Simple Linear Regression

G. Jay Kerns

September 9, 2011

1 Simple Linear Regression

What do I want them to know?

- basic philosophy of SLR and the regression assumptions
- point and interval estimation of the model parameters, and how to use it to make predictions
- point and interval estimation of future observations from the model
- regression diagnostics, including R^2 and basic residual analysis
- the concept of influential versus outlying observations, and how to tell the difference

1.1 Basic Philosophy

Here we have two variables X and Y. For our purposes, X is not random (so we will write x), but Y is random. We believe that Y depends in *some* way on x. Some typical examples of (x, Y) pairs are

- x = study time and Y = score on a test.
- x = height and Y = weight.
- x = smoking frequency and Y = age of first heart attack.

Given information about the relationship between x and Y, we would like to *predict* future values of Y for particular values of x. This turns out to be a difficult problem [1] , so instead we first tackle an easier problem: we estimate $\mathbb{E}Y$. How can we accomplish this? Well, we know that Y depends somehow on x, so it stands to reason that

$$\mathbb{E}Y = \mu(x), \text{ a function of } x. \tag{1.1.1}$$

But we should be able to say more than that. To focus our efforts we impose some structure on the functional form of μ. For instance,

- if $\mu(x) = \beta_0 + \beta_1 x$, we try to estimate β_0 and β_1.

[1] Yogi Berra once said, "It is always difficult to make predictions, especially about the future."

- if $\mu(x) = \beta_0 + \beta_1 x + \beta_2 x^2$, we try to estimate β_0, β_1, and β_2.
- if $\mu(x) = \beta_0 e^{\beta_1 x}$, we try to estimate β_0 and β_1.

This helps us in the sense that we concentrate on the estimation of just a few parameters, β_0 and β_1, say, rather than some nebulous function. Our *modus operandi* is simply to perform the random experiment n times and observe the n ordered pairs of data (x_1, Y_1), (x_2, Y_2), $\ldots$, (x_n, Y_n). We use these n data points to estimate the parameters.

More to the point, there are *three simple linear regression* (SLR) assumptions that will form the basis for the rest of this chapter:

Assumption 1.1. *We assume that μ is a linear function of x, that is,*

$$\mu(x) = \beta_0 + \beta_1 x, \tag{1.1.2}$$

where β_0 and β_1 are unknown constants to be estimated.

Assumption 1.2. *We further assume that Y_i is $\mu(x_i)$, a "signal", plus some "error" (represented by the symbol ϵ_i):*

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, 2, \ldots, n. \tag{1.1.3}$$

Assumption 1.3. *We lastly assume that the errors are IID normal with mean 0 and variance σ^2:*

$$\epsilon_1, \epsilon_2, \ldots, \epsilon_n \sim \mathsf{norm}(\mathtt{mean} = 0,\ \mathtt{sd} = \sigma). \tag{1.1.4}$$

Remark 1.4. We assume both the normality of the errors ϵ and the linearity of the mean function μ. Recall from Proposition **??** of Chapter **??** that if $(X, Y) \sim \mathsf{mvnorm}$ then the mean of $Y|x$ is a linear function of x. This is not a coincidence. In more advanced classes we study the case that both X and Y are random, and in particular, when they are jointly normally distributed.

1.1.1 What does it all mean?

See Figure **??**. Shown in the figure is a solid line, the regression line μ, which in this display has slope 0.5 and y-intercept 2.5, that is, $\mu(x) = 2.5 + 0.5x$. The intuition is that for each given value of x, we observe a random value of Y which is normally distributed with a mean equal to the height of the regression line at that x value. Normal densities are superimposed on the plot to drive this point home; in principle, the densities stand outside of the page, perpendicular to the plane of the paper. The figure shows three such values of x, namely, $x = 1$, $x = 2.5$, and $x = 4$. Not only do we assume that the observations at the three locations are independent, but we also assume that their distributions have the same spread. In mathematical terms this means that the normal densities all along the line have

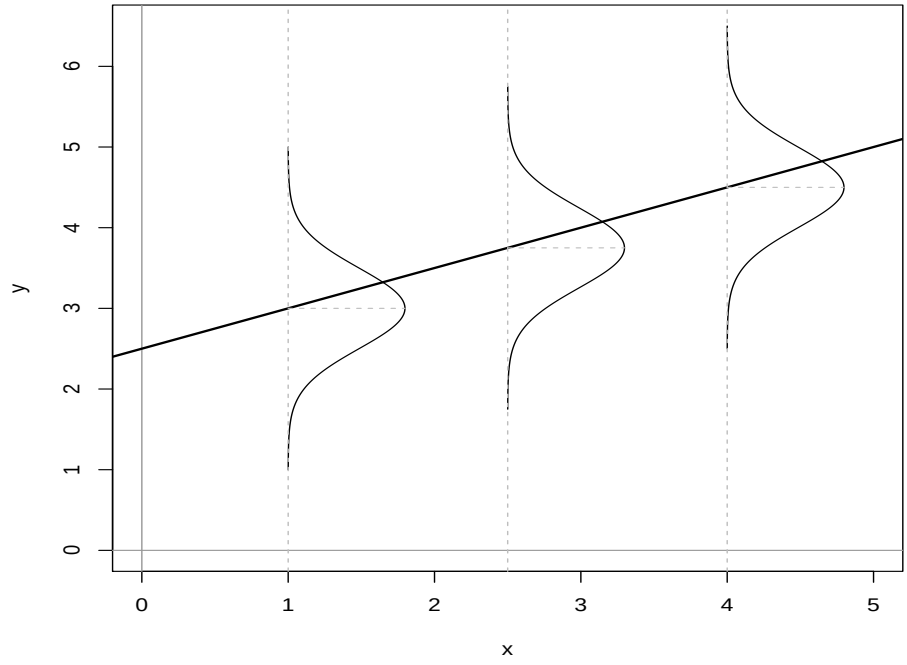

Figure 1.1.1: Philosophical foundations of SLR.

identical standard deviations – there is no "fanning out" or "scrunching in" of the normal densities as x increases [2].

Example 1.5. Speed and stopping distance of cars. We will use the data frame `cars` from the `datasets` package. It has two variables: `speed` and `dist`. We can take a look at some of the values in the data frame:

```
head(cars)
```

```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
```

[2] In practical terms, this constant variance assumption is often violated, in that we often observe scatterplots that fan out from the line as x gets large or small. We say under those circumstances that the data show *heteroscedasticity*. There are methods to address it, but they fall outside the realm of SLR.

```
5     8   16
6     9   10
```

The `speed` represents how fast the car was going (x) in miles per hour and `dist` (Y) measures how far it took the car to stop, in feet. We can make a simple scatterplot of the data with the `qplot` command in the `ggplot2` package.

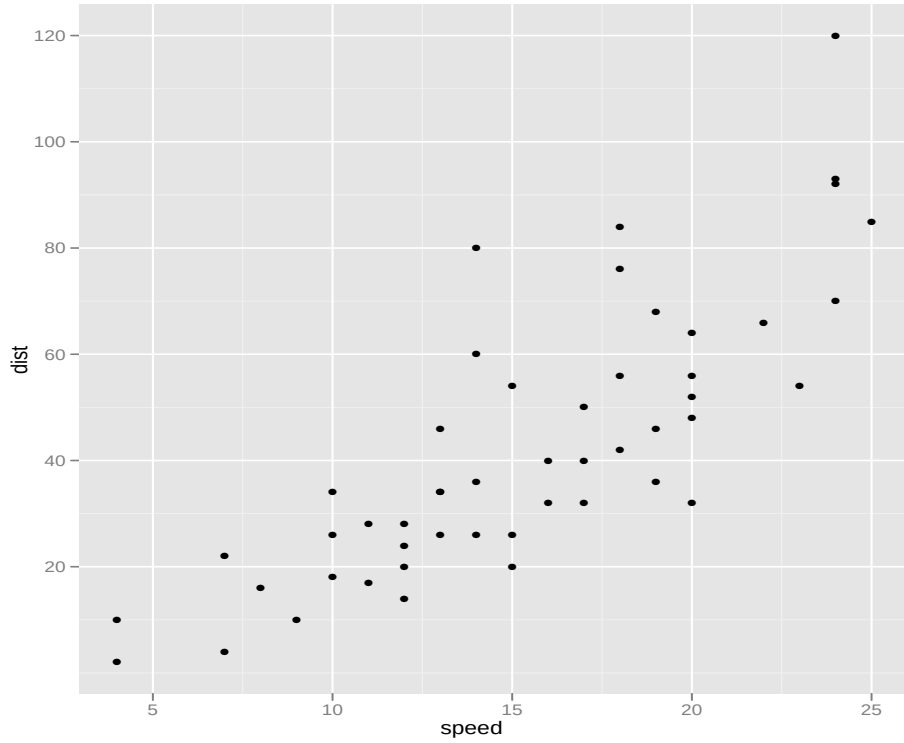

Figure 1.1.2: A scatterplot of `dist` versus `speed` for the `cars` data. There is clearly an upward trend to the plot which is approximately linear.

```
qplot(speed, dist, data = cars)
```

You can see the output in Figure **??**, which was produced by the following code.

```
plot(dist ~ speed, data = cars)
```

There is a pronounced upward trend to the data points, and the pattern looks approximately linear. There does not appear to be substantial fanning out of the points or extreme values.

