

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

Thomas Scofield

May 05, 2020

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

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

or equivalent stated as

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

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

Scatterplots; calculating b_1 , r in R

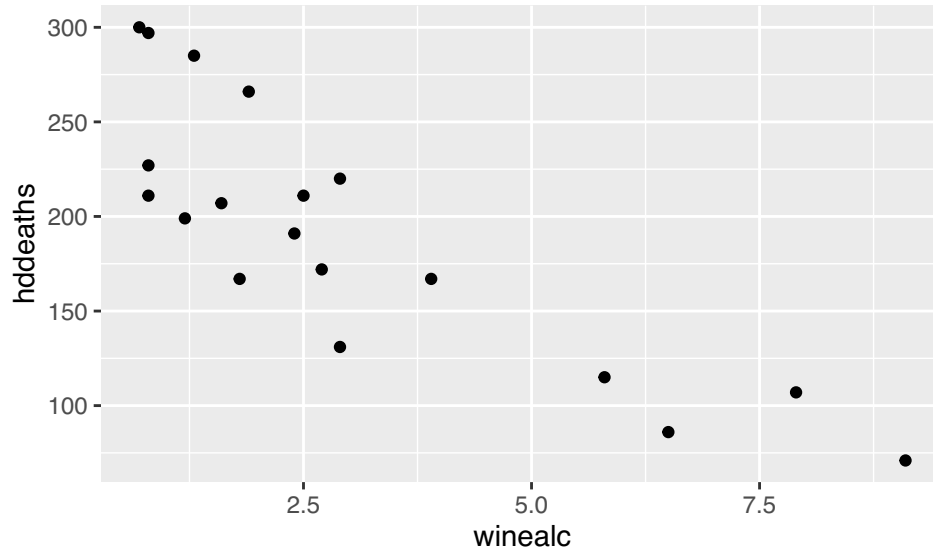
The data set found at <http://scofield.site/teaching/data/csv/heartDiseaseDeathsAndWine.csv> contains a variable `winealc` that measures wine consumption (measured in liters per person per year) in various countries and another variable `hddeaths` which measures heart disease mortality rates (deaths 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
```

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
```

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.

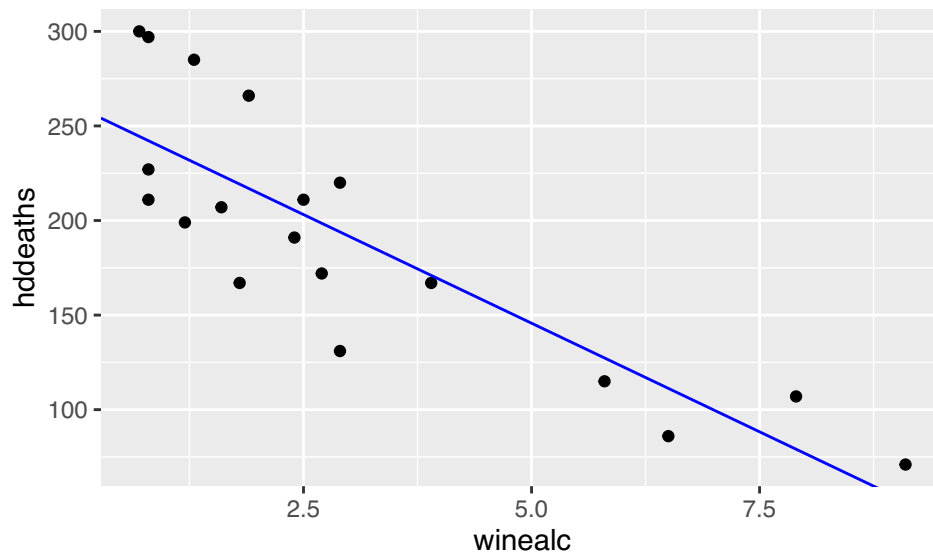
```
lm(hddeaths ~ winealc, data=hdAndWine)$coefficients[2]
```

```
##      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")
```



