Lab2: Linear Regression

Claire Jellison

9/11/2019

```
data(quakes)
str(quakes)
```

```
## 'data.frame':    1000 obs. of  5 variables:
##  $ lat     : num  -20.4 -20.6 -26 -18 -20.4 ...
##  $ long    : num  182 181 184 182 182 ...
##  $ depth   : int  562 650 42 626 649 195 82 194 211 622 ...
##  $ mag     : num  4.8 4.2 5.4 4.1 4 4 4.8 4.4 4.7 4.3 ...
##  $ stations: int  41 15 43 19 11 12 43 15 35 19 ...
```

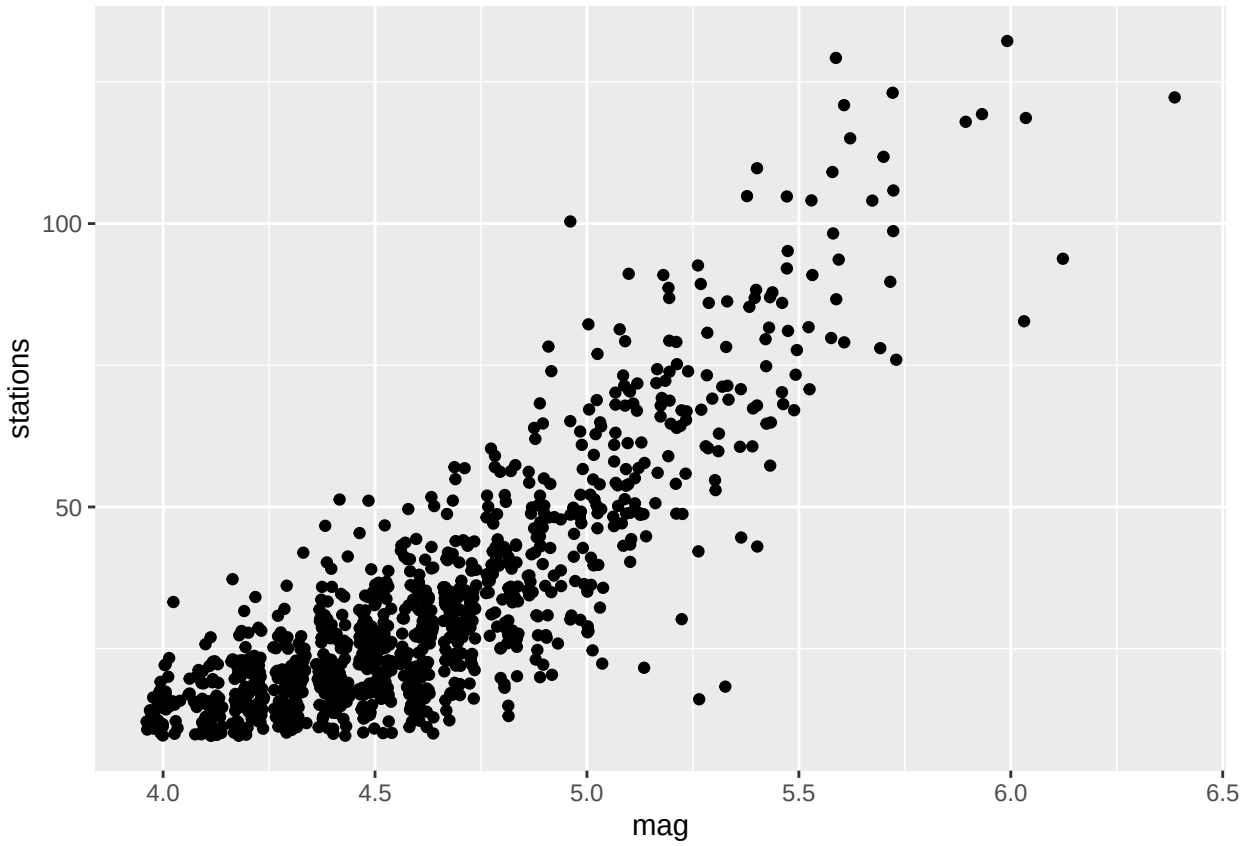
```
?quakes
```

Earthquake detection Included in the data set is a column recording the number of stations that detected each earthquake. This refers to a global network of seismographs and it stands to reason that the larger the quake, the more widely it will be detected.

Exercise 1

Create a plot of the relationship between stations and magnitude. How would you characterize the relationship? (If you see overplotting, you may want to add jitter to your points or make them transparent by playing with the alpha value.)

```
library(ggplot2)
ggplot(quakes, (aes(x = mag, y = stations))) + geom_point(position = "jitter")
```



It appears as though there is a fairly linear relationship between the magnitude of the earthquake and the number of stations that it is recorded at (though it could be slightly concave up), with the greater the magnitude the more stations detecting it.

