11

Linear regression

In Chapter 3 we looked at the simple linear regression model,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

as a way to summarize a linear relationship between pairs of data (x_i, y_i). In this chapter we return to this model. We begin with a review and then further the discussion using the tools of statistical inference. Additionally, we will see that the methods developed for this model extend readily to the multiple linear regression model where there is more than one predictor.[1]

11.1 The simple linear regression model

Many times we assume that an increase in a predictor variable will correspond to an increase (or decrease) in the response variable. A basic model for this is the simple linear regression model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i.$$

The y variable is called the response variable and the x variable the predictor variable, covariate, or regressor.

As a statistical model, this says that the value of y_i depends on three things: that of x_i, the function $\beta_0 + \beta_1 x$, and the value of the random variable ϵ_i. The model says that for a given value of x, the corresponding value of y can be found by first applying the function to x and then adding the random error term ϵ_i.

To be able to make statistical inference, we assume that the error terms, ϵ_i, are *i.i.d.* and have a $\mathsf{Normal}(0, \sigma)$ distribution. This assumption can be rephrased as an assumption on the randomness of the response variable. If the x values are fixed, then the distribution of y_i is normal with mean $\mu_{y|x} = \beta_0 + \beta_1 x_i$ (depending of the values of x) and variance σ^2 (not depending on the values of x). This can be expressed as y_i has a $\mathsf{Normal}(\beta_0 + \beta_1 x_i, \sigma)$

[1] There is a large literature on using R for modeling such as described here and related extensions. For example, all of these books are quite informative: [57], [29], [22], [25], [20], [21], and [48]. The text [36] introduces R through a modeling approach.

distribution. If the x values are random, the model assumes that, conditionally on knowing these random values, the same is true about the distribution of the y_i.

Estimating the parameters in simple linear regression

One goal when modeling is to "fit" the model by estimating the parameters based on the sample. For the regression model the method of least squares is used. With an eye toward a more general usage, suppose we have several predictors, $x_1, x_2, \ldots, x_k$; several parameters, $\beta_0, \beta_1, \ldots, \beta_p$; and some function, f, which gives the mean for the variables y_i. That is, the statistical model

$$y_i = f(x_{1i}, x_{2i}, \ldots, x_{ki} \,|\, \beta_1, \beta_2, \ldots, \beta_p) + \epsilon_i.$$

The method of least squares finds values for the β's that minimize the squared difference between the actual values, y_i, and those predicted by the function f. That is, the following sum is minimized:

$$\sum_i \left[y_i - f(x_{1i}, x_{2i}, \ldots, x_{ki} \,|\, \beta_0, \beta_1, \ldots, \beta_p)\right]^2.$$

For the simple linear regression model, the formulas are not difficult to write (they are given below). For the more general model, even if explicit formulas are known, we don't present them.

The simple linear regression model for y_i has three parameters, β_0, β_1, and σ^2. The least-squares estimators for these are

$$\widehat{\beta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}, \tag{11.1}$$

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}, \qquad \text{and} \tag{11.2}$$

$$\widehat{\sigma}^2 = \frac{1}{n-2} \sum [y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_i)]^2. \tag{11.3}$$

We call $\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$ the prediction line; a value $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$ the predicted value for x_i; and the difference between the actual and predicted values, $e_i = y_i - \widehat{y}_i$, the *residual*. The *residual sum of squares* is denoted RSS and is equal to $\sum_i e_i^2$.

Quickly put, the regression line is chosen to minimize the residual sum of squares, RSS; it has slope $\widehat{\beta}_1$, intercept $\widehat{\beta}_0$, and goes through the point $(\bar{x}, \bar{y})$. Furthermore, the estimate for σ^2 is $\widehat{\sigma}^2 = \text{RSS}/(n-2)$.

Figure 11.1 shows a data set simulated from the equation $y_i = 1 + 2x_i + \epsilon_i$, where $\beta_0 = 1, \beta_1 = 2$, and $\sigma^2 = 3$. Both the line $y = 1 + 2x$ and the regression line $\widehat{y} = 0.329 + 2.158 \cdot x$, predicted by the data, are drawn. They are different, of course, as one of them depends on the random sample. Keep in mind

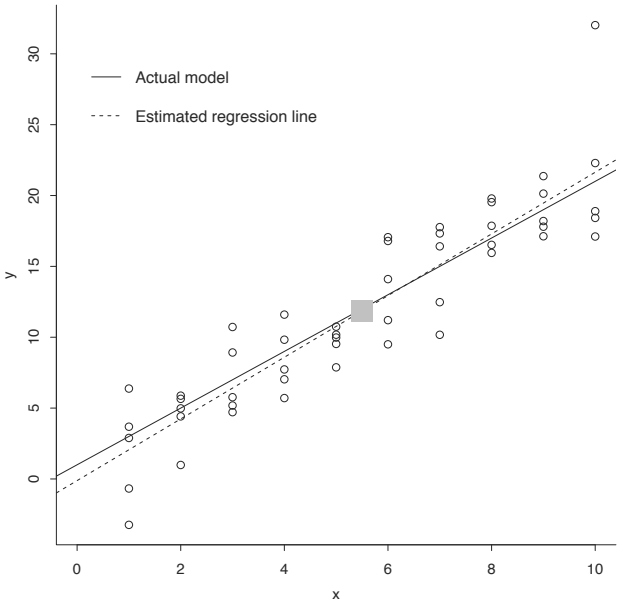

Figure 11.1: Simulation of model $y_i = 1 + 2x_i + \epsilon_i$. The regression line based on the data is drawn with dashes. The big square marks the value $(\bar{x}, \bar{y})$.

that the data is related by the true model, but if all we have is the data, the estimated model is given by the regression line. Our task of inference is to decide how much the regression line can tell us about the underlying true model.

Using `lm` to find the estimates

In Chapter 3 we learned how to fit the simple linear regression model using `lm`. The basic usage is of the form

```
lm(formula, data=..., subset=...)
```

Linear models are fit using R's model formulas, of which we have already seen a few examples.

The basic format for a formula is

```
response ~ predictor
```

The ~ (tilde) is read "is modeled by" and is used to separate the response from the predictor(s). The response variable can have regular mathematical expressions applied to it, but for the predictor variables the regular notations

+, -, *, /, and ^ have different meanings. A + means to add another term to the model, - means to drop a term, more or less coinciding with the symbols' common usage. But *, /, and ^ are used differently. If we want to use regular mathematical notation for the predictor we must insulate the symbols' usage with the I function, as in I(x^2).

As is usual with functions using model formulas, the data argument allows the variable names to reference those in the specified data frame, and the subset argument can be used to restrict the indices of the variables used by the modeling function.

By default, the lm function will print out the estimates for the coefficients. Much more is returned, but needs to be explicitly asked for. Usually, we store the results of the model in a variable, so that it can subsequently be queried for more information.

In Chapter 3 we fit a regression model to maximum heart rate by age with:

```
res.mhr <- lm(maxrate ~ age, data=heartrate)
res.mhr

##
## Call:
## lm(formula = maxrate ~ age, data = heartrate)
##
## Coefficients:
## (Intercept)          age
##     210.048       -0.798
```

These coefficients can be used directly for predictions. For example, a 50-year-old male would have a predicted maximum heart rate of:

```
208.36 - 0.76 * 50

## [1] 170.4
```

Extractor functions for lm

The lm function is reticent, but we can coax out more information as needed. This is done using extractor functions. Useful ones are summarized in Table 11.1.

These functions are passed an object returned by a modeling function, such as lm. These are "generic functions" which may have slightly different implementations depending on what type of model object is passed as the first object.

To illustrate, the estimate for σ^2 can be found using the resid function to retrieve the residuals from the model fitting:

Function	Description
`summary`	returns summary information about the regression
`plot`	makes diagnostic plots
`coef`	returns the coefficients
`confint`	returns confidence intervals for the coefficients
`vcov`	estimated covariance between parameter estimates
`residuals`	returns the residuals (can be abbreviated `resid`)
`fitted`	returns fitted values, $\widehat{y}_i$
`deviance`	returns RSS
`predict`	performs predictions
`anova`	finds various sums of squares
`AIC`	is used for model selection
`model.matrix`	matrix used to fit model mathematically

Table 11.1: Generic extractor functions for many of R's modeling functions, including `lm`.

```
n <- length(heartrate$age)
sum( resid(res)^2 ) / (n-2)

## [1] 31.05
```

Or, the RSS part can be found directly through `deviance`:

```
deviance(res)/ (n - 2)

## [1] 31.05
```

Problems

11.1 For the `Cars93` (`MASS`) data set, answer the following:

1. For `MPG.highway` modeled by `Horsepower`, find the simple regression coefficients. What is the predicted mileage for a car with 225 horsepower?
2. Fit the linear model with `MPG.highway` modeled by `Weight`. Find the predicted highway mileage of a 6,400 pound HUMMER H2 and a 2,524 pound MINI Cooper.
3. Fit the linear model `Max.Price` modeled by `Min.Price`. Why might you expect the slope to be around 1?

Can you think of any other linear relationships among the variables?

Age 2 (in.)	39	30	32	34	35	36	36	30
Adult (in.)	71	63	63	67	68	68	70	64

Table 11.2: Height as two-year-old and as an adult.

11.2 For the data set `MLBattend` (`UsingR`) concerning Major League Baseball attendance, fit a linear model of `attendance` modeled by `wins`. What is the predicted increase in attendance if a team that won 80 games last year wins 90 this year?

11.3 People often predict children's future height by using their 2-year-old height. A common rule is to double the height. Table 11.2 contains data for eight people's heights as 2-year-olds and as adults. Using the data, what is the predicted adult height for a 2-year-old who is 33 inches tall?