1.2 Estimation

1.2.1 Point Estimates of the Parameters

Where is $\mu(x)$? In essence, we would like to "fit" a line to the points. But how do we determine a "good" line? Is there a *best* line? We will use maximum likelihood to find it. We know:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \ldots, n, \tag{1.2.1}$$

where the ϵ_i are IID $\texttt{norm}(\texttt{mean} = 0, \texttt{sd} = \sigma)$. Thus $Y_i \sim \texttt{norm}(\texttt{mean} = \beta_0 + \beta_1 x_i, \texttt{sd} = \sigma)$, $i = 1, \ldots, n$. Furthermore, $Y_1, \ldots, Y_n$ are independent – but not identically distributed. The likelihood function is:

$$L(\beta_0, \beta_1, \sigma) = \prod_{i=1}^{n} f_{Y_i}(y_i), \tag{1.2.2}$$

$$= \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} \exp\left\{\frac{-(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right\}, \tag{1.2.3}$$

$$= (2\pi\sigma^2)^{-n/2} \exp\left\{\frac{-\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\right\}. \tag{1.2.4}$$

We take the natural logarithm to get

$$\ln L(\beta_0, \beta_1, \sigma) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}. \tag{1.2.5}$$

We would like to maximize this function of β_0 and β_1. See Appendix **??** which tells us that we should find critical points by means of the partial derivatives. Let us start by differentiating with respect to β_0:

$$\frac{\partial}{\partial\beta_0} \ln L = 0 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} 2(y_i - \beta_0 - \beta_1 x_i)(-1), \tag{1.2.6}$$

and the partial derivative equals zero when $\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0$, that is, when

$$n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i. \tag{1.2.7}$$

Moving on, we next take the partial derivative of $\ln L$ (Equation **??**) with respect to β_1 to get

$$\frac{\partial}{\partial\beta_1} \ln L = 0 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} 2(y_i - \beta_0 - \beta_1 x_i)(-x_i), \tag{1.2.8}$$

$$= \frac{1}{\sigma^2} \sum_{i=1}^{n} \left(x_i y_i - \beta_0 x_i - \beta_1 x_i^2\right), \tag{1.2.9}$$

and this equals zero when the last sum equals zero, that is, when

$$\beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i. \tag{1.2.10}$$

Solving the system of equations **??** and **??**

$$n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{1.2.11}$$

$$\beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \tag{1.2.12}$$

for β_0 and β_1 (in Exercise **??**) gives

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right) \Big/ n}{\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2 \Big/ n} \tag{1.2.13}$$

and

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}. \tag{1.2.14}$$

The conclusion? To estimate the mean line

$$\mu(x) = \beta_0 + \beta_1 x, \tag{1.2.15}$$

we use the "line of best fit"

$$\hat{\mu}(x) = \hat{\beta}_0 + \hat{\beta}_1 x, \tag{1.2.16}$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are given as above. For notation we will usually write $b_0 = \hat{\beta}_0$ and $b_1 = \hat{\beta}_1$ so that $\hat{\mu}(x) = b_0 + b_1 x$.

Remark 1.6. The formula for b_1 in Equation **??** gets the job done but does not really make any sense. There are many equivalent formulas for b_1 that are more intuitive, or at the least are easier to remember. One of the author's favorites is

$$b_1 = r\frac{s_y}{s_x}, \tag{1.2.17}$$

where r, s_y, and s_x are the sample correlation coefficient and the sample standard deviations of the Y and x data, respectively. See Exercise **??**. Also, notice the similarity between Equation **??** and Equation **??**.

How to do it with R Here we go. R will calculate the linear regression line with the `lm` function. We will store the result in an object which we will call `cars.lm`. Here is how it works:

```
cars.lm <- lm(dist ~ speed, data = cars)
```

The first part of the input to the `lm` function, `dist ~ speed`, is a *model formula*, read as "`dist` is described (or modeled) by `speed`". The `data = cars` argument tells R where to look for the variables quoted in the model formula. The output object `cars.lm` contains a multitude of information. Let's first take a look at the coefficients of the fitted regression line, which are extracted by the `coef` function (alternatively, we could just type `cars.lm` to see the same thing):

```
coef(cars.lm)
```

```
 (Intercept)       speed
  -17.579095    3.932409
```

The parameter estimates b_0 and b_1 for the intercept and slope, respectively, are shown above. The regression line is thus given by $\hat{\mu}(\texttt{speed}) = -17.58 + 3.93 \cdot \texttt{speed}$.

It is good practice to visually inspect the data with the regression line added to the plot. To do this we first scatterplot the original data and hen follow with a call to the `abline` function. The inputs to `abline` are the coefficients of `cars.lm`; see Figure **??**.

```
ggplot(cars, aes(x = speed, y = dist)) + geom_point(shape = 19) +
       geom_smooth(method = lm)
```

To calculate points on the regression line we may simply plug the desired x value(s) into $\hat{\mu}$, either by hand, or with the `predict` function. The inputs to `predict` are the fitted linear model object, `cars.lm`, and the desired x value(s) represented by a data frame. See the example below.

Example 1.7. Using the regression line for the `cars` data:

1. What is the meaning of $\mu(60) = \beta_0 + \beta_1(8)$? This represents the average stopping distance (in feet) for a car going 8 mph.

2. Interpret the slope β_1. The true slope β_1 represents the increase in average stopping distance for each mile per hour faster that the car drives. In this case, we estimate the car to take approximately 3.93 additional feet to stop for each additional mph increase in speed.

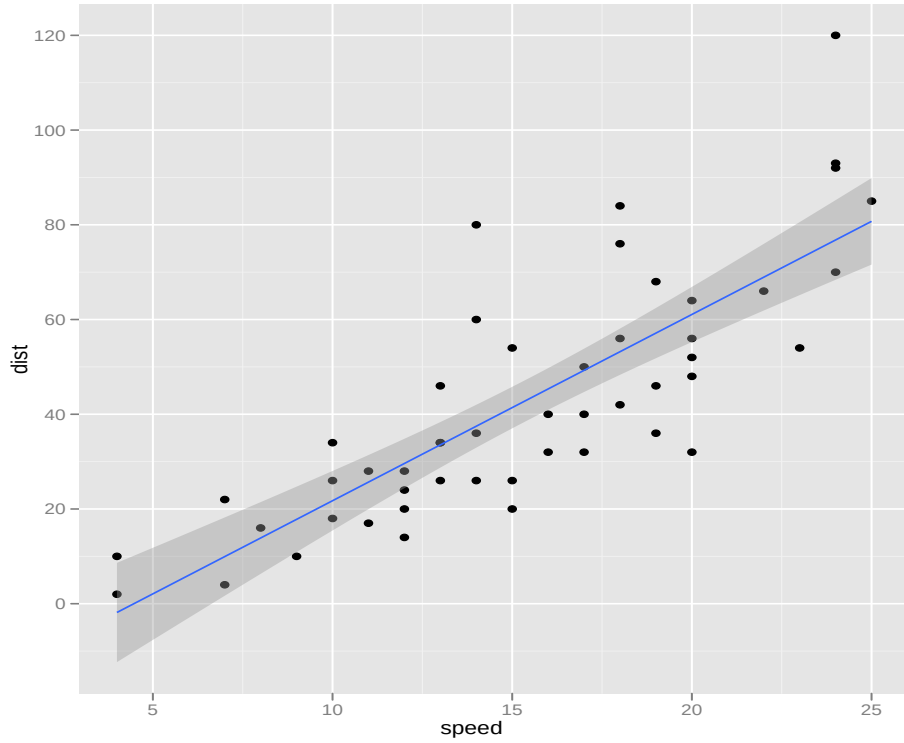

Figure 1.2.1: A scatterplot with an added regression line for the `cars` data.

3. Interpret the intercept β_0. This would represent the mean stopping distance for a car traveling 0 mph (which our regression line estimates to be -17.58). Of course, this interpretation does not make any sense for this example, because a car travelling 0 mph takes 0 ft to stop (it was not moving in the first place)! What went wrong? Looking at the data, we notice that the smallest speed for which we have measured data is 4 mph. Therefore, if we predict what would happen for slower speeds then we would be *extrapolating*, a dangerous practice which often gives nonsensical results.

1.2.2 Point Estimates of the Regression Line

We said at the beginning of the chapter that our goal was to estimate $\mu = \mathbb{E}Y$, and the arguments in Section **??** showed how to obtain an estimate $\hat{\mu}$ of μ when the regression assumptions hold. Now we will reap the benefits of our work in more ways than we previously disclosed. Given a particular value x_0, there are two values we would like to estimate:

1. the mean value of Y at x_0, and

2. a future value of Y at x_0.

The first is a number, $\mu(x_0)$, and the second is a random variable, $Y(x_0)$, but our point estimate is the same for both: $\hat{\mu}(x_0)$.

Example 1.8. We may use the regression line to obtain a point estimate of the mean stopping distance for a car traveling 8 mph: $\hat{\mu}(15) = b_0 + (8)(b_1) \approx -17.58 + (8)(3.93) \approx 13.88$. We would also use 13.88 as a point estimate for the stopping distance of a future car traveling 8 mph.

Note that we actually have observed data for a car traveling 8 mph; its stopping distance was 16 ft as listed in the fifth row of the `cars` data (which we saw in Example **??**).

```
cars[5, ]
```

```
 [1] 3.93
 [1] -17.58
 [1] -17.58
 [1] 3.93
   speed dist
5     8   16
```

There is a special name for estimates $\hat{\mu}(x_0)$ when x_0 matches an observed value x_i from the data set. They are called *fitted values*, they are denoted by $\hat{Y}_1$, $\hat{Y}_2$, ..., $\hat{Y}_n$ (ignoring repetition), and they play an important role in the sections that follow.

