Inference for Regression: Test for Linear Association Between Two Quantitative Variables

Thomas Scofield

November 21, 2021

The Model Utility Test

There are things you can do whenever you have bivariate quantitative data, such as

- produce a scatterplot of the data.
- calculate the (sample) correlation r.
- find the slope b_1 and intercept b_0 (both *sample statistics*) of the least squares regression line.

As we discussed in Chapter 2, the correlation is not always *meaningful*. But, in this chapter, we will assume that it is—that we have variables X and Y where the average response value $\mu_Y(x)$ at any particular value of X is given linearly as

$$\mu_Y(X) = \beta_0 + \beta_1 X,$$

making it meaningful to discuss the true correlation ρ.

An association between X and Y exists if the true slope $\beta_1 \neq 0$ or, equivalently, if the true correlation $\rho \neq 0$. Otherwise the variables are independent, meaning X has no value in predicting Y. To conduct a test, called the **model utility test**, of

$$\mathbf{H}_0: \ \beta_1 = 0 \qquad \text{vs.} \qquad \mathbf{H}_a: \ \beta_1 \neq 0,$$

or equivalent stated as

$$\mathbf{H}_0: \ \rho = 0 \qquad \text{vs.} \qquad \mathbf{H}_a: \ \rho \neq 0,$$

we will need sample data, producing a sample slope b_1 or sample correlation r.

Scatterplots; calculating b_1, r in R

The data set found at http://scofield.site/teaching/data/csv/heartDiseaseDeathsAndWine.csv contains a variable `winealc` that measures wine consumption (measured in liters per person per year) in various countries and another variable `hddeaths` which measures heart disease mortality rates (deaths per thousand). We import this data, view a scatterplot, and calculate the sample values.

```
hdAndWine <- read.csv("http://scofield.site/teaching/data/csv/heartDiseaseDeathsAndWine.csv")
head(hdAndWine)
```

```
##       country winealc hddeaths
## 1   Australia     2.5      211
## 2 Netherlands     1.8      167
## 3     Austria     3.9      167
## 4 New Zealand     1.9      266
## 5     Belgium     2.9      131
## 6      Norway     0.8      227
```

from sampled data we can glean statistics like

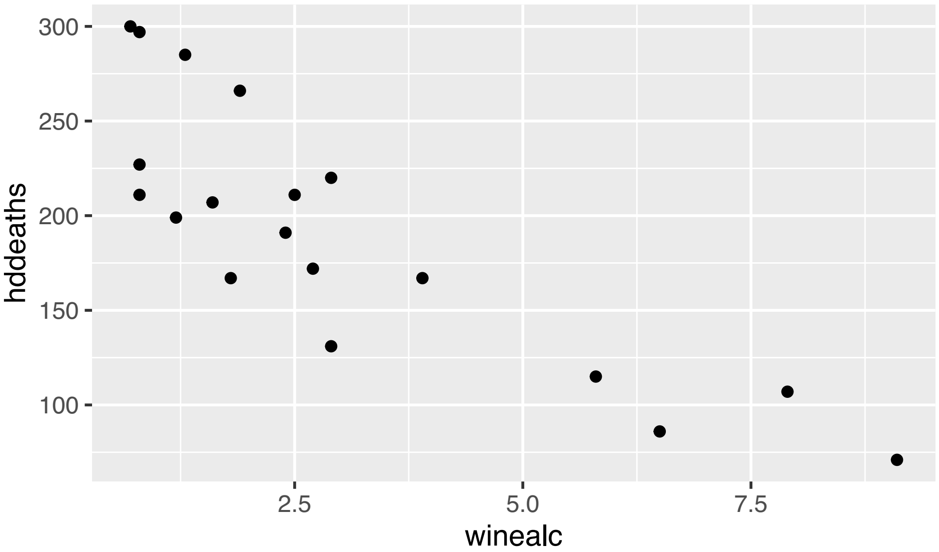
r (sample correlation)

b_0 (sample y-intcpt)

b_1 (sample slope)

In making a scatterplot, we must decide which variable (between `winealc` and `hddeaths`) to consider explanatory, placing it on the horizontal axis. Either could serve in that role, but in most discussions involving these variables, it is the alcohol consumption that people generally adopt as explanatory. So, it appears on the right side of the tilde in the command

```
gf_point(hddeaths ~ winealc, data=hdAndWine)
```



We can get the coefficients (intercept b_0 and slope b_1) of the best-fit line for the data via the command

```
lm(hddeaths ~ winealc, data=hdAndWine)
```

```
##
## Call:
## lm(formula = hddeaths ~ winealc, data = hdAndWine)
##
## Coefficients:
## (Intercept)      winealc
##      260.56       -22.97
```

b_0 b_1

Note that, by adding `$coefficients`, the output is less “wordy”,

```
lm(hddeaths ~ winealc, data=hdAndWine)$coefficients
```

```
## (Intercept)     winealc
##   260.56338   -22.96877
```

and by altering this to ‘*coefficients*[2] we obtain just the slope.

```
lm(hddeaths ~ winealc, data=hdAndWine)$coefficients[2]
```

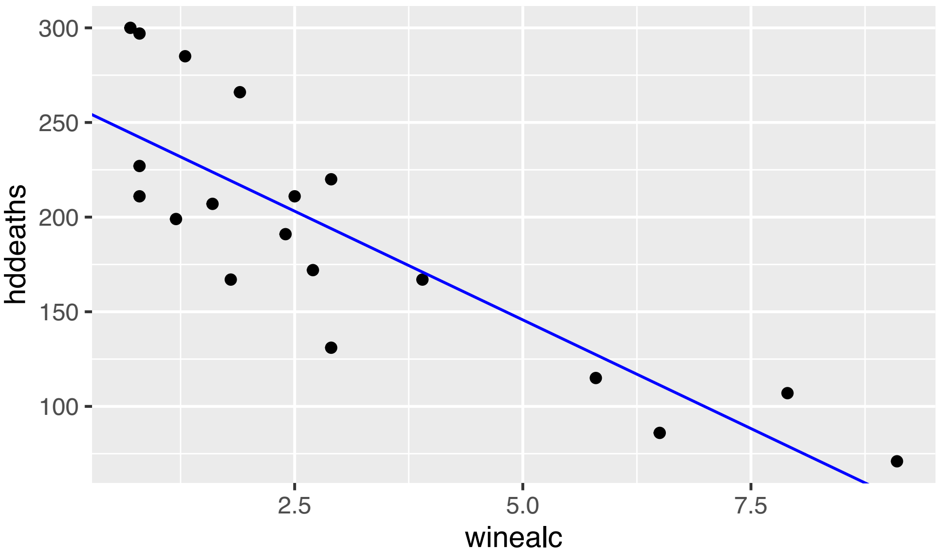
```
##   winealc
## -22.96877
```

This last version, isolating the response to *slope* only, will be helpful in generating randomization distributions for b_1.