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)
## -9.162597
```

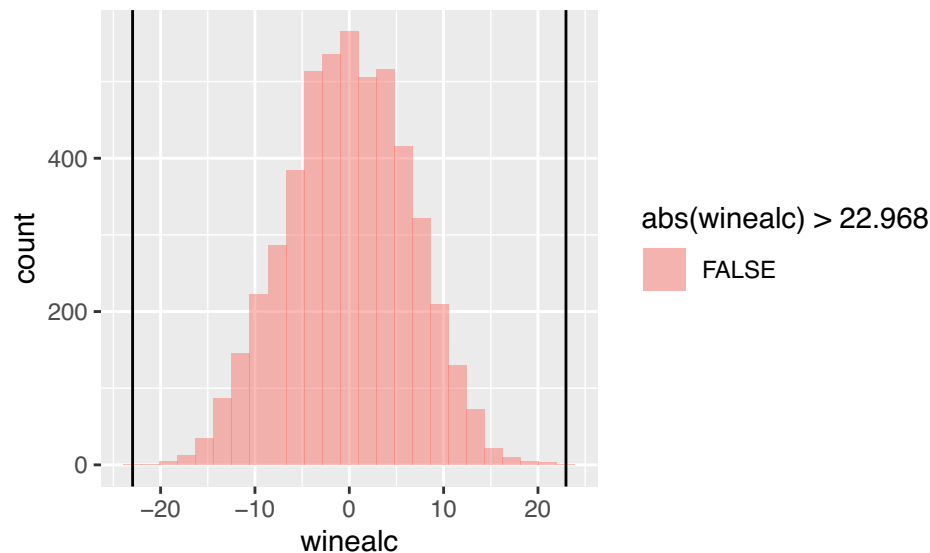
We get an approximate P -value when we generate lots of these randomization statistics, locate our test statistic (the slope for the original data), and determining how often something that extreme (or more so) occurs:

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

```
##      winealc
## 1  0.9532254
## 2 -3.1437312
## 3  1.2522282
## 4 -3.6473614
## 5  6.1765143
## 6 -4.0275096
```

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)
```

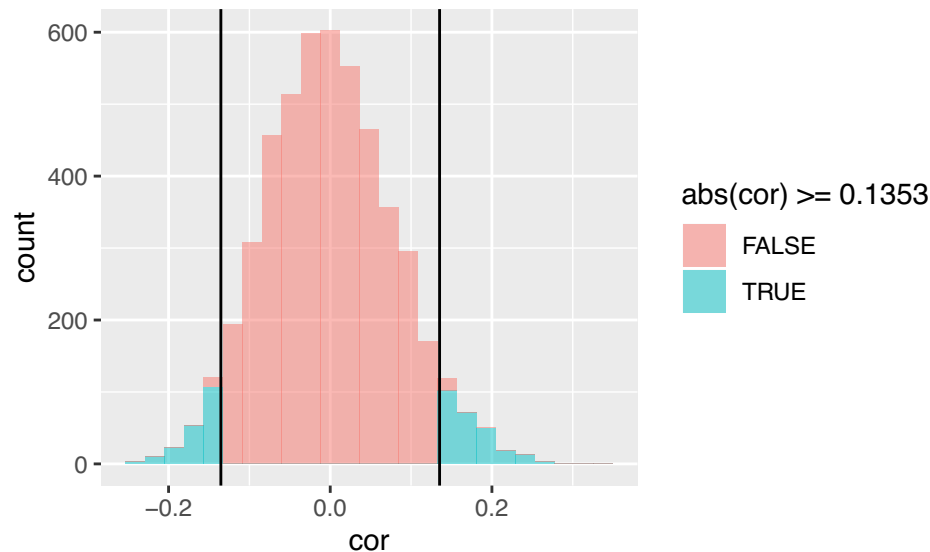
```
## [1] -0.05667181
```

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 -0.094804600
## 2  0.002920893
## 3  0.051602954
## 4  0.022790217
## 5 -0.073243585
## 6 -0.048819004
```

```
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] 0.0916
```

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 .

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
```

Results are suspect —
they rely on certain conditions,
"the simple linear regression
model", being met.