In an abuse of notation we will sometimes write $\hat{Y}$ or $\hat{Y}(x_0)$ to denote a point on the regression line even when x_0 does not belong to the original data if the context of the statement obviates any danger of confusion.

We saw in Example **??** that spooky things can happen when we are cavalier about point estimation. While it is usually acceptable to predict/estimate at values of x_0 that fall within the range of the original x data, it is reckless to use $\hat{\mu}$ for point estimates at locations outside that range. Such estimates are usually worthless. *Do not extrapolate* unless there are compelling external reasons, and even then, temper it with a good deal of caution.

How to do it with R The fitted values are automatically computed as a byproduct of the model fitting procedure and are already stored as a component of the `cars.lm` object. We may access them with the `fitted` function (we only show the first five entries):

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

```
         1          2          3          4          5
-1.849460 -1.849460  9.947766  9.947766 13.880175
```

Predictions at x values that are not necessarily part of the original data are done with the `predict` function. The first argument is the original `cars.lm` object and the second argument `newdata` accepts a dataframe (in the same form that was used to fit `cars.lm`) that contains the locations at which we are seeking predictions. Let us predict the average stopping distances of cars traveling 6 mph, 8 mph, and 21 mph:

```
predict(cars.lm, newdata = data.frame(speed = c(6, 8, 21)))
```

```
        1         2         3
 6.015358 13.880175 65.001489
```

Note that there were no observed cars that traveled 6 mph or 21 mph. Also note that our estimate for a car traveling 8 mph matches the value we computed by hand in Example **??**.

1.2.3 Mean Square Error and Standard Error

To find the MLE of σ^2 we consider the partial derivative

$$\frac{\partial}{\partial \sigma^2} \ln L = \frac{n}{2\sigma^2} - \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2, \tag{1.2.18}$$

and after plugging in $\hat{\beta}_0$ and $\hat{\beta}_1$ and setting equal to zero we get

$$\hat{\sigma^2} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 = \frac{1}{n} \sum_{i=1}^{n} [y_i - \hat{\mu}(x_i)]^2. \tag{1.2.19}$$

We write $\hat{Y}i = \hat{\mu}(x_i)$, and we let $E_i = Y_i - \hat{Y}_i$ be the i^{th} *residual*. We see

$$n\hat{\sigma^2} = \sum_{i=1}^{n} E_i^2 = SSE = \text{ the sum of squared errors.} \tag{1.2.20}$$

For a point estimate of σ^2 we use the *mean square error* S^2 defined by

$$S^2 = \frac{SSE}{n-2}, \tag{1.2.21}$$

and we estimate σ with the *standard error* [3] $S = \sqrt{S^2}$.

[3] Be careful not to confuse the mean square error S^2 with the sample variance S^2 in Chapter **??**. Other notation the reader may encounter is the lowercase s^2 or the bulky MSE.

How to do it with R The residuals for the model may be obtained with the `residuals` function; we only show the first few entries in the interest of space:

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

```
        1         2         3         4         5
3.849460 11.849460 -5.947766 12.052234  2.119825
```

In the last section, we calculated the fitted value for $x = 8$ and found it to be approximately $\hat{\mu}(8) \approx 13.88$. Now, it turns out that there was only one recorded observation at $x = 8$, and we have seen this value in the output of `head(cars)` in Example **??**; it was `dist` = 16 ft for a car with `speed` = 8 mph. Therefore, the residual should be $E = Y - \hat{Y}$ which is $E \approx 16 - 13.88$. Now take a look at the last entry of `residuals(cars.lm)`, above. It is not a coincidence.

The estimate S for σ is called the `Residual standard error` and for the `cars` data is shown a few lines up on the `summary(cars.lm)` output (see How to do it with R in Section **??**). We may read it from there to be $S \approx 15.38$, or we can access it directly from the `summary` object.

```
carsumry <- summary(cars.lm)
carsumry$sigma
```

```
[1] 13.88
[1] 13.88
[1] 15.38
[1] 15.37959
```

1.2.4 Interval Estimates of the Parameters

We discussed general interval estimation in Chapter **??**. There we found that we could use what we know about the sampling distribution of certain statistics to construct confidence intervals for the parameter being estimated. We will continue in that vein, and to get started we will determine the sampling distributions of the parameter estimates, b_1 and b_0.

To that end, we can see from Equation **??** (and it is made clear in Chapter **??**) that b_1 is just a linear combination of normally distributed random variables, so b_1 is normally distributed too. Further, it can be shown that

$$b_1 \sim \mathsf{norm}\left(\mathtt{mean} = \beta_1,\ \mathtt{sd} = \sigma_{b_1}\right) \tag{1.2.22}$$

where

$$\sigma_{b_1} = \frac{\sigma}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}} \tag{1.2.23}$$

is called *the standard error of* b_1 which unfortunately depends on the unknown value of σ. We do not lose heart, though, because we can estimate σ with the standard error S from the last section. This gives us an estimate S_{b_1} for σ_{b_1} defined by

$$S_{b_1} = \frac{S}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}}. \tag{1.2.24}$$

Now, it turns out that b_0, b_1, and S are mutually independent (see the footnote in Section **??**). Therefore, the quantity

$$T = \frac{b_1 - \beta_1}{S_{b_1}} \tag{1.2.25}$$

has a $\mathsf{t}(\mathtt{df} = n - 2)$ distribution and a $100(1-\alpha)\%$ confidence interval for β_1 is given by

$$b_1 \pm \mathsf{t}_{\alpha/2}(\mathtt{df} = n - 1)\, S_{b_1}. \tag{1.2.26}$$

It is also sometimes of interest to construct a confidence interval for β_0 in which case we will need the sampling distribution of b_0. It is shown in Chapter **??** that

$$b_0 \sim \mathsf{norm}\left(\mathtt{mean} = \beta_0,\ \mathtt{sd} = \sigma_{b_0}\right), \tag{1.2.27}$$

where σ_{b_0} is given by

$$\sigma_{b_0} = \sigma\sqrt{\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}, \tag{1.2.28}$$

and which we estimate with the S_{b_0} defined by

$$S_{b_0} = S\sqrt{\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}. \tag{1.2.29}$$

Thus the quantity

$$T = \frac{b_0 - \beta_0}{S_{b_0}} \tag{1.2.30}$$

has a $\mathsf{t}(\mathtt{df} = n - 2)$ distribution and a $100(1-\alpha)\%$ confidence interval for β_0 is given by

$$b_0 \pm \mathsf{t}_{\alpha/2}(\mathtt{df} = n - 1)\, S_{b_0}. \tag{1.2.31}$$

How to do it with R Let us take a look at the output from `summary(cars.lm)`:

```
summary(cars.lm)
```

```
Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511,Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

In the `Coefficients` section we find the parameter estimates and their respective standard errors in the second and third columns; the other columns are discussed in Section **??**. If we wanted, say, a 95% confidence interval for β_1 we could use $b_1 = 3.932$ and $S_{b_1} = 0.416$ together with a $\mathrm{t}_{0.025}(\mathtt{df} = 23)$ critical value to calculate $b_1 \pm \mathrm{t}_{0.025}(\mathtt{df} = 23) \cdot S_{b_1}$. Or, we could use the `confint` function.

```
confint(cars.lm)
```

```
[1] 3.932
[1] 0.416
                 2.5 %    97.5 %
(Intercept) -31.167850 -3.990340
speed         3.096964  4.767853
```

With 95% confidence, the random interval $[3.097, 4.768]$ covers the parameter β_1.

1.2.5 Interval Estimates of the Regression Line

We have seen how to estimate the coefficients of regression line with both point estimates and confidence intervals. We even saw how to estimate a value $\hat{\mu}(x)$ on the regression line for a given value of x, such as $x = 15$.

But how good is our estimate $\hat{\mu}(15)$? How much confidence do we have in *this* estimate? Furthermore, suppose we were going to observe another value of Y at $x = 15$. What could we say?

Intuitively, it should be easier to get bounds on the mean (average) value of Y at x_0 – called a *confidence interval for the mean value of Y at x_0* – than it is to get bounds on a future observation of Y (called a *prediction interval for Y at x_0*). As we shall see, the intuition serves us well and confidence intervals are shorter for the mean value, longer for the individual value.

Our point estimate of $\mu(x_0)$ is of course $\hat{Y} = \hat{Y}(x_0)$, so for a confidence interval we will need to know $\hat{Y}$'s sampling distribution. It turns out (see Section) that $\hat{Y} = \hat{\mu}(x_0)$ is distributed

$$\hat{Y} \sim \texttt{norm}\left(\texttt{mean} = \mu(x_0),\ \texttt{sd} = \sigma\sqrt{\frac{1}{n} + \frac{(x_0 - \overline{x})^2}{\sum_{i=1}^n (x_i - \overline{x})^2}}\right). \tag{1.2.32}$$

Since σ is unknown we estimate it with S (we should expect the appearance of a $\texttt{t}(\texttt{df} = n-2)$ distribution in the near future).

A $100(1-\alpha)\%$ *confidence interval (CI) for* $\mu(x_0)$ is given by

$$\hat{Y} \pm \texttt{t}_{\alpha/2}(\texttt{df} = n-2)\, S\, \sqrt{\frac{1}{n} + \frac{(x_0 - \overline{x^2})}{\sum_{i=1}^n (x_i - \overline{x})^2}}. \tag{1.2.33}$$

Prediction intervals are a little bit different. In order to find confidence bounds for a new observation of Y (we will denote it Y_{new}) we use the fact that

$$Y_{\text{new}} \sim \texttt{norm}\left(\texttt{mean} = \mu(x_0),\ \texttt{sd} = \sigma\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \overline{x})^2}{\sum_{i=1}^n (x_i - \overline{x})^2}}\right). \tag{1.2.34}$$

Of course, σ is unknown so we estimate it with S and a $100(1-\alpha)\%$ prediction interval (PI) for a future value of Y at x_0 is given by

$$\hat{Y}(x_0) \pm \texttt{t}_{\alpha/2}(\texttt{df} = n-1)\, S\, \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \overline{x})^2}{\sum_{i=1}^n (x_i - \overline{x})^2}}. \tag{1.2.35}$$

We notice that the prediction interval in Equation **??** is wider than the confidence interval in Equation **??**, as we expected at the beginning of the section.

How to do it with R Confidence and prediction intervals are calculated in R with the `predict` function, which we encountered in Section **??**. There we neglected to take advantage of its additional `interval` argument. The general syntax follows.

Example 1.9. We will find confidence and prediction intervals for the stopping distance of a car travelling 5, 6, and 21 mph (note from the graph that there are no collected data for these speeds). We have computed `cars.lm` earlier, and we will use this for input to the `predict` function. Also, we need to tell R the values of x_0 at which we want the predictions made, and store the x_0 values in a data frame whose variable is labeled with the correct name. *This is important.*

