Section 11
Multiple Linear Regression

11.1 – Simulation in Linear Regression

Before we dive into multiple linear regression, our main topic for this section, let's revisit what we did in class this week with the simulation for heart rate and caffeine, and try seeing what that might look like in R as well.

Hopefully you drew similarities between what you saw in simulating data under the null hypothesis for a difference in two means/proportions and for linear regression! Both involve looking at two variables, and seeing how re-pairing values from each variable in some way might impact the strength of the relationship. Before, we measured the strength of the relationship by how different the means or proportions were, as if there is a meaningful relationship, say, between chunking letters and the number of letters memorized, the difference in means would be large.

Now, we have two numerical variables that we want to understand the relationship between. To summarize how strongly related they are, we used the slope value, as a steeper slope would indicate a stronger relationship. Simulation now allows us to rule out whether the slope we got could have just been a product of random chance, or if it is representing a meaningful relationship, like between the amount of caffeine given and the change in heart rate.

Let's revisit this scenario through the data in **caffeine.csv**. To remind ourselves, let's find the slope and intercept for the least squares line again!

```
model = lm(hr_change ~ caffeine, data=caffeine)
summary(model)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.51649    2.39681   1.050   0.2990
caffeine     0.03802    0.01636   2.324   0.0244 *
```

Thus, we have our actual least squares line of $y = 2.51649 + 0.03802x$. We'd like to simulate now what might happen by random chance by re-simulating the random assignment process that the researchers of this caffeine study used, by randomly assigning different amounts of caffeine to each of the participants. By assuming the null hypothesis is true, we can say that the changes in heart rate represent the participants themselves, as if there was no effect of caffeine, they'd exhibit the same change in heart rate.

```
caff_rand = sample(caffeine$caffeine, 50, replace=F)
hr_rand = sample(caffeine$hr_change, 50, replace=F)
data_rand = data.frame(caff_rand, hr_rand)
```

With this randomized data, we want to collect the slope value for the least squares line that we would fit to the data.

```
m_rand = lm(hr_rand~caff_rand, data=data_rand)
m_rand$coefficients["caff_rand"]
```

And with this structure to generate a randomized slope, we can use our for-loop structure to simulate many possible randomly generated slopes!

```
slopes = rep(0, 1000)
for (i in 1:1000) {
  caff_rand = sample(caffeine$caffeine, 50, replace=F)
  hr_rand = sample(caffeine$hr_change, 50, replace=F)
  data_rand = data.frame(caff_rand, hr_rand)
  m_rand = lm(hr_rand~caff_rand, data=data_rand)
  slopes[i] = m_rand$coefficients["caff_rand"]
}

hist(slopes)
abline(v=0.03802, col="red")
mean(slopes >= 0.03802 | slopes <= -0.03802)
```

The simulated *p*-value we get here lines up pretty well with the value we got from the *t*-test!

11.2 – Multiple Linear Regression

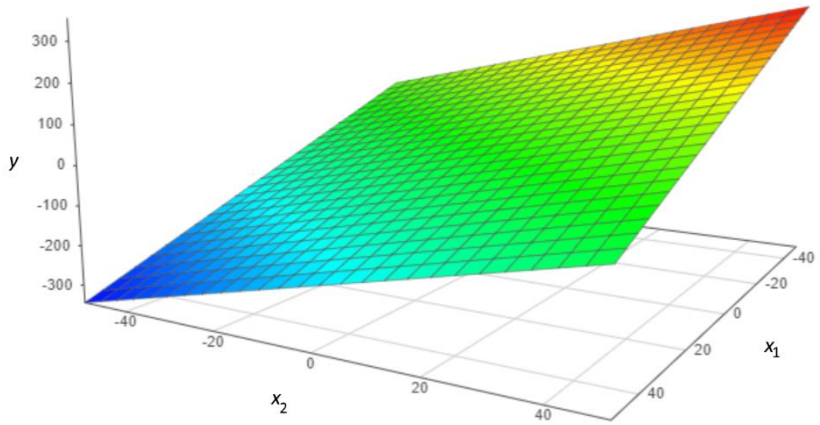
A linear model for multiple predictors

We need not limit ourselves to just one predictor variable in a linear model – often, it is more useful to consider a multitude of predictor variables. This can allow for a narrower prediction, as well as allow for the use of categorical variables as predictors.

