Stat/Math 415 Homework 5

Due Friday Oct 25; Joseph Sepich (jps6444)

1 Problem 7.4-3

Problem information:

- $\overline{x} = 6.09$
- $s = 0.02$

1.1 Part a

Sample size calculation: sample size needed for $\epsilon = 0.001$ with confidence of $\alpha = 0.1$

Assuming standard deviation is the same for the population we can use the Z value of 1.645 for alpha of 0.1 and plug the numbers into our equation based off a confidence interval:

$$n \geq \frac{z_{\alpha/2} S^2}{\epsilon^2} = \frac{1.645^2 * 0.02^2}{0.001^2} = 1082.41$$

Remember we must round up, so we need the sample size to be at least **1,083** samples.

1.2 Part b

With the following numbers we have a decent sample size, and we don't know the distribution, so we use central limit theorem and approximate with a normal distribution:

- $n = 1219$
- $\overline{x} = 6.048$
- $s = 0.022$

Since we looking at the confidecnce level from the last problem, we are still using the same Z value of 1.645. We plug this into our confidence interval formula:

$$\overline{x} + -Z_{\alpha/2} \frac{s}{\sqrt{n}} = 6.048 + -1.645 * \frac{0.022}{\sqrt{1219}}$$

This gives us a confidence interval for mu of [**6.0470, 6.0490**]

1.3 Part c

The problem states that for every 0.01 pounds less the company would save \$14,000 per year. the original sample mean was 6.09 and the new one is around 6.048. This would be a savings of roughly 4.2 * 14,000 = \$58,800.

1.4 Part d

To estimate the proportion of boxes that will weight less than 6 pounds, we want to know P(X < 6). We can use our estimated normal distribution from CLT to come up with a z score of $\frac{6-6.048}{0.022} \approx -2.18$.

$$P(X < 6) = P(Z < -2.18) = 1 - P(Z < 2.18) = 1 - 0.9854 = 0.0146$$

The proportion of boxes measured to be under 6 pounds is now **0.0146**.

2 Problem 7.4-7

2.1 Part a

What we know:

- $\epsilon = 0.03$
- $\alpha = 0.05$

What we don't have is any idea of what the actual proportion or a pilot sample proportion would be. We would have to assume the worst with p = 0.5 to maximize possible variance. Z score for a value of 0.025 (half of alpha) is 1.96. We plug this into our formula:

$$n = \frac{Z_{\alpha/2}^2 * p(1-p)}{\epsilon^2} = \frac{1.96^2 * 0.25}{0.03^2} = 1067.11$$

Rounding up we would require a sample size of at least **1,068**.

2.2 Part b

What we know:

- $\epsilon = 0.02$
- $\alpha = 0.05$

What we don't have is any idea of what the actual proportion or a pilot sample proportion would be. We would have to assume the worst with p = 0.5 to maximize possible variance. Z score for a value of 0.025 (half of alpha) is 1.96. We plug this into our formula:

$$n = \frac{Z_{\alpha/2}^2 * p(1-p)}{\epsilon^2} = \frac{1.96^2 * 0.25}{0.02^2} = 2401$$

We would require a sample size of at least **2,401**.

2.3 Part c

What we know:

- $\epsilon = 0.03$
- $\alpha = 0.1$

What we don't have is any idea of what the actual proportion or a pilot sample proportion would be. We would have to assume the worst with p = 0.5 to maximize possible variance. Z score for a value of 0.5 (half of alpha) is 1.645. We plug this into our formula:

$$n = \frac{Z_{\alpha/2}^2 * p(1-p)}{\epsilon^2} = \frac{1.645^2 * 0.25}{0.03^2} = 751.674$$

Rounding up we would require a sample size of at least **752**.

3 Problem 7.4-8

What we know:

- $n = 137$
- $y = 54$
- $p^* = 0.3942$
- $\epsilon = 0.04$
- $\alpha = 0.1$

Since we have a study already we can figure out how many samples we need using the variance based off the point estime from the first study. With a Z score value of 1.645 for alpha /2 of 0.5 we can plug into our formula:

$$n = \frac{Z_{\alpha/2}^2 * p(1-p)}{\epsilon^2} = \frac{1.645^2 * 0.3942 * 0.6058}{0.04^2} = 403.885$$

Rounding up we would require a sample size of at least **404**.