```
new <- data.frame(speed = c(5, 6, 21))
```

```
 [1] 3.097
 [1] 4.768
```

Next we instruct R to calculate the intervals. Confidence intervals are given by

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

```
        fit       lwr      upr
 1  2.082949 -7.644150 11.81005
 2  6.015358 -2.973341 15.00406
 3 65.001489 58.597384 71.40559
```

Prediction intervals are given by

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

```
        fit       lwr      upr
 1  2.082949 -30.33359 34.49948
 2  6.015358 -26.18731 38.21803
 3 65.001489  33.42257 96.58040
```

The type of interval is dictated by the `interval` argument (which is `none` by default), and the default confidence level is 95% (which can be changed with the `level` argument).

Example 1.10. Using the `cars` data,

1. Report a point estimate of and a 95% confidence interval for the mean stopping distance for a car travelling 5 mph. The fitted value for $x = 5$ is 2.08, so a point estimate would be 2.08 ft. The 95% CI is given by $[-7.64, 11.81]$, so with 95% confidence the mean stopping distance lies somewhere between -7.64 ft and 11.81 ft.

2. Report a point prediction for and a 95% prediction interval for the stopping distance of a hypothetical car travelling 21 mph. The fitted value for $x = 21$ is 65, so a point prediction for the stopping distance is 65 ft. The 95% PI is $[33.42, 96.58]$, so with 95% confidence we may assert that the hypothetical stopping distance for a car travelling 21 mph would lie somewhere between 33.42 ft and 96.58 ft.

1.2.6 Graphing the Confidence and Prediction Bands

We earlier guessed that a bound on the value of a single new observation would be inherently less certain than a bound for an average (mean) value; therefore, we expect the CIs for the mean to be tighter than the PIs for a new observation. A close look at the standard deviations in Equations **??** and **??** confirms our guess, but we would like to see a picture to drive the point home.

We may plot the confidence and prediction intervals with one fell swoop using the `ci.plot` function from the `HH` package [**?**]. The graph is displayed in Figure **??**.

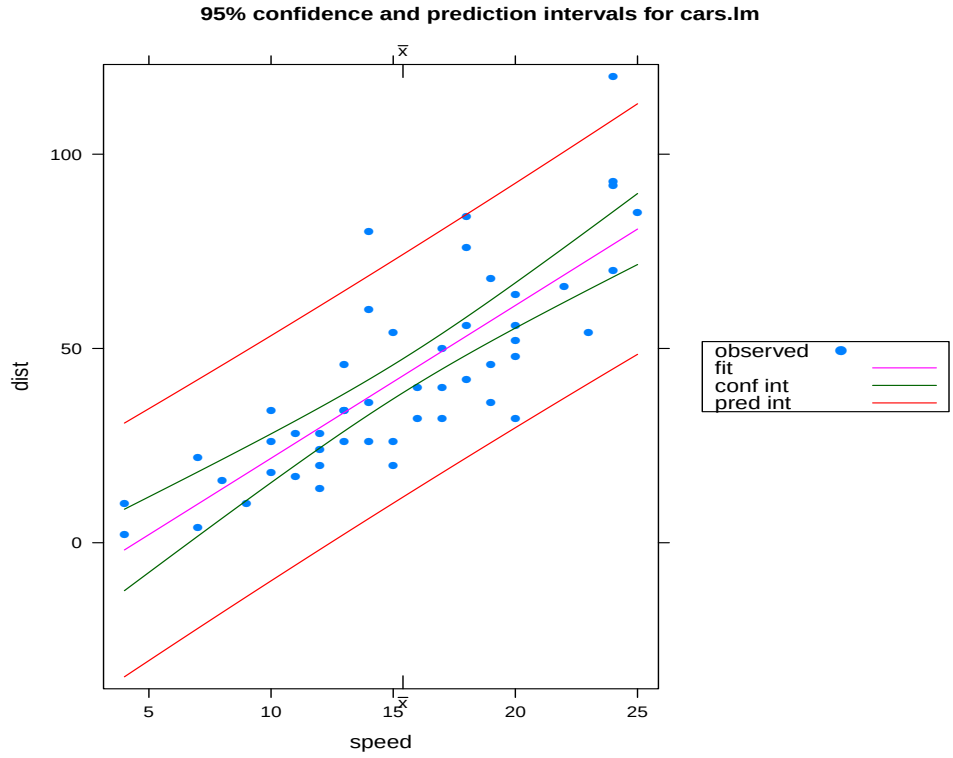

Figure 1.2.2: A scatterplot with confidence/prediction bands for the `cars` data.

```
library(HH)
ci.plot(cars.lm)
```

Notice that the bands curve outward from the regression line as the x values move away from the center. This is expected once we notice the $(x_0 - \overline{x})^2$ term in the standard deviation formulas in Equations **??** and **??**.

1.3 Model Utility and Inference

1.3.1 Hypothesis Tests for the Parameters

Much of the attention of SLR is directed toward β_1 because when $\beta_1 \neq 0$ the mean value of Y increases (or decreases) as x increases. It is really boring when $\beta_1 = 0$, because in that case the mean value of Y remains the same, regardless of the value of x (when the regression assumptions hold, of course). It is thus very important to decide whether or not $\beta_1 = 0$. We address the question with a statistical test of the null hypothesis $H_0 : \beta_1 = 0$ versus the alternative hypothesis $H_1 : \beta_1 \neq 0$, and to do that we need to know the sampling distribution of b_1 when the null hypothesis is true.

To this end we already know from Section **??** that the quantity

$$T = \frac{b_1 - \beta_1}{S_{b_1}} \tag{1.3.1}$$

has a t(df $= n - 2$) distribution; therefore, when $\beta_1 = 0$ the quantity b_1/S_{b_1} has a t(df $= n - 2$) distribution and we can compute a p-value by comparing the observed value of b_1/S_{b_1} with values under a t(df $= n - 2$) curve.

Similarly, we may test the hypothesis $H_0 : \beta_0 = 0$ versus the alternative $H_1 : \beta_0 \neq 0$ with the statistic $T = b_0/S_{b_0}$, where S_{b_0} is given in Section **??**. The test is conducted the same way as for β_1.

How to do it with R Let us take another look at the output from `summary(cars.lm)`:

```
summary(cars.lm)

null device
          1

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511,Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

In the `Coefficients` section we find the t statistics and the p-values associated with the tests that the respective parameters are zero in the fourth and fifth columns. Since the p-values are (much) less than 0.05, we conclude that there is strong evidence that the parameters $\beta_1 \neq 0$ and $\beta_0 \neq 0$, and as such, we say that there is a statistically significant linear relationship between `dist` and `speed`.

1.3.2 Simple Coefficient of Determination

It would be nice to have a single number that indicates how well our linear regression model is doing, and the *simple coefficient of determination* is designed for that purpose. In what follows, we observe the values $Y_1, Y_2, \ldots, Y_n$, and the goal is to estimate $\mu(x_0)$, the mean value of Y at the location x_0.