Exercise 2

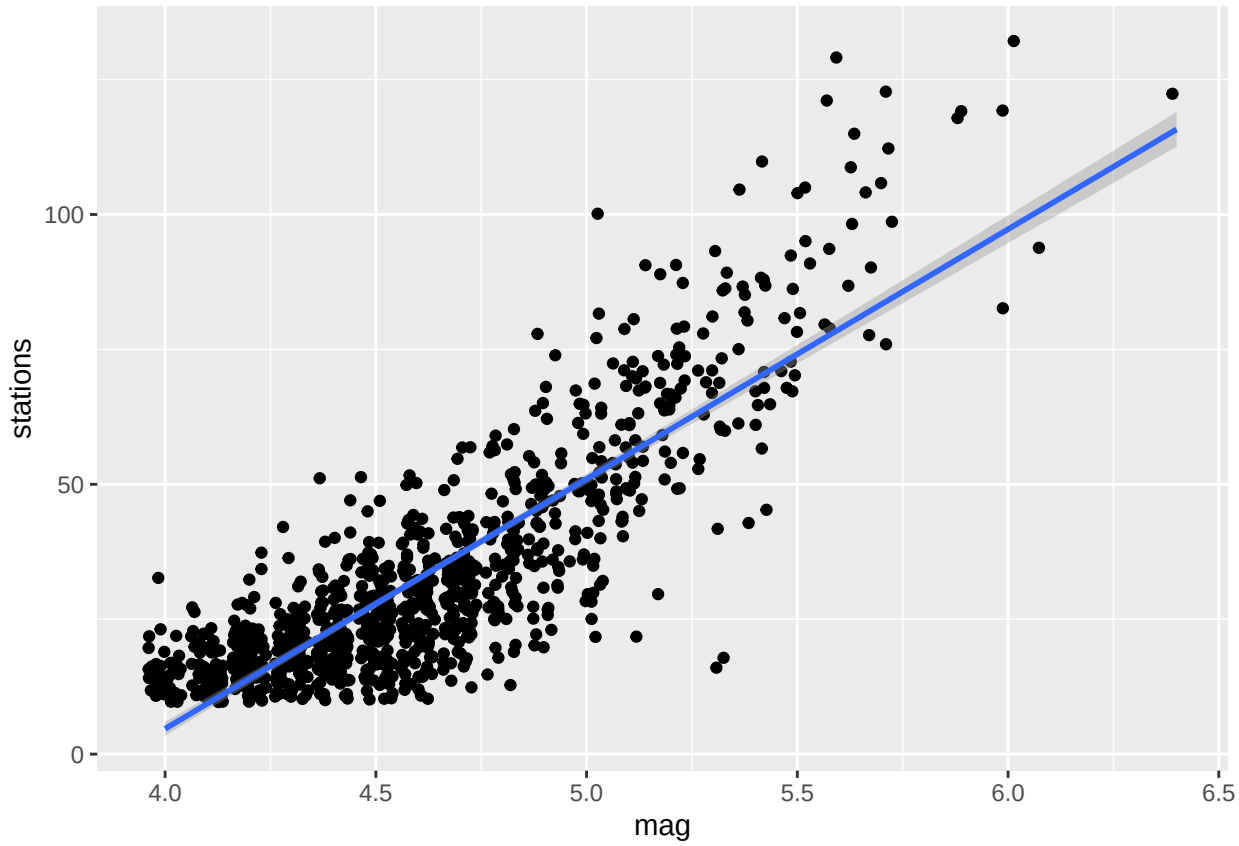
Before you go ahead and fit a linear model to this trend, if in fact there was no relationship between the two, what would you expect the slope to be? What about the intercept?

If there was no relationship between the two variables, we would expect the slope to be zero (since there is no positive or negative correlation between them) and the y intercept should be at the mean number of stations since the mean is a pretty optimal prediction given that the other variable doesn't really provide any useful information.

Exercise 3

Ok, now go ahead and fit a linear model called m1 to the trend and add that line to the plot from exercise 1. Interpret your slope and intercept in the context of the problem.

```
m1 <- lm(stations ~ mag, data = quakes)
ggplot(quakes, (aes(x = mag, y = stations))) + geom_point(position = "jitter") +
  geom_smooth(method = "lm")
```




```
summary(m1)
```

```
##
## Call:
## lm(formula = stations ~ mag, data = quakes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.871  -7.102  -0.474   6.783  50.244
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -180.4243     4.1899  -43.06   <2e-16 ***
## mag           46.2822     0.9034   51.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.5 on 998 degrees of freedom
## Multiple R-squared:  0.7245, Adjusted R-squared:  0.7242
## F-statistic:  2625 on 1 and 998 DF,  p-value: < 2.2e-16
```

Here, the y-intercept is around -180 stations indicating that for an earthquake of 0 magnitude -180 stations would detect it, clearly this is outside the realm of possibility and results from extrapolating to far outside the range of our data. The slope is about 46 indicating that for each additinal unit increase in magnitude the model predicts that 46 more stations will detect the quake.