4 Problem 6.5-3

4.1 Part a

```
midterms <- c(70, 74, 80, 84, 80, 67, 70, 64, 74, 82)
finals <- c(87, 79, 88, 98, 96, 73, 83, 79, 91, 94)

meanx <- sum(midterms) / 10
meany <- sum(finals) / 10

beta <- sum((finals - meany) * (midterms - meanx)) / sum((midterms - meanx)^2)

print(paste0('Sample mean x: ',meanx))
```

```
## [1] "Sample mean x: 74.5"
```

```
print(paste0('Sample mean y(alpha): ',meany))
```

```
## [1] "Sample mean y(alpha): 86.8"
```

```
print(paste0('Beta: ',beta))
```

```
## [1] "Beta: 1.01568154402895"
```

Suppose x denotes the midterm score, which indicates the final score y.

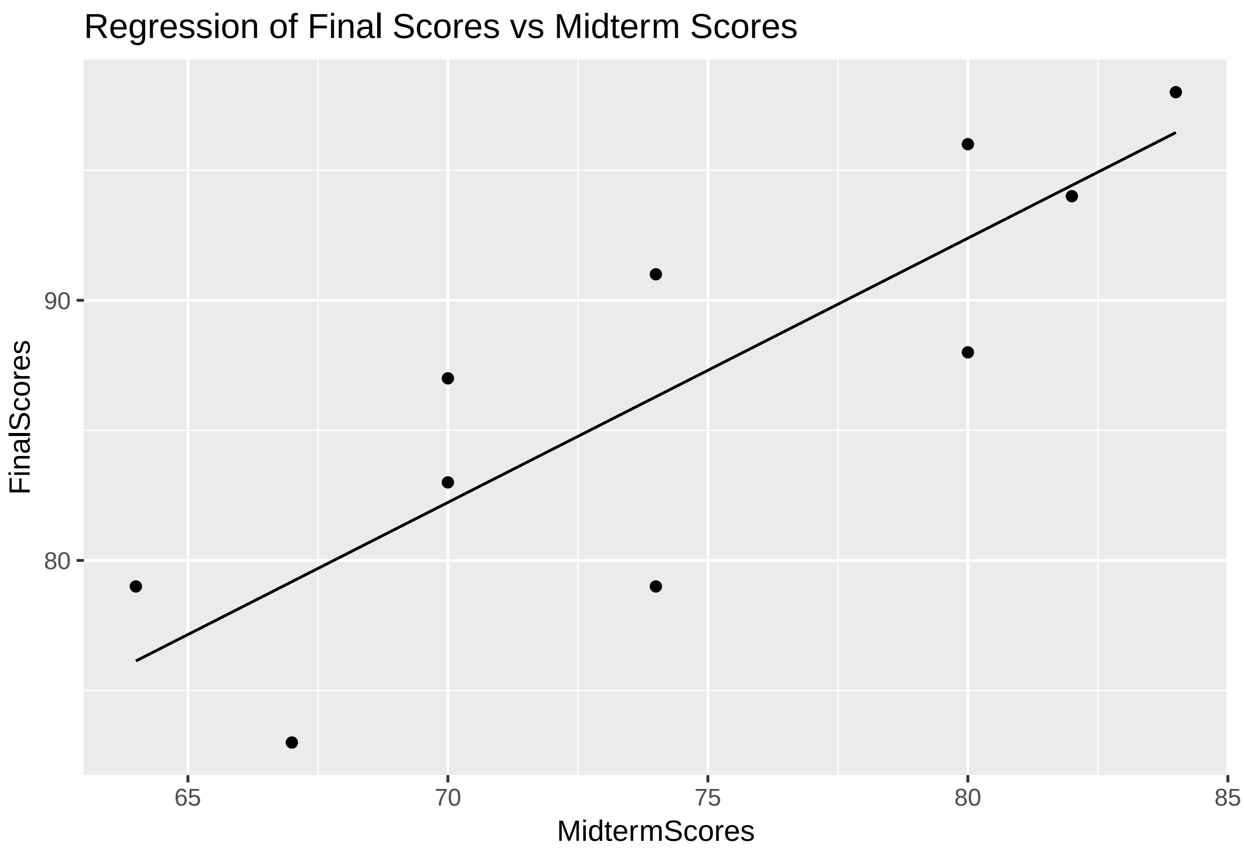
$$y = \alpha + \beta(x - \overline{x})$$
$$\overline{x} = \frac{1}{n}\Sigma_{i=1}^{n} x_i = \frac{1}{10} * (70 + 74 + ... + 74 + 82) = 74.5$$
$$\alpha = \overline{y} = \frac{1}{n}\Sigma_{i=1}^{n} y_i = \frac{1}{10} * (87 + 79 + ... + 91 + 94) = 86.8$$
$$\beta = \frac{\Sigma_{i=1}^{n}(y_i - \overline{y})(x_i - \overline{x})}{\Sigma_{i=1}^{n}(x_i - \overline{x})^2} = 1.0157$$