If we disregard the dependence of Y and x and base our estimate only on the Y values then a reasonable choice for an estimator is just the MLE of μ, which is $\overline{Y}$. Then the errors incurred by the estimate are just $Y_i - \overline{Y}$ and the variation about the estimate as measured by the sample variance is proportional to

$$SSTO = \sum_{i=1}^{n} (Y_i - \overline{Y})^2. \tag{1.3.2}$$

The acronym $SSTO$ stands for *total sum of squares*. And we have additional information, namely, we have values x_i associated with each value of Y_i. We have seen that this information leads us to the estimate $\hat{Y}_i$ and the errors incurred are just the residuals, $E_i = Y_i - \hat{Y}_i$. The variation associated with these errors can be measured with

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2. \tag{1.3.3}$$

We have seen the SSE before, which stands for the *sum of squared errors* or *error sum of squares*. Of course, we would expect the error to be less in the latter case, since we have used more information. The improvement in our estimation as a result of the linear regression model can be measured with the difference

$$(Y_i - \overline{Y}) - (Y_i - \hat{Y}_i) = \hat{Y}_i - \overline{Y},$$

and we measure the variation in these errors with

$$SSR = \sum_{i=1}^{n} (\hat{Y}_i - \overline{Y})^2, \tag{1.3.4}$$

also known as the *regression sum of squares*. It is not obvious, but some algebra proved a famous result known as the **ANOVA Equality**:

$$\sum_{i=1}^{n}(Y_i - \overline{Y})^2 = \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2 + \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1.3.5}$$

or in other words,

$$SSTO = SSR + SSE. \tag{1.3.6}$$

This equality has a nice interpretation. Consider $SSTO$ to be the *total variation* of the errors. Think of a decomposition of the total variation into pieces: one piece measuring the reduction of error from using the linear regression model, or *explained variation* (SSR), while the other represents what is left over, that is, the errors that the linear regression model doesn't explain, or *unexplained variation* (SSE). In this way we see that the ANOVA equality merely partitions the variation into

$$\text{total variation} = \text{explained variation} + \text{unexplained variation}.$$

For a single number to summarize how well our model is doing we use the *simple coefficient of determination* r^2, defined by

$$r^2 = 1 - \frac{SSE}{SSTO}. \tag{1.3.7}$$

We interpret r^2 as the proportion of total variation that is explained by the simple linear regression model. When r^2 is large, the model is doing a good job; when r^2 is small, the model is not doing a good job.

Related to the simple coefficient of determination is the sample correlation coefficient, r. As you can guess, the way we get r is by the formula $|r| = \sqrt{r^2}$. The sign of r is equal the sign of the slope estimate b_1. That is, if the regression line $\hat{\mu}(x)$ has positive slope, then $r = \sqrt{r^2}$. Likewise, if the slope of $\hat{\mu}(x)$ is negative, then $r = -\sqrt{r^2}$.

How to do it with R The primary method to display partitioned sums of squared errors is with an *ANOVA table*. The command in R to produce such a table is `anova`. The input to `anova` is the result of an `lm` call which for the `cars` data is `cars.lm`.

```
anova(cars.lm)
```

```
Analysis of Variance Table

Response: dist
          Df Sum Sq Mean Sq F value   Pr(>F)
speed      1  21186 21185.5  89.567 1.49e-12 ***
Residuals 48  11354   236.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output gives

$$r^2 = 1 - \frac{SSE}{SSR + SSE} = 1 - \frac{11353.5}{21185.5 + 11353.5} \approx 0.65.$$

The interpretation should be: "The linear regression line accounts for approximately 65% of the variation of `dist` as explained by `speed`".

The value of r^2 is stored in the `r.squared` component of `summary(cars.lm)`, which we called `carsumry`.

```
carsumry$r.squared
```

```
[1] 0.6510794
```

We already knew this. We saw it in the next to the last line of the `summary(cars.lm)` output where it was called `Multiple R-squared`. Listed right beside it is the `Adjusted R-squared` which we will discuss in Chapter **??**. For the `cars` data, we find r to be

```
sqrt(carsumry$r.squared)
```

```
[1] 0.8068949
```

We choose the principal square root because the slope of the regression line is positive.

1.3.3 Overall *F* statistic

There is another way to test the significance of the linear regression model. In SLR, the new way also tests the hypothesis $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$, but it is done with a new test statistic called the *overall F statistic*. It is defined by

$$F = \frac{SSR}{SSE/(n-2)}. \tag{1.3.8}$$

Under the regression assumptions and when H_0 is true, the F statistic has an `f(df1 = 1, df2 =` $n-2$) distribution. We reject H_0 when F is large – that is, when the explained variation is large relative to the unexplained variation.

All this being said, we have not yet gained much from the overall F statistic because we already knew from Section **??** how to test $H_0 : \beta_1 = 0 \ldots$ we use the Student's t statistic. What is worse is that (in the simple linear regression model) it can be proved that the F in Equation **??** is exactly the Student's t statistic for β_1 squared,

$$F = \left(\frac{b_1}{S_{b_1}}\right)^2. \tag{1.3.9}$$

So why bother to define the F statistic? Why not just square the t statistic and be done with it? The answer is that the F statistic has a more complicated interpretation and plays a more important role in the multiple linear regression model which we will study in Chapter **??**. See Section **??** for details.

How to do it with R The overall F statistic and p-value are displayed in the bottom line of the `summary(cars.lm)` output. It is also shown in the final columns of `anova(cars.lm)`:

```
anova(cars.lm)

 Analysis of Variance Table

 Response: dist
           Df Sum Sq Mean Sq F value   Pr(>F)
 speed      1  21186 21185.5  89.567 1.49e-12 ***
 Residuals 48  11354   236.5
 ---
 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we see that the F statistic is 89.57 with a p-value very close to zero. The conclusion: there is very strong evidence that $H_0 : \beta_1 = 0$ is false, that is, there is strong evidence that $\beta_1 \neq 0$. Moreover, we conclude that the regression relationship between `dist` and `speed` is significant.

Note that the value of the F statistic is the same as the Student's t statistic for `speed` squared.

1.4 Residual Analysis

We know from our model that $Y = \mu(x) + \epsilon$, or in other words, $\epsilon = Y - \mu(x)$. Further, we know that $\epsilon \sim$ norm(mean $= 0$, sd $= \sigma$). We may estimate ϵ_i with the *residual* $E_i = Y_i - \hat{Y}_i$, where $\hat{Y}_i = \hat{\mu}(x_i)$. If the regression assumptions hold, then the residuals should be normally distributed. We check this in Section **??**. Further, the residuals should have mean zero with constant variance σ^2, and we check this in Section **??**. Last, the residuals should be independent, and we check this in Section **??**.

In every case, we will begin by looking at residual plots – that is, scatterplots of the residuals E_i versus index or predicted values $\hat{Y}_i$ – and follow up with hypothesis tests.

1.4.1 Normality Assumption

We can assess the normality of the residuals with graphical methods and hypothesis tests. To check graphically whether the residuals are normally distributed we may look at histograms or q-q plots. We first examine a histogram in Figure **??**. There we see that the distribution of the residuals appears to be mound shaped, for the most part. We can plot the order statistics of the sample versus quantiles from a norm(mean $= 0$, sd $= 1$) distribution with the command `plot(cars.lm, which = 2)`, and the results are in Figure **??**. If the

assumption of normality were true, then we would expect points randomly scattered about the dotted straight line displayed in the figure. In this case, we see a slight departure from normality in that the dots show systematic clustering on one side or the other of the line. The points on the upper end of the plot also appear begin to stray from the line. We would say there is some evidence that the residuals are not perfectly normal.

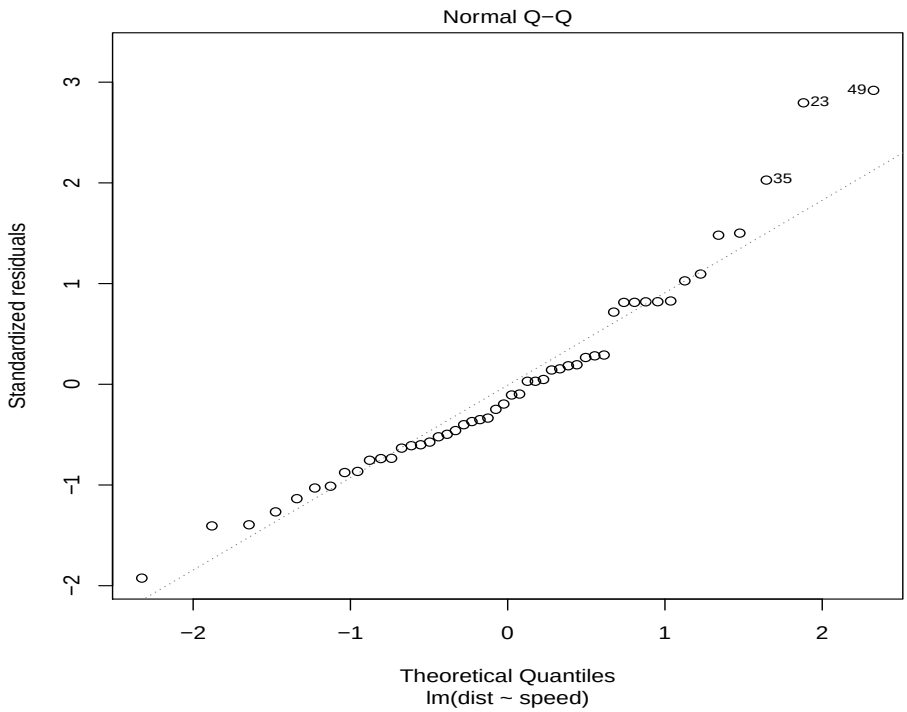

Figure 1.4.1: Used for checking the normality assumption. Look out for any curvature or substantial departures from the straight line; hopefully the dots hug the line closely.

```
plot(cars.lm, which = 2)
```

Testing the Normality Assumption Even though we may be concerned about the plots, we can use tests to determine if the evidence present is statistically significant, or if it could have happened merely by chance. There are many statistical tests of normality. We will use the Shapiro-Wilk test, since it is known to be a good test and to be quite powerful. However, there are many other fine tests of normality including the Anderson-Darling test and the Lillefors test, just to mention two of them.

