Week 7 Exercise 2

Kaylar Fullington

Create the Data Frame from Survey Data

```
readingvstv <- read.csv(file = 'data/student-survey.csv')
```

Find the Covariance

Use R to calculate the covariance of the survey variables and provide an explanation of why you would use this calculation and what the results indicate.

The variables collected in the survey include: *Time Reading*, *Time Watching TV*, *Happiness*, and *Gender*. Calculating the covariance between each variable and one other variable will give us an idea of whether a linear relationship between the two is positive or negative.

```
cov(readingvstv)
```

```
##              TimeReading       TimeTV  Happiness      Gender
## TimeReading   3.05454545 -20.36363636 -10.350091 -0.08181818
## TimeTV      -20.36363636 174.09090909 114.377273  0.04545455
## Happiness   -10.35009091 114.37727273 185.451422  1.11663636
## Gender       -0.08181818   0.04545455   1.116636  0.27272727
```

I used the cov() function to calculate the covariance between each variable and each other variable. The resulting table shows numbers that indicate whether there is a positive or negative linear relationship between each pair of variables. For instance, the covariance for *TimeReading* and *TimeTV* is negative - meaning that as one variable increases, the other decreases (and vice versa).

Pros and Cons of Covariance and the cov() function

Pros This is a good first step in evaluating a set of data, it gives the user an idea of what kind of relationships each pair of variables might have, positive or negative.

Cons Unfortunately, the covariance calculation does not account for different measurement scales. We cannot confidently examine how strongly two variables correlate if the measurement methods are not consistent. There is a way to fix this, however, and it is with correlation coefficients.

Find the Correlation Coefficient

The covariance needs to be standardized to remove the impact of measurement methods. The cor() function will calculate the correlation coefficient of each variable with each other variable. It automatically chooses to give the *Pearson Coefficient*. This coefficient is calculated by dividing the covariance of two variables by the standard deviations of each variable. See below.

Perform a correlation analysis of all variables.

```
cor(readingvstv, use = "everything")
```

```
##             TimeReading       TimeTV  Happiness       Gender
## TimeReading  1.00000000 -0.883067681 -0.4348663 -0.089642146
## TimeTV      -0.88306768  1.000000000  0.6365560  0.006596673
## Happiness   -0.43486633  0.636555986  1.0000000  0.157011838
## Gender      -0.08964215  0.006596673  0.1570118  1.000000000
```

The resulting chart shows how each pair correlates._Use = "everything"_ causes the program to use all data entires, placing an "NA" for rows with missing data. Each score can be multiplied by 100 to get a percentage for how strongly the variables correlate. Each variable correlates 100% with itself. We can also see that *TimeReading* and *TimeTV* have a very strong negative correlation that comes out to around 88%. As another example, we can see that *TimeTV* and *Happiness* have a relatively strong positive correlation at around 64%.

Perform a correlation analysis of just one pair of variables.

```
cor.test(readingvstv$TimeReading, readingvstv$Happiness, use = "everything")
```

```
##
##  Pearson's product-moment correlation
##
## data:  readingvstv$TimeReading and readingvstv$Happiness
## t = -1.4488, df = 9, p-value = 0.1813
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8206596  0.2232458
## sample estimates:
##        cor
## -0.4348663
```

The *cor.test()* function was chosen because it gives more extended statistics than the *cor()* function does.The correlation between *TimeReading* and *Happiness* (under the word "cor") is negative and around 43%. This implies that as more time is spent reading, the happiness score goes down and vice versa. Additionally, it shows the confidence interval, which is a range of *z-scores* (a z-score is the measure of how far a sample entry is from the population mean) assuming a normal distribution.(The majority of population samples of above 30 entries will have a normal distribution). Z-scores fall into a range of -3 (on the left side of a normal distribution) to 3 (on the right side of the distribution). A 95% confidence interval means that we expect 95% of the z-scores will fall between -1.96 and 1.96, with a range of 3.92. In this case, the upper bound is 0.2232458 and the lower bound is -0.8206596.

Let's see what happens when we increase that score to 99%.