This gives us the least squares regression line:

$$y = 86.8 + 1.0157(x - 74.5) = 11.132 + 1.0157x$$

4.2 Part b

```
xVals <- seq(min(midterms), max(midterms), 0.01)
yVals <- meany + beta * (xVals - meanx)
df1 <- data.frame("MidtermScores"=midterms, "FinalScores"=finals)
df2 <- data.frame("RegressionX"=xVals, "RegressionY"=yVals)
df1 %>%
  ggplot(aes(x=MidtermScores,y=FinalScores)) +
  geom_point() +
  geom_line(data=df2,aes(x=RegressionX,y=RegressionY)) +
  ggtitle('Regression of Final Scores vs Midterm Scores')
```



4.3 Part c

The point estimate of the variance for our regression is the sum of squared errors divided by the sample size:

$$\hat{\sigma^2} = \frac{\Sigma_{i=1}^{n}(y_i - \hat{y_i})}{n} = \frac{179.9981}{10} = 17.99981$$

```
yHats <- meany + beta * (midterms - meanx)
sumError <- sum((finals - yHats)^2)
sigma <- sumError / 10

print(paste0('The sum of squared errors: ', sumError))
```

```
## [1] "The sum of squared errors: 179.998069963812"
```

```
print(paste0('The variance estimate: ', sigma))
```

```
## [1] "The variance estimate: 17.9998069963812"
```

5 Problem 6.5-5

```
n <- 14
horsepower <- c(230, 225, 375, 322, 190, 150, 178, 282, 300, 220, 250, 315, 200, 300)
acceleration <- c(8.1, 7.8, 4.7, 6.6, 8.4, 8.4, 7.2, 6.2, 6.4, 7.7, 7.0, 5.3, 6.2, 5.5)
weight <- c(3516, 3690, 2976, 4215, 3761, 2940, 2818, 3627, 3892, 3377, 3625, 3230, 2657, 3518)
```

5.1 Part

Calculate the least squares regression line for 0-60 vs horsepower:

```
meanHorse <- sum(horsepower) / n
meanAccel <- sum(acceleration) / n

beta <- sum((acceleration - meanAccel) * (horsepower - meanHorse)) /
  sum((horsepower - meanHorse)^2)


print(paste0('Sample mean x: ',meanHorse))
```

```
## [1] "Sample mean x: 252.642857142857"
```

```
print(paste0('Sample mean y(alpha): ',meanAccel))
```

```
## [1] "Sample mean y(alpha): 6.82142857142857"
```

```
print(paste0('Beta: ',beta))
```

```
## [1] "Beta: -0.014993624475684"
```

Suppose x denotes the horsepower, which affects 0-60 y.

$$y = \alpha + \beta(x - \overline{x})$$

$$\overline{x} = \frac{1}{n}\Sigma_{i=1}^{n} x_i = \frac{1}{10} * (230 + 225 + ... + 200 + 300) = 252.64$$

$$\alpha = \overline{y} = \frac{1}{n}\Sigma_{i=1}^{n} y_i = \frac{1}{10} * (8.1 + 7.8 + ... + 6.2 + 5.5) = 6.82$$