The Shapiro-Wilk test is based on the statistic

$$W = \frac{\left(\sum_{i=1}^{n} a_i E_{(i)}\right)^2}{\sum_{j=1}^{n} E_j^2}, \tag{1.4.1}$$

where the $E_{(i)}$ are the ordered residuals and the a_i are constants derived from the order statistics of a sample of size n from a normal distribution. See Section **??**. We perform the Shapiro-Wilk test below, using the `shapiro.test` function from the `stats` package. The hypotheses are

$$H_0: \text{ the residuals are normally distributed}$$

versus

$$H_1: \text{ the residuals are not normally distributed.}$$

The results from R are

```
shapiro.test(residuals(cars.lm))
```

```
 null device
           1


  Shapiro-Wilk normality test

data:  residuals(cars.lm)
W = 0.9451, p-value = 0.02153
```

For these data we would reject the assumption of normality of the residuals at the $\alpha = 0.05$ significance level, but do not lose heart, because the regression model is reasonably robust to departures from the normality assumption. As long as the residual distribution is not highly skewed, then the regression estimators will perform reasonably well. Moreover, departures from constant variance and independence will sometimes affect the quantile plots and histograms, therefore it is wise to delay final decisions regarding normality until all diagnostic measures have been investigated.

1.4.2 Constant Variance Assumption

We will again go to residual plots to try and determine if the spread of the residuals is changing over time (or index). However, it is unfortunately not that easy because the residuals do not have constant variance! In fact, it can be shown that the variance of the residual E_i is

$$\text{Var}(E_i) = \sigma^2(1 - h_{ii}), \quad i = 1, 2, \ldots, n, \tag{1.4.2}$$

where h_{ii} is a quantity called the *leverage* which is defined below. Consequently, in order to check the constant variance assumption we must standardize the residuals before plotting.

We estimate the standard error of E_i with $s_{E_i} = s\sqrt{(1-h_{ii})}$ and define the *standardized residuals* R_i, $i = 1, 2, \ldots, n$, by

$$R_i = \frac{E_i}{s\sqrt{1-h_{ii}}}, \quad i = 1, 2, \ldots, n. \tag{1.4.3}$$

For the constant variance assumption we do not need the sign of the residual so we will plot $\sqrt{|R_i|}$ versus the fitted values. As we look at a scatterplot of $\sqrt{|R_i|}$ versus $\hat{Y}_i$ we would expect under the regression assumptions to see a constant band of observations, indicating no change in the magnitude of the observed distance from the line. We want to watch out for a fanning-out of the residuals, or a less common funneling-in of the residuals. Both patterns indicate a change in the residual variance and a consequent departure from the regression assumptions, the first an increase, the second a decrease.

In this case, we plot the standardized residuals versus the fitted values. The graph may be seen in Figure **??**. For these data there does appear to be somewhat of a slight fanning-out of the residuals.

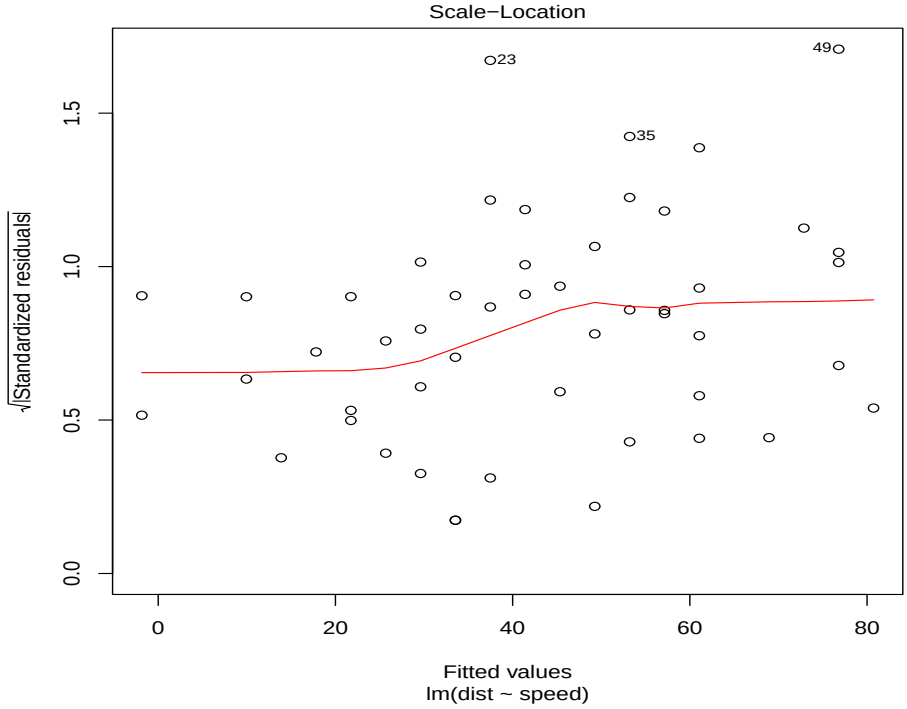

Figure 1.4.2: Used for checking the constant variance assumption. Watch out for any fanning out (or in) of the dots; hopefully they fall in a constant band.

```
plot(cars.lm, which = 3)
```

Testing the Constant Variance Assumption We will use the Breusch-Pagan test to decide whether the variance of the residuals is nonconstant. The null hypothesis is that the variance is the same for all observations, and the alternative hypothesis is that the variance is not the same for all observations. The test statistic is found by fitting a linear model to the centered squared residuals,

$$W_i = E_i^2 - \frac{SSE}{n}, \quad i = 1, 2, \ldots, n. \tag{1.4.4}$$

By default the same explanatory variables are used in the new model which produces fitted values $\hat{W}_i$, $i = 1, 2, \ldots, n$. The Breusch-Pagan test statistic in R is then calculated with

$$BP = n \sum_{i=1}^{n} \hat{W}_i^2 \div \sum_{i=1}^{n} W_i^2. \tag{1.4.5}$$

We reject the null hypothesis if BP is too large, which happens when the explained variation i the new model is large relative to the unexplained variation in the original model. We do it in R with the `bptest` function from the `lmtest` package [**?**].

```
library(lmtest)
bptest(cars.lm)
```

```
 null device
           1
```

```
  studentized Breusch-Pagan test
```

```
 data:  cars.lm
 BP = 3.2149, df = 1, p-value = 0.07297
```

For these data we would not reject the null hypothesis at the $\alpha = 0.05$ level. There is relatively weak evidence against the assumption of constant variance.

1.4.3 Independence Assumption

One of the strongest of the regression assumptions is the one regarding independence. Departures from the independence assumption are often exhibited by correlation (or autocorrelation, literally, self-correlation) present in the residuals. There can be positive or negative correlation.

Positive correlation is displayed by positive residuals followed by positive residuals, and negative residuals followed by negative residuals. Looking from left to right, this

is exhibited by a cyclical feature in the residual plots, with long sequences of positive residuals being followed by long sequences of negative ones.

On the other hand, negative correlation implies positive residuals followed by negative residuals, which are then followed by positive residuals, *etc*. Consequently, negatively correlated residuals are often associated with an alternating pattern in the residual plots. We examine the residual plot in Figure **??**. There is no obvious cyclical wave pattern or structure to the residual plot.

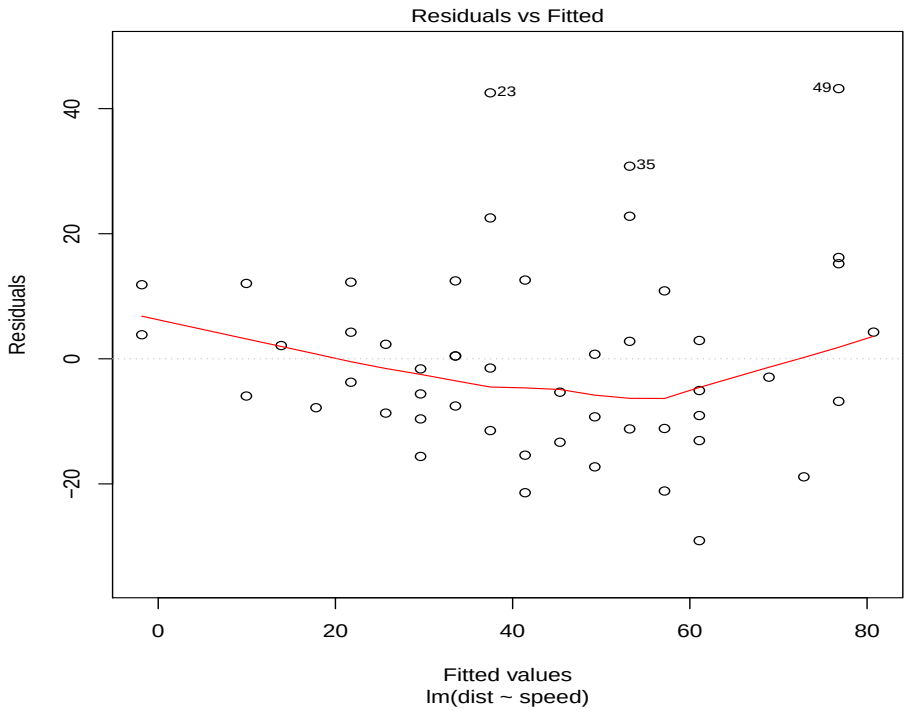

Figure 1.4.3: Used for checking the independence assumption. Watch out for any patterns or structure; hopefully the points are randomly scattered on the plot.

```
plot(cars.lm, which = 1)
```

Testing the Independence Assumption We may statistically test whether there is evidence of autocorrelation in the residuals with the Durbin-Watson test. The test is based on the statistic

$$D = \frac{\sum_{i=2}^{n}(E_i - E_{i-1})^2}{\sum_{j=1}^{n} E_j^2}. \tag{1.4.6}$$

Exact critical values are difficult to obtain, but R will calculate the *p-value* to great accuracy. It is performed with the `dwtest` function from the `lmtest` package. We will conduct a two sided test that the correlation is not zero, which is not the default (the default is to test that the autocorrelation is positive).

```
library(lmtest)
dwtest(cars.lm, alternative = "two.sided")
```

```
 null device
           1
```

```
  Durbin-Watson test

 data:  cars.lm
 DW = 1.6762, p-value = 0.1904
 alternative hypothesis: true autocorrelation is not 0
```

In this case we do not reject the null hypothesis at the $\alpha = 0.05$ significance level; there is very little evidence of nonzero autocorrelation in the residuals.