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

Our point estimate b_0 for β_0 is 15.50965, has $SE_{b_0} = 0.73956$, 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 $SE_{b_1} = 0.02871$, 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:

$$t = \frac{0.04881 - 0}{0.02871} = 1.700.$$

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:

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

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**, a topic we will not cover in this course. Since we are considering only 1 predictor variable, the number of degrees of freedom is $df = n - 2$. 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
```

check conditions of
SLRM before trusting wholeheartedly

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** :

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

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

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}},$$

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?

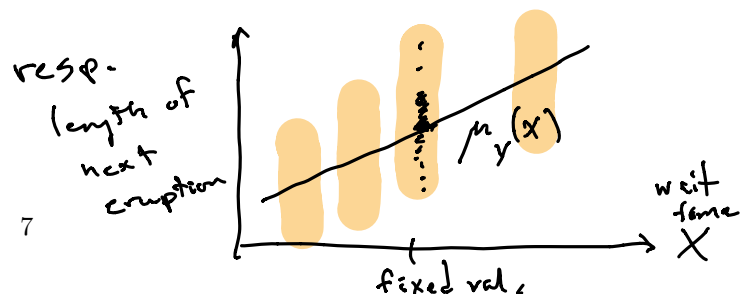
Caution about using results based on a theoretical t -distribution: As in the previous chapters, beginning with Chapter 5, whenever we have turned to a theoretical distribution (a normal distribution, a t distribution, a chi-square distribution, or an F distribution) to compute a P -value, there have 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:



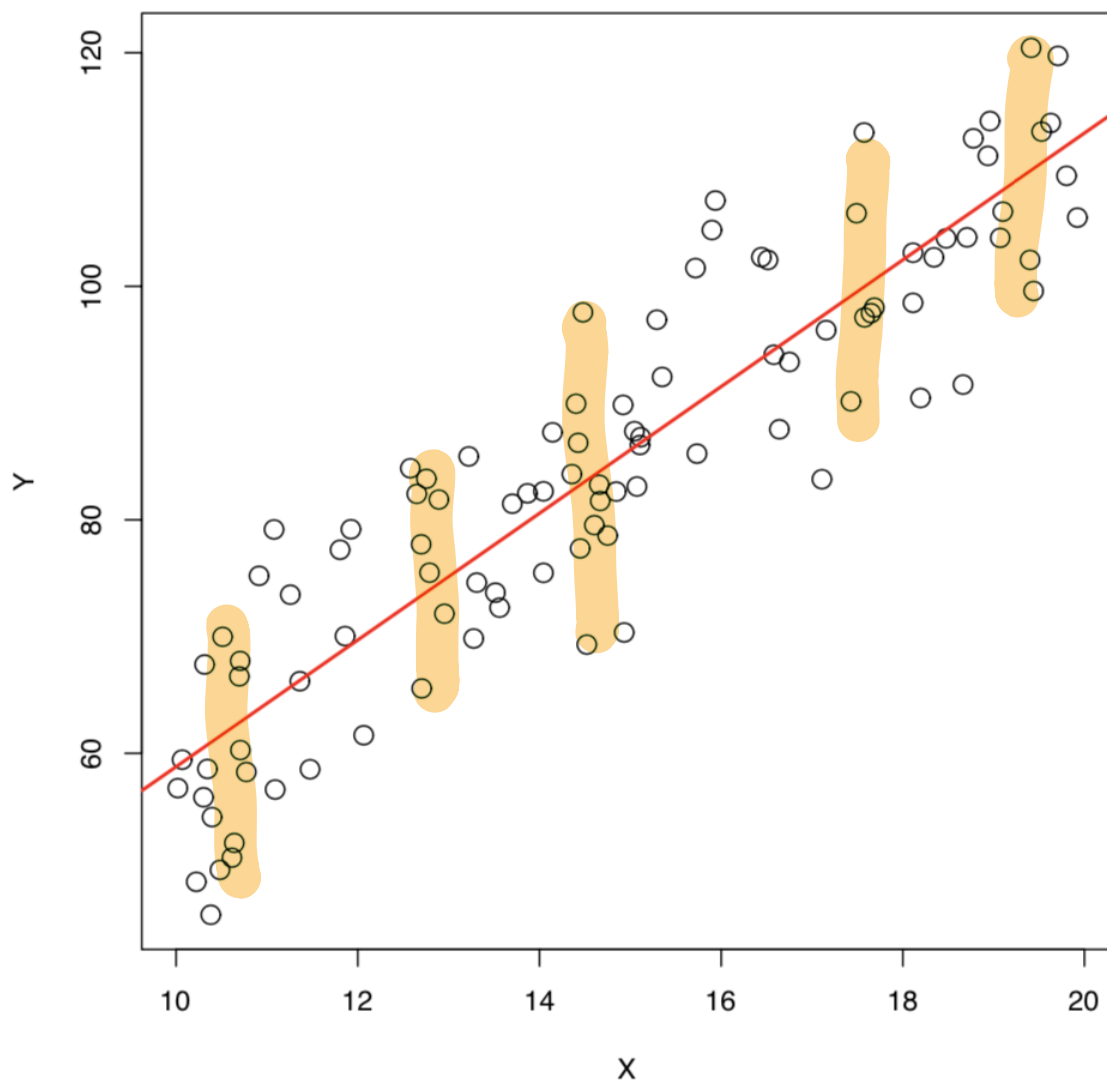


Figure 1: Good: conditions appear to be met

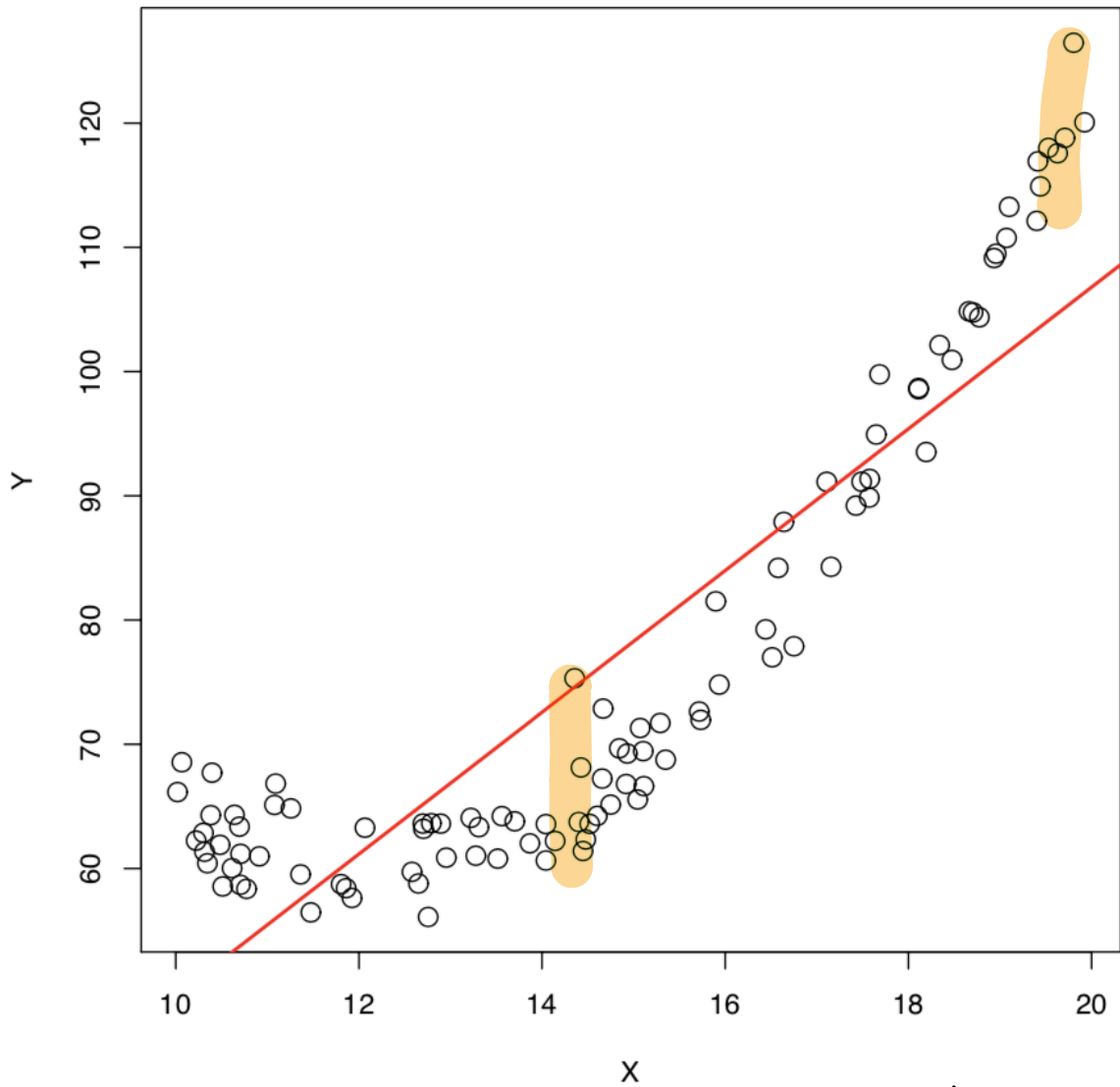


Figure 2: Bad: Not a linear association

Assumptions of
SLRM aren't met

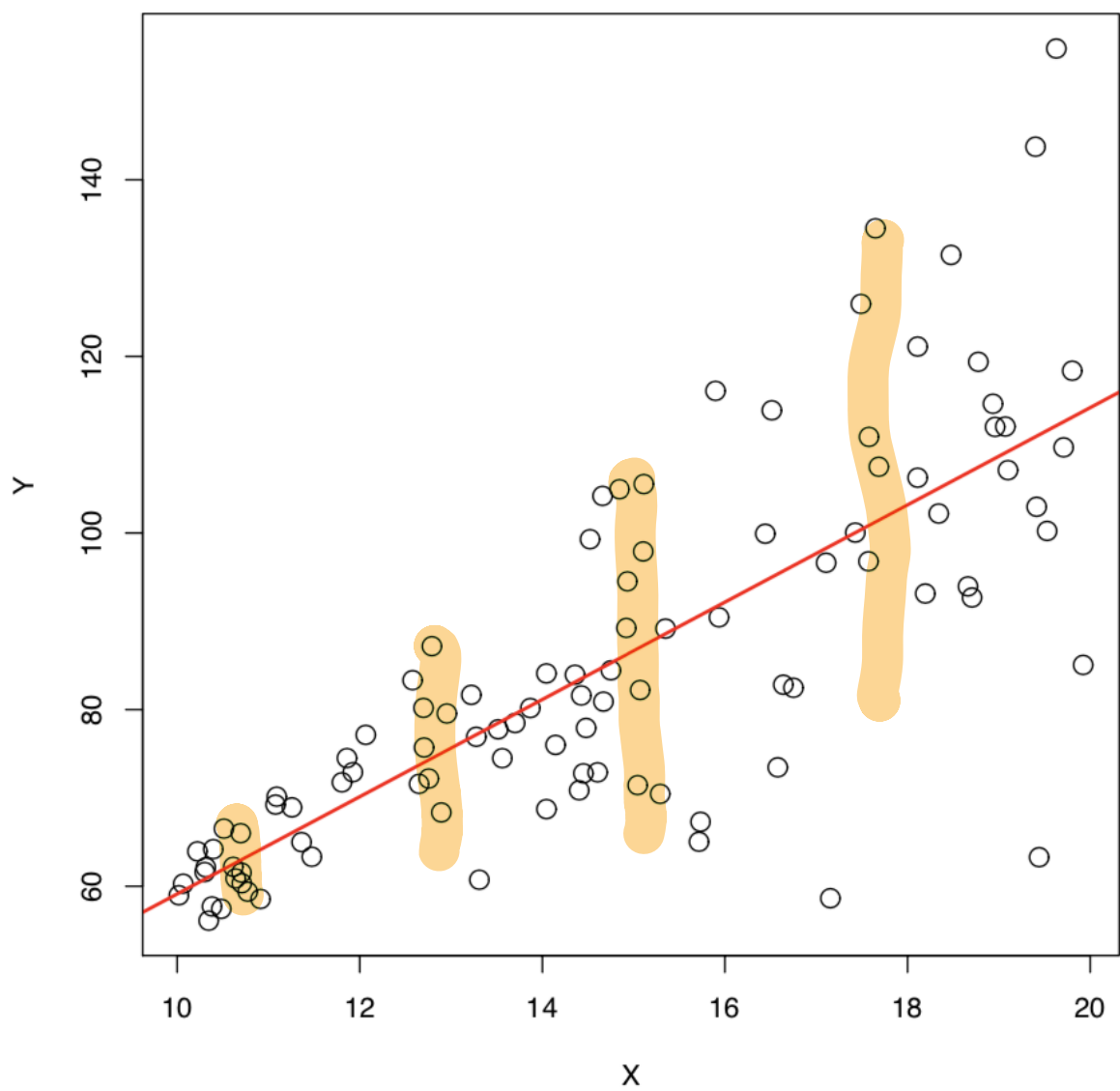


Figure 3: Bad: standard deviation changes with X

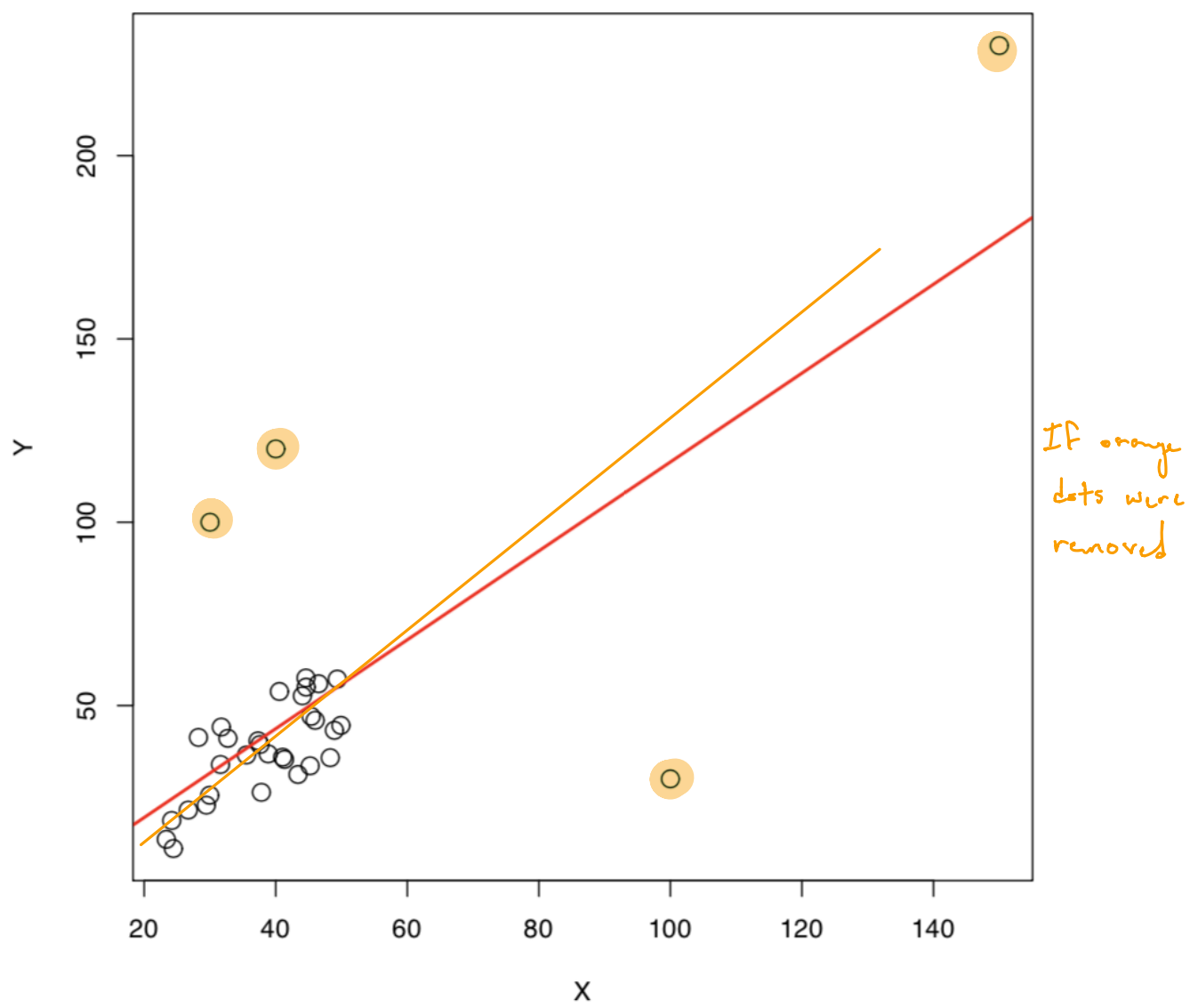


Figure 4: Bad: Outliers/influential points are present

ANOVA for regression

We can consider regression whenever we have bivariate quantitative data, a collection of n points which we can label

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

Since Y is treated as a response variable, we call the y_1, y_2, \dots, y_n **observed** values.

One can ignore the x -values and calculate things like the mean response-value