We can overlay the best-fit line by “piping” the scatterplot to the `gf_abline()` command with specified slope and intercept:

```
gf_point(hddeaths ~ winealc, data=hdAndWine) %>%
  gf_abline(slope = ~-22.97, intercept = ~260.56, color="blue")
```

can replace this part with the simpler gf_lm(type = "lm")



It is simpler, and achieves the same thing as above, to pipe the scatterplot to `gf_lm()`.

```
gf_point(hddeaths ~ winealc, data=hdAndWine) %>% gf_lm(color="blue")
```

To calculate the sample correlation, instead, we change `lm()` to `cor()`:

```
cor(hddeaths ~ winealc, data=hdAndWine)
```

```
## [1] -0.8428127
```

Question

Would we get the same slope and intercept if we exchanged the roles of the variables, in this case making `hddeaths` the explanatory variable? Would we get the same correlation?

Randomization

Randomization distributions are meant to simulate the null distribution—what sort of values we expect, and how frequently, out of our sample statistic when the null hypothesis (no association between the quantitative variables) holds. We simulate it by shuffling one of the variables.

Randomization distribution for b_1**:** In the context of our *wine-and-heart-disease-deaths* data, one randomization statistic b_1 arises from

```
lm(hddeaths ~ shuffle(winealc), data=hdAndWine)$coefficients[2]
```

```
## shuffle(winealc)
##         4.161994
```

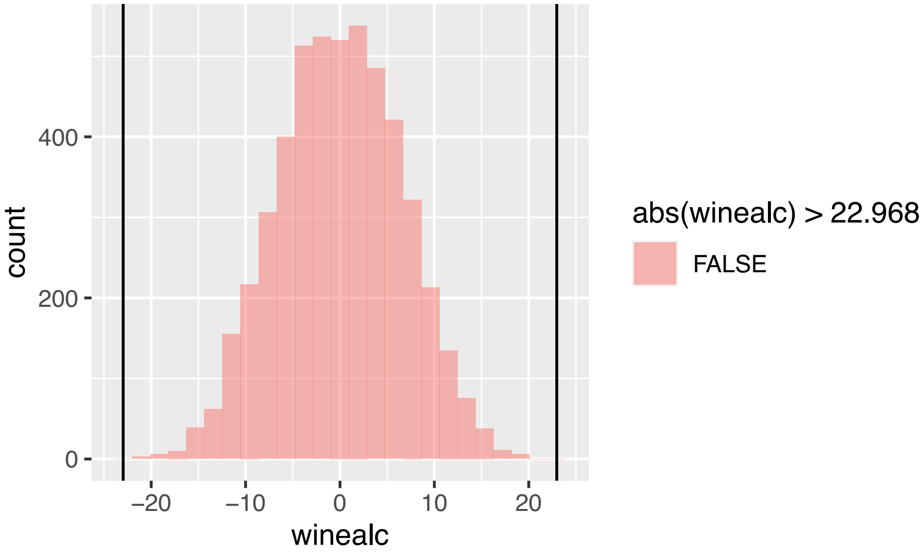
We get an approximate P-value when we generate lots of these randomization statistics, locate our test statistic (the slope for the original data), and determining how often something that extreme (or more so) occurs:

```
manyb1s <- do(5000) * lm(hddeaths ~ shuffle(winealc), data=hdAndWine)$coefficients[2]
head(manyb1s)
```

```
##      winealc
## 1  12.158335
## 2  -4.762228
## 3 -10.205138
## 4   7.933487
## 5 -10.321564
## 6   3.098284
```

This randomization distribution with the test statistic can be visualized in a manner like we have used before:

```
gf_histogram(~winealc, data=manyb1s, fill = ~abs(winealc)>22.968) %>%
  gf_vline(xintercept = ~22.968) %>%
  gf_vline(xintercept = ~-22.968)
```



This graph makes it appear that occurrences of $b_1 = 22.96877$, or even further distant from 0, are extremely rare. Indeed, if we count how often it happened in our 5000 tries, we have

```
2 * nrow( filter(manyb1s, winealc < -22.968) ) / 5000
```

```
## [1] 0
```

Randomization distribution for r: Consider another dataset, the **RestaurantTips** data from the `Lock5withR` package. The question here is whether there is an association between a bill for the meal at a restaurant, and the tip as a percentage-of-the-bill-as-tip (variable name `PctTip`). There are other ways to state this question of association. In the text, the Locks state it (roughly) as, “Is `Bill` an effective predictor of the size of the tip as a percentage of the bill?” The null hypothesis says, “no, it isn’t”, or $\rho = 0$.

To generate a randomization distribution for r under this null hypothesis, we start with the command that generates the r from the original data:

```
cor(PctTip ~ Bill, data=RestaurantTips)
```

```
## [1] 0.1352976
```

That is our test statistic.

The generation of a single randomization statistic r comes from shuffling one of the variables:

```
cor(PctTip ~ shuffle(Bill), data=RestaurantTips)
```

