Lab 2: Linear Regression

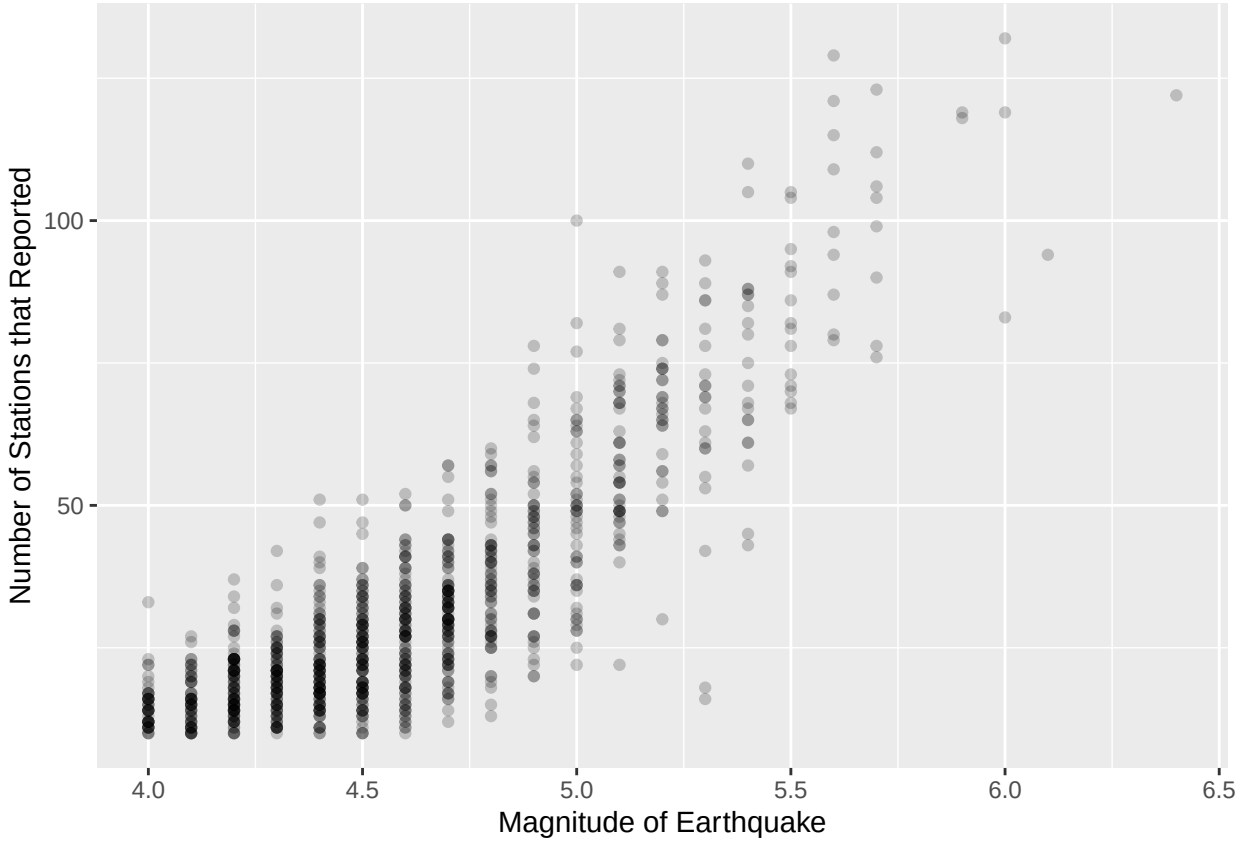
An Island Never Cries

Alyssa Andrichik

Exercise 1

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.)

```
ggplot(data = quakes, mapping = aes(x = mag,
y = stations)) +
geom_point(alpha = 0.2) +
  labs(x = "Magnitude of Earthquake", y = "Number of Stations that Reported")
```



There is a positive correlation between the number of stations reporting and the Magnitude of the Earthquake. This makes sense because the larger the earthquake, the greater amount of land mass it effects, and thus the more stations will feel the quake enough to report it.

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 Magnitude of the earthquake and the number of stations reporting, I would expect the slope of the data to be 0 since the data points would be spread sporadically, but evenly, on the plot. The y-intercept would be around the mean of the number of stations recording.

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.