$$\bar{y} = \frac{1}{n}(y_1 + y_2 + \dots + y_n) = \frac{1}{n} \sum y_i,$$

or the **variance** (the square of the standard deviation)

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

And, since standard deviation and variance both measure how much variation there is between observed values, so does the quantity $(n-1)s_y^2$, which we again call SST , or

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

We can describe this relationship between SST and variance by saying that variance is SST divided by number of degrees of freedom $n-1$.

The computed regression line with intercept b_0 and slope b_1 has been used by us back in Chapter 2 to compute **predicted**, also known as **fitted**, values. We use a “hat” to distinguish these values from the observed ones:

$$\hat{y}_1 = b_0 + b_1 x_1, \hat{y}_2 = b_0 + b_1 x_2, \dots, \hat{y}_n = b_0 + b_1 x_n.$$

It is perhaps not so surprising that these fitted values have the same mean as the observed ones—that is

$$\frac{1}{n}(\hat{y}_1 + \hat{y}_2 + \dots + \hat{y}_n) \quad \text{also equals } \bar{y}.$$

So, if we did the same type of sum-of-squares calculation as above, but using fitted values instead of observed ones, that would give a measure of the variability among fitted values. We will call that SSM , or

$$SSModel = \sum (\hat{y}_i - \bar{y})^2.$$

(called SSG in Ch. 8)
SSTreatment

Our fitted values all lie on the line, and the difference between an observed value y_i and its predicted value \hat{y}_i at x_i ,

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

is what we call a **residual**. You may remember that, in finding the best-fit line, we chose, out of all possible lines, the one that had the smallest possible sum-of-squares-of-residuals, sometimes called $SSResid$, or SSE (since *error* and *residual* are synonyms):

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

a.k.a. SSResiduals

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

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

SSE = 0 corresp.
to every obs $y = \hat{y}$
pred.

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
```