11.4 The `galton` (`UsingR`) data set contains data collected by Francis Galton in 1885 concerning the influence a parent's height has on a child's height. Fit a linear model for a child's height modeled by his parent's height. Make a scatterplot with a regression line. (Is this data set a good candidate for using `jitter`?) What is the value of $\widehat{\beta}_1$, and why is this of interest?

11.5 Formulas (11.1), (11.2), and the prediction line equation can be rewritten in terms of the correlation coefficient, r, as

$$\frac{\widehat{y}_i - \bar{y}}{s_y} = r\frac{x_i - \bar{x}}{s_x}.$$

Thus the five summary numbers: the two means, the standard deviations, and the correlation coefficient are fundamental for regression analysis.

This is interpreted as follows. Scaled differences of $\widehat{y}_i$ from the mean $\bar{y}$ are less than the scaled differences of x_i from $\bar{x}$, as $|r| \leq 1$. That is, "regression" toward the mean, as unusually large differences from the mean are lessened in their prediction for y.

For the data set `galton` (`UsingR`) use `scale` on the variables `parent` and `child`, and then model the height of the child by the height of the parent. What are the estimates for r and β_1?

11.2 Statistical inference for simple linear regression

If the simple regression model is appropriate for our data, then statistical inferences can be made about the unknown parameters.

Statistical inferences

If the linear model seems appropriate for the data, statistical inference is possible. What is needed is an understanding of the sampling distribution of the estimators.

To investigate these sampling distributions, we performed simulations of the model $y_i = x_i + \epsilon_i$, using `x <- rep(1:10,10)` and `y <- rnorm(100, x, 5)`. Figure 11.2 shows the resulting regression lines for the different simulations. For reference, a single result of the simulation is plotted using a scatterplot. There is wide variation among the regression lines. In addition, histograms of the simulated values of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are shown.

We see from the figure that the estimators are random but not arbitrary. Both $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are normally distributed, with respective means β_0 and β_1. Furthermore, $(n-2)\widehat{\sigma}^2/\sigma^2$ has a χ^2-distribution with $n-2$ degrees of freedom.

We will use the fact that the following statistics have a t-distribution with $n-2$ degrees of freedom:

$$\frac{\widehat{\beta}_0 - \beta_0}{\mathsf{SE}(\widehat{\beta}_0)}, \qquad \frac{\widehat{\beta}_1 - \beta_1}{\mathsf{SE}(\widehat{\beta}_1)}. \tag{11.4}$$

The standard errors are found from the known formulas for the variances of the $\widehat{\beta}_i$:

$$\mathsf{SE}(\widehat{\beta}_0) = \widehat{\sigma}\left(\sum \frac{x_i^2}{\sum(x_i - \bar{x})^2}\right)^{1/2}, \quad \mathsf{SE}(\widehat{\beta}_1) = \frac{\widehat{\sigma}}{\sqrt{\sum(x_i - \bar{x})^2}}. \tag{11.5}$$

(Recall that, $\widehat{\sigma}^2 = \mathsf{RSS}/(n-2)$.)

Marginal t-tests

We can find confidence intervals and construct significance tests from the statistics in (11.4) and (11.5). For example, a significance test for

$$H_0 : \beta_1 = b, \quad H_A : \beta_1 \neq b$$

is carried out with the test statistic

$$T = \frac{\widehat{\beta}_1 - \beta_1}{\mathsf{SE}(\widehat{\beta}_1)}.$$

Under H_0, T has the t-distribution with $n-2$ degrees of freedom.

A similar test for β_0 would use the test statistic $(\widehat{\beta}_0 - \beta_0)/\mathsf{SE}(\widehat{\beta}_0)$.

When the null hypothesis is $\beta_1 = 0$ or $\beta_0 = 0$ we call these tests *marginal t-tests*, as they test whether the parameter is necessary for the model without consideration of the other parameters involved.

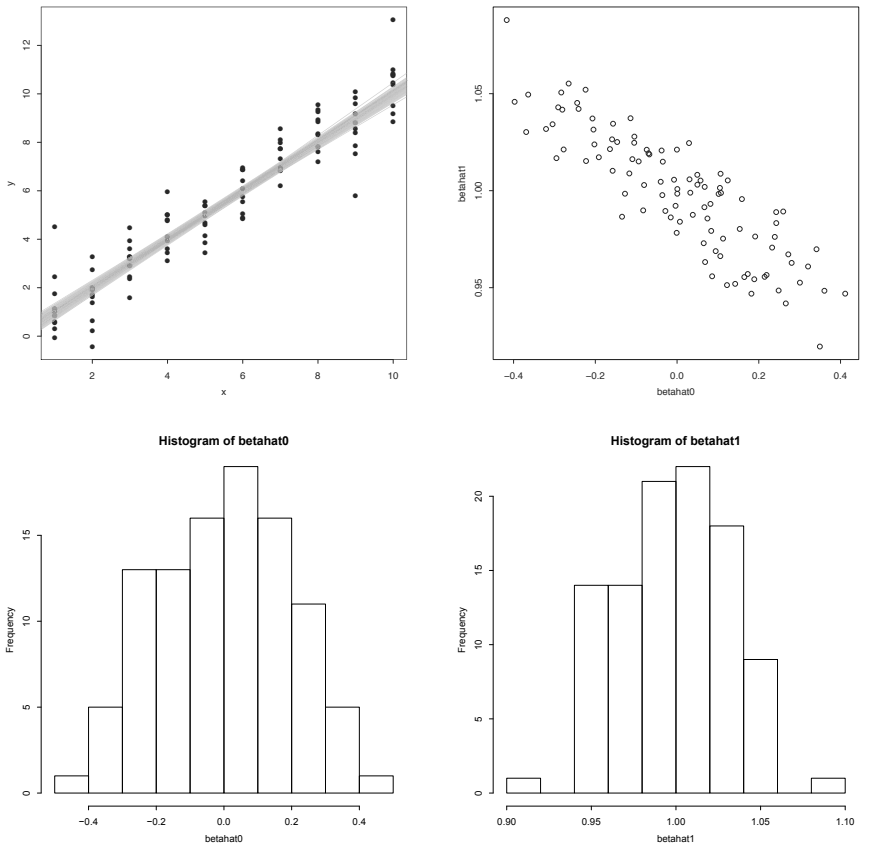

Figure 11.2: Four plots produced from a simulation finding the least squares regression coefficients from a known model. The upper left plot regression lines for 100 simulations from the model $y_i = x_i + \epsilon_i$. The plotted points show a single realization of the paired data during the simulation. The upper right plot shows a scatterplot of the points $(\widehat{\beta}_0, \widehat{\beta}_1)$. The lower left and right plots are histograms of $\widehat{\beta}_0$ and $\widehat{\beta}_1$.

The F-test

An alternate test for the null hypothesis $\beta_1 = 0$ can be done using a different but related approach that generalizes to the multiple-regression problem.

One goal of modeling is the attempt to explain the variation in the response variable using one or more predictor variables. The total variation in the y values about the mean is

$$\text{SST} = \text{total sum of squares} = \sum (y_i - \bar{y})^2.$$

Algebraically (or geometrically), this can be shown to be the sum of two easily interpreted terms:

$$\sum(y_i - \bar{y})^2 = \sum(y_i - \widehat{y}_i)^2 + \sum(\widehat{y}_i - \bar{y})^2. \tag{11.6}$$

The first term is the residual sum of squares, or RSS. The second is the total variation for the fitted model about the mean and is called the regression sum of squares, SSReg. Equation 11.6 becomes

$$\mathsf{SST} = \mathsf{RSS} + \mathsf{SSReg}.$$

For each term, a number—called the *degrees of freedom*—is assigned that depends on the sample size and the number of estimated values in the term. For the SST there are n data points and one estimated value, $\bar{y}$, leaving $n-1$ degrees of freedom. For RSS there are again n data points but two estimated values, $\widehat{\beta}_0$ and $\widehat{\beta}_1$, so $n-2$ degrees of freedom. This leaves 1 degree of freedom for the SSReg, as the degrees of freedom are additive in this case. When a sum of squares is divided by its degrees of freedom it is referred to as a *mean sum of squares*.

We rewrite the form of the prediction line to:

$$\widehat{y}_i = \bar{y} + \widehat{\beta}_1(x_i - \bar{x}).$$

If $\widehat{\beta}_1$ is close to 0, $\widehat{y}_i$ and $\bar{y}$ are similar in size, so we would have $\mathsf{SST} \approx \mathsf{RSS}$. In this case SSReg would be small. Whereas, if $\widehat{\beta}_1$ is not close to 0, then SSReg is not small. So, SSReg would be a reasonable test statistic for the hypothesis $H_0 : \beta_1 = 0$. What do small and big mean? As usual, we need to scale the value by the appropriate factor. The F-statistic is the ratio of the mean regression sum of squares divided by the mean residual sum of squares.

$$F = \frac{\mathsf{SSReg}/1}{\mathsf{RSS}/(n-2)} = \frac{\mathsf{SSReg}}{\widehat{\sigma}^2}. \tag{11.7}$$

Under the null hypothesis $H_0 : \beta_1 = 0$, the sampling distribution of F is known to be the F-distribution with 1 and $n-2$ degrees of freedom.

This allows us to make the following significance test.

F-test for $\beta_1 = 0$

A significance test for the hypotheses

$$H_0 : \beta_1 = 0, \qquad H_A : \beta_1 \neq 0$$

can be made with the the test statistic

$$F = \frac{\mathsf{SSReg}}{\widehat{\sigma}^2}.$$

Under the null hypothesis, the statistic F has the F-distribution with 1 and $n-2$ degrees of freedom. Larger values of F are more extreme, so the p-value for this test is computed from $\mathsf{P}(F \geq \text{ observed value } | H_0)$.