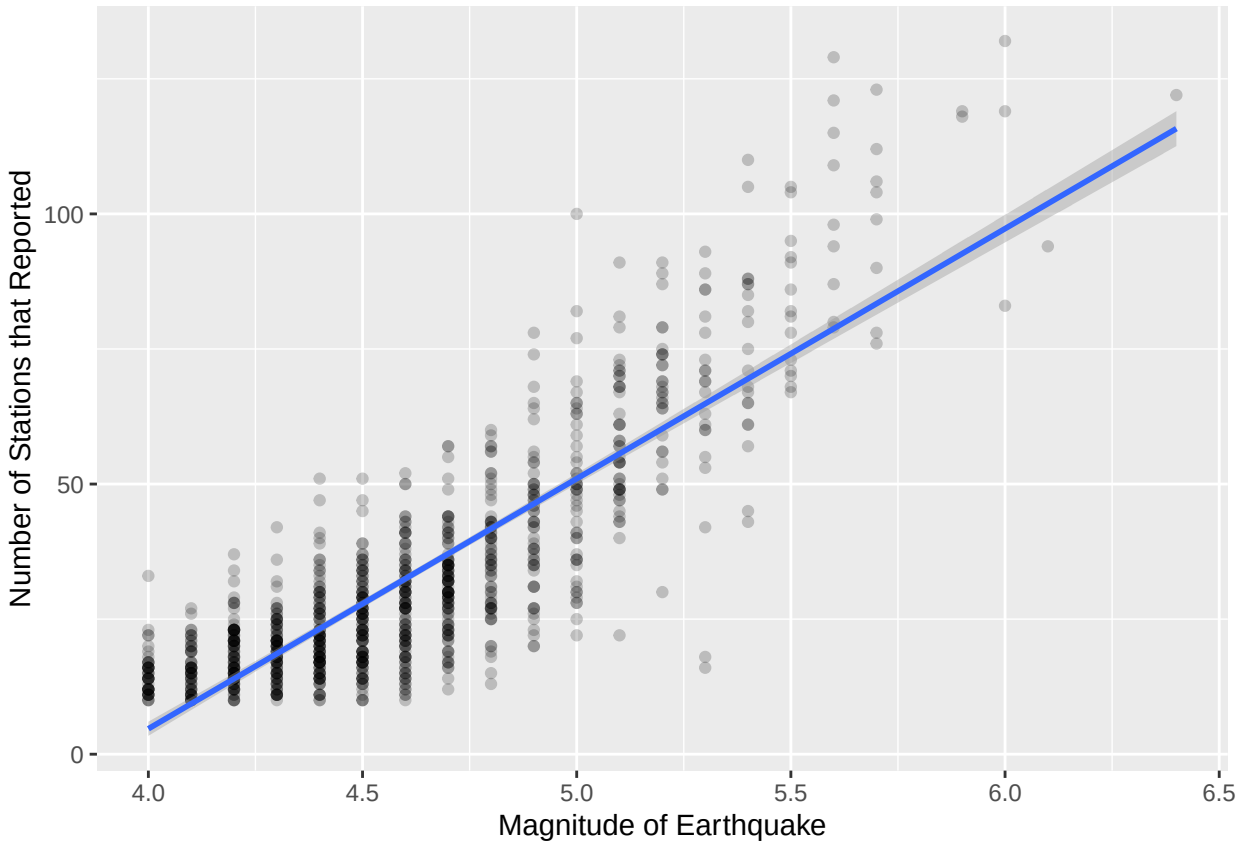
```
m1 <- lm(stations ~ mag, data = quakes)
m1
```

```
##
## Call:
## lm(formula = stations ~ mag, data = quakes)
##
## Coefficients:
## (Intercept)          mag
##     -180.42        46.28
```

```
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
```

```
ggplot(data = quakes, mapping = aes(x = mag,
y = stations)) +
geom_point(alpha = 0.2) +
  labs(x = "Magnitude of Earthquake", y = "Number of Stations that Reported") +
  geom_smooth(method = "lm")
```



The slope is 46.28. Every increase in 1 point of magnitude, the average number of stations that recorded that earthquake goes up 46.28. The y-intercept is -180.42. This would mean that the average number of stations recording an earthquake of 0 magnitude would be -180.42. This is obviously impossible because one cannot observe a negative number of stations recording. And also, an earthquake of 0 magnitude is not an earthquake at all. The reason for this discrepancy of the linear model is because the data has only recordings of a magnitude of 4 or higher. This might be because the scientific instruments required to detect earthquakes cannot detect vibrations that are lower than a 4.0 magnitude.