If we have a data set with a response variable, represented by data $y_1, y_2, \ldots, y_n$, as well as a collection of other variables that are used to predict this variable where one case in the data set looks like $(x_{1,i}, x_{2,i}, \ldots x_{n,i})$, we can write out the multiple linear regression model as:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_k x_{i,k} + \varepsilon_i$$

In essence, we are now working with multiple "slopes" for each variable that we are trying to predict. In the case of two predictor variables, this could visually be represented by a plane of best fit for the three variables y, x_1 and x_2. To the right is a visualization of an example plane: $y = 5-4x_1+3x_2$.



Using R for multiple linear regression

The act of creating a multiple linear regression model is not just as simple as throwing a bunch of variables into a model. Sometimes it might actually make sense to remove variables from your model to improve it. It's hard to come up with well-defined rules for these sorts of things, so let's start by working through an example.

Example: Measuring the body fat of a human can be done with electronic instruments, but what if one wanted to estimate their body fat based on other measurements that can be done with simple tools on the body? Three such simple measurements that are often associated with body fat are a skinfold measurement, thigh circumference, and midarm circumference. Data on measurements of 20 people can be found in **body.csv**. Create a multiple linear regression model that can predict one's body fat based on these simple measurements, then predict the body fat for a person who has a skinfold measurement of 24, thigh circumference of 42, and arm circumference of 23.

```
pairs(body)
bmodel1 = lm(bf~skinfold+thigh_circ+arm_circ, data=body)
bmodel2 = lm(bf~skinfold+arm_circ, data=body)
bmodel3 = lm(bf~skinfold+arm_circ+skinfold:arm_circ, data=body)
```

When checking for the assumption of constant residuals in multiple regression, rather than checking against each predictor variable for constant variance, we check against the predicted/fitted values in the model. There is a built-in functionality that does this, simply by plotting the model in R.

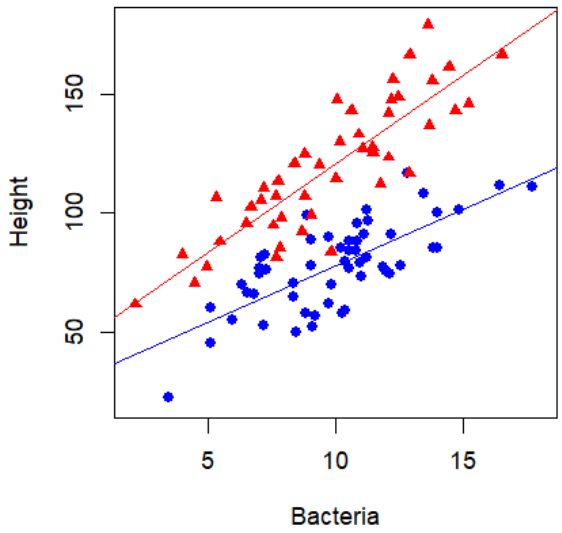
```
plot(bmodel2)
```

Lessons learned from this example:

- The idea of _____ is when two predictor variables in a linear regression model have a strong relationship with each other. This can cause issues with interpreting coefficients of the model, as they tend to "fight" each other for importance in predicting the response variable.
- Creating a multiple linear regression model is a balancing act of having enough predictors to explain your response variable without being too complicated. Achieving this goal would produce a _____ model.
- When working with multiple predictor variables, there may be variables that in tandem produce an effect that is larger than the sum of their parts. While we did not see this in this example, such an effect would be called an _____. Let's examine an example of this now!