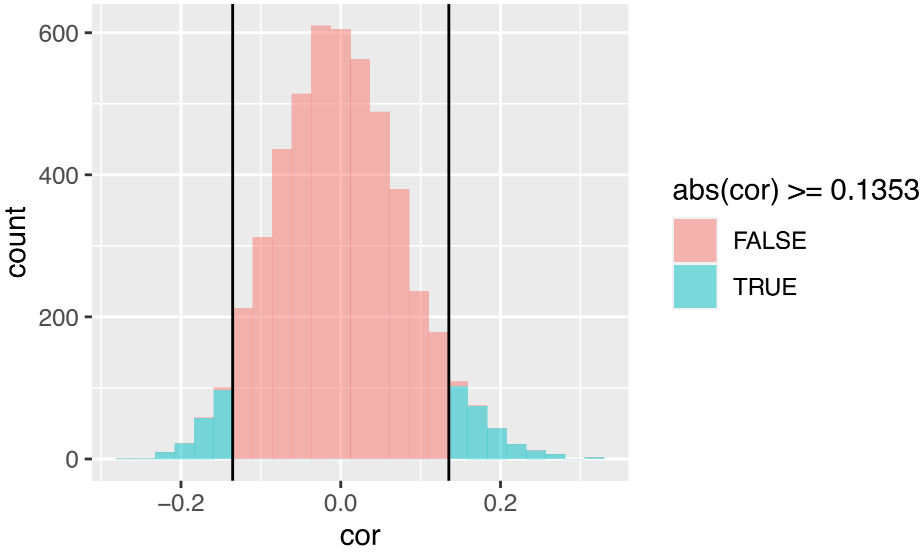
```
## [1] 0.001997988
```

If we do this often, we get a randomization distribution for r. We locate our test statistic on this distribution and use it as a boundary to determine the P-value.

```
manyCors <- do(5000) * cor(PctTip ~ shuffle(Bill), data=RestaurantTips)
head(manyCors)
```

```
##          cor
## 1 -0.04941720
## 2  0.01103168
## 3 -0.09646720
## 4 -0.07947382
## 5 -0.09839163
## 6  0.04777958
```

```
gf_histogram(~cor, data=manyCors, fill= ~abs(cor) >= 0.1353) %>%
  gf_vline(xintercept = ~0.1353) %>%
  gf_vline(xintercept = ~-0.1353)
```



```
nrow( filter(manyCors, abs(cor) >= 0.13529) ) / 5000
```

```
## [1] 0.0902
```

This P-value is not smaller than $\alpha = 0.05$, so at that level we fail to reject the null hypothesis, that the true correlation ρ (and the true slope β_1) is zero. That is, if someone tends to think that patrons of restaurants tip the same percentage regardless of the overall tab, we have not found here evidence sufficient to conclude otherwise.

Exercise

Throughout the discussion above, it has been implied that it didn't matter which test statistic you randomized, r or b_1. Convince yourself that this is the case by playing with randomization distributions and P-values for bivariate quantitative data in StatKey. The confirmation that "it does not matter" would be that, when working with a fixed dataset, when you generate a P-value corresponding to your test statistic r, it is roughly the same as the P-value corresponding to your test b_1.

Jeffrey, Valeria,
Natalie F

This reflects where class ended on Tues., Nov. 23

Model Utility Test

- context is bivariate quant. data (can view scatterplot)
- already decided data appears linear
- already computed b_1 or r

  probably not zero (either one)

  $\Rightarrow$? population β_1 or ρ are 0 ?

$$\begin{aligned} H_0 &: \beta_1 = 0 \\ H_a &: \beta_1 \neq 0 \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} H_0 &: \rho = 0 \\ H_a &: \rho \neq 0. \end{aligned}$$

Before break, discussed using randomization to conduct this test.

Without randomization? Context

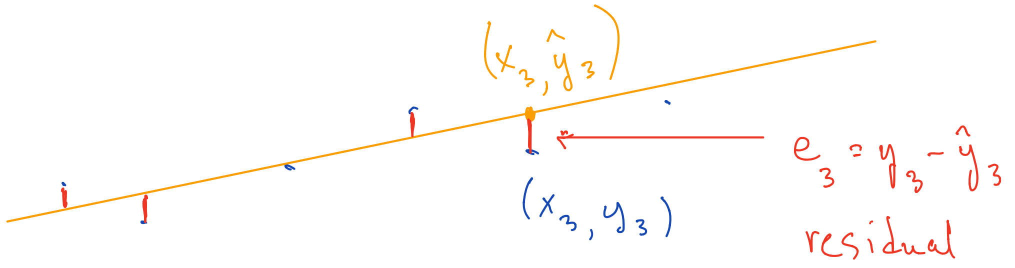
$$b_1 = r \frac{s_y}{s_x}, \qquad b_0 = \bar{y} - b_1 \bar{x}$$

Data:

$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Fitted values

$\hat{y}_i = b_0 + b_1 x_i$ — generates points on the regression line



Model Utility Test without randomization: Simple Linear Regression model

People have been conducting the Model Utility Test for a long while, since before computers were on every desktop, before it was feasible to generate randomization distributions, when only hand calculations were possible. Upon request, R will generate the kind of results those hand calculations produced. Naturally the `lm()` command is used, but so is `summary()`. In the case of the *PctTip-and-Bill* data, the request goes like this:

```
summary( lm(PctTip ~ Bill, data=RestaurantTips) )
```

```
##
## Call:
## lm(formula = PctTip ~ Bill, data = RestaurantTips)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9927 -2.3096 -0.6455  1.4679 25.5335
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.50965    0.73956   20.97   <2e-16 ***
## Bill         0.04881    0.02871    1.70   0.0911 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.36 on 155 degrees of freedom
## Multiple R-squared:  0.01831,    Adjusted R-squared:  0.01197
## F-statistic:  2.89 on 1 and 155 DF,  p-value: 0.09112
```

b_1

SE_{b_1}

95% CI for β_1

$$\begin{pmatrix}\text{point}\\ \text{est.}\end{pmatrix} \pm \begin{pmatrix}\text{critical}\\ \text{value}\end{pmatrix}(SE)$$

$$b_1 \pm (t^*)(SE_{b_1})$$

$$(0.0488) \pm (1.975)(0.0287)$$

$$\uparrow$$

qt(0.975, n-2)

$$\uparrow$$

always used in regression when just one expl. var.

Notice the result contains information about both coefficients. The `(Intercept)` row says

> Our point estimate b_0 for β_0 is 15.50965, has $\text{SE}_{b_0} = 0.73956$, and standardized t-score $t = 20.97$.

The other row, the one about *slope*, is always of greater interest to us. It says

> Our point estimate b_1 for β_1 is 0.04881, has $\text{SE}_{b_1} = 0.02871$, and standardized t-score $t = 1.70$.

We see that the process our forbears devised in lieu of randomization led them to deal, again, with t-distributions. Specifically, since in the Model Utility Test we hypothesize that $\beta_1 = 0$, standardizing leads to the reported t-score:

$$t = \frac{0.04881 - \overset{\text{null value}}{0}}{0.02871} = 1.700.$$

To get the resulting P-value the way they (and this command) did, we find how often a t-score is as extreme or more so than this one—i.e., compute the tail area and double it. We get the tail area using a t-distribution, but with how many degrees of freedom? If there are n cases in the dataset, regression computes degrees of freedom in this way:

$$\text{df} = n - 1 - (\text{number of predictor variables}).$$

What is implied here is that it is possible to consider more than 1 explanatory/predictor variable. When we do, it is called **multiple regression**. Since we are considering only 1 predictor variable, the number of degrees of freedom is $\text{df} = n - 2$. So, in the case of *PctTip-as-predicted-by-Bill* data, which has $n = 157$ cases, R determined its P-value above by doing what the following command does:

```
2 * (1 - pt(1.7, df=155))
```

```
## [1] 0.09113667
```