Exercise 4

Verify the way that lm() has computed your slope correctly by using R to do the calculation using the equation for the slope based on X and Y.

```
x = quakes$mag
y = quakes$stations
slope <- (sd(y)/sd(x)) * cor(x,y)
slope <- m1$coefficients[2]
```

From these calculations, we see that the slope we found in the previous problem lie in between the interval above.

Exercise 5

Using R, calculate a 95% confidence interval for the slope of the model that you fit in exercise 3. Confirm the calculation using confint().

```
SE_slope <- summary(m1)$coef[2,2]
n <- nrow(quakes)
t_stat <- qt(.025, df = n-2)
LB <- slope - t_stat * SE_slope
UB <- slope + t_stat * SE_slope
c(LB, UB)
```

```
##      mag      mag
## 48.05498 44.50944
```

```
confint(m1, level = 0.95)
```

```
##                   2.5 %     97.5 %
## (Intercept) -188.64628 -172.20238
## mag           44.50944   48.05498
```

We see that the two confidence intervals for the slope have the same range.

Exercise 6

How many stations do you predict would be able to detect an earthquake of magnitude 7.0?

```
stations = -180.4243 + 46.2822*(7.0)
stations
```

```
## [1] 143.5511
```

The linear model would predict that around 144 stations would detect an earthquake of magnitude 7.0

Exercise 7

Questions 1 - 6 in this lab involve elements of data description, inference, and/or prediction. Which was the dominant goal in each question?

Question 1 focused on data description because we are just trying to get a visualization of what the data looks like. Question 2 focused on data description because we are considering whether the variables could actually be independent and what that would imply. Question 3 focused on inference because by creating a linear model we assess the relationship between the two variables. Question 4 focused on inference as well because we are examining the relationship between the variables. Question 5 focused on inference because we are assessing the goodness of fit of our model. Quesiton 6 focused on prediciton because we are predicting the number of stations based the linear model we constructed.

Simulation

One good way to assess whether your fitted model seems appropriate is to simulate data from it and see if it looks like the data that you observed. For the following questions it will be useful to reference the R code provided in the previous two lectures.

Exercise 9

Please simulate a data set that has the same number of observations as quakes. To start, generate a vector of x's. You can either generate your own x's or use the exact same x's as the quakes data.

```
n1 <- nrow(quakes)
n1
```

```
## [1] 1000
```

```
meanmag = mean(quakes$mag)
meanmag
```

```
## [1] 4.6204
```

```
stdmag = sd(quakes$mag)
stdmag
```

```
## [1] 0.402773
```

```
xsim = rnorm(n = n1, mean = meanmag, sd = stdmag)
xactual = quakes$mag
mean(xsim)
```

```
## [1] 4.606003
```

Exercise 10

Next, generate your

$$\hat{y}$$

's (the value of the mean function at the observed x's). Please generate them by writing your own function of the form:

```
x1 <- quakes$mag

f_hat1 <- function(x1) {
  f_hatgen <- 46.2822*(x1) -180.4243
  return(f_hatgen)
}

f_hatsim <- function(xsim) {
  f_hatgsim <- 46.2822*(xsim) -180.4243
  return(f_hatgsim)
  }

f_hat1(5)
```

```
## [1] 50.9867
```

```
f_hatsim(5)
```

```
## [1] 50.9867
```

Exercise 11

Now, generate the y's. Note that you'll need an estimate of

$$\sigma^2$$

, for which you can use

$$\hat{\sigma}^2 = RSS/n - 2$$

. You can extract the vector of residuals with m1$res.

```
sqresid <- m1$res^2
var_hat <- sum(sqresid)/(1000 - 2)
stdev = sqrt(var_hat)
error <- rnorm(n = 1000, mean = 0, sd = stdev)
mean(error)
```