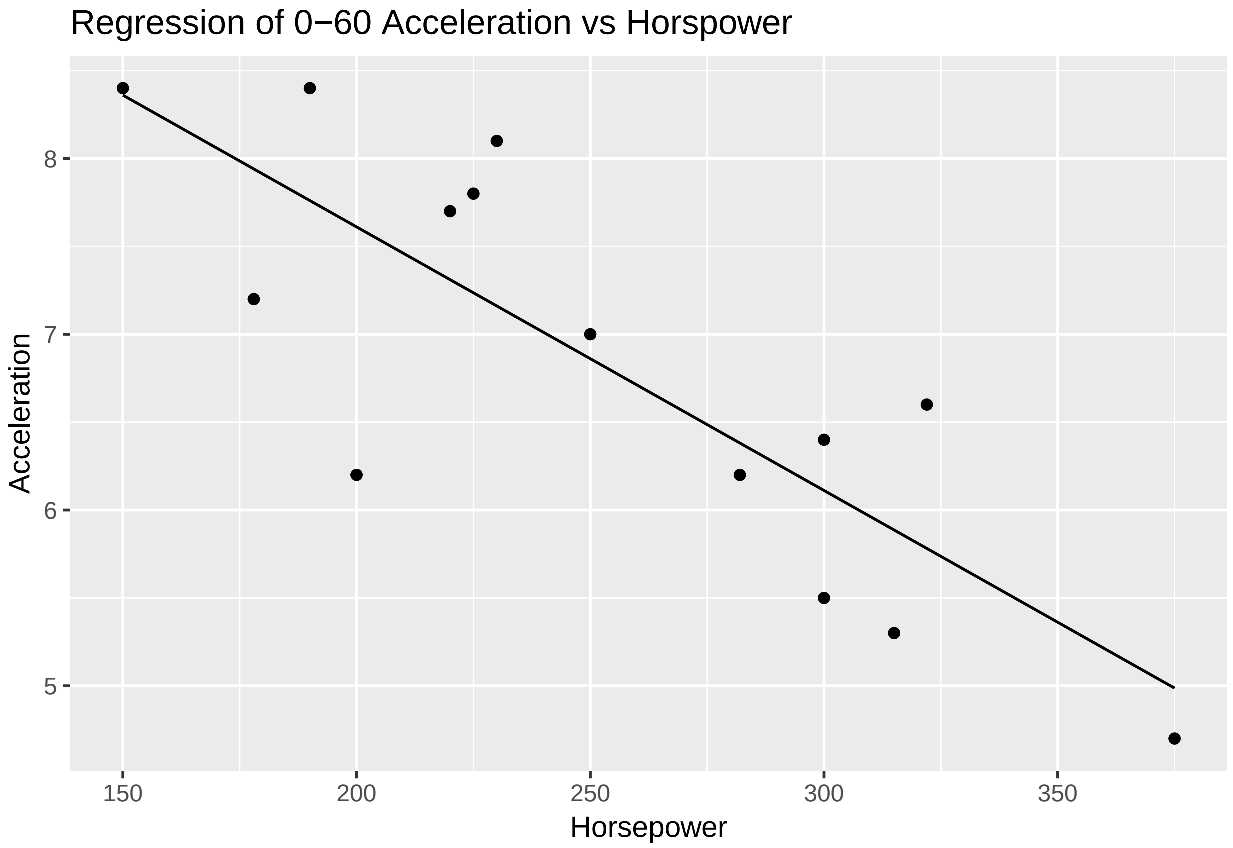
$$\beta = \frac{\Sigma_{i=1}^{n}(y_i - \overline{y})(x_i - \overline{x})}{\Sigma_{i=1}^{n}(x_i - \overline{x})^2} = -0.01499$$

This gives us the least squares regression line:

$$y = 6.82 - 0.01499(x - 252.64) = 10.609 - 0.01499x$$

5.2 Part b

```
xVals <- seq(min(horsepower), max(horsepower), 0.01)
yVals <- meanAccel + beta * (xVals - meanHorse)
df1 <- data.frame("Horsepower"=horsepower, "Acceleration"=acceleration)
df2 <- data.frame("RegressionX"=xVals, "RegressionY"=yVals)
df1 %>%
  ggplot(aes(x=Horsepower,y=Acceleration)) +
  geom_point() +
  geom_line(data=df2,aes(x=RegressionX,y=RegressionY)) +
  ggtitle('Regression of 0-60 Acceleration vs Horspower')
```



5.3 Part c

Calculate the least squares regression line for 0-60 vs weight:

```
meanWeight <- sum(weight) / n

beta <- sum((acceleration - meanAccel) * (weight - meanWeight)) /
  sum((weight - meanWeight)^2)


print(paste0('Sample mean x: ',meanWeight))
```

```
## [1] "Sample mean x: 3417.28571428571"
```

```r
print(paste0('Sample mean y(alpha): ',meanAccel))
```

```
## [1] "Sample mean y(alpha): 6.82142857142857"
```

```r
print(paste0('Beta: ',beta))
```

```
## [1] "Beta: 0.000395096548870521"
```

Suppose x denotes the horsepower, which affects 0-60 y.

$$y = \alpha + \beta(x - \overline{x})$$

$$\overline{x} = \frac{1}{n}\Sigma_{i=1}^{n} x_i = \frac{1}{10} * (3516 + 3690 + ... + 2657 + 3518) = 3417.286$$

$$\alpha = \overline{y} = \frac{1}{n}\Sigma_{i=1}^{n} y_i = \frac{1}{10} * (8.1 + 7.8 + ... + 6.2 + 5.5) = 6.82$$