This standard error SE_{b_1} can be used the way we have used standard errors in the past, not only in the calculation of a standardized t-statistic above, but also in the **construction of a confidence interval for the true slope β_1**:

$$(\text{point est.}) \pm t^* \text{ SE}, \qquad \text{or} \qquad b_1 \pm t^* \text{ SE}_{b_1}.$$

You choose the critical value t^* as in Chapter 6, but now using $n-2$ degrees of freedom. So, to get a 92% CI for β_1 for the *PctTip-as-predicted-by-bill* data, with sample size

```
nrow(RestaurantTips)
```

```
## [1] 157
```

we get our critical value

```
tstar <- qt(0.96, df=155); tstar
```

```
## [1] 1.76224
```

and our confidence interval

```
lm(PctTip ~ Bill, data=RestaurantTips)$coefficients[2] + c(-1,1) * tstar * 0.02871
```

```
## [1] -0.001781462  0.099406336
```

As we have seen before, there is a connection between the P-value of a Model Utility Test and this confidence interval for β_1. Since the null value, 0, is inside a 92% CI, we would have expected correspondingly that the P-value of the Model Utility Test to be *larger* than 0.08, and above we found it to be 0.091, confirming that.

Question: Our forbears also developed the formula for the standardized t-statistic of the sample correlation r to be

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}, \qquad \Bigg\} \ \text{Needs only } r, n$$

which also has $n-2$ degrees of freedom when there is one predictor variable. What is the resulting P-value, and what hypotheses would we be testing?

Caution about using results based on a theoretical t-distribution: As in the previous chapters, beginning with Chapter 5, whenever we have turned to a theoretical distribution (a normal distribution, a t distribution, a chi-square distribution, or an F distribution) to compute a P-value, there have been conditions which validate the approach and, in the absence of such, leave us in some doubt about the conclusions. The same is true with the results above.

Conditions for the **Simple Linear Regression Model**. What has been assumed, and validates the approach, may be described as follows. Many different observed Y-values are possible for any fixed X-value, but

- they are normally distributed with mean $\mu_Y(X) = \beta_0 + \beta_1 X$, and
- the standard deviation is σ, a number that doesn't change with X.

The output from `summary( lm( ...) )` above includes an estimate for σ. It is the number reported as `Residual standard error`. For the *PctTip-as-predicted-by-Bill* data, the estimate for σ is 4.36

Some pictures from the textbook:

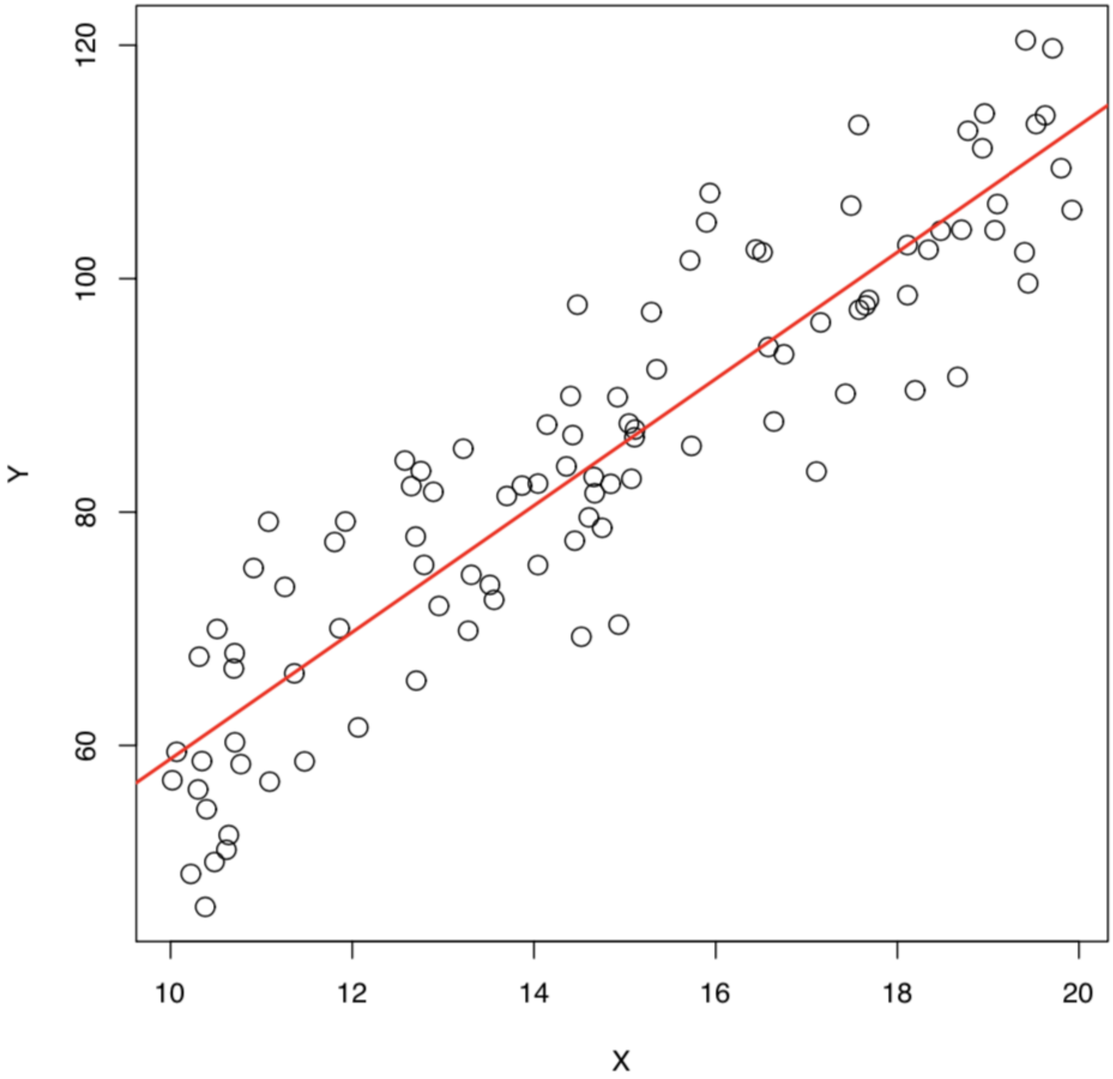


Figure 1: Good: conditions appear to be met

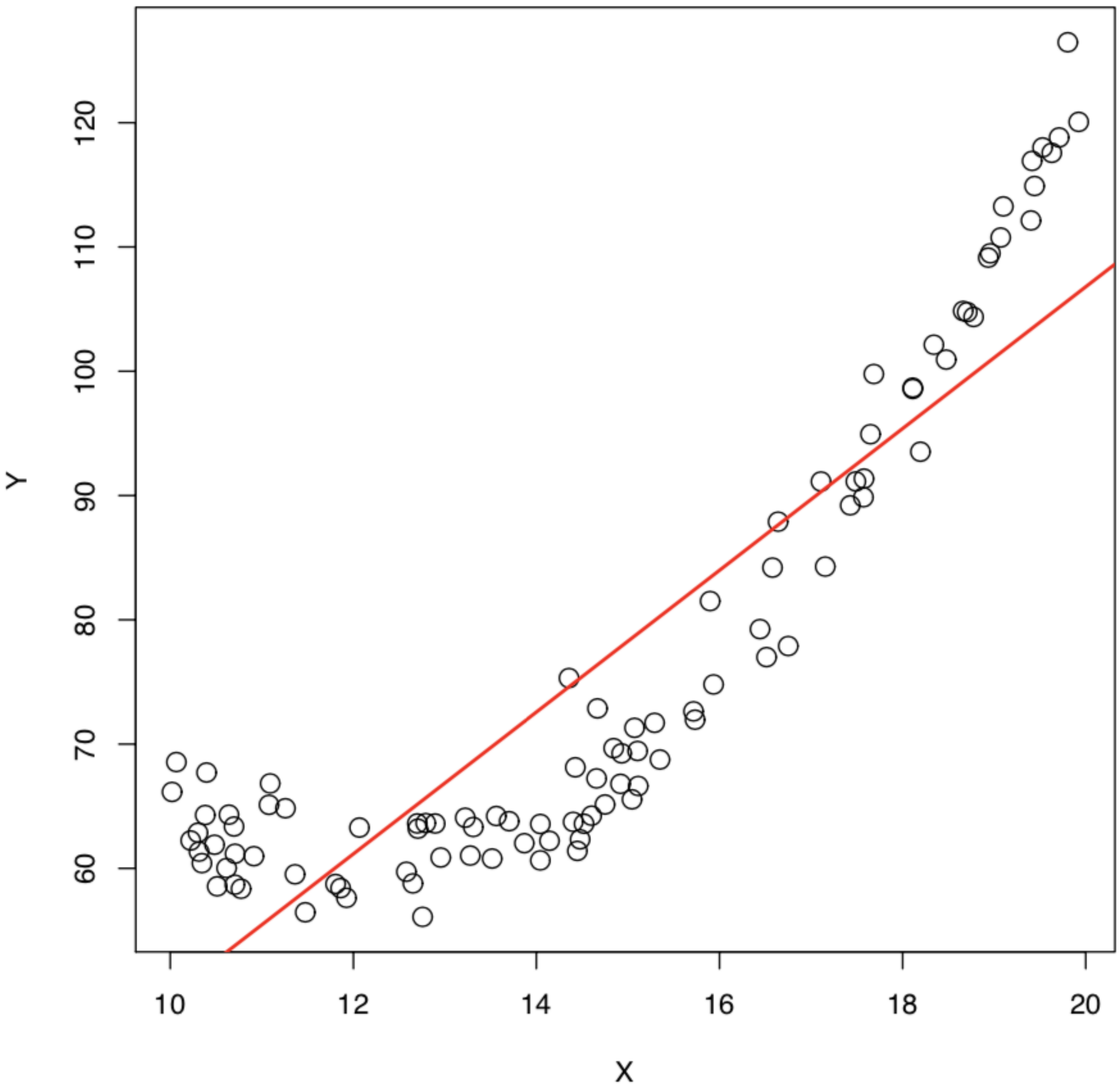


Figure 2: Bad: Not a linear association

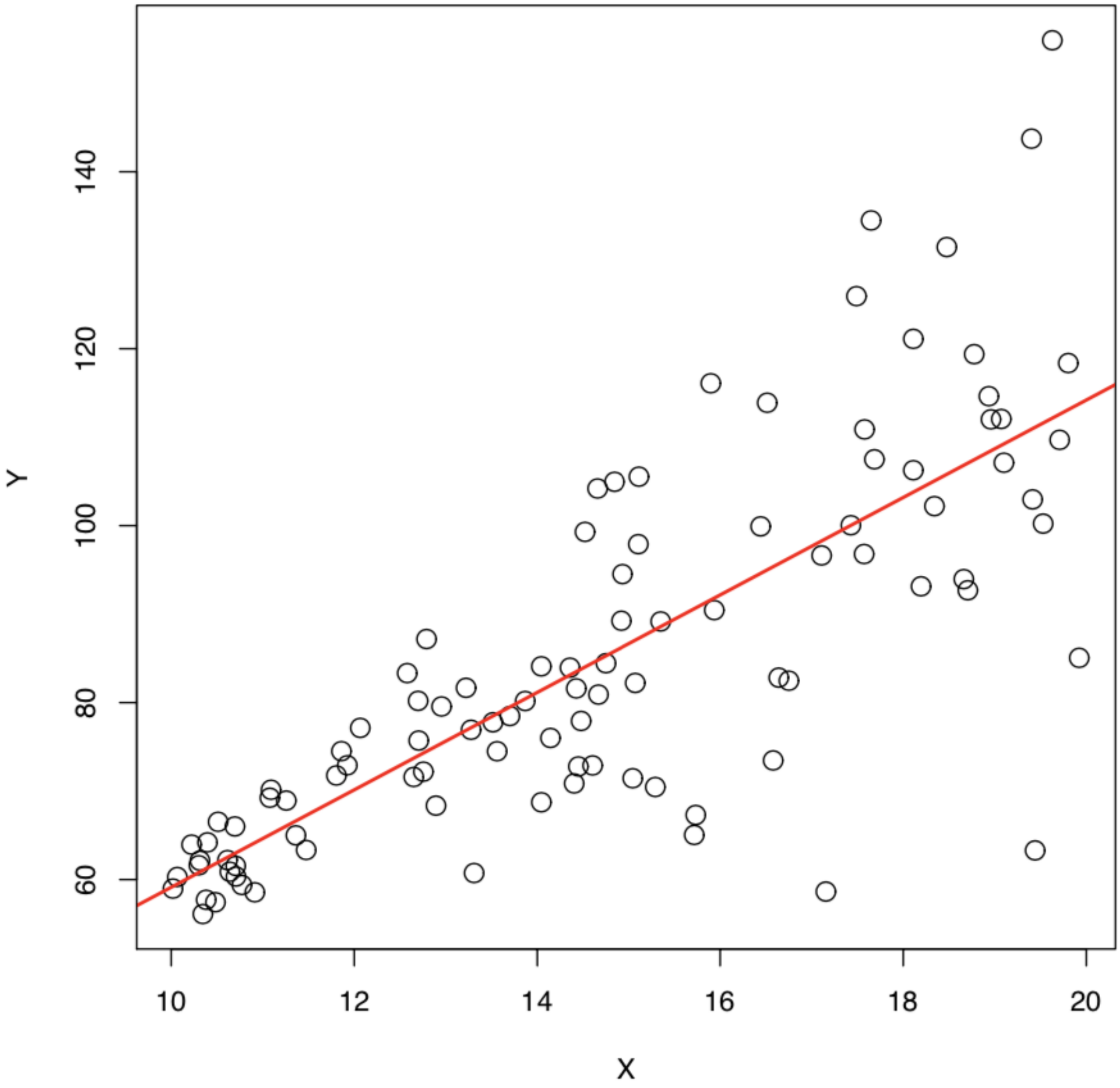


Figure 3: Bad: standard deviation changes with X

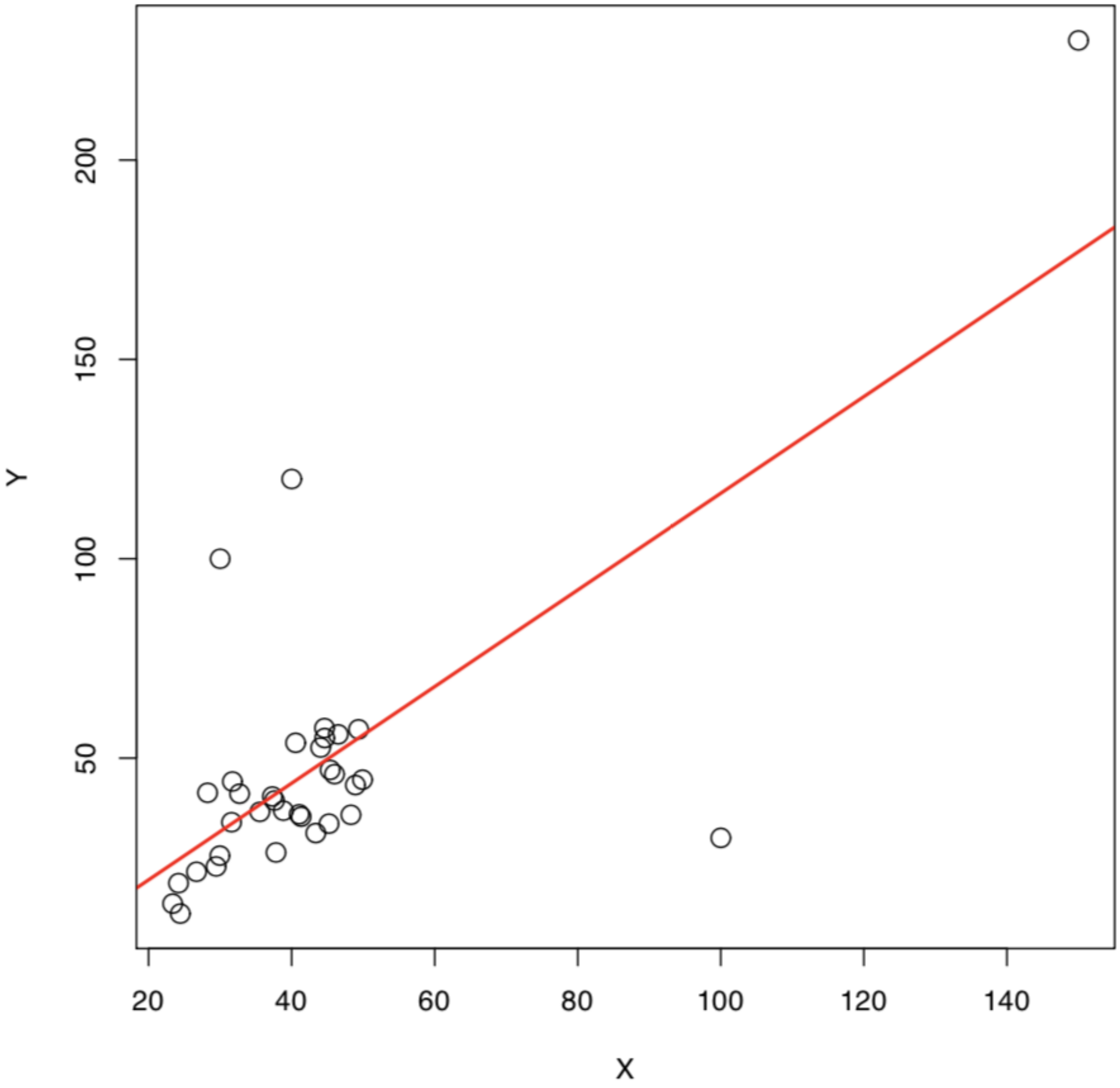


Figure 4: Bad: Outliers/influential points are present

ANOVA for regression

In Chapter 8, the quantitative response variable was used to generate SSG, SSE, and SST. In regression, we continue to have a quantitative response, the y-coordinates (known as **observed** values) of the data points

$$(x_1, y_1),\ \ (x_2, y_2),\ \ (x_3, y_3),\ \ \ldots\ \ (x_n, y_n).$$

We can, therefore, produce similar sum-of-squares values such as

$$SSTotal\ =\ \sum (y_i - \overline{y})^2\ =\ (n-1)s_y^2.$$

Here, $\overline{y}$ is the mean of observed responses,

$$\overline{y}\ =\ \frac{1}{n}(y_1 + y_2 + \cdots + y_n)\ =\ \frac{1}{n}\sum y_i.$$

As before, this $SSTotal$ breaks into two parts, $SSModel$, or SSM (formerly called SSG)

$$SSModel\ =\ \sum (\widehat{y}_i - \overline{y})^2,$$

and $SSResid$ (or SSE)

$$SSE\ =\ \sum (y_i - \widehat{y}_i)^2.$$

Here, $\widehat{y}_i$ refers to a **predicted** or **fitted value**; more specifically, the subscript i is associated with a point (x_i, y_i) in the data, and the x_i from that point is what generates our point estimate of the mean response at that $X = x_i$, namely

$$\widehat{y}_i = b_0 + b_1 x_i.$$

Note, from Chapter 2 we defined a **residuals** to be the difference between an observed and predicted value,

$$\epsilon_i\ =\ y_i - \widehat{y}_i,$$

and $SSResid$ is simply the quantity we sought to make as small as possible in order to choose our least squares regression line.