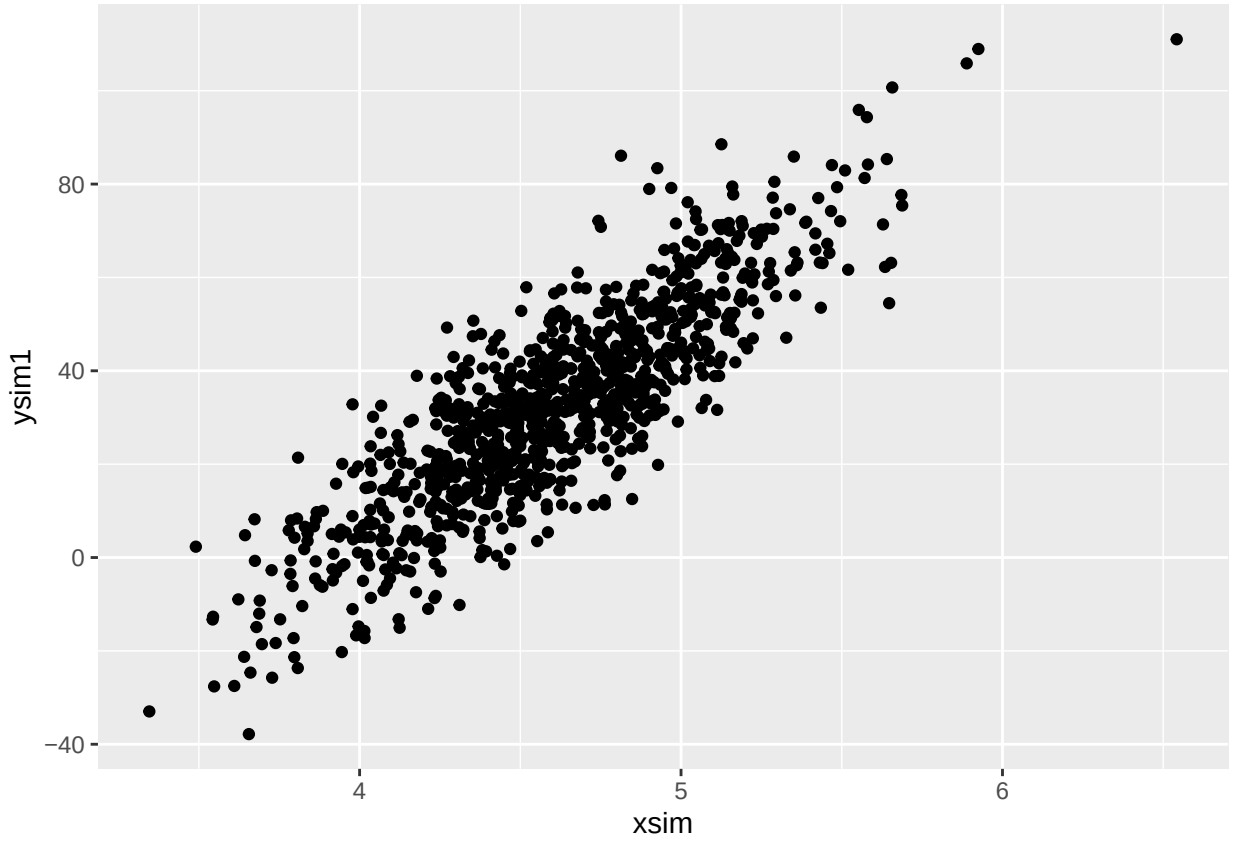
```
## [1] -0.1269407
```

```
ysim1 = f_hat1(xsim) + error
ysim2 = f_hat1(x1) + error
```