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.

```
n <- nrow(quakes)
x <- quakes$mag
y <- quakes$stations
slope <- sd(y)/sd(x) * cor(x,y)
slope
```

```
## [1] 46.28221
```

The lm() calculation matches the equation calculation!

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().

```
m1$coefficients[2]
```

```
##      mag
## 46.28221
```

```
SE_slope <- summary(m1)$coef[2,2]
SE_slope
```

```
## [1] 0.9033955
```

```
t_stat <- qt(.025, df = n - 2)
t_stat
```

```
## [1] -1.962344
```

```
LB <- slope + t_stat * SE_slope
LB
```

```
## [1] 44.50944
```

```
UB <- slope - t_stat * SE_slope
UB
```

```
## [1] 48.05498
```

```
c(LB, UB)
```

```
## [1] 44.50944 48.05498
```

```
confint(m1)
```

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

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

```
predict(m1, data.frame(mag = 7))
```

```
##        1
## 143.5511
```

R calculates that around 143.6, rounded up to 144, stations would be to calculate an earthquake with a magnitude of 7.0. This, however, is based on a data set where the largest magnitude is not even 6.5, so the model does not account for a larger earthquake than that. This estimate is assuming that it is a linear correlation between magnitude and number of stations recording. This is most likely an exponential trend.

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's goal was data description. Question 2's goal was data description. Question 3's goal was data description. Question 4's goal was to interpret. Question 5's goal was to interpret. Question 6's goal was to predict.

Simulation

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.

```
simulation <- rnorm(1000)
```

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:

```
f_hat <- mean(x)
f_hat
```

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

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

$$(\sigma^2),$$

for which you can use

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

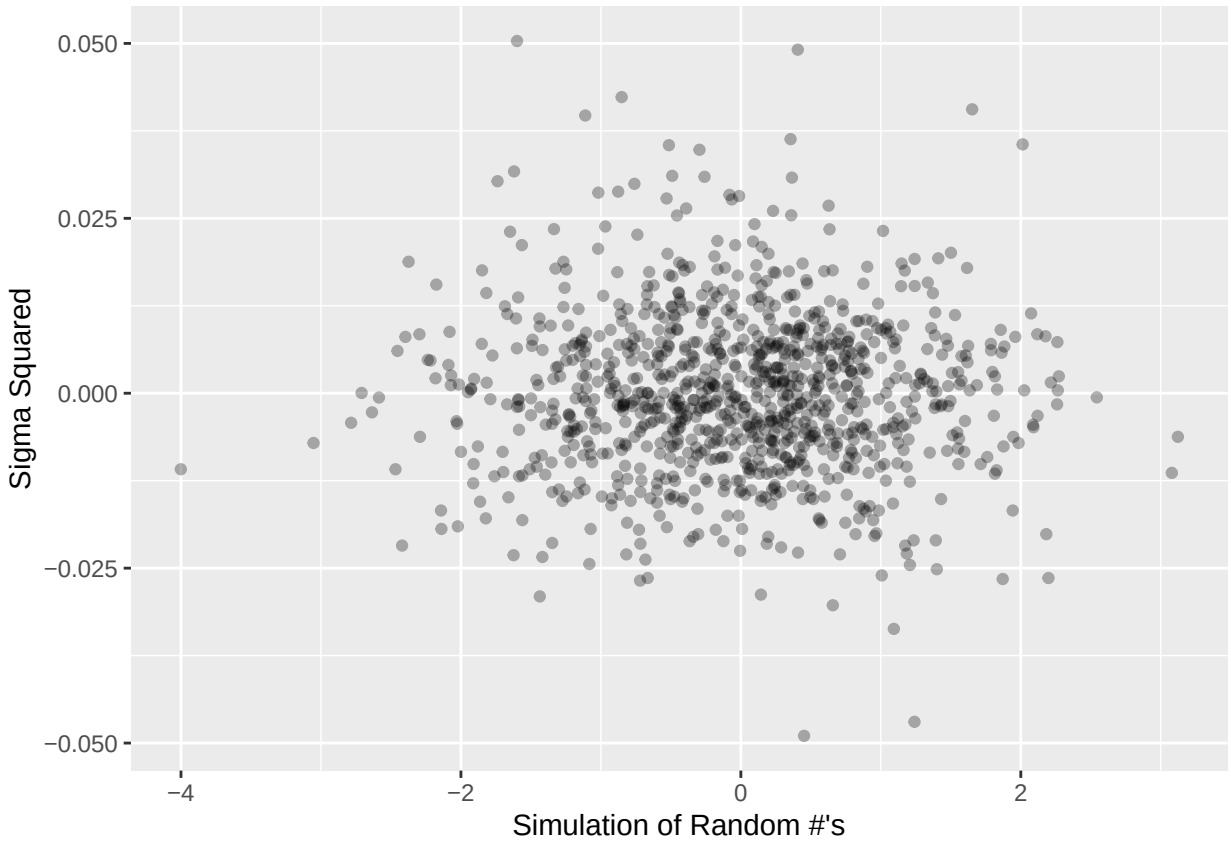
You can extract the vector of residuals with m1$res.

```
sigma_sq <- m1$res/(n-2)
```