Repeat your correlation test in step 2 but set the confidence interval at 99%.

```
cor.test(readingvstv$TimeReading, readingvstv$Happiness, use = "everything", conf.level = 0.99)
```

```
##
##  Pearson's product-moment correlation
##
## data:  readingvstv$TimeReading and readingvstv$Happiness
## t = -1.4488, df = 9, p-value = 0.1813
## alternative hypothesis: true correlation is not equal to 0
## 99 percent confidence interval:
##  -0.8801821  0.4176242
## sample estimates:
##        cor
## -0.4348663
```

A 99% confidence interval means that 99% of z-scores will be in the range between -2.58 and 2.58, which is a larger range than 95%. It's a little less precise. In this case, the confidence interval changed. Now, the upper bound is 0.4176242 and the lower bound is -0.8801821. This is a larger range than that was returned at a confidence interval of 95%. What does this mean? The probability that a score will fall within the confidence interval has gone up. In turn, this means it's more likely to contain the true mean. Interestingly, both confidence intervals are equally accurate in finding the range in which the true mean exists.The main difference between the two is that IF the true mean falls within the 95% confidence interval, it is more likely that you will *find* the true mean since there are fewer observations in that interval. However, it is more likely that the true mean will be in the confidence interval of 99% because there are more observations.

The other calculations in the corr.test() function.

The corr.test() function also produces, in no particular order, the *t-statistic*(t), the *degrees of freedom* (df), the *p-value*(p-value), and the *correlation coefficient*(cor). Lets define the *t-statistic* and the *p-value*.

The t-statistic

This statistic is calculated by dividing the difference between the estimated mean and they hypothesized mean by the standard error of the estimated mean. If the hypothesized mean is correct, we must assume that the t-statistic will fall around the middle of the t distribution. This would be about 2 standard deviations from the mean.

In the case of our *readingvstv* data frame, we can conclude that the hypothesized mean COULD be correct, since the t-statistic is within 2 standard deviations.

The p-value

This statistic represents the probability, if the null hypothesis is correct, that it would return a similar or more extreme estimated mean. If it too extreme, we will conclude that we must reject the null hypothesis. If the p-value is 0.05 or lower, we can conclude that the result is significant. If not, the result is not significant.

In the case of our *readingvstv* data frame, our p-value of 0.1813 is above that 0.05 value, so we can conclude that this p-value is not significant. *******

The Coefficient of Determination

This coefficient is calculated by performing r^2, or r squared, with r being the correlation coefficient.

```
model <- lm(readingvstv$TimeTV~readingvstv$TimeReading, data = readingvstv)
summary(model)
```

```
##
## Call:
## lm(formula = readingvstv$TimeTV ~ readingvstv$TimeReading, data = readingvstv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.333 -4.167 -1.667  1.667 11.667
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                98.333      4.723  20.818 6.37e-09 ***
## readingvstv$TimeReading    -6.667      1.181  -5.646 0.000315 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.526 on 9 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7553
## F-statistic: 31.87 on 1 and 9 DF,  p-value: 0.0003153
```

This model shows the Multiple R-Squared (Correlation of Determination) of 0.7798. In order to get the level that variation in time reading was caused by time watching TV, we multiply by 100. Thus, according to the Coefficient of Determination, about 78% of the variation in reading time can be explained by the number of hours spent watching TV.

Based on my own intuition and the statistics I've run today, I can say that there *may* be an impact by time watching TV on time reading. Having spent time as a student, however, I would say time spent reading would be impacted heavily by what type of reading was done. Someone who loves light fiction novels would likely spend a greater proporation of their time reading. However, if dry textbooks are the only reading these students do, I'm not surprised that they would spend more time watching TV. More research into the type of reading would be needed before I can make any strong determinations.

Partial Correlation

```
readingvstv2 <- readingvstv[c("TimeTV", "TimeReading", "Happiness")]
library(ggm)
pc <- pcor(c("TimeTV", "TimeReading", "Happiness"), var(readingvstv2))
pc^2
```

```
## [1] 0.762033
```

First, a second data frame, *readingvstv2* was created with only three variables, excluding sex. Here, the output of the function pcor(), assigned to the variable "pc", shows the partial correlation for *TimeTv* and *TimeReading*, controlling for *Happiness*.

```
pcor.test(pc, 1, 11)
```

```
## $tval
## [1] -5.061434
##
## $df
## [1] 8
##
## $pvalue
## [1] 0.0009753126
```