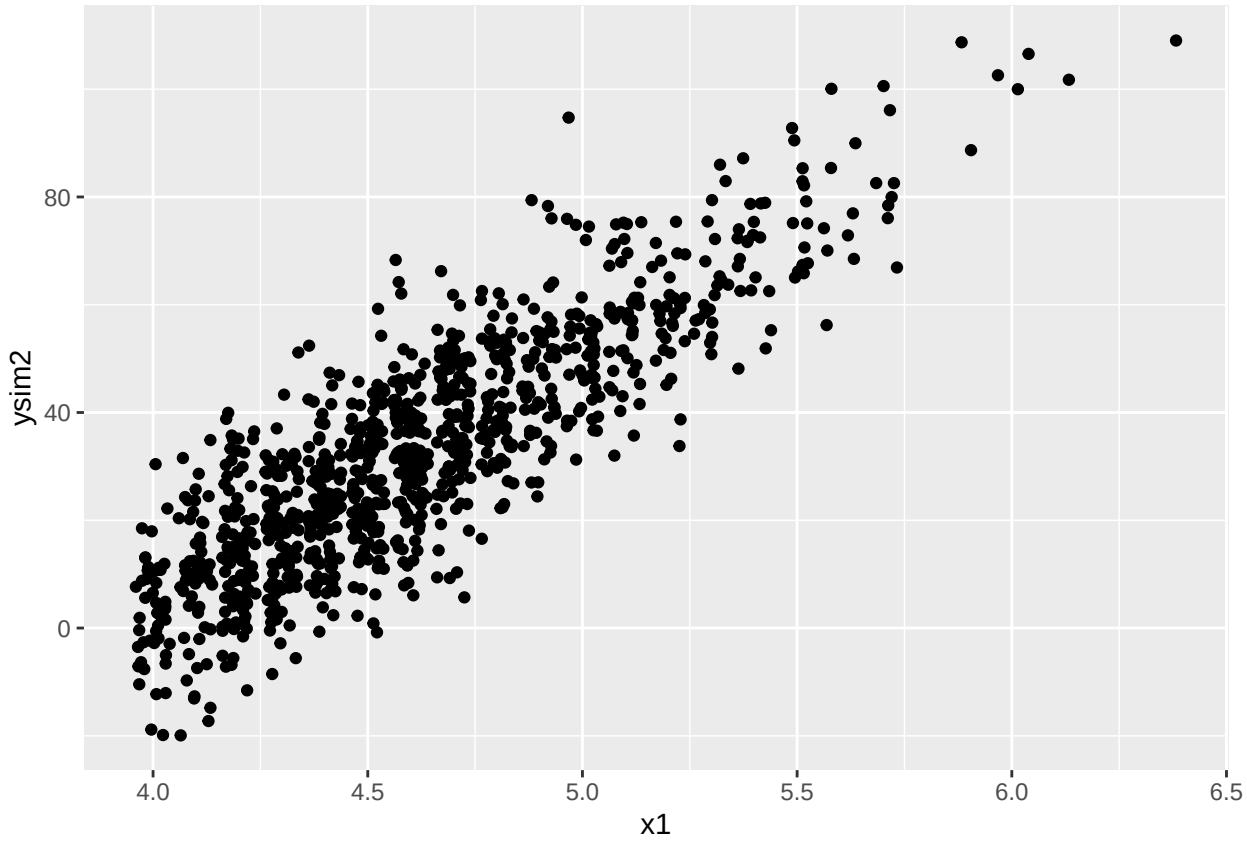
Exercise 12

Finally, make a plot of your simulated data. How is it similar to the original data? How is it different? How might you change your model to make it more consistent with the data?

```
ggplot(quakes, (aes(x = xsim, y = ysim1))) + geom_point(position = "jitter")
```



```
ggplot(quakes, (aes(x = x1, y = ysim2))) + geom_point(position = "jitter")
```