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?

```
simplot <- data.frame(x = simulation, y = sigma_sq)

ggplot(data = simplot, mapping = aes(x = simulation,
y = sigma_sq)) +
geom_point(alpha = 0.3) +
  labs(x = "Simulation of Random #'s", y = "Sigma Squared")
```



The simulation and the original data have the same number of observations (1000), but, besides that, the data sets are very different. The range of both the X and Y are different. The original data set had no negatives for either variable while the simulated data has nearly an even distribution of positive and negative. There seems to be no correlation at all with the simulated data, except perhaps that it is more concentrated around the point (0,0), while the original data has a clear positive correlation. I might change my simulated model by giving it parameters that match the original data set. Match the ranges and make it the same function.

Problem Set #2

1. Describe the null hyptheses to which the p-values given in Table 3.4 correspond. Explain what conclusions you can draw based on these p-values. Your explanaiton should be phrased in terms of sales, TV, radio, and newspaper, rather than in terms of the coefficients of the linear model.

The null hypothesis is that no combination of the units sold of radio, TV, or newspapers will have an effect on the sales of a product. The p-values for the intercept, TV, and radio are <0.0001 when all other variable are held constant, which means that it is a very small probability that this data was due to chance and does indeed have an effect on sales. This means that we can reject the null hypothesis. Newspaper on the other hand, when all other variables are held constant, has a p-value that is above 0.05, which means that there is a high chance that the results were by chance and that the number of sales is not effected by newspaper advertising.

4.

(a) We would expect that the training RSS to be higher for the liner regression since the true relationship is linear. The training RSS for the cubic regression would then be lower than the linear because it is

likely there would be excess noise in the cubic regression.

(b) We would expect that the test RSS would be lower for the cubic regression in relation to linear regression as well. The overfit would lead to more error than in the linear regression.

(c) There is not enough information to know for sure, but it is most likely that a cubic regression–it being more flexible than a linear regression–would be lower. The cubic regression is more likely to follow the points more closely and reduce the train RSS.

(d) "The true relationship between X and Y is not linear, but we don't know how far it is from linear" is not enough information to know whether linear or cubic regression would have a lower test RSS. It is impossible to know which flexbility of either regression would fit better.

5.Consider the fitted values that result from performing linear regression without an intercept. In this setting, the ith fitted value takes the form where

$$\hat{y}[i] = x[i]\hat{\beta}[i]$$

where

$$\hat{\beta} = (\sum i = 1- > n(x[i]y[i]))/(\sum j = 1- > n(x[j]^2)).$$

Show that we can write

$$\hat{y}[i] = \sum j = 1- > n(a[j]y[j]).$$

What is

$$a[j]?$$

My awnser:

$$\hat{y}[i] = x[i] * (\sum i = 1- > n(x[i]y[i]))/(\sum j = 1- > n(x[j]^2))$$

$$\hat{y}[i] = x[i] * (\sum i = 1- > n((x[i]y[i])/x[i]^2))$$

$$\hat{y}[i] = x[i] * (\sum i = 1- > n((1/x[i]^2) * x[i] * y[i])$$

$$\hat{y}[i] = \sum i = 1- > n((1/n) * x[i] * (1/x[i]^2) * x[i] * y[i])$$

$$\hat{y}[i] = \sum j = 1- > n(a[j]y[j])$$

where

$$a[j] = (1/n) * x[j] * (1/x[j]^2) * x[j]$$

$$a[j] = x[j] * (1/n * *x[j]^2) * x[j]$$

$$a[j] = (1/n * *x[j]^2) * x[j]^2$$

$$a[j] = (x[j]^2)/(n * *x[j]^2)$$

$$a[j] = 1/n$$

Additional Exercise

The k-nearest neighbor regression was defined as:

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

This is one of the few models that has a closed form for the bias and the variance. Calculate each and use them to write a decomposition of the expected test MSE its three components. This is an eminently google-able result, but doing so will teach you nothing, so work to do this yourself (or working with each other). It's not an extensive derivation at all, you just have to be sure you're careful in your notation. If you're new to , draw examples from the source files on the website. Please post to slack if you have questions.

