elementary school that the volume of a rectangle is $V = lwh$ and the volume of a cylinder (which is closer to what a black cherry tree looks like) is

$$V = \pi r^2 h \quad \text{or} \quad V = 4\pi dh, \tag{1.9.1}$$

where r and d represent the radius and diameter of the tree, respectively. With this in mind, it would seem that a more appropriate model for μ might be

$$\mu(x_1, x_2) = \beta_0 x_1^{\beta_1} x_2^{\beta_2}, \tag{1.9.2}$$

where β_1 and β_2 are parameters to adjust for the fact that a black cherry tree is not a perfect cylinder.

How can we fit this model? The model is not linear in the parameters any more, so our linear regression methods will not work... or will they? In the `trees` example we may take the logarithm of both sides of Equation **??** to get

$$\mu^*(x_1, x_2) = \ln\left[\mu(x_1, x_2)\right] = \ln\beta_0 + \beta_1 \ln x_1 + \beta_2 \ln x_2, \tag{1.9.3}$$

and this new model μ^* is linear in the parameters $\beta_0^* = \ln\beta_0$, $\beta_1^* = \beta_1$ and $\beta_2^* = \beta_2$. We can use what we have learned to fit a linear model `log(Volume) ~ log(Girth) + log(Height)`, and everything will proceed as before, with one exception: we will need to be mindful when it comes time to make predictions because the model will have been fit on the log scale, and we will need to transform our predictions back to the original scale (by exponentiating with `exp`) to make sense.

```
treesNonlin.lm <- lm(log(Volume) ~ log(Girth) + log(Height),
                     data = trees)
summary(treesNonlin.lm)
```

```

Call:
lm(formula = log(Volume) ~ log(Girth) + log(Height), data = trees)

Residuals:
      Min         1Q     Median         3Q        Max
-0.168561 -0.048488  0.002431  0.063637  0.129223
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.63162    0.79979  -8.292 5.06e-09 ***
log(Girth)   1.98265    0.07501  26.432  < 2e-16 ***
log(Height)  1.11712    0.20444   5.464 7.81e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08139 on 28 degrees of freedom
Multiple R-squared: 0.9777,Adjusted R-squared: 0.9761
F-statistic: 613.2 on 2 and 28 DF,  p-value: < 2.2e-16
```

This is our best model yet (judging by R^2 and $\overline{R}^2$), all of the parameters are significant, it is simpler than the quadratic or interaction models, and it even makes theoretical sense. It rarely gets any better than that.

We may get confidence intervals for the parameters, but remember that it is usually better to transform back to the original scale for interpretation purposes :

```
exp(confint(treesNonlin.lm))
```

```
                    2.5 %      97.5 %
 (Intercept) 0.0002561078 0.006783093
 log(Girth)  6.2276411645 8.468066317
 log(Height) 2.0104387829 4.645475188
```

(Note that we did not update the row labels of the matrix to show that we exponentiated and so they are misleading as written.) We do predictions just as before. Remember to transform the response variable back to the original scale after prediction.

```
new <- data.frame(Girth = c(9.1, 11.6, 12.5), Height = c(69, 74, 87))
exp(predict(treesNonlin.lm, newdata = new, interval = "confidence"))
```

```
        fit      lwr      upr
 1 11.90117 11.25908 12.57989
 2 20.82261 20.14652 21.52139
 3 28.93317 27.03755 30.96169
```

The predictions and intervals are slightly different from those calculated earlier, but they are close. Note that we did not need to transform the `Girth` and `Height` arguments in the dataframe `new`. All transformations are done for us automatically.

1.9.2 Real Nonlinear Regression

We saw with the `trees` data that a nonlinear model might be more appropriate for the data based on theoretical considerations, and we were lucky because the functional form of μ allowed us to take logarithms to transform the nonlinear model to a linear one. The same trick will not work in other circumstances, however. We need techniques to fit general models of the form

$$\mathbf{Y} = \mu(\mathbf{X}) + \epsilon, \tag{1.9.4}$$

where μ is some crazy function that does not lend itself to linear transformations.

There are a host of methods to address problems like these which are studied in advanced regression classes. The interested reader should see Neter *et al* [**?**] or Tabachnick and Fidell [**?**].

It turns out that John Fox has posted an Appendix to his book [**?**] which discusses some of the methods and issues associated with nonlinear regression; see here for more. Here is an example of how it works, based on a question from R-help.

```
# fake data
set.seed(1)
x <- seq(from = 0, to = 1000, length.out = 200)
y <- 1 + 2*(sin((2*pi*x/360) - 3))^2 + rnorm(200, sd = 2)
# plot(x, y)
acc.nls <- nls(y ~ a + b*(sin((2*pi*x/360) - c))^2,
               start = list(a = 0.9, b = 2.3, c = 2.9))
summary(acc.nls)
#plot(x, fitted(acc.nls))
```

```
Formula: y ~ a + b * (sin((2 * pi * x/360) - c))^2

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  0.95884    0.23097   4.151 4.92e-05 ***
b  2.22868    0.37114   6.005 9.07e-09 ***
c  3.04343    0.08434  36.084  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.865 on 197 degrees of freedom

Number of iterations to convergence: 3
Achieved convergence tolerance: 6.546e-08
```

1.9.3 Multicollinearity

A multiple regression model exhibits *multicollinearity* when two or more of the explanatory variables are substantially correlated with each other. We can measure multicollinearity by having one of the explanatory play the role of "dependent variable" and regress it on the remaining explanatory variables. The the R^2 of the resulting model is near one, then we say that the model is multicollinear or shows multicollinearity.

Multicollinearity is a problem because it causes instability in the regression model. The instability is a consequence of redundancy in the explanatory variables: a high R^2 indicates a strong dependence between the selected independent variable and the others. The redundant information inflates the variance of the parameter estimates which can cause them to be statistically insignificant when they would have been significant otherwise. To wit, multicollinearity is usually measured by what are called *variance inflation factors*.

Once multicollinearity has been diagnosed there are several approaches to remediate it. Here are a couple of important ones.

Principal Components Analysis. This approach casts out two or more of the original explanatory variables and replaces them with new variables, derived from the original ones, that are by design uncorrelated with one another. The redundancy is thus eliminated and we may proceed as usual with the new variables in hand. Principal Components Analysis is important for other reasons, too, not just for fixing multicollinearity problems.

Ridge Regression. The idea of this approach is to replace the original parameter estimates with a different type of parameter estimate which is more stable under multicollinearity. The estimators are not found by ordinary least squares but rather a different optimization procedure which incorporates the variance inflation factor information.

We decided to omit a thorough discussion of multicollinearity because we are not equipped to handle the mathematical details. Perhaps the topic will receive more attention in a later edition.

- What to do when data are not normal
  - Bootstrap (see Chapter **??**).

1.9.4 Akaike's Information Criterion

$$AIC = -2 \ln L + 2(p + 1)$$

1.10 Chapter Exercises

Exercise 1.1. Use Equations **??**, **??**, and **??** to prove the Anova Equality:

$$SSTO = SSE + SSR.$$