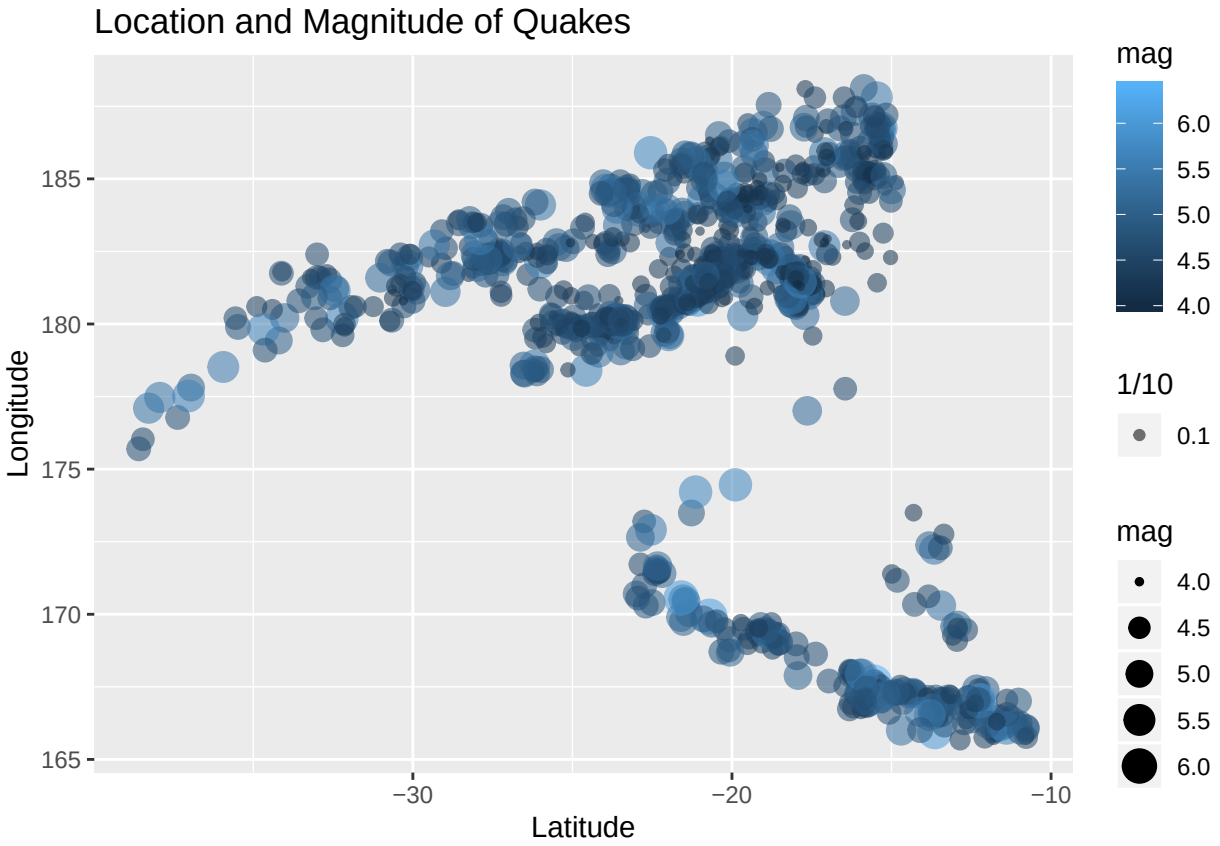


In the original data there are far more observations around the lower magnitudes.In general, the data appears more linear here, than originally unsurprisingly. The residuals also seem more constant across the magnitude values in the simulated data than in the original data.

Challenge Problem

Use the latitude and longitude data to plot each of these earthquakes in quakes on a map with their magnitude mapped to the size of the plotting character. You may need to add some transparency to prevent overplotting.

```
ggplot(quakes, aes(x=lat, y=long)) + geom_point(aes(size = mag, colour = mag, alpha = 1/10)) + labs(tit
```



Problem Set 2

CH3 Exercise 1

The null hypothesis for TV is that for an increase in spending in TV advertising there is no effect on sales. The null hypothesis for radio is that for an increase in spending in radio advertising there is no effect on sales. The null hypothesis for newspaper is that for an increase in spending in newspaper advertising there is no effect on sales. Based on the p values, at the 99% confidence level we can reject the null hypothses that the intercept, TV advertising spending, and Radio advertising spending do not have an effect on sales. However, the much higher p value for newspaper, does not allow us to reject the null hypothesis that a change in spending on newspaper advertising has no effect on sales.

CH3 Exercise 4

I collect a set of data (n = 100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression.

(a) Suppose that the true relationship between X and Y is linear. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

There is not enough information to be entirely sure. Since the true relationship is linear it is possible that the linear model could have a lower RSS, despite the fact that it is the less flexible model. Then again, the cubic regression could overfit the data in a way that could result in a lower RSS depite the true linear relationship.

(b) Answer (a) using test rather than training RSS.

With the test, RSS we would expect the linear model to have a lower RSS. THis is because the cubic model likely overfit the training data especially outliers, resulting in a higher RSS with the test data.

(c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

We would probably expect the training RSS for the cubic regression to be lower since it has more flexibility to curve and get closer to the data points, while the linear model is very rigid.

(d) Answer (c) using test rather than training RSS.

Given that the relationship is not linear, I would expect the test RSS for the cubic regression to be lower, since it has more ability to conform to the true relationship whatever that may be.

CH3 Exercise 5

$$\hat{y} = (\sum_{i=1}^{n} x_i y_i / (\sum_{i^1=1}^{n} x_i^2)) x_i \quad = (x_i x_1 y_1 + x_i x_2 y_2 + \cdots + x_i x_n y_n)/(x_1^2 + x_2^2 + \cdots + x_n^2) = \sum_{i^1=1}^{n} (\frac{x_i \sum_{i=1}^{n} x_i}{x_{i^1}^2}) y_{i^1} = \sum_{i^1=1}^{n} a_{i^1} y_{i^1}$$

Expanding the two sums, we see that we can rearrange them as shown above. So a_i is equal to everything inside the parenthesis in the second to the right equality.

Additional Exercise

The k-nearest neighbor regression was defined as:

$$[\hat{f}(x) = \frac{1}{k} \sum_{x_i \in \mathcal{N}(x)} y_i]$$

$$MSE = E[(y - \hat{f})^2] = Var(\hat{f}) + [f - E(\hat{f})]^2 + Var(\epsilon)$$

We will break up the right hand side into each of the parts.

$$Var(\hat{f}) = \frac{\sigma^2}{k}$$

This is by the definition of the variance of the sample mean. We see from the equation above that it is decreasing in k.

$$[f - E(\hat{f})]^2 = [f - E(\frac{1}{k} \sum_{x_i \in \mathcal{N}(x)} y_i)]^2 = [f - (\frac{1}{k} \sum_{x_i \in \mathcal{N}(x)} E(y_i)]^2 = [f - (\frac{1}{k} \sum_{x_i \in \mathcal{N}(x)} f(x_i)]^2$$