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:

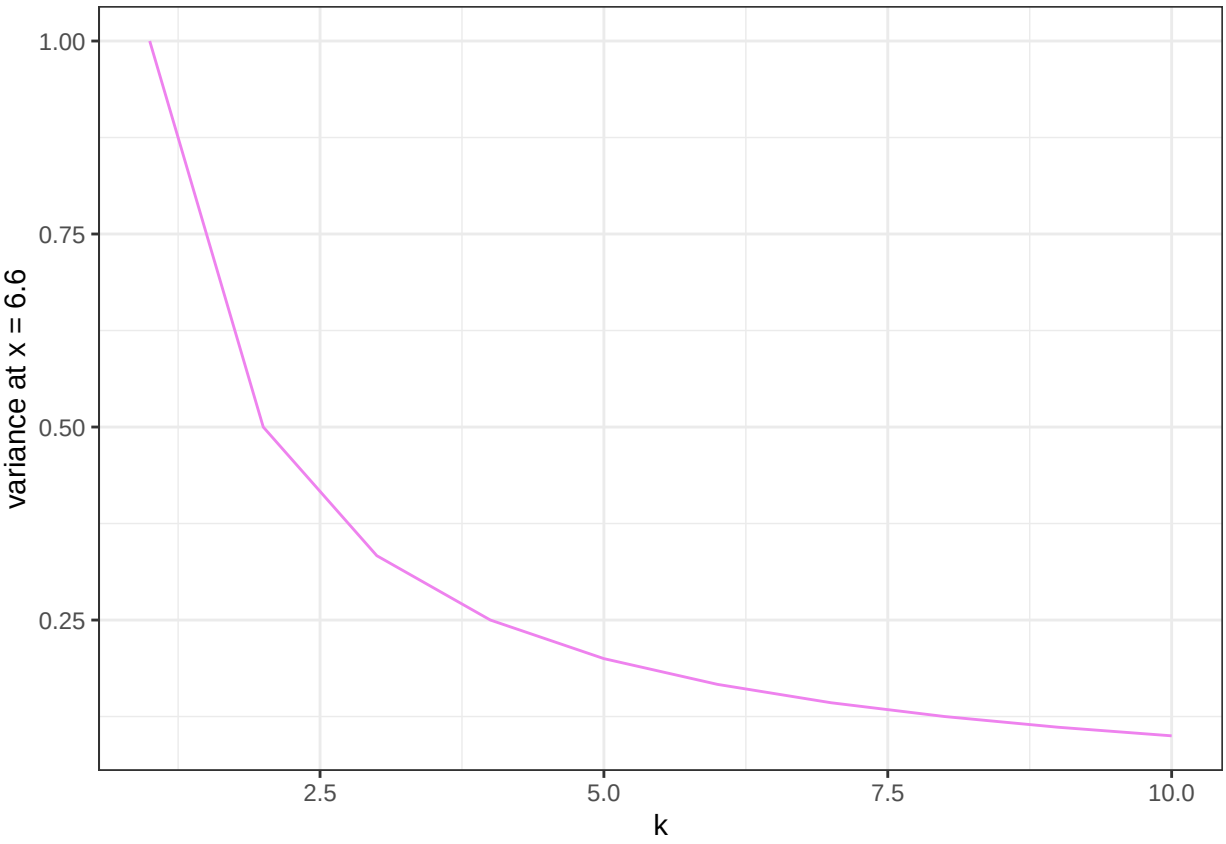
```
x <- c(1:3, 5:12)
y <- c(-7.1, -7.1, .5, -3.6, -2, -1.7, -4, -.2, -1.2, -1.2, -3.5)
```

And the following example of how to plot a function of (k) (with the training data fixed).

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

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

```
variance <- tibble(k = k, f_k = f_k)
ggplot(variance, aes(x = k, y = 1/k)) +
  geom_line(col = "violet") +
  theme_bw() +
  ylab("variance at x = 6.6")
```


1.00
0.75
0.50
0.25
variance at x = 6.6
2.5
5.0
7.5
10.0
k