The F-statistic can be rewritten as

$$F = \left(\frac{\widehat{\beta}_1}{\mathsf{SE}(\widehat{\beta}_1)} \right)^2.$$

Under the assumption $\beta_1 = 0$, this is the square of one of the t-distributed random variables of Equation 11.4. For simple linear regression the two tests of $H_0 : \beta_1 = 0$, the marginal t-test and the F-test, are equivalent. However, we will see that with more predictors, the two tests are different.

R^2—the coefficient of determination

The decomposition of the total sum of squares into the residual sum of squares and the regression sum of squares in Equation 11.6 allows us to interpret how well the regression line fits the data. If the regression line fits the data well, then the residual sum of squares, $\sum(y_i - \widehat{y}_i)^2$, will be small. If there is a lot of scatter about the regression line, then RSS will be big. To quantify this, we can divide by the total sum of squares, leading to the definition of the *coefficient of determination*:

$$R^2 = 1 - \frac{\mathsf{RSS}}{\mathsf{SST}} = 1 - \frac{\sum(y_i - \widehat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} = \frac{\sum(\widehat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}. \tag{11.8}$$

This is close to 1 when the linear regression fit is good and close to 0 when it is not.

When the simple linear regression model is appropriate this value is interpreted as the proportion of the total response variation explained by the regression. That is, $R^2 \cdot 100\%$ of the variation is explained by the regression line. When R^2 is close to 1, most of the variation is explained by the regression line, and when R^2 is close to 0, not much is.

This interpretation is similar to that given for the Pearson correlation coefficient, r, in Chapter 3. This is no coincidence: for the simple linear regression model $r^2 = R^2$.

The *adjusted* R^2 divides the sums of squares by their degrees of freedom. For the simple regression model, these are $n-2$ for RSS and $n-1$ for SST. This is done to penalize models that get better values of R^2 by using more predictors. This is of interest when multiple predictors are used.

Using `lm` to find values for a regression model

Here we illustrate how R can be used to directly compute these values and, alternatively, how these values are returned by the `lm` object and its extractor methods.

Confidence intervals

For example, based on the distribution of $\widehat{\beta}_0$, a 95% confidence interval for β_0 can be found with:

$$\widehat{\beta}_0 \pm t^* \mathsf{SE}(\widehat{\beta}_0).$$

Using the values in our example, this could be found with

```
res.mhr <- lm(maxrate ~ age, data=heartrate)

betahat0 <- coef(res.mhr)[1]       # first coefficient
n <- nrow(heartrate)
sigmahat <- sqrt( sum(resid(res.mhr)^2) / (n - 2))
SE <- with(heartrate,
           sigmahat*sqrt(sum(age^2) / (n*sum((age - mean(age))^2)))
           )
SE

## [1] 2.867

tstar <- qt(1 - 0.05/2, df=n - 2)

betahat0 + c(-1, 1) * tstar * SE

## [1] 203.9 216.2
```

Standard error

The `summary` method for `lm` objects provides most of the values related to the model, including, for example, the standard error just computed. Find SE in the `Coefficients:` part of the output under the column labeled `Std. Error`.

```
summary(res.mhr)

##
## Call:
## lm(formula = maxrate ~ age, data = heartrate)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -8.926 -2.538  0.388  3.187  6.624
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  210.048      2.867    73.3  < 2e-16 ***
## age           -0.798      0.070   -11.4  3.8e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.58 on 13 degrees of freedom
## Multiple R-squared:  0.909,Adjusted R-squared:  0.902
## F-statistic:  130 on 1 and 13 DF,  p-value: 3.85e-08
```

By reading the standard error from this output, a 95% confidence interval for β_1 may be more easily found than the one for β_0 above:

```
betahat1 <- coef(res.mhr)[2]            # second coefficient
SE <- 0.06996281                        # read from summary
tstar <- qt(1 - 0.05/2, df=n - 2)
betahat1 + c(-1, 1) * tstar * SE

## [1] -0.9489 -0.6466
```

The two coefficients in this model are returned by the `coef` method:

```
coef(res.mhr)

## (Intercept)         age
##    210.0485     -0.7977
```

The `coef` method called on the *summary* of the model returns a matrix with the standard errors included:

```
coef(summary(res.mhr))

##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 210.0485    2.86694   73.27 2.124e-18
## age          -0.7977    0.06996  -11.40 3.848e-08
```

which can be used to programmatically extract the standard errors, as with:

```
coef(summary(res.mhr))["age", "Std. Error"]

## [1] 0.06996
```

The above shows how to do the work piece-by-piece. If that isn't of interest, the `confint` method can do both of these computations directly:

```
confint(res.mhr)

##                2.5 %   97.5 %
## (Intercept) 203.8548 216.2421
## age          -0.9489  -0.6466
```

Significance tests

The summary function for lm objects displays more than the standard errors. For each coefficient a marginal t-test is performed. This is a two-sided hypothesis test of the null hypothesis that $\beta_i = 0$ against the alternative that $\beta_i \neq 0$. We see in this case that both are rejected with very low p-values (as to be expected as we expect an intercept around 220 and slope around -1). These small p-values are flagged in the output of summary with significance stars.

Other t-tests are possible. For example, we can test the null hypothesis that the slope is -1 with the commands

```
mu0 <- -1
T.obs <- (betahat1 - mu0)/SE
T.obs

##   age
## 2.891

2*pt(abs(T.obs), df=n-2, lower.tail=FALSE)

##     age
## 0.01262
```

This is a small p-value, indicating that the model with slope -1 is unlikely to have produced this data or anything more extreme than it.

Finding $\widehat{\sigma}^2$, R^2

The estimate for $\widehat{\sigma}$ is marked Residual standard error and is labeled with $13 = 15 - 2$ degrees of freedom. The degrees of freedom are contained in the df.residual component of the model object. The estimate for $\widehat{\sigma}$ can be computed directly with:

```
sigma2 <- sum(resid(res.mhr)^2) / res.mhr$df.residual
sqrt(sigma2)                              # sigma hat

## [1] 4.578
```

The value of $R^2 =$ cor(age,mhr)^2 is given in the output along with an adjusted value.

F-test for $\beta_1 = 0$

Finally, the F-statistic is calculated. As this is given by $(\widehat{\beta}_1/\mathsf{SE}(\widehat{\beta}_1))^2$, it can be found directly with:

```
(-0.7595 / 0.0561)^2

## [1] 183.3
```

The significance test $H_0 : \beta_1 = 0$ with two-sided alternative is performed and again returns a tiny p-value.

The sum of squares to compute F are also given as the output of the `anova` extractor function.

```
anova(res.mhr)

## Analysis of Variance Table
##
## Response: maxrate
##           Df Sum Sq Mean Sq F value  Pr(>F)
## age        1   2725    2725     130 3.8e-08 ***
## Residuals 13    272      21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These values in the column headed `Sum Sq` are SSReg and RSS. The total sum of squares, SST, would be the sum of the two.

A short summary The `summary` function can feel a bit verbose at times. The following function will be used in the sequel to tighten the display up to show just the coefficient information:

```
short_summary <- function(x) {
  x <- summary(x)
  cmat <- coef(x)
  printCoefmat(cmat)
}
```

Predicting the response with `predict`

The function `predict` is used to make different types of predictions.

A template for its usage with `lm` objects is

```
predict(res, newdata=..., interval=..., level = ...)
```

The value of `res` is the output of an `lm` model. We call this `res` below, but we can use any valid name. Any changes to the values of the predictor are given to the argument `newdata` in the form of a data frame with names that match those used in the model formula. The arguments `interval` and `level` are set when prediction or confidence intervals are desired.

The simplest usage, `predict(res)`, returns the predicted values (the $\widehat{y}_i$'s) for the data. Predictions for other values of the predictor are specified using a data frame whose variable names match the variables used in the predictor side of the model, as this example illustrates:

```
res.mhr <- lm(maxrate ~ age, data=heartrate)
predict(res.mhr, newdata=data.frame(age=42))