Since the inside of the sum is constant the whole thing is incerasing in k, since as k increases the total quantity goes up.

$$Var(\epsilon) = \sigma^2$$

$$MSE = \frac{\sigma^2}{k} + [f - (\frac{1}{k} \sum_{x_i \in \mathcal{N}(x)} f(x_i)]^2 + \sigma^2$$

One you have a form for each of these terms, construct a plot that shows the decomposition of the MSE into its components as a function of (k) (this is a formal version of your sketch from the handout). Use the following as your training data set:

```
x <- c(1:3, 5:12)
y <- c(-7.1, -7.1, .5, -3.6, -2, -1.7, -4, -.2, -1.2, -1.2, -3.5)

sigmasqr = (sd(y))^2
sigmasqr
```

```
## [1] 6.416182
```

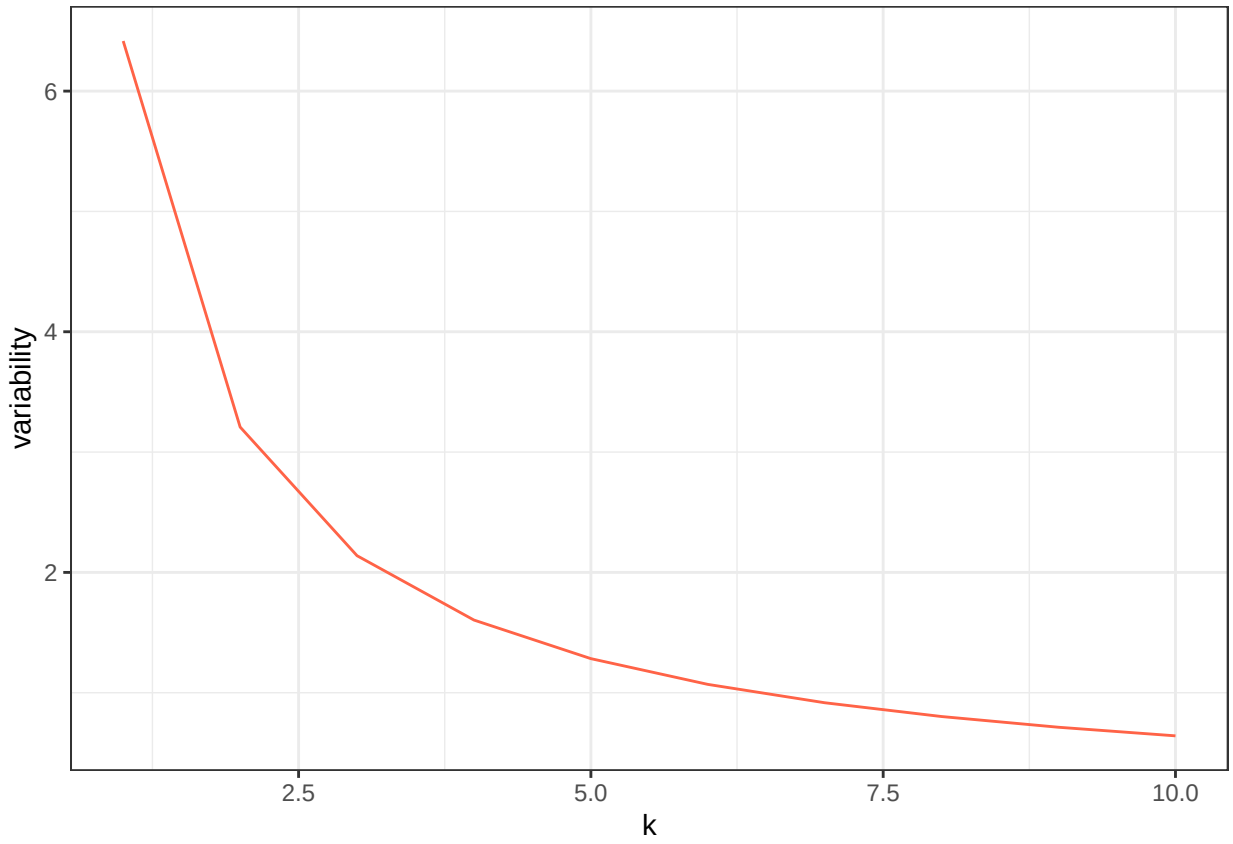
```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v tibble  2.1.1       v purrr   0.3.2
## v tidyr   0.8.3       v dplyr   0.8.0.1
## v readr   1.3.1       v stringr 1.4.0
## v tibble  2.1.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

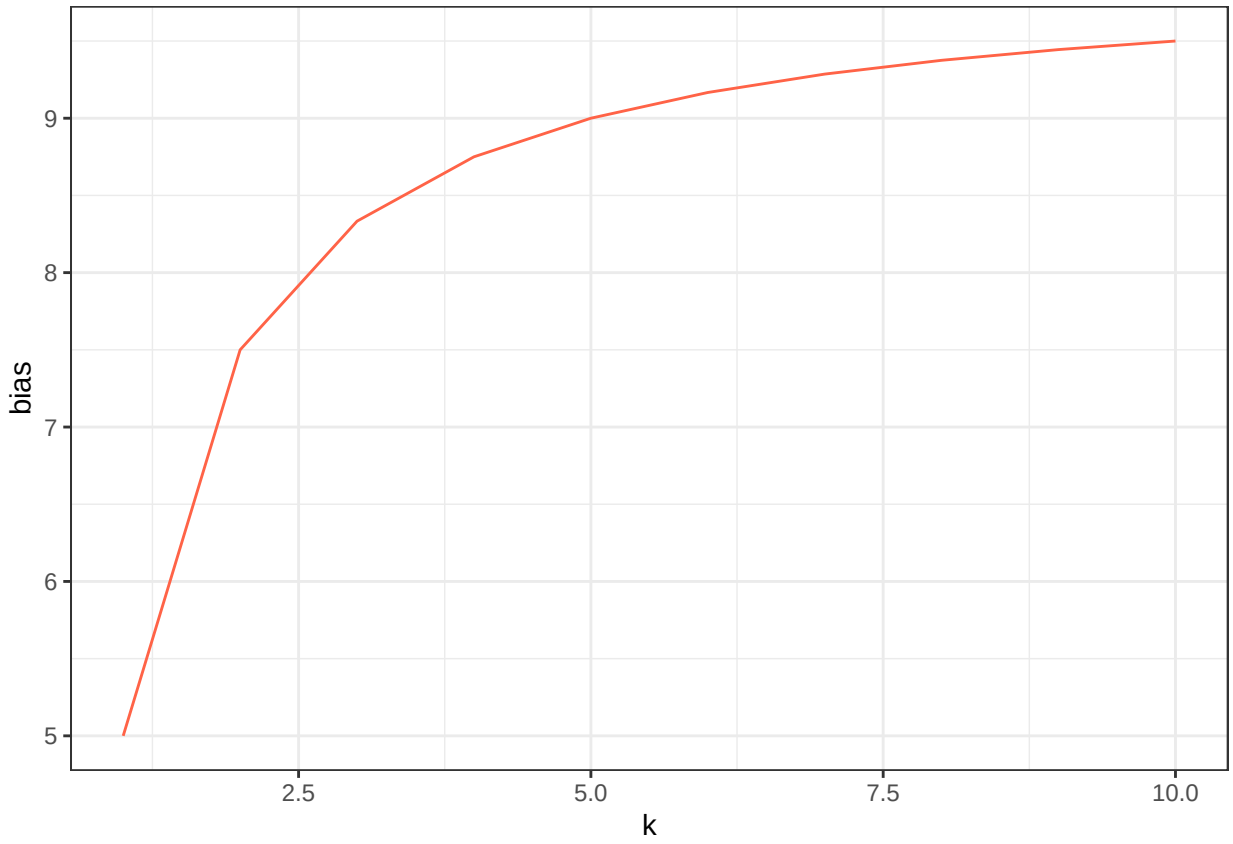