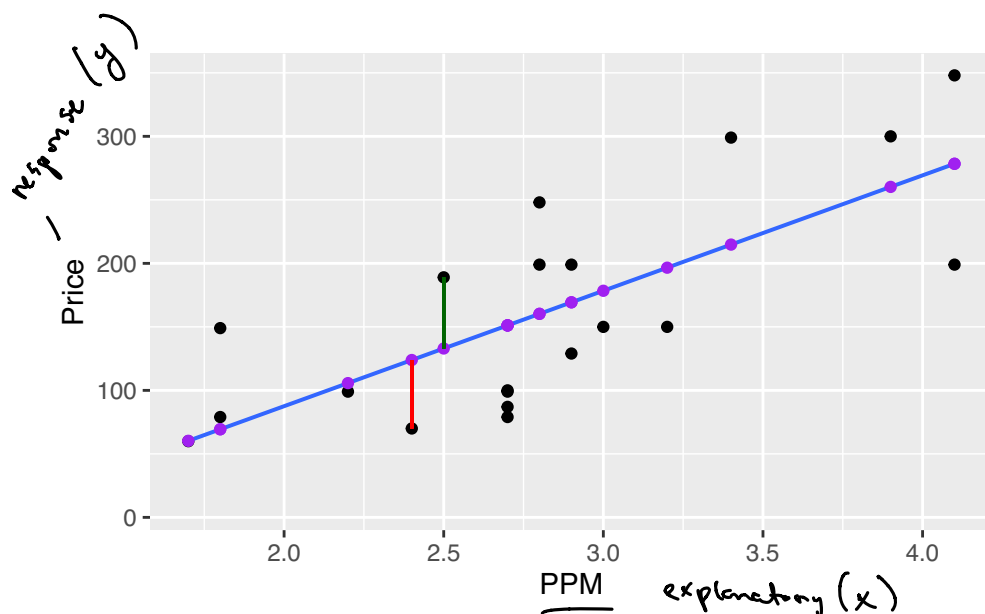
	1	2	3	4	5	6
## 39.7972965	29.6753642	-72.1490222	-40.3246358	-53.8856019	69.6216830	
## 7	8	9	10	11	12	
## 84.2363304	87.7631710	-28.4124425	-46.5880561	-64.1490222	-51.1490222	
## 13	14	15	16	17	18	
## -6.7099884	56.0265913	-52.1490222	-0.2709545	38.7631710	79.6412387	
## 19	20					
## 9.6412387	-79.3783170					