1.4.4 Remedial Measures

We often find problems with our model that suggest that at least one of the three regression assumptions is violated. What do we do then? There are many measures at the statistician's disposal, and we mention specific steps one can take to improve the model under certain types of violation.

Mean response is not linear We can directly modify the model to better approximate the mean response. In particular, perhaps a polynomial regression function of the form

$$\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2$$

would be appropriate. Alternatively, we could have a function of the form

$$\mu(x) = \beta_0 e^{\beta_1 x}.$$

Models like these are studied in nonlinear regression courses.

Error variance is not constant Sometimes a transformation of the dependent variable will take care of the problem. There is a large class of them called *Box-Cox transformations*. They take the form

$$Y^* = Y^\lambda, \tag{1.4.7}$$

where λ is a constant. (The method proposed by Box and Cox will determine a suitable value of λ automatically by maximum likelihood). The class contains the transformations

$$\begin{aligned} \lambda = 2, &\quad Y^* = Y^2 \\ \lambda = 0.5, &\quad Y^* = \sqrt{Y} \\ \lambda = 0, &\quad Y^* = \ln Y \\ \lambda = -1, &\quad Y^* = 1/Y \end{aligned}$$

Alternatively, we can use the method of *weighted least squares*. This is studied in more detail in later classes.

Error distribution is not normal The same transformations for stabilizing the variance are equally appropriate for smoothing the residuals to a more Gaussian form. In fact, often we will kill two birds with one stone.

Errors are not independent There is a large class of autoregressive models to be used in this situation which occupy the latter part of Chapter **??**.

1.5 Other Diagnostic Tools

There are two types of observations with which we must be especially careful:

Influential observations are those that have a substantial effect on our estimates, predictions, or inferences. A small change in an influential observation is followed by a large change in the parameter estimates or inferences.

Outlying observations are those that fall fall far from the rest of the data. They may be indicating a lack of fit for our regression model, or they may just be a mistake or typographical error that should be corrected. Regardless, special attention should be given to these observations. An outlying observation may or may not be influential.

We will discuss outliers first because the notation builds sequentially in that order.

1.5.1 Outliers

There are three ways that an observation (x_i, y_i) might be identified as an outlier: it can have an x_i value which falls far from the other x values, it can have a y_i value which falls far from the other y values, or it can have both its x_i and y_i values falling far from the other x and y values.

1.5.2 Leverage

Leverage statistics are designed to identify observations which have x values that are far away from the rest of the data. In the simple linear regression model the leverage of x_i is denoted by h_{ii} and defined by

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{k=1}^{n}(x_k - \overline{x})^2}, \quad i = 1, 2, \ldots, n. \tag{1.5.1}$$

The formula has a nice interpretation in the SLR model: if the distance from x_i to $\overline{x}$ is large relative to the other x's then h_{ii} will be close to 1.

Leverages have nice mathematical properties; for example, they satisfy

$$0 \leq h_{ii} \leq 1, \tag{1.5.2}$$

and their sum is

$$\begin{aligned} \sum_{i=1}^{n} h_{ii} &= \sum_{i=1}^{n}\left[\frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{k=1}^{n}(x_k - \overline{x})^2}\right], & (1.5.3)\\ &= \frac{n}{n} + \frac{\sum_i (x_i - \overline{x})^2}{\sum_k (x_k - \overline{x})^2}, & (1.5.4)\\ &= 2. & (1.5.5) \end{aligned}$$

A rule of thumb is to consider leverage values to be large if they are more than double their average size (which is $2/n$ according to Equation **??**). So leverages larger than $4/n$ are suspect. Another rule of thumb is to say that values bigger than 0.5 indicate high leverage, while values between 0.3 and 0.5 indicate moderate leverage.

1.5.3 Standardized and Studentized Deleted Residuals

We have already encountered the *standardized residuals* r_i in Section **??**; they are merely residuals that have been divided by their respective standard deviations:

$$R_i = \frac{E_i}{S\sqrt{1 - h_{ii}}}, \quad i = 1, 2, \ldots, n. \tag{1.5.6}$$

Values of $|R_i| > 2$ are extreme and suggest that the observation has an outlying y-value.

Now delete the i^{th} case and fit the regression function to the remaining $n - 1$ cases, producing a fitted value $\hat{Y}_{(i)}$ with *deleted residual* $D_i = Y_i - \hat{Y}_{(i)}$. It is shown in later classes that

$$\text{Var}(D_i) = \frac{S_{(i)}^2}{1 - h_{ii}}, \quad i = 1, 2, \ldots, n, \tag{1.5.7}$$

so that the *studentized deleted residuals* t_i defined by

$$t_i = \frac{D_i}{S_{(i)}/(1-h_{ii})}, \quad i = 1, 2, \ldots, n, \tag{1.5.8}$$

have a $\mathsf{t}(\mathtt{df} = n-3)$ distribution and we compare observed values of t_i to this distribution to decide whether or not an observation is extreme.

The folklore in regression classes is that a test based on the statistic in Equation **??** can be too liberal. A rule of thumb is if we suspect an observation to be an outlier *before* seeing the data then we say it is significantly outlying if its two-tailed p-value is less than α, but if we suspect an observation to be an outlier *after* seeing the data then we should only say it is significantly outlying if its two-tailed p-value is less than α/n. The latter rule of thumb is called the *Bonferroni approach* and can be overly conservative for large data sets. The responsible statistician should look at the data and use his/her best judgement, in every case.

How to do it with R We can calculate the standardized residuals with the `rstandard` function. The input is the `lm` object, which is `cars.lm`.

```
sres <- rstandard(cars.lm)
sres[1:5]
```

```
         1          2          3          4          5
 0.2660415  0.8189327 -0.4013462  0.8132663  0.1421624
```

We can find out which observations have studentized residuals larger than two with the command

```
sres[which(abs(sres) > 2)]
```

```
       23        35        49
 2.795166 2.027818 2.919060
```

In this case, we see that observations 23, 35, and 49 are potential outliers with respect to their y-value. We can compute the studentized deleted residuals with `rstudent`:

```
sdelres <- rstudent(cars.lm)
sdelres[1:5]
```

```
         1          2          3          4          5
 0.2634500  0.8160784 -0.3978115  0.8103526  0.1407033
```

We should compare these values with critical values from a t(df = $n - 3$) distribution, which in this case is t(df = $50 - 3 = 47$). We can calculate a 0.005 quantile and check with

```
t0.005 <- qt(0.005, df = 47, lower.tail = FALSE)
sdelres[which(abs(sdelres) > t0.005)]
```

```
       23       49
 3.022829 3.184993
```

This means that observations 23 and 49 have a large studentized deleted residual. The leverages can be found with the `hatvalues` function:

```
leverage <- hatvalues(cars.lm)
leverage[which(leverage > 4/50)]
```

```
          1           2          50
 0.11486131 0.11486131 0.08727007
```

Here we see that observations 1, 2, and 50 have leverages bigger than double their mean value. These observations would be considered outlying with respect to their x value (although they may or may not be influential).

1.5.4 Influential Observations

DFBETAS **and** *DFFITS* Any time we do a statistical analysis, we are confronted with the variability of data. It is always a concern when an observation plays too large a role in our regression model, and we would not like or procedures to be overly influenced by the value of a single observation. Hence, it becomes desirable to check to see how much our estimates and predictions would change if one of the observations were not included in the analysis. If an observation changes the estimates/predictions a large amount, then the observation is influential and should be subjected to a higher level of scrutiny.

We measure the change in the parameter estimates as a result of deleting an observation with $DFBETAS$. The $DFBETAS$ for the intercept b_0 are given by

$$(DFBETAS)_{0(i)} = \frac{b_0 - b_{0(i)}}{S_{(i)}\sqrt{\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}}, \quad i = 1, 2, \ldots, n. \tag{1.5.9}$$

and the $DFBETAS$ for the slope b_1 are given by

$$(DFBETAS)_{1(i)} = \frac{b_1 - b_{1(i)}}{S_{(i)}\left[\sum_{i=1}^{n}(x_i - \overline{x})^2\right]^{-1/2}}, \quad i = 1, 2, \ldots, n. \tag{1.5.10}$$

See Section **??** for a better way to write these. The signs of the $DFBETAS$ indicate whether the coefficients would increase or decrease as a result of including the observation. If the $DFBETAS$ are large, then the observation has a large impact on those regression coefficients. We label observations as suspicious if their $DFBETAS$ have magnitude greater 1 for small data or $2/\sqrt{n}$ for large data sets. We can calculate the $DFBETAS$ with the `dfbetas` function (some output has been omitted):

```
dfb <- dfbetas(cars.lm)
head(dfb)
```

```
  (Intercept)       speed
1  0.09440188 -0.08624563
2  0.29242487 -0.26715961
3 -0.10749794  0.09369281
4  0.21897614 -0.19085472
5  0.03407516 -0.02901384
6 -0.11100703  0.09174024
```

We see that the inclusion of the first observation slightly increases the `Intercept` and slightly decreases the coefficient on `speed`.