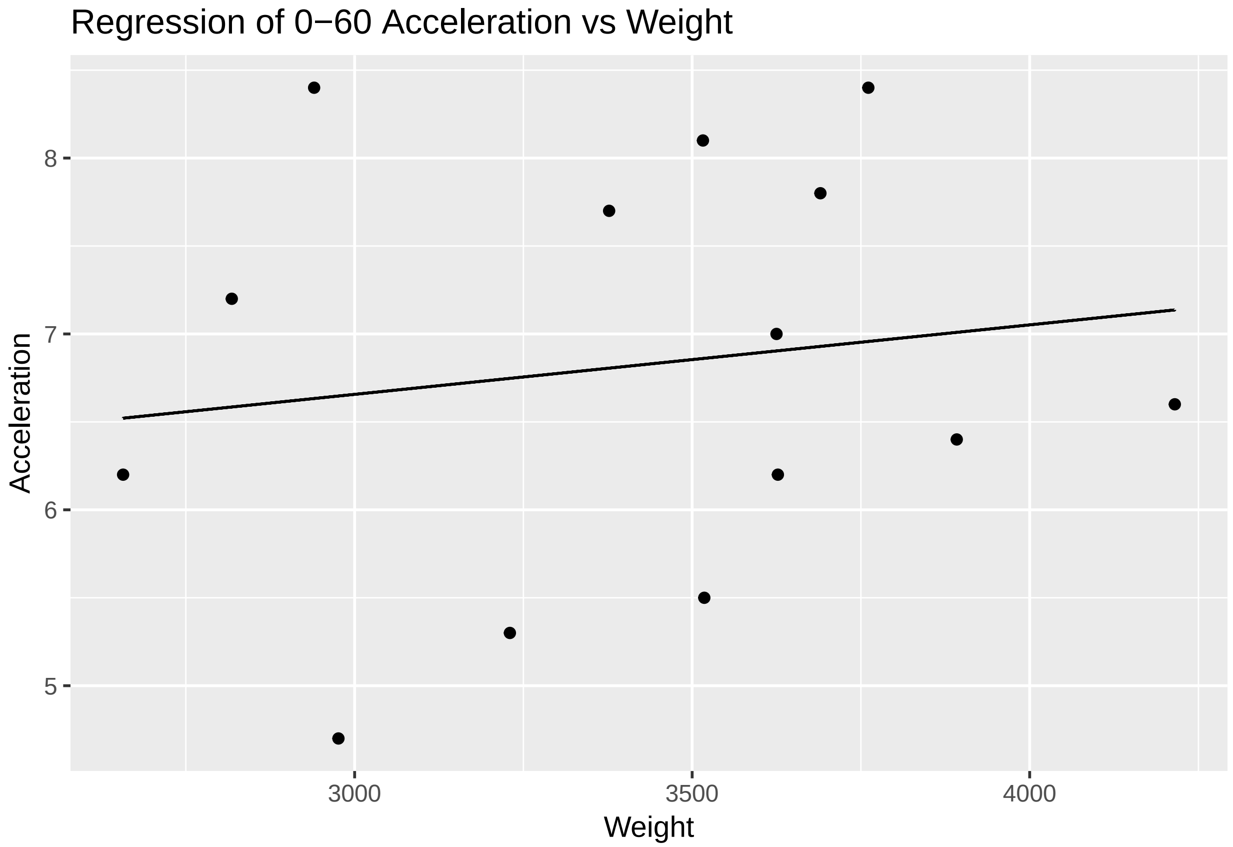
$$\beta = \frac{\Sigma_{i=1}^{n}(y_i - \overline{y})(x_i - \overline{x})}{\Sigma_{i=1}^{n}(x_i - \overline{x})^2} = 0.000395$$

This gives us the least squares regression line:

$$y = 6.82 + 0.000395(x - 3417.286) = 5.471 + 0.000395x$$

5.4 Part d

```r
xVals <- seq(min(weight), max(weight), 0.01)
yVals <- meanAccel + beta * (xVals - meanWeight)
df1 <- data.frame("Weight"=weight, "Acceleration"=acceleration)
df2 <- data.frame("RegressionX"=xVals, "RegressionY"=yVals)
df1 %>%
  ggplot(aes(x=Weight,y=Acceleration)) +
  geom_point() +
  geom_line(data=df2,aes(x=RegressionX,y=RegressionY)) +
  ggtitle('Regression of 0-60 Acceleration vs Weight')
```



5.5 Part e

Horsepower has more of an affect on 0-60 acceleration time. The slope of the regression line was steeper for horsepower vs 0-60, which means beta for that regression had a larger absolute value than for weight. This implies there is a greater change in y or 0-60 time due to horsepower than weight, because of a larger beta value.