As when we defined these quantities in Chapter 8, the relationship between them is

$$SSTotal = SSModel + SSResid, \qquad \text{or} \qquad SST = SSM + SSE.$$

When SSE is small in comparison with SST, that is indicative of a strong linear relationship between the variables; the line does a good job of explaining the variation in observed values. A good measure of how well the variability in response values Y is explained by the linear model $b_0 + b_1 X$ is the ratio

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

known as the **coefficient of determination**. It might have been better to use a lower-case r, and call it r^2, since the coefficient of determination is equal to the square of the correlation.

Example: `InkjetPrinters`. In the text, the Locks propose using `PPM`, the number of pages a printer can turn out per minute, to explain `Price`. The first few rows of the raw data are as follows.

```
head(InkjetPrinters)
```

```
##                                 Model PPM PhotoTime Price CostBW CostColor
## 1 HP Photosmart Pro 8500A e-All-in-One 3.9        67   300    1.6       7.2
## 2                   Canon Pixma MX882 2.9        63   199    5.2      13.4
## 3                 Lexmark Impact S305 2.7        43    79    6.9       9.0
## 4              Lexmark Interpret S405 2.9        42   129    4.9      13.9
## 5                 Epson Workforce 520 2.4       170    70    4.9      14.4
## 6                 Brother MFC-J6910DW 4.1       143   348    1.7       7.9
```

A scatterplot makes a linear relationship appear reasonable. The black points represent the data, while the purple points, lying on the line, are the *fits*. By storing the result of the `lm()` command, R can be asked to provide the fitted values (in the same order as the original data)

```
lmRes <- lm(Price ~ PPM, data=InkjetPrinters)
lmRes$fitted
```

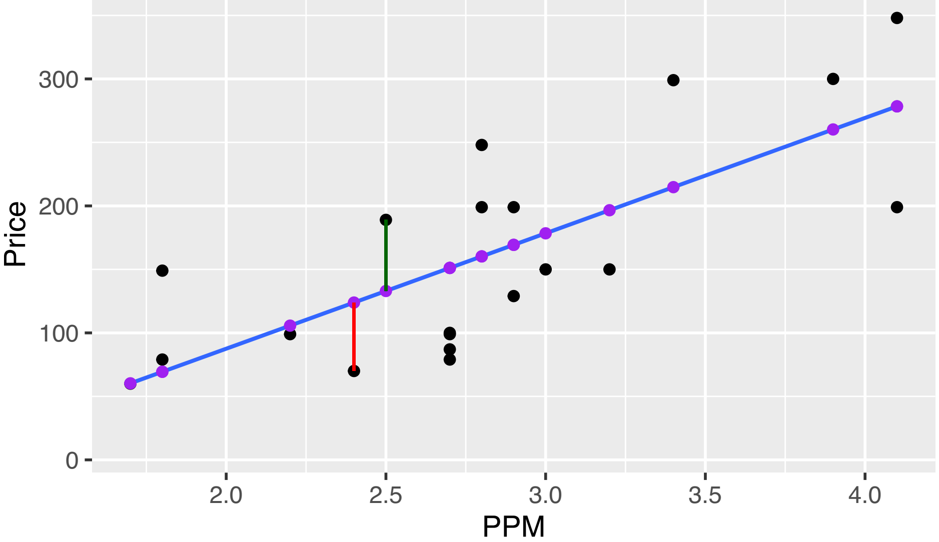
```
##         1         2         3         4         5         6         7         8 
## 260.20270 169.32464 151.14902 169.32464 123.88560 278.37832 214.76367 160.23683 
##         9        10        11        12        13        14        15        16 
## 178.41244 196.58806 151.14902 151.14902 105.70999 132.97341 151.14902  60.27095 
##        17        18        19        20 
## 160.23683  69.35876  69.35876 278.37832 
```

as well as the residuals

```
lmRes$residuals
```

```
##           1           2           3           4           5           6 
##  39.7972965  29.6753642 -72.1490222 -40.3246358 -53.8856019  69.6216830 
##           7           8           9          10          11          12 
##  84.2363304  87.7631710 -28.4124425 -46.5880561 -64.1490222 -51.1490222 
##          13          14          15          16          17          18 
##  -6.7099884  56.0265913 -52.1490222  -0.2709545  38.7631710  79.6412387 
##          19          20 
##   9.6412387 -79.3783170 
```

The vertical green line is from the 14th observed value, the observed price of \$189 for a Dell V715 w inkjet printer, down to its fitted price of \$132.97, a positive residual of $189 - 132.97 = 56.03$. The red vertical line is from the 5th observed value, the price of \$70 for an Epson Workforce 520 up to its fitted price of \$123.89, a negative residual of $70 - 123.89 = -53.89$.



Take a look at the output from the following commands applied to this data. First the summary from `lm()` (recall that `lmResult` stores output from `lm()`).

```
summary(lmRes)
```

```
## 
## Call:
## lm(formula = Price ~ PPM, data = InkjetPrinters)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79.38 -51.40  -3.49  43.85  87.76
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -94.22      56.40  -1.671 0.112086
## PPM            90.88      19.49   4.663 0.000193 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.55 on 18 degrees of freedom
## Multiple R-squared:  0.5471, Adjusted R-squared:  0.522
## F-statistic: 21.75 on 1 and 18 DF,  p-value: 0.0001934
```

See there is a number reported at the bottom with the label `Multiple R-squared`. That is the coefficient of determination, R^2, we defined above. It says about 55% of the variability in sampled printer prices is "explained" by the variable `PPM` through the linear model

$$\widehat{\text{Price}} = -94.22 + 90.88(\text{PPM}).$$

Next look at the correlation:

```
cor(Price ~ PPM, data=InkjetPrinters)
```

```
## [1] 0.7396862
```

Squaring this correlation

$$(0.7396862)^2 \doteq 0.5471,$$

yields the same number as `Multiple R-squared` reported above.