##     1
## 176.5
```

This finds the predicted maximum heart rate for a 42-year-old. The `age` part of the data frame call is important. Variable names in the data frame supplied to the `newdata` argument must exactly match the variable names used when the model object was produced.

To assess whether the simple regression model is appropriate for the data we use a graphical approach.

Testing the model assumptions

The simple linear regression model, $y_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i + \epsilon_i = \mu_{y|x} + \epsilon_i$, places assumptions on the data set that we should verify before proceeding with any statistical inference. In particular, the linear model should be appropriate for the mean value of the y_i, and the error distribution should be normally distributed and independent.

Just as we looked at graphical evidence when investigating assumptions about normally distributed populations when performing a t-test, we will consider graphical evidence to assess the appropriateness of a regression model for the data. The `plot` method for `lm` (`?plot.lm`) objects can be used to plot 6 different diagnostic plots. We consider the four that are produced by default.[2]

The biggest key to assessing the aptness of the model is found in the residuals. The residuals are not an *i.i.d.* sample, as they sum to 0 and they do not have the same variance. The *standardized residuals* rescale the residuals to have unit variance.

[2]In using `plot` to produce the diagnostic plots it is convenient to first issue the command `par(mfrow=c(2,2))`. This sets up the plot device to have four panes for graphics added row by row.

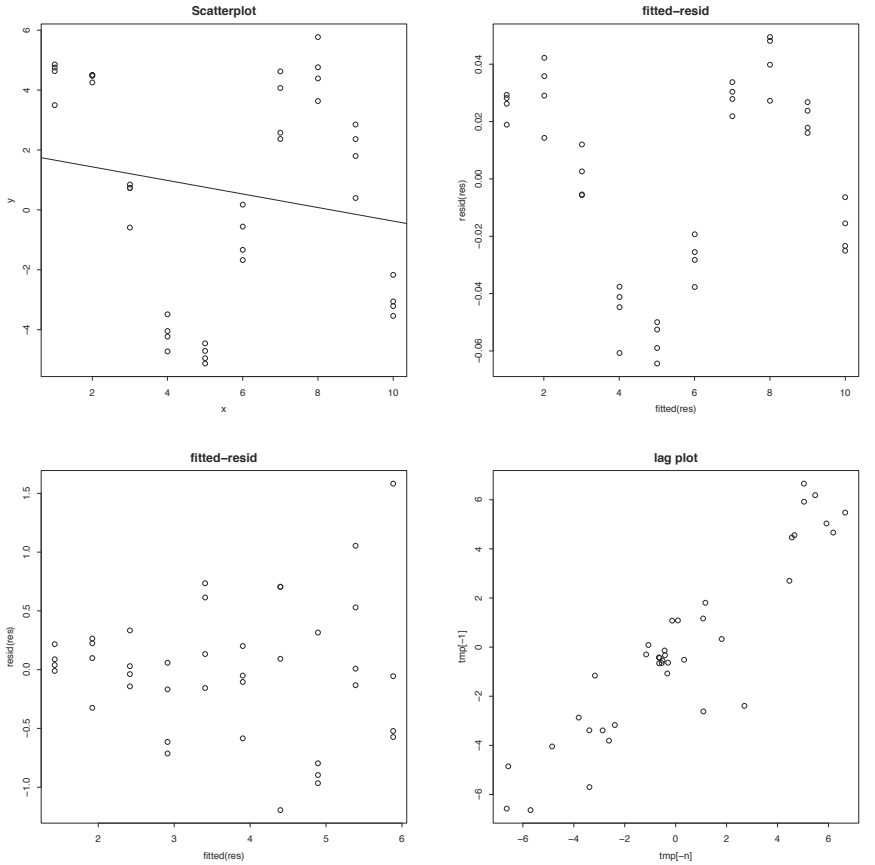

Figure 11.3: Four graphs showing problematic linear models. Scatterplot in upper left shows linear model is incorrect. Fitted versus residual plot in upper right shows a nonlinear trend. Fitted versus residual plot in lower left shows non-constant variance. Lag plot in lower right shows correlations in error terms.

Assessing the linear model for the mean

A scatterplot of the data with the regression line can show quickly whether the linear model seems appropriate for the data. If the general trend is not linear, either a transformation or a different model is called for. An example of a cyclical trend (which calls for a transformation of the data) is the upper-left plot in Figure 11.3 and is made with these commands:

```
x <- rep(1:10,4)
y <- rnorm(40, mean=5*sin(x), sd=1)
```

```
plot(y ~ x)
abline(lm(y ~ x))
```

When there is more than one predictor variable, a scatterplot will not be as useful.

A residual plot can also show whether the linear model is appropriate and can be made with more than one predictor. As well, it can detect small deviations from the model that may not show up in a scatterplot. The upper-right plot in Figure 11.3 shows a residual plot that finds a sinusoidal trend that will not show up in a scatterplot. It was simulated with these commands:

```
x <- rep(1:10, 4)
y <- rnorm(40, mean=x + 0.05 * sin(x), sd=0.01) # small trend
res <- lm(y ~ x)
plot(fitted(res), resid(res))
```

The residual plot is one of the four diagnostic plots produced by `plot`.

Assessing the residuals

The residuals are used to assess whether the error terms in the model are normally distributed. As mentioned, the residuals are correlated as they add to 0, we treat them as if they are the actual error terms in the model. For example, we use either a histogram or, preferably, a quantile-normal plot to investigate if a normal assumption is appropriate. For the quantile-normal plot, deviations from a straight line indicate non-normality. One of the diagnostic plots produced by `plot` is a quantile-normal plot of the standardized residuals. Though normality is not essential for prediction, the sampling distributions of the coefficients depend on the error terms not being too skewed or long-tailed.

In addition to normality, an assumption of the model is also that the error terms have a common variance. A residual plot can show whether this is the case. When it is, the residuals show scatter about a horizontal line. In many data sets, the variance increases for larger values of the predictor. The commands below create a simulation of this. The graph showing the effect is in the lower-left of Figure 11.3. The absence of equal variance can sometimes be addressed by transformations or weighted least squares, though we don't pursue that here.

```
x <- rep(1:10, 4)
y <- rnorm(40, mean=1 + 1/2*x, sd=x/10)
res <- lm(y ~ x)
plot(fitted(res), resid(res))
```

The scale-location plot is one of the four diagnostic plots produced by the defaults of the `plot` method. This graphic also shows the residuals, but in

terms of the square root of the absolute value of the standardized residuals. The graph should show points scattered along the y-axis, as we scan across the x-axis, but the spread of the scattered points should not get larger or smaller.

In some data sets, there is a lack of independence in the residuals. For example, the errors may accumulate. A lag plot, where the data is plotted against previous values of the data, may be able to show this effect. For an independent sequence, the lag plot should be scattered, whereas many dependent sequences will show some pattern. This is illustrated in the lower-right plot in Figure 11.3, which was made as follows:

```
x <- rep(1:10, 4)
epsilon <- rnorm(40, mean=0, sd=1)
y <- 1 + 2*x + cumsum(epsilon) # cumsum() correlates errors
res <- lm(y ~ x)
tmp <- resid(res)
n <- length(tmp)
plot(tmp[-n], tmp[-1])         # lag plot
```

Influential points

As we observed in Chapter 3, the regression line can be greatly influenced by a single observation that is far from the trend set by the data. The difference in slopes between the regression line with all the data and the regression line with the ith point missing will mostly be small, except for influential points. The Cook's distance is based on the difference of the predicted values of y_i for a given x_i when the point (x_i, y_i) is and isn't included in the calculation of the regression coefficients. Comparing predicted amounts, as opposed to change in slope, allows the method to generalize to more than one predictor. The Cook's distance is computed by the extractor function `cooks.distance`.

One of the diagnostic plots produced by the default `plot` method for `lm` objects will show the Cook's distance for the data points plotted using spikes. Another way to display this information graphically is to make the size of the points in the scatterplot depend on this distance using the `cex` argument. This type of plot is referred to as a bubble plot and is illustrated using the `emissions` (`UsingR`) data set in Figure 11.4. The graphic is made with the following commands:

```
res <- lm(CO2 ~ perCapita, emissions)
plot(CO2 ~ perCapita, emissions,
     cex=10*sqrt(cooks.distance(res)),
     main=expression(                    # make subscript on CO2
       paste("bubble plot of ",CO[2],
             " emissions by per capita GDP")
       ))
```

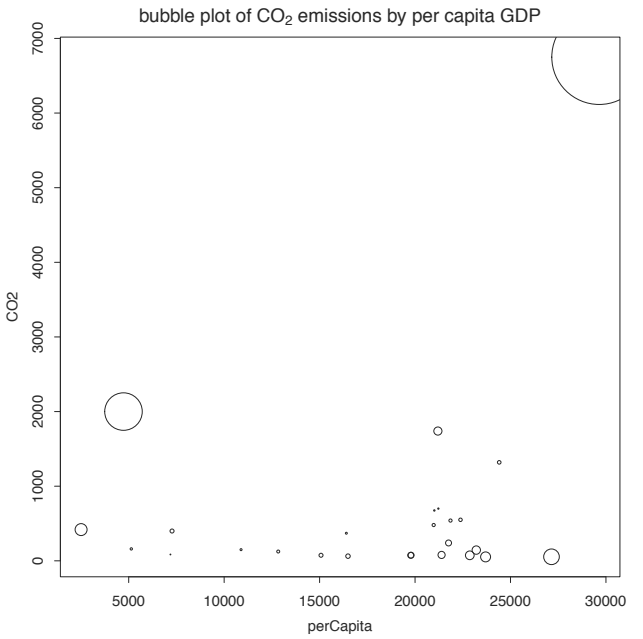

Figure 11.4: Bubble plot of CO_2 emissions by per capita GDP with area of points proportional to Cook's distance.

The square root of the distances is used, so the area of the points is proportional to Cook's distance rather than to the radius.[3]

For the maximum-heart-rate data, the four diagnostic plots produced by R with the command `plot(res.mhr)` are in Figure 11.5.

Prediction intervals

The value of $\widehat{y}$ can be used to predict two different things: the value of a single estimate of y for a given x or the average value of many values of y for a given x. If we think of a model with replication (repeated y's for a given x, such as in Figure 11.2), then the difference is clear: one is a prediction for a given point, the other a prediction for the average of the points.

Statistical inference about the predicted value of y based on the sample is done with a *prediction interval*. As y is not a parameter, we don't call this a confidence interval. The form of the prediction interval is similar to that of a confidence interval:

[3]The argument to `main` illustrates how to use mathematical notation in the title of a graphic. See the help page `?plotmath` for details.

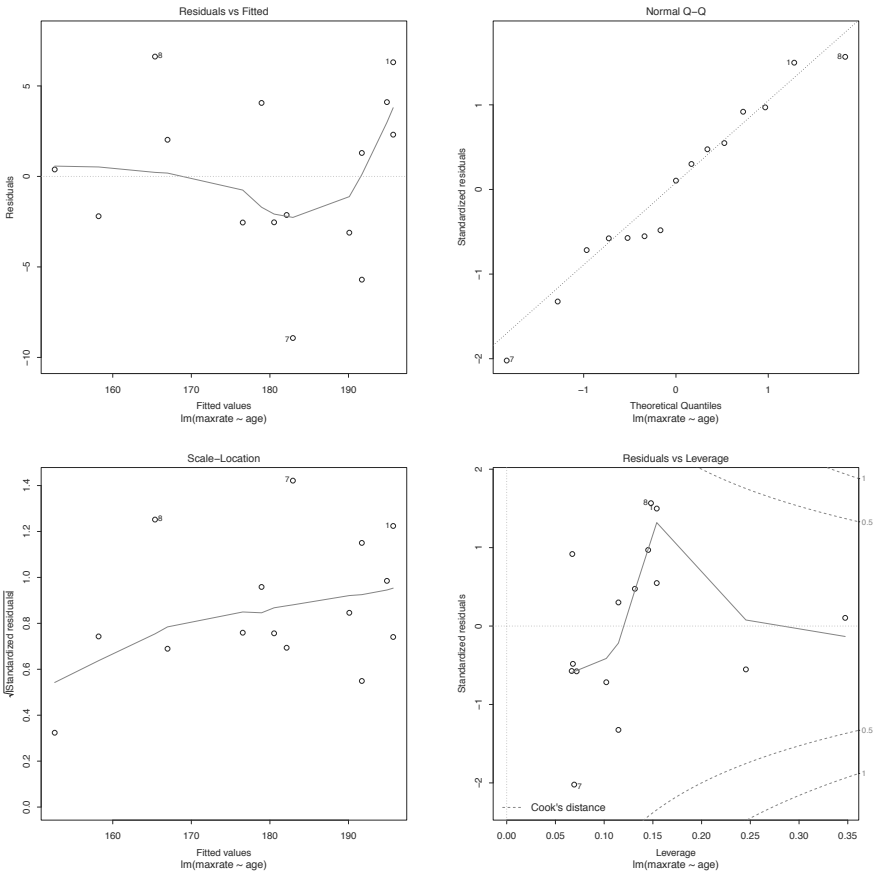

Figure 11.5: Four diagnostic plots for the maximum-heart-rate data produced by the extractor function `plot`.

$$\widehat{y} \pm t^* \mathsf{SE}.$$

For the prediction interval, the standard error depends on x and is given by

$$\mathsf{SE} = \widehat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{s_{xx}}}. \tag{11.9}$$

The value of t^* comes from the t-distribution with $n - 2$ degrees of freedom.

The prediction interval holds for all x simultaneously. Meaning, there is a $(1 - \alpha)100\%$ chance that a new data point chosen from the model will be

within these bounds. These values are usually plotted using two lines on the scatterplot to show the upper and lower limits.

The `predict` function will return the lower and upper endpoints for each value of the predictor. We specify `interval="prediction"` (which can be shortened) and a confidence level with `level`. (The default is 0.95.)

For the heart-rate example we have:

```
pred.res <- predict(res.mhr, int = "pred")

## Warning: predictions on current data refer to _future_ responses

head(pred.res, n=3)

##     fit   lwr   upr
## 1 195.7 185.1 206.3
## 2 191.7 181.3 202.1
## 3 190.1 179.7 200.5
```

A matrix is returned with columns giving the data we want. We cannot access these with the data frame notation `pred.res$lwr`, as the return value is not a data frame. Rather we can access the columns by name, like `pred.res[,'lwr']`, or by column number, as in

```
head(pred.res[, 2])                # the 'lwr' column

##     1     2     3     4     5     6
## 185.1 181.3 179.7 171.9 147.2 156.5
```

We want to plot both the lower and upper limits. In our example, we have the predicted values for the given values of age. As the age variable is not sorted, simply plotting will make a real mess. To remedy this, we specify the values of the age variable for which we make a prediction. We use the values `sort(unique(age))`, which gives just the x values in increasing order.

```
age.sort <- sort(unique(heartrate$age))
pred.res <- predict(res.mhr, newdata = data.frame(age = age.sort),
                    int="pred")
pred.res[,2]

##     1     2     3     4     5     6     7     8     9    10
## 185.1 184.3 181.3 179.7 172.7 171.9 170.3 168.7 166.3 156.5
##    11    12    13
## 154.8 147.2 141.1
```

Now we can add the prediction intervals to the scatterplot with the `lines` function (`matlines` offers a one-step alternative). The result is Figure 11.6.

```
plot(maxrate ~ age, data=heartrate)
abline(res.mhr)
lines(age.sort, pred.res[,2], lty=2) # lower curve
lines(age.sort, pred.res[,3], lty=2) # upper curve
```

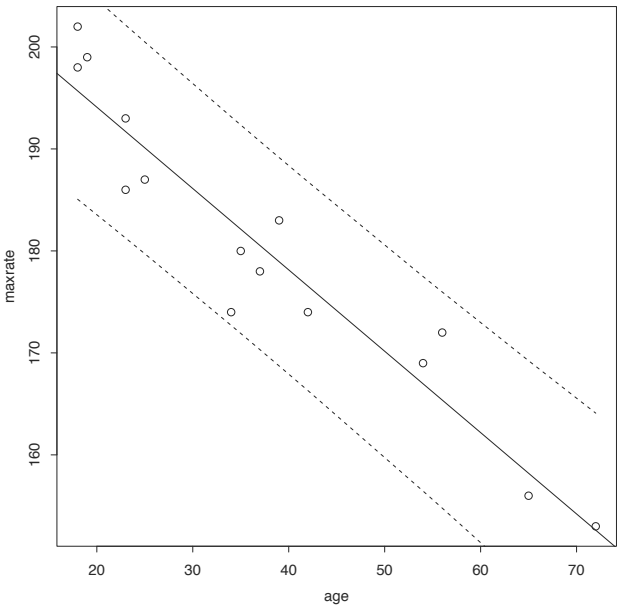

Figure 11.6: Regression line with 95% prediction intervals drawn for age versus maximum heart rate.

There is a slight curve in the lines drawn, which is hinted at in Equation 11.9. This implies that estimates near the value $(\bar{x}, \bar{y})$ have a smaller variance. This is expected: there is generally more data near this value, so the variances should be smaller.

Confidence intervals for $\mu_{y|x}$

A confidence interval for the mean value of y for a given x is given by

$$\widehat{y} \pm t^* \mathsf{SE}(\widehat{y}).$$

Again, t^* is from the t-distribution with $n-2$ degrees of freedom. The standard error used is now

$$\mathsf{SE}(\widehat{y}) = \widehat{\sigma}\sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{s_{xx}}}.$$

The standard error for the prediction interval differs by an extra term of plus 1 inside the square root. This may appear minor, but is not. If we had so much data (large n) that the estimates for the β's have small variance, we would not have much uncertainty in predicting the mean amount, but we would still have uncertainty in predicting a single deviation from the mean due to the error term in the model.

The values for this confidence interval are also returned by `predict`. In this case, we use the argument `interval="confidence"`.

Problems

11.6 The cost of a home is related to the number of bedrooms it has. Suppose the following table contains data recorded for homes in a given town.

price	\$300	\$250	\$400	\$550	\$317	\$389	\$425	\$289	\$389
bedrooms	3	3	4	5	4	3	6	3	4

Make a scatterplot, and fit the data with a regression line. On the same graph, test the hypothesis that an extra bedroom is worth \$60,000 versus the alternative that it is worth more.

11.7 The more beer you drink, the more your blood alcohol level (BAL) rises. The following table contains a data set on beer consumption.

beers	5	2	9	8	3	7	3	5	3	5
BAL	0.10	0.03	0.19	0.12	0.04	0.095	0.07	0.06	0.02	0.05

Make a scatterplot with a regression line and 95% prediction intervals drawn. Test the hypothesis that one beer raises your BAL by 0.02% against the alternative that it raises it less. (A formula from `wikipedia.org` specifies a model for the mean with

$$\frac{0.906 \cdot d \cdot 1.2}{(0.49 + 0.09 \cdot 1_{\text{a male}}) \cdot w} - 0.017 \cdot t$$

where d is the number of drinks, w the weight in kilograms, and t the time since drinking.)

11.8 For the same blood-alcohol data as the previous exercise perform a significance test that the intercept is 0 with a two-sided alternative.

11.9 The lapse rate is the rate at which temperature drops as you increase elevation. Some hardy students were interested in checking empirically whether the lapse rate of 9.8°C/km was accurate. To investigate, they grabbed their thermometers and their Suunto® wrist altimeters and recorded the data from their hike in this table:

elevation (ft)	600	1000	1250	1600	1800	2100	2500	2900
temperature (°F)	56	54	56	50	47	49	47	45

Draw a scatterplot with regression line and investigate whether the lapse rate is 9.8°C/km. (It helps to convert to the rate of change °F per feet, which is 5.34 degrees per 1,000 feet.) Test the hypothesis that the lapse rate is 5.34 degrees per 1,000 feet against a two-sided alternative.

11.10 For the `homedata` (`UsingR`) data set, find the regression equation to predict the year-2000 value of a home from its year-1970 value. Make a prediction for an $80,000 home in 1970. Comment on the appropriateness of the regression model by investigating the residuals.

11.11 A seal population is counted over a ten-year period. The counts are reported in this table:

year	pop.	year	pop.	year	pop	year	pop
1952	724	1955	1,392	1958	1,212	1961	1,980
1953	176	1956	1,392	1959	1,672	1962	2,116
1954	920	1957	1,448	1960	2,068		

Make a scatterplot with population on the y-axis and year on the x-axis. Find the regression line. What is the predicted value for 1963? Would you use this to predict the population in 2014? Why or why not?

11.12 The `deflection` (`UsingR`) data set contains deflection measurements for various loads. Fit a linear model to `Deflection` as a function of `load`. Plot the data and the regression line. How well does the line fit? Investigate with a residual plot.

11.13 The `alaska.pipeline` (`UsingR`) data set contains measurements of defects on the Alaska pipeline that are taken first in the field and then in the laboratory. The measurements are done in six batches. Fit a linear model for the lab-defect size as modeled by the field-defect size. Find the coefficients. Discuss the appropriateness of the model.

11.14 In athletic events in which people of various ages participate, performance is sometimes related to age. Multiplying factors are used to compare

the performance of a person of a given age to another person of a different age. The data set `best.times` (`UsingR`) features world records by age and distance in track and field.

We split the records by distance, allowing us to compare the factors for several distances.