```
my_fun <- function(k, x, y) {
  f_k <- rep(NA, length(k))
  for (i in 1:length(k)) {
    f_k[i] <- sigmasqr/k[i]
  }
  f_k
}

k <- 1:10
f_k <- my_fun(k, x, y)

df <- tibble(k = k, f_k = f_k)
ggplot(df, aes(x = k, y = f_k)) +
  geom_line(col = "tomato") +
  theme_bw() +
  ylab("variability")
```




```
x <- c(1:3, 5:12)
y <- c(-7.1, -7.1, .5, -3.6, -2, -1.7, -4, -.2, -1.2, -1.2, -3.5)

sigmasqr = (sd(y))^2
sigmasqr
```

```
## [1] 6.416182
```

```
library(tidyverse)
my_fun <- function(k, x, y) {
  f_k <- rep(NA, length(k))
  for (i in 1:length(k)) {
    f_k[i] <- 10 - (1/k[i])*5 #this eqn isn't quite right I'm just using constants
  }
  f_k
}

k <- 1:10
f_k <- my_fun(k, x, y)

df <- tibble(k = k, f_k = f_k)
ggplot(df, aes(x = k, y = f_k)) +
  geom_line(col = "tomato") +
  theme_bw() +
  ylab("bias")
```


bias
9
8
7
6
5
2.5
5.0
7.5
10.0
k