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 $70 - 123.89 = -53.89$.

$$x_1 = 3.9, \quad y_1 = 300$$

$$\hat{y}_1 = b_0 + b_1(3.9) = 260.203$$

Regression line
 $b_0 = -94.22, \quad b_1 = 90.88$



data frame

InkjetPrinters

black dots

(x_i, y_i)

observed values

purple dots

(x_i, \hat{y}_i)

fitted, predicted values

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
```

roughly 54.7% of variability in Price of our printers is explained by PPM

2x pt (4.663, df=18)

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

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

Coeff. of determination

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
```

$$\frac{SSM}{SSM+SSE} = \frac{74540}{74540+61697} = 0.547$$

area under theoretical F-dist.
to the right of our test stat
 $1 - pf(21.747, df1=1, df2=18)$

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:

$$MS_{Model} = \frac{SS_{Model}}{1}, \quad MSE = \frac{SSE}{n-2}, \quad \text{and} \quad F = \frac{MSM}{MSE},$$

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.

rejected $H_0: \beta_1 = 0$ ($p = 0$)
in favor of
there
being
an assoc.

⇒ my explanatory
variable is helpful
in predicting response

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

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

gives single number estimates used to answer both 1 and 2.

The best *single number* to

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

$$-94.22 + 90.88(3.5) = \underline{223.86},$$

or \$223.86.

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

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
```

(184.57, 263.13)

— used to
give, w/
95% conf.
an answer
to 1.

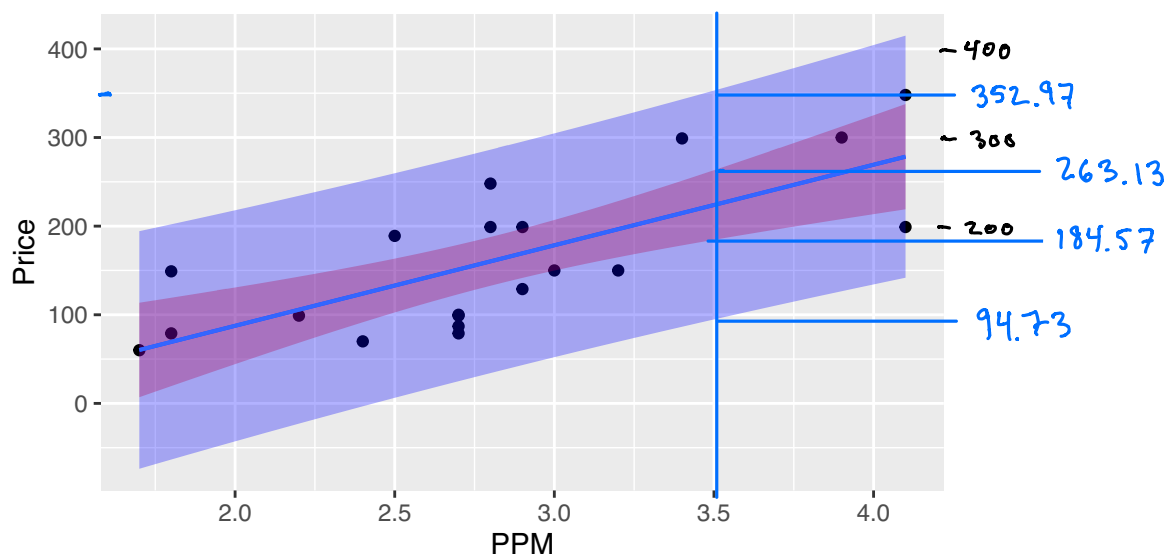
```
predict(lmResult, newdata=data.frame(PPM=3.5), interval="prediction", level = 0.95)
```

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

Produces a 95% prediction interval
(resp. to Question 2)

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.