```
by.dist <- split(best.times, as.factor(best.times$Dist))
```

This returns a list of data frames, one for each distance. We can plot the times in the 800-meter run:

```
plot(Time ~ age, by.dist[["800"]])
```

It is actually better to apply `scale` first, so that we can compare times.

Through age 70, a linear regression model seems to fit. It can be found with

```
lm(scale(Time) ~ age, by.dist[["800"]], subset = age < 70)

##
## Call:
## lm(formula = scale(Time) ~ age, data = by.dist[["800"]], subset = age <
##     70)
##
## Coefficients:
## (Intercept)          age
##     -1.2933       0.0136
```

Using the above technique, compare the data for the 100-meter dash, the 400-meter dash, and the 10,000-meter run. Are the slopes similar?

11.15 The `galton` (`UsingR`) data set contains data collected by Francis Galton in 1885 concerning the influence a parent's height has on a child's height. Fit a linear model modeling a child's height by his parents'. Do a test of significance to see whether β_1 equals 1 against a two-sided alternative.

11.16 Find and plot both the prediction and the confidence intervals for the heart-rate model: `res.mhr <- lm(maxrate ~ age, data=heartrate)`.

11.17 The `alaska.pipeline` (`UsingR`) data set appears appropriate for a linear model, but the assumption of equal variances does not seem appropriate. A log-transformation of each variable does seem to have equal variances. Fit the model

$$\log(\texttt{lab.defect}) = \beta_0 + \beta_1 \cdot \log(\texttt{field.defect}) + \epsilon.$$

Investigate the residuals and determine whether the assumption of equal variance seems appropriate.

11.18 The following commands will simulate the regression model $y_i = 1 + 2x_i + \epsilon_i$:

```
m <- 200
x <- rep(1:10, 4)
res <- replicate(m, {
  y <- rnorm(40, 1 + 2*x, 3)
  coef(lm(y ~ x))
})
plot(res[1,], res[2,])
```

Run the simulation and comment on the shape of the scatterplot. What does it say about the correlation between $\widehat{\beta}_0$ and $\widehat{\beta}_1$?

11.19 In a simple linear regression, confidence intervals for β_0 and β_1 are given separately in terms of the t-distribution as $\widehat{\beta}_i \pm t^* \mathsf{SE}(\widehat{\beta}_i)$. They can also be found *jointly*, giving a *confidence ellipse* for the parameters as a pair. This can be found easily in R with the `ellipse` package.[4]

If `res` is the result of the `lm` function, then `plot(ellipse(res), type="l")` will draw the confidence ellipse.

For the `deflection` (`UsingR`) data set, find the confidence ellipse for `Deflection` modeled by `Load`.

11.20 The linear regression model $y_i = \mu_{y|x_i} + \epsilon_i$ is flexible enough to accommodate some of the other models already encountered. The basic t-test is modeled by `y ~ 1`. The paired t-test becomes $y_i = \mu + x_i + \epsilon_i$ which can be modeled with `y ~ offset(x)`. The two-sample t-test can be modeled with a predictor which is 1 for one population and 0 for the other via `y ~ x`.

Let's see the latter. The `normtemp` (`UsingR`) data set has normal body temperature measurements for both men and women. A two-sample t-test can be employed to perform a significance test of difference between gender, via:

```
t.test(temperature ~ factor(gender), data=normtemp)
```

Find the corresponding p-value in the output of this model:

```
lm(temperature ~ factor(gender), data=normtemp)
```

[4]The `ellipse` package is not part of the standard R installation, but it is on CRAN. You can install it with the command `install.packages("ellipse")`.

11.3 Multiple linear regression

Multiple linear regression allows for more than one regressor to predict the value of y. Lots of possibilities exist. These regressors may be separate variables, products of separate variables, powers of the same variable, or functions of the same variable. In the next chapter, we will consider regressors that are not numeric but categorical. They all fit together in the same model, but there are additional details. We see, though, that much of the background for the simple linear regression model carries over to the multiple regression model.

Types of models

Let y be a response variable and let x_1, x_2,…, x_p be p variables that we will use for predictors. For each variable we have n values recorded. The multiple regression model we discuss here is

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \epsilon_i.$$

There are $p + 1$ parameters in the model labeled $\beta_0, \beta_1, \ldots, \beta_p$. They appear in a linear manner, just like a slope or intercept in the equation of a line. The x_i's are predictor variables, or covariates. They may be random; they may be related, such as powers of each other; or they may be correlated. As before, it is assumed that the ϵ_i values are an *i.i.d.* sample from a normal distribution with mean 0 and unknown variance σ^2. In terms of the y variable, the values y_i are an independent sample from a normal distribution with mean $\beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$ and common variance σ^2. If the x variables are random, this is true after conditioning on their values.

Interpretation For the simple linear regression model, the slope parameter, β_1, is easily interpreted, as one-unit change in the predictor variable will correspond to a predicted change in the mean response by β_1 units. For the multiple regression model, a similar interpretation is possible: a one-unit change in the ith predictor corresponds to a β_i-unit change in the predicted mean response *if the other predictors are held constant*. This is not always possible in practice.

• Example 11.1: What influences a baby's birth weight?
A child's birth weight depends on many things, among them the parents' genetic makeup, gestation period, and mother's activities during pregnancy. The `babies` (`UsingR`) data set lets us investigate some of these relationships.

This data set contains many variables to consider. We first look at the quantitative variables as predictors. These are gestation period; mother's age, height, and weight; and father's age, height, and weight.

A first linear model might incorporate all of these at once:

$$\text{wt} = \beta_0 + \beta_1 \cdot \text{gestation} + \beta_2 \cdot \text{mother's age} + \cdots + \beta_7 \cdot \text{father's weight} + \epsilon_i.$$

Why should this have a linear model? It seems intuitive that birth weight would vary monotonically with the variables, so a linear model might be a fairly good approximation. We'll want to look at some plots to make sure our model seems appropriate. ••

• Example 11.2: Polynomial regression

In 1609, Galileo proved mathematically that the horizontal distance traveled by an object with an initial horizontal velocity is a parabola. He based his insight on an experimental setup consisting of a ball placed at a certain height on a ramp and then released. The distance traveled was then measured. This experiment was chosen to reduce the effects of friction.[5] The data consists of two variables. Let's call them y for distance traveled and x for initial height. Galileo may have considered any of these polynomial models:

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_i + \epsilon_i, \\ y_i &= \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i, \text{ or} \\ y_i &= \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i. \end{aligned}$$

The ϵ_i would cover error terms that are presumably independent and normally distributed. The quadratic model (the second model) is correct under perfect conditions, as Galileo demonstrated, but the data may suggest a different model if the conditions are not perfect. ••

• Example 11.3: Predicting classroom performance

College admissions offices are faced with the problem of predicting future performance based on a collection of measures, such as grade-point average and standardized test scores. These values may be correlated. There may also be other variables that describe why a student does well, such as type of high school attended or student's work ethic.

Initial student placement is also a big issue. If a student does not place into the right class, he may become bored and leave the school. Successful placement is key to retention. For New York City high school graduates, available at time of placement are SAT scores and Regents Exam scores. High school grade-point average may be unreliable or unavailable.

The data set `stud.recs` (`UsingR`) contains test scores and initial grades in a math class for several randomly selected students. What can we predict about the initial grade based on the standardized scores?

[5] This example appears in Ramsey and Schafer [49], where a schematic of the experimental apparatus is drawn.

An initial model might be to fit a linear model for grade with all the other terms included. Other restricted models might be appropriate. For example, are the verbal SAT scores useful in predicting grade performance in a future math class? ••

Fitting the multiple regression model using `lm`

As seen previously, the method of least squares is used to estimate the parameters in the multiple regression model. We don't give formulas for computing the $\widehat{\beta}$'s but note that, since there are $p+1$ estimated parameters, the estimate for the variance changes to

$$\widehat{\sigma}^2 = \frac{\text{RSS}}{n-(p+1)}.$$

To find these estimates in R, again the `lm` function is used. The syntax for the model formula varies depending on the type of terms in the model. For these problems, we use `+` to add terms to a model, `-` to drop terms, and `I` to insulate terms so that the usual math notations apply.

For example, if x, y, and z are variables, then the following statistical models have the given R counterparts:

$z_i = \beta_0 + \beta_1 x_i + \beta_2 y_i + \epsilon_i$	is expressed as	`z ~ x + y`
$z_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$	is expressed as	`z ~ x + I(x^2)`

Once the model is specified, the `lm` function follows this familiar format:

```
lm(formula, data=..., subset=...)
```

To illustrate with an artificial example, we simulate the relationship $z_i = \beta_0 + \beta_1 x_i + \beta_2 y_i + \epsilon_i$ and then find the estimated coefficients:

```
x <- 1:10
y <- rchisq(10,3)
z <- 1 + x + y + rnorm(10)
lm(z ~ x + y)

##
## Call:
## lm(formula = z ~ x + y)
##
## Coefficients:
## (Intercept)            x            y
##      -0.367        1.179        0.990
```

The output of `lm` stores much more than is seen initially (which is just the formula and the estimates for the coefficients). It is recommended that the return value be stored. Afterward, the different extractor functions can be used to view the results.

• **Example 11.4: Finding the regression estimates for baby's birth weight** Fitting the birth-weight model is straightforward. The basic model formula is

```
wt ~ gestation + age + ht + wt1 + dage + dht + dwt
```

We've seen with this data set that the variables have some missing values that are coded not with `NA` but with very large values that are obvious when plotted, but not when we blindly use the functions. In particular, gestation should be less than 350 days, mother's age and height less than 99, and weight less than 999, etc. We can avoid these cases by using the `subset` argument as illustrated. Recall that we combine logical expressions with `&` for "and" and `|` for "or."