Now look at an ANOVA table:

```
anova(lmRes)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df Sum Sq Mean Sq F value    Pr(>F)
## PPM        1  74540   74540  21.747 0.0001934 ***
## Residuals 18  61697    3428
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The reported sum-of-squares values are $SSM = 74540$ and $SSE = 61697$, which means $SST = 74540+61697 = 136237$. We defined R^2 to be the ratio

$$\frac{SSM}{SST} = \frac{74540}{136237} = 0.5471,$$

again matching `Multiple R-squared`.

Question: Consider the command

```
sum( lmRes$residuals^2 )
```

```
## [1] 61696.79
```

Can you guess what number this will give and find its value using the ANOVA table? Run the command and check that you are correct.

Looking more carefully at this ANOVA table, we see that, out of the $n = 20$ cases (printers) in the **InkjetPrinters** dataset, $n - 2 = 18$ degrees of freedom have been "assigned" to the `Residuals`, and 1 degree of freedom to `PPM`, for a total of $18 + 1 = 19 = n - 1$. In simple linear regression (i.e., regression with just 1 predictor variable), the number of degrees of freedom on the residual row is always $n - 2$. The calculations of quantities such as MSM (we called it MSG in Chapter 8), MSE and F which appear in the ANOVA table are done exactly as in 1-way ANOVA:

$$MSModel = \frac{SSModel}{1}, \qquad MSE = \frac{SSE}{n-2}, \qquad \text{and} \qquad F = \frac{MSM}{MSE},$$

and the resulting P-value, obtained with the command like

```
1 - pf(fstatistic, df1=1, df2=n-2)
```

is exactly the same as P-value from the Model Utility Test, representing another way to compute it.

Prediction and confidence intervals

If the conditions for the simple linear model are met, and if we have rejected the null hypothesis in the Model Utility Test in favor of the alternative, that the explanatory variable has some usefulness as a predictor of values of the response variable, it is typical to see the model used that way. There are two sorts of *prediction*-type questions we might ask.

1. What is the average response Y at a particular X? We denote this number by $\mu_Y(X)$, which is a parameter specific to the subpopulation of response one can see for that particular value of X.
2. What is the *next* response Y I expect to see for a particular X?

Example. For predicting `Price` of an inkjet printer from its `PPM`, we have the coefficients

```
lm(Price ~ PPM, data=InkjetPrinters)
```

```
##
## Call:
## lm(formula = Price ~ PPM, data = InkjetPrinters)
##
## Coefficients:
## (Intercept)          PPM
##      -94.22        90.88
```

which we express as a linear model

$$\widehat{\text{Price}} = -94.22 + 90.88(\text{PPM}).$$

The best *single number* to

1. estimate the average price for an inkjet printer that prints 3.5 pages per minute is

   $$-94.22 + 90.88(3.5) \;=\; 223.86,$$

   or $223.86.
2. estimate the price of the next inkjet printer that prints 3.5 pages is, likewise, $-94.22+90.88(3.5) = 223.86$.

But this number is most likely wrong, as it merely *estimates* answers to these questions. We would prefer to give an interval of values, along the lines of a confidence interval,

$$(\text{point estimate}) \pm (\text{margin of error}).$$

As one might expect, the margin of error is larger when predicting the *next* response value at X than it is for the *average* response. Formulas are available for these two margins of errors, but they are ugly. (You can find them incorporated into the interval formulas in the box on p. 553.)

We will use software to generate these intervals, and the easiest approach I know in R is to use the (stored) model to make an *estimator* function:

```
lmResult <- lm(Price ~ PPM, data=InkjetPrinters)
printerPriceEstimator <- makeFun( lmResult )
```

The resulting function `printerPriceEstimator()` (it seemed an appropriate name, given the situation), can be used to repeat my calculation of a single-number estimate above:

```
printerPriceEstimator(PPM = 3.5)
```

```
##        1
## 223.8515
```

It can also be used to generate a **confidence interval for the mean response** when `PPM` equals 3.5, a better answer to Question 1 than any single number can be:

```
printerPriceEstimator(PPM = 3.5, interval="confidence")
```

```
##        fit      lwr      upr
## 1 223.8515 184.5706 263.1324
```

Finally, the same estimator function, with switch `interval="prediction"`, gives an interval that responds to Question 2, known as a **prediction interval for a response value** at a given choice of the explanatory variable.

```
printerPriceEstimator(PPM = 3.5, interval="prediction")
```

```
##        fit      lwr      upr
## 1 223.8515 94.73146 352.9715
```

Not surprisingly, both the confidence interval and the prediction interval are centered at 223.85 (the single-number estimate), but the prediction interval is (much) wider. For both answers, the level of confidence was 95%. At the time of writing this, I do not know how to set the level of confidence to some other value.

An alternate command that one can use to produce these same intervals (and can be tweaked to change the level of confidence) is demonstrated below.

```
predict(lmResult, newdata=data.frame(PPM=3.5), interval="confidence", level = 0.95)
```

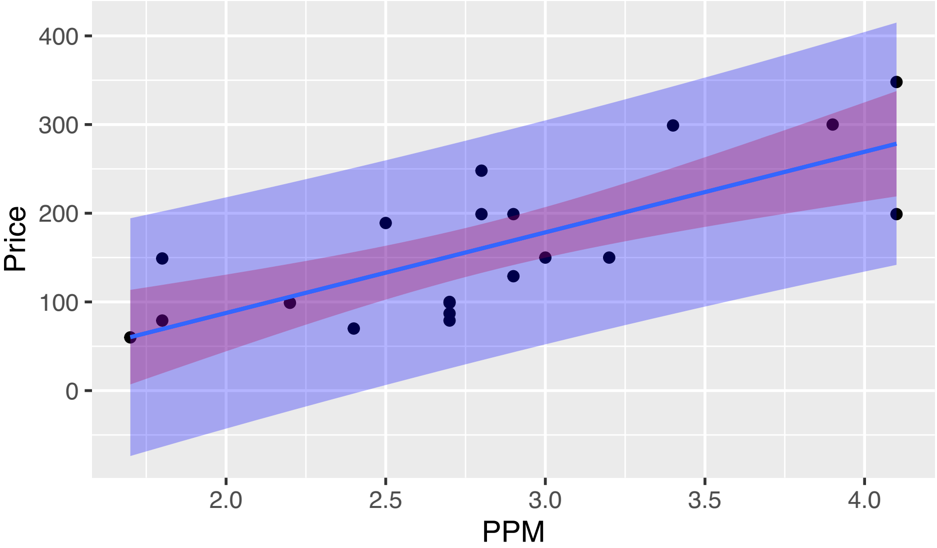
```
##        fit      lwr      upr
## 1 223.8515 184.5706 263.1324
```

```
predict(lmResult, newdata=data.frame(PPM=3.5), interval="prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 223.8515 94.73146 352.9715
```

It can be instructive to envision confidence and prediction intervals spread out around the regression line.

```
gf_point(Price ~ PPM, data=InkjetPrinters) %>%
  gf_lm(interval="confidence", fill="red") %>%
  gf_lm(interval="prediction",fill="blue")
```



Exercise: Use commands like those above to both a confidence interval for the mean response, and a prediction interval when `PPM = 3.0`. Compare your answers to those given in Example 9.19 on p. 554.