Example: A study was done on a type of shrub to determine factors that impact its height. Each shrub had its soil examined for the thousands of bacteria per mL of soil, and it was also noted whether the shrub was in direct or partial sunlight. Create a multiple linear regression model that predicts the height of the plant using the **plants.csv** data file. Interpret the interaction term that is used in the model.

```
plot(plants$bacteria, plants$height)
pmodel1 = lm(height~sunlight+bacteria, data=plants)
pmodel2 = lm(height~sunlight+bacteria+sunlight:bacteria, data=plants)
```



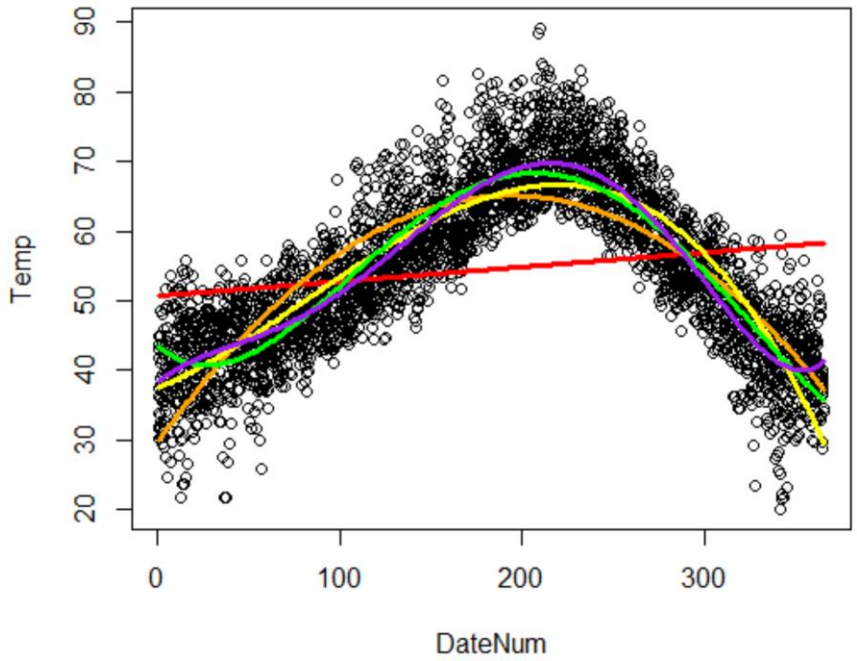
Polynomial Terms

It's important to keep in mind that all of our methods only apply to linear relationships. However, we can apply methods that allow us to fit quadratic (and beyond!) terms to our model through an extension of multiple linear regression. We simply just define new variables as powers of existing ones. This results in a linear regression model defined as:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_n x_i^n + \varepsilon_i$$

Example: Data over the last 10 years (2009-2018) recorded at the weather station at Portland International Airport are provided in the data file **pdxweather.csv**. Create a polynomial linear regression model that predicts the average daily temperature for a given day in the year.

```
plot(pdxweather$DateNum, pdxweather$Temp)
wmodel1 = lm(Temp~poly(DateNum, 2, raw=T), data=pdxweather)
```



11.3 – Additional Practice

Example: A random sample of 95 sprinters was taken, and the following information was collected on them:

- **time**: The athlete's best 100m sprint time.
- **cal**: The athlete's average daily caloric intake.
- **height**: The athlete's height in inches.
- **hr**: The athlete's resting heart rate in beats per minute.
- **Train**: The athlete's average number of hours spent training weekly.

This data can be found in **runners.csv**. In the following questions, we will use linear models to predict the time of sprinters.

Fit a multiple linear regression model that predicts time based on all other predictors. Which predictors are significant in predicting the 100m time?

Fit a new multiple linear regression model based only on those predictors that are significant. Write out the equation for this linear model.

Fit an interaction effect for the remaining predictors into your linear model. Is it significant? What does the slope for this interaction effect mean?

Create a 90% prediction interval for an athlete's 100m time that eats 3500 calories daily, is 74 inches tall, has a resting heart rate of 50 bpm, and spends 30 hours per week training. Use the interaction in your prediction if it was a significant predictor. Interpret the interval you find.