```
res.lm <- lm(wt ~ gestation + age + ht + wt1 + dage + dht + dwt,
 data=babies,
 subset=gestation < 350 & age < 99 & ht < 99 & wt1 < 999 &
        dage < 99 & dht < 99 & dwt < 999)
res.lm

##
## Call:
## lm(formula = wt ~ gestation + age + ht + wt1 + dage + dht + dwt,
##     data = babies, subset = gestation < 350 & age < 99 & ht <
##         99 & wt1 < 999 & dage < 99 & dht < 99 & dwt < 999)
##
## Coefficients:
## (Intercept)    gestation          age           ht          wt1
##   -105.4576       0.4625       0.1384       1.2161       0.0289
##        dage          dht          dwt
##      0.0590      -0.0663       0.0782
```

A residual plot (not shown) shows nothing too unusual:

```
plot(fitted(res.lm), resid(res.lm))
```

The diagnostic plots found with `plot(res.lm)` indicate that observation 261 might be a problem. Looking at `babies[261,]`, it appears that this case is an outlier, as it has a very short gestation period. ••

Using `update` with model formulas

When comparing models, we may be interested in adding or subtracting a term and refitting. Rather than typing in the entire model formula again, R provides a way to add or drop terms from a model and have the new model fit. This process is called updating and is done with the `update` function. The usage is

```
update(model.object, formula = . ~ . + new.terms - old.terms)
```

The `model.object` is the output of some modeling command, such as `lm`. The `formula` argument uses a `.` to represent the previous value. In the template above, the `.` to the left of the `~` indicates that the previous left side of the model formula should be reused. The right-hand-side `.` refers to the previous right-hand side. In the template, the `+new.terms` means to add term and `-old.terms` is used to drop terms.

• Example 11.5: Discovery of the parabolic trajectory

The data set `galileo` (`UsingR`) contains two variables measured by Galileo (described previously). One is the initial height and one the horizontal distance traveled.

A plot of the data illustrates why Galileo may have thought to prove that the correct shape is described by a parabola. Clearly a straight line does not fit the data well. However, with modern computers, we can investigate whether a cubic term is warranted for this data.

To do so we fit three polynomial models. The `update` function is used to add terms to the previous model to give the next model. To avoid a different interpretation of `^`, the powers are insulated with `I`.

```
init.h <- c(600,700,800,950,1100,1300,1500)
h.d <- c(253, 337, 395, 451, 495, 534, 573)
res.lm  <- lm(h.d ~ init.h)
res.lm2 <- update(res.lm,  . ~ . + I(init.h^2))
res.lm3 <- update(res.lm2, . ~ . + I(init.h^3))
```

To plot these, we will use `curve`, but first we define a function which evaluates a polynomial given its coefficients:

```
polynomial <- Vectorize(function(x, ps) {
  n <- length(ps)
  sum(ps*x^(1:n-1))
}, "x")
```

Then we can plot as follows (Figure 11.7).

```
plot(h.d ~ init.h)
curve(polynomial(x, coef(res.lm )), add=TRUE, lty=1)
curve(polynomial(x, coef(res.lm2)), add=TRUE, lty=2)
curve(polynomial(x, coef(res.lm3)), add=TRUE, lty=3)
legend(1200, 400, legend=c("linear", "quadratic", "cubic"), lty=1:3)
```

The linear model is a poor fit, but both the quadratic and cubic fits seem reasonable. ●●

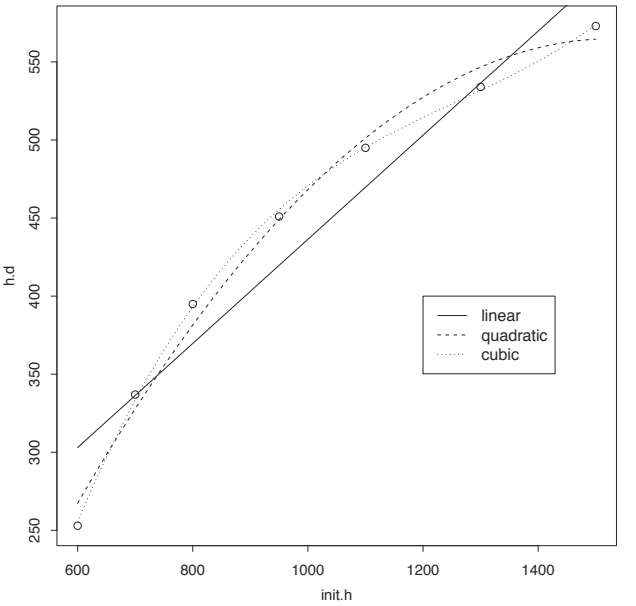

Figure 11.7: Three polynomial models fit to the Galileo data.

Interpreting the regression parameters

As mentioned, interpretation in simple regression is usually straightforward. Changes in the predictor variable correspond to changes in the response variable in a linear manner: a unit change in the predictor corresponds to a $\widehat{\beta}_1$-unit change in the response.

However, in multiple regression this picture may not be applicable, as we may not be able to change just a single variable. As well, when more

variables are added to a model, if the variables are correlated then the sign of the coefficients can change, leading to a different interpretation.

The language often used is that we "control" the other variables while seeking a primary predictor variable.

• **Example 11.6: Does taller mean higher paid?**
A University of Florida press release from October 16, 2003, read:

> "Height matters for career success..."
> The reported study, which controlled for gender, weight, and age, found that mere inches cost thousands of dollars. Each inch in height amounted to about $789 more a year in pay, the study found.

The mathematical model mentioned would be

$$\text{pay} = \beta_0 + \beta_1 \text{ height} + \beta_2 \text{ gender} + \beta_3 \text{ weight} + \beta_4 \text{ age} + \epsilon.$$

(In the next chapter we see how to interpret the term involving the categorical variable `gender`.) The data gives rise to the estimate $\widehat{\beta}_1 = 789$. The authors interpret this to mean that each extra inch of height corresponds to a $789 increase in expected pay. So someone who is 4 inches taller, say 6 feet versus 5 feet 8 inches, would be expected to earn $3,156 more annually. ($\widehat{y}$ is used to predict expected values.) The word "controlled" means that they included these variables in the model.

There are few caveats to this interpretation. First, unlike in a science experiment, where we may be able to specify the value of a variable, a person cannot simply grow an inch to see if his salary goes up. As well, it isn't realistic to imagine a person growing an inch without some change in their weight, say. So it is hard to hold all other variables equal when interpreting the coefficient. Further, as this is an observational study, causal interpretations are not necessarily valid. ••

Statistical inferences

As in the simple linear regression case, if the model is correct, statistical inference can be made about the coefficients. In general, the estimators for a linear model are unbiased and normally distributed; from this, t-tests and confidence intervals can be constructed for the estimators, once we learn the standard errors. As before, these are output by the `summary` function.

• **Example 11.7: Galileo, continued**
For the Galileo data example, the `summary` of the quadratic fit contains

```
short_summary(res.lm2)

##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.40e+02   6.90e+01   -3.48   0.0253 *
## init.h       1.05e+00   1.41e-01    7.48   0.0017 **
## I(init.h^2) -3.44e-04   6.68e-05   -5.15   0.0068 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For each $\widehat{\beta}$, the standard errors are given, as is the marginal t-test, which tests for the null hypothesis that the $\widehat{\beta}$ is 0. All three have small p-values and are flagged as such with significance stars.

Finding a confidence interval for the parameters is straightforward, as the values $(\widehat{\beta}_i - \beta_i)/\mathsf{SE}(\widehat{\beta}_i)$ have a t-distribution with $n - (p + 1)$ degrees of freedom if the linear model applies.

For example, a 95% confidence interval for β_1 would be

```
alpha <- 0.05
tstar <- qt(1 - alpha/2, df=4)          # n=7; p=2; df=n-(p+1)
beta1 <- 1.05
SE <- 0.141
beta1 + c(-1,1 ) * tstar * SE

## [1] 0.6585 1.4415
```

••

Model selection

Modeling is done for many reasons. One is to shine the focus on the important predictors to explain as much variation in the response as possible while avoiding the noise of unimportant factors. Doing this requires some means for determining when a predictor variable contributes sufficiently to the description of the response as to be warranted. For this we discuss a few criteria below that are easily used within R.

Before proceeding with methods to remove variables from consideration, we paraphrase some practical, general principles on building regression models for prediction provided in Section 4.6 of [25]:

- Include all input variables that might be expected to be important in predicting the response.
- Sometimes, predictors can be combined into other variables. For example, using BMI instead of both height and weight.

- For decisions on which variables to exclude:
  - If a predictor is not statistically significant and has the expected sign it is generally fine to leave it in (though the methods below will exclude it).
  - Consider removing predictors which are not statistically significant and do not have the expected sign.
  - If a predictor is statistically significant and has the expected sign, leave it in.
  - If a predictor is statistically significant and *does not* have the expected sign, then think hard about its inclusion. It might point to lurking variables, or underlying correlations with other predictors.

Partial F-test

The partial F-test is used to discriminate between two models with one being nested in the other. For example,

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki} + \epsilon_i \\ y_i &= \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki} + \beta_{k+1} x_{(k+1)i} + \cdots + \beta_p x_{pi} + \epsilon_i. \end{aligned} \tag{11.10}$$

The first model has $k+1$ parameters, and the second has $p+1$ with $p > k$ (not counting σ). Recall that the residual sum of squares, RSS, measures the variation between the data and the model. For the model with p predictors, $\mathsf{RSS}(p)$ can be no more than $\mathsf{RSS}(k)$ for the model with k predictors. Call the difference the *extra sum of squares*.

