Covariance and Correlation

Load in Some Data

For this example, I'll be using data about board games. The file "BoardGames.csv" can be found on the UBlearns site for the lab. Download the file so it is on your computer and then load it into R. Go to "Import Dataset" in R Studio to load in the data. Attach the data to let R know you're working with it.

```
attach(BoardGames)
```

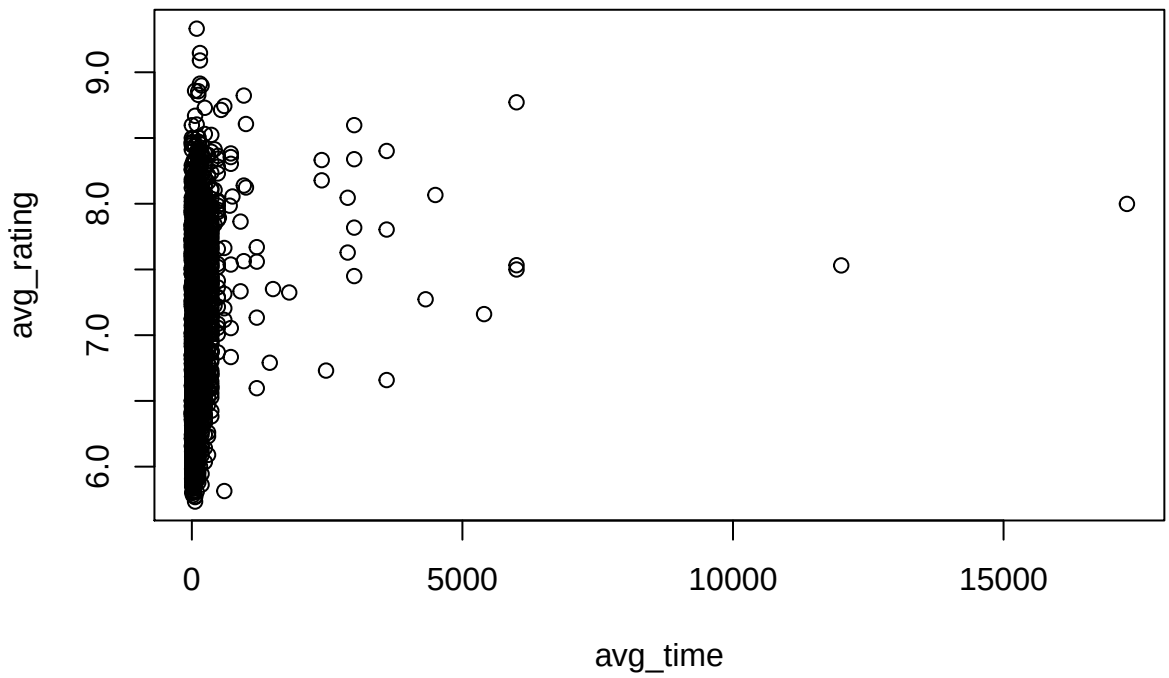
Use the names command to see the names of the variables included in the data.

```
names(BoardGames)
```

Covariance

Let's understand the relationship between average time it takes to play the game and the rating people give it. We could first begin by making a scatter plot of the two variables. This can be done by using the plot command.

```
plot(avg_time, avg_rating)
```



Well that looks kind of dumb. There are lots of board games with 0 average time so that's why. Normally, we'd remove those because those are probably missing data, but let's leave them in because it's easier.

Next, let's figure out the covariance. We can do this the slow way by following the equation:

$$cov_{xy} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{n}$$

Let's try that way. First we'll need some means and n. I'm giving these numbers names so we can use them in the equation.

```
mean.rating <- mean(avg_rating)
mean.time <- mean(avg_time)
n <- nrow(BoardGames)

covariance <- sum((avg_time - mean.time)*(avg_rating - mean.rating))/n
covariance
```

```
## [1] 34.62641
```

We can check that by using the command in R for covariance which is cov.

```
covcheck <- cov(avg_time, avg_rating)
covcheck
```

```
## [1] 34.63334
```

Hooray. They are basically the same.

Correlation

Covariance is kind of lame though. Using a correlation coefficient will help put that covariance number on a useful scale between -1 and 1.

We can calculate the correlation the slow way first. The equation is:

$$Correlation = \frac{covariance_{xy}}{\sigma_x \sigma_y}$$

Remember that sigma x and sigma y are the standard deviations. The slow way in R will look like this:

```
correlation <- covariance/(sd(avg_time)*sd(avg_rating))
correlation
```

```
## [1] 0.1567415
```

Then we can check it with the command for correlation in R which is cor.

```
corcheck <- cor(avg_time, avg_rating)
corcheck
```

```
## [1] 0.1567728
```

Excellent.

Testing for Significant Correlation

Lastly, we can check to see if this correlation is statistically significant by using cor.test.

```
cor.test(avg_time, avg_rating)
```

```
##
##  Pearson's product-moment correlation
##
## data:  avg_time and avg_rating
## t = 11.221, df = 4997, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1296141 0.1836965
## sample estimates:
##       cor
## 0.1567728
```

You can see in the information that we get a p-value, a 95% confidence interval for the correlation, and lastly the correlation is reported as 0.16 or so. The very small p-value (2.2e-16) tells us that there is a statistically significant positive correlation between how the average time it takes to play a board game and how well it gets rated. Since the correlation is positive, the longer it takes to play the game, the more people like it.

See if you can repeat this process for other variables like max_players, age, or weight. Additionally, you can see how rank is negatively correlated with average rating and geek_rating.