We can measure the influence that an observation has on its fitted value with $DFFITS$. These are calculated by deleting an observation, refitting the model, recalculating the fit, then standardizing. The formula is

$$(DFFITS)_i = \frac{\hat{Y}_i - \hat{Y}_{(i)}}{S_{(i)}\sqrt{h_{ii}}}, \quad i = 1, 2, \ldots, n. \tag{1.5.11}$$

The value represents the number of standard deviations of $\hat{Y}_i$ that the fitted value $\hat{Y}_i$ increases or decreases with the inclusion of the i^{th} observation. We can compute them with the `dffits` function.

```
dff <- dffits(cars.lm)
dff[1:5]
```

```
         1          2          3          4          5
0.09490289 0.29397684 -0.11039550 0.22487854 0.03553887
```

A rule of thumb is to flag observations whose $DFFIT$ exceeds one in absolute value, but there are none of those in this data set.

Cook's Distance The $DFFITS$ are good for measuring the influence on a single fitted value, but we may want to measure the influence an observation has on all of the fitted values simultaneously. The statistics used for measuring this are Cook's distances which may be calculated[fn:coodefine Cook's distances are actually defined by a different formula than the one shown. The formula in Equation **??** is algebraically equivalent to the defining formula and is, in the author's opinion, more transparent.] by the formula

$$D_i = \frac{E_i^2}{(p+1)S^2} \cdot \frac{h_{ii}}{(1-h_{ii})^2}, \quad i = 1, 2, \ldots, n. \tag{1.5.12}$$

It shows that Cook's distance depends both on the residual E_i and the leverage h_{ii} and in this way D_i contains information about outlying x and y values.

To assess the significance of D, we compare to quantiles of an f(`df1` = 2, `df2` = n − 2) distribution. A rule of thumb is to classify observations falling higher than the 50$^{\text{th}}$ percentile as being extreme.

How to do it with R We can calculate the Cook's Distances with the `cooks.distance` function.

```
cooksD <- cooks.distance(cars.lm)
cooksD[1:4]
```

```
          1           2           3           4
0.004592312 0.043513991 0.006202350 0.025467338
```

We can look at a plot of the Cook's distances with the command `plot(cars.lm, which = 4)`.

```
plot(cars.lm, which = 4)
```

Observations with the largest Cook's D values are labeled, hence we see that observations 23, 39, and 49 are suspicious. However, we need to compare to the quantiles of an f(`df1` = 2, `df2` = 48) distribution:

```
F0.50 <- qf(0.5, df1 = 2, df2 = 48)
any(cooksD > F0.50)
```

```
 null device
           1
 [1] FALSE
```

We see that with this data set there are no observations with extreme Cook's distance, after all.

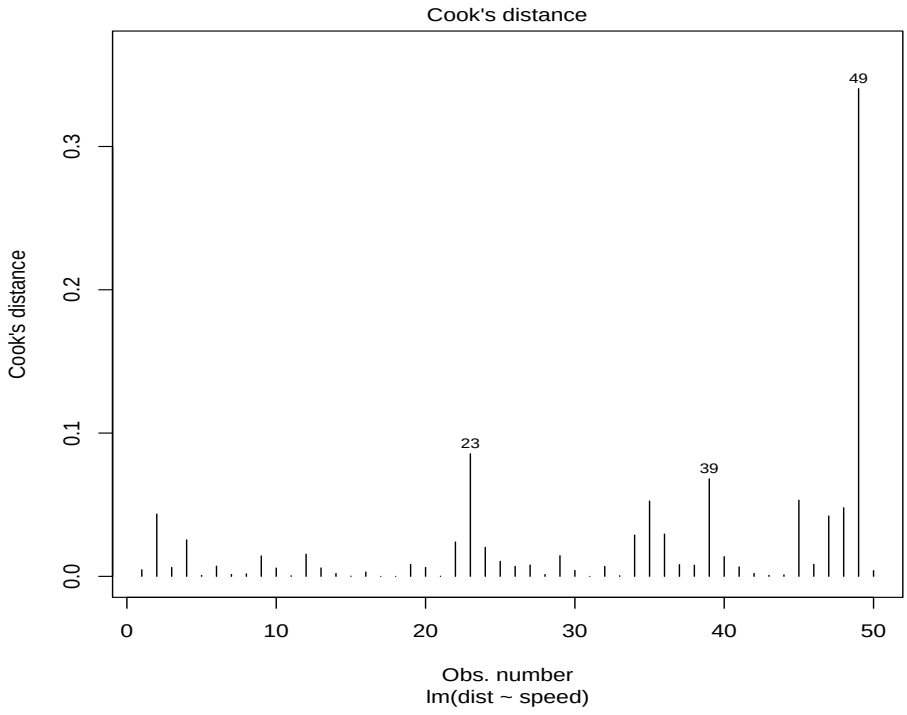

Figure 1.5.1: Used for checking for influential and/our outlying observations. Values with large Cook's distance merit further investigation.

1.5.5 All Influence Measures Simultaneously

We can display the result of diagnostic checking all at once in one table, with potentially influential points displayed. We do it with the command `influence.measures(cars.lm)`:

```
influence.measures(cars.lm)
```

The output is a huge matrix display, which we have omitted in the interest of brevity. A point is identified if it is classified to be influential with respect to any of the diagnostic measures. Here we see that observations 2, 11, 15, and 18 merit further investigation.

We can also look at all diagnostic plots at once with the commands

```
par(mfrow = c(2,2))
plot(cars.lm)
par(mfrow = c(1,1))
```

The `par` command is used so that $2 \times 2 = 4$ plots will be shown on the same display. The diagnostic plots for the `cars` data are shown in Figure **??**:

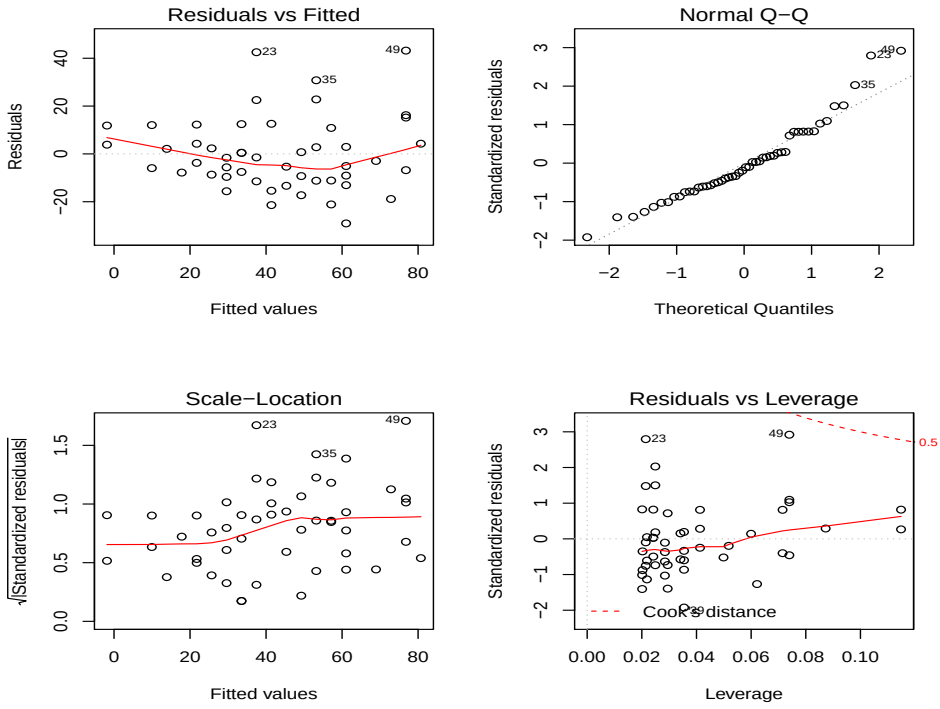

Figure 1.5.2: Diagnostic plots for the `cars` data.

We have discussed all of the plots except the last, which is possibly the most interesting. It shows Residuals vs. Leverage, which will identify outlying y values versus outlying x values. Here we see that observation 23 has a high residual, but low leverage, and it turns out that observations 1 and 2 have relatively high leverage but low/moderate leverage (they are on the right side of the plot, just above the horizontal line). Observation 49 has a large residual with a comparatively large leverage.

We can identify the observations with the `identify` command; it allows us to display the observation number of dots on the plot. First, we plot the graph, then we call `identify`:

```
plot(cars.lm, which = 5)   # std'd resids vs lev plot
identify(leverage, sres, n = 4)   # identify 4 points
```

The graph with the identified points is omitted (but the plain plot is shown in the bottom right corner of Figure **??**). Observations 1 and 2 fall on the far right side of the plot, near the horizontal axis.

1.6 Chapter Exercises

Exercise 1.1. Prove the ANOVA equality, Equation **??**. *Hint*: show that

$$\sum_{i=1}^{n}(Y_i - \hat{Y}_i)(\hat{Y}_i - \overline{Y}) = 0.$$

Exercise 1.2. Solve the following system of equations for β_1 and β_0 to find the MLEs for slope and intercept in the simple linear regression model.

$$\begin{aligned} n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i \\ \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 &= \sum_{i=1}^{n} x_i y_i \end{aligned}$$

Exercise 1.3. Show that the formula given in Equation **??** is equivalent to

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right) \Big/ n}{\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2 \Big/ n}.$$