If the new parameters are not really important, then there should be little difference between the sums of squares when computed with or without the new parameters. If they are important, then there should be a big difference. To measure big or small, we can divide by the residual sum of squares for the full model. That is,

$$\frac{\mathsf{RSS}(k) - \mathsf{RSS}(p)}{\mathsf{RSS}(p)}$$

should measure the influence of the extra parameters. If we divide the extra sum of squares by $p-k$ and the residual sum of squares by $n-(p+1)$ (the respective degrees of freedom), then the statistic becomes

$$F = \frac{(\mathsf{RSS}(k) - \mathsf{RSS}(p))/(p-k)}{\mathsf{RSS}(p)/(n-(p+1)))} = \frac{(\mathsf{RSS}(k) - \mathsf{RSS}(p))/(p-k)}{\widehat{\sigma}^2}. \tag{11.11}$$

This statistic is actually a more general example of that in Equation 11.7 and has a similar sampling distribution. Under the null hypothesis that the

extra β's are 0 ($\beta_{k+1} = \cdots = \beta_p = 0$), and the ϵ_i are *i.i.d.* with a Normal$(0,\sigma^2)$ distribution, F will have the F-distribution with $p-k$ and $n-(p+1)$ degrees of freedom.

This leads to the following significance test.

Partial F-test for null hypothesis of no effect

For the nested models of Equation 11.10, a significance test for the hypotheses

$$H_0 : \beta_{k+1} = \beta_{k+2} = \cdots = \beta_p = 0 \quad \text{and} \quad H_A : \text{ at least one } \beta_j \neq 0 \text{ for } j > k$$

can be performed with the test statistic (11.11):

$$F = \frac{\text{extra sum of squares}/(p-k)}{\widehat{\sigma}^2}.$$

Under H_0, F has the F-distribution with $p-k$ and $n-(p+1)$ degrees of freedom. Large values of F are in the direction of the alternative. This test is called the *partial F-test*.

The anova function will perform the partial F-test. If `res.lm1` and `res.lm2` are the return values of two nested models, then

```
anova(res.lm1, res.lm2)
```

will perform the test and produce an analysis of variance table.

• Example 11.8: Discovery of the parabolic trajectory revisited
In Example 11.3 we fit the data with three polynomials, graphing each. Referring to Figure 11.7, we see that the parabola and cubic clearly fit better than the linear. But which of those two fits better? We use the partial F-test to determine whether the extra cubic term is significant.

To do this, we use the anova function on the two results `res.lm2` and `res.lm3`. This yields

```
anova(res.lm2,res.lm3)

## Analysis of Variance Table
##
## Model 1: h.d ~ init.h + I(init.h^2)
## Model 2: h.d ~ init.h + I(init.h^2) + I(init.h^3)
##   Res.Df RSS Df Sum of Sq    F Pr(>F)
## 1      4 744
## 2      3  48  1       696 43.3 0.0072 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-test is significant ($p = 0.0072$), indicating that the null hypothesis ($\beta_3 = 0$) does not describe the data well. This suggests that the underlying relationship from Galileo's data is cubic and not quadratic. Perhaps the apparatus introduced drag. ••

The Akaike information criterion

In the partial F-test, the trade-off between adding more parameters to improve the model fit and making a more complex model appears in the $n - (p + 1)$ divisor. Another common criterion with this trade-off is Akaike's information criterion (AIC). The AIC is computed in R with the `AIC` extractor function. The details of the statistic involve the likelihood function, a more advanced concept, but the usage is straightforward: models with lower AICs are preferred. An advantage to the AIC is that it can be used to compare models that are not nested, a restriction of the partial F-test.

The extractor function `AIC` will compute the value for a given model, but the convenient `stepAIC` function from the `MASS` package will step through the submodels and do the comparisons for us.

• Example 11.9: Predicting grades based on standardized tests
The data set `stud.recs` (`UsingR`) contains five standardized test scores and a numeric value for the initial grade in a subsequent math course. The goal is to use the test-score data to predict the grade that a student will get. If the grade is predicted to be low, perhaps an easier class should be recommended.

First, we view the data using paired scatterplots

```
pairs(stud.recs)
```

The figure (not shown) indicates strong correlations among the variables.

We begin by fitting the entire model. In this case, the convenient `.` syntax on the right-hand side is used to indicate all the remaining variables.

```
d <- subset(stud.recs, select=-letter.grade)
res.lm <- lm(num.grade ~ ., data = d)
res.lm

##
## Call:
## lm(formula = num.grade ~ ., data = d)
##
## Coefficients:
```

```
## (Intercept)        seq.1        seq.2        seq.3        sat.v
##    -0.73953     -0.00394     -0.00272      0.01565     -0.00125
##       sat.m
##     0.00590
```

Some terms are negative, which seems odd. (Why?) Looking at the summary of the regression model we have

```
short_summary(res.lm)

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.73953    1.21128   -0.61    0.543
## seq.1       -0.00394    0.01457   -0.27    0.787
## seq.2       -0.00272    0.01503   -0.18    0.857
## seq.3        0.01565    0.00941    1.66    0.099 .
## sat.v       -0.00125    0.00163   -0.77    0.443
## sat.m        0.00590    0.00267    2.21    0.029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The marginal t-tests for whether the given parameter is 0 or not are "rejected" only for the `seq.3` (for this sample of students, sequential 3 was the last high school test taken) and `sat.m` (the math SAT score). It is important to remember that these are tests concerning whether the value is 0 given the other predictors. They can change if correlated predictors are removed.

The `stepAIC` function can step through the various submodels and rank them by AIC. This gives

```
library(MASS)                 # load in MASS package for stepAIC
stepAIC(res.lm, trace=0)      # trace=0 suppresses intermediate output

##
## Call:
## lm(formula = num.grade ~ seq.3 + sat.m, data = d)
##
## Coefficients:
## (Intercept)        seq.3        sat.m
##    -1.14078      0.01371      0.00479
```

The submodel with just two predictors is selected. As expected, the verbal scores on the SAT are not a useful indicator of performance. ●●

Problems

11.21 Following the example with Galileo's data, fit a fourth-degree polynomial to the `galileo` (`UsingR`) data and compare to the cubic polynomial using a partial F-test. Is the new coefficient significant?

11.22 For the data set `trees`, model the `Volume` by the `Girth` and `Height` variables. Does the model fit the data well?

11.23 The data set `MLBattend` (`UsingR`) contains attendance data for Major League Baseball for the years 1969 to 2000. Fit a linear model of `attendance` modeled by `year`, `runs.scored`, `wins`, and `games.behind`. Which variables are flagged as significant? Look at the diagnostic plots and comment on the validity of the model.

11.24 For the `deflection` (`UsingR`) data set, fit the quadratic model

$$\texttt{Deflection} = \beta_0 + \beta_1 \texttt{Load} + \beta_2 \texttt{Load}^2 + \epsilon.$$

How well does this model fit the data? Compare to the linear model.

11.25 The data set `kid.weights` contains age, weight, and height measurements for several children. Fit the linear model

$$\texttt{weight} = \beta_0 + \beta_1 \texttt{age} + \beta_2 \texttt{height} + \beta_3 \texttt{height}^2 + \beta_4 \texttt{height}^3 + \beta_5 \texttt{height}^4$$

Use the partial F-test to select between this model and the nested models found by using only first-, second-, and third-degree polynomials for `height`.

11.26 The data set `fat` (`UsingR`) contains several body measurements that can be done using a scale and a tape measure. These can be used to predict the body-fat percentage (`body.fat`). Measuring body fat requires a special apparatus; if our resulting model fits well, we have a low-cost alternative.

Fit the variable `body.fat` using each of the variables `age`, `weight`, `height`, `BMI`, `neck`, `chest`, `abdomen`, `hip`, `thigh`, `knee`, `ankle`, `bicep`, `forearm`, and `wrist`. Use the `stepAIC` function to select a submodel. For this submodel, what is the adjusted R^2?

11.27 The data set `Cars93` (`MASS`) contains data on cars sold in the United States in the year 1993. Fit a regression model with `MPG.city` modeled by the numeric variables `EngineSize`, `Weight`, `Passengers`, and `price`. Which variables are marked as statistically significant by the marginal t-tests? Which model is selected by the AIC?

11.28 We can simulate the data to see how often the partial F-test or AIC works. For example, a single simulation can be done with the commands

```
x <- 1:10
y <- rnorm(10, 1 + 2*x + 3*x^2, 4)
require(MASS)
stepAIC(lm(y ~ x + I(x^2)), trace=0)

##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Coefficients:
## (Intercept)            x       I(x^2)
##      -0.583        3.494        2.846
```

Do a few simulations to see how often the correct model is selected.

11.29 The data set baycheck (UsingR) contains estimated populations for a variety of Bay Checkerspot butterflies near California. A common model for population dynamics is the Ricker model, for which t is time in years:

$$N_{t+1} = aN_t e^{bN_t} W_t,$$

where a and b are parameters and W_t is a lognormal multiplicative error. This can be turned into a regression model by dividing by N_t and then taking logs of both sides to give

$$\log(\frac{N_{t+1}}{N_t}) = \log(a) + bN_t + \epsilon_t.$$

Let y_t be the left-hand side. This may be written as

$$y_t = r(1 - \frac{N_t}{K}) + \epsilon_t,$$

because r can be interpreted as an unconstrained growth rate and K as a carrying capacity.

Fit the model to the baycheck data set and find values for r and K. To find y_t you can do the following:

```
d <- with(baycheck, {
  n <- length(year)
  yt <- log(Nt[-1]/Nt[-n])
  nt <- Nt[-n]
  data.frame(yt, nt)
})
```

Recall that a negative index means all but that index.