HW1

Shay Lebovitz

1/14/2021

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v dplyr   1.0.2
## v tibble  3.0.4     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Problem 1.5

No, I do not agree with the student.

$E[Y_i] = E[B_0 + B_1 * x_i + e_i]$

$E[Y_i] = B_0 + B_1 * x_i + E[e_i]$, because B_0, B_1 and x_i are non-random numbers.

Because $E[e_i] = 0$, we see that $E[Y_i] = B_0 + B_1 * x_i$

Thus, the student was wrong because they forgot that the expected value of the error term is 0.

Problem 1.19

```
gpa_data <- read_delim('data/CH01PR19.txt', delim = ' ', col_names = F)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character()
## )
```

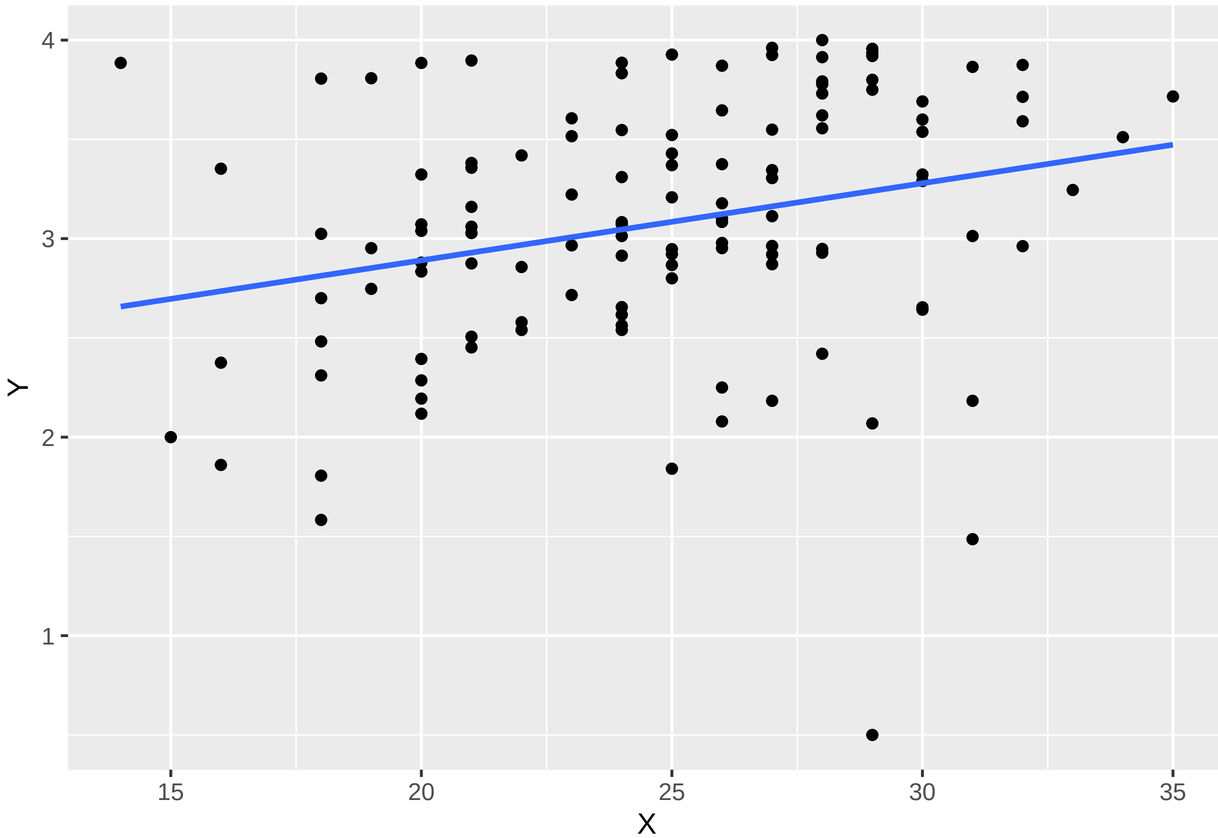
```
gpa_data <- lapply(gpa_data, as.numeric)
gpa_data <- as_tibble(gpa_data)
gpa_data <- gpa_data %>%
  transmute(X = X2, Y = X1)

#a
beta_1 <- cov(gpa_data$X, gpa_data$Y)/var(gpa_data$X)
beta_0 <- mean(gpa_data$Y) - beta_1 * mean(gpa_data$X)
```

$Y_i = 2.1140 + 0.0388 * x_i$

```r
#b
gpa_data %>%
  ggplot(aes(X, Y)) +
  geom_point() +
  geom_smooth(method = lm, se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



No, the estimated regression line does not fit the data very well. It shows that ACT score is not a very good predictor of freshman year GPA.

c $2.1140 + 0.0388 * 30 = 3.2789$

d The point estimate of the change in GPA when the ACT score increases by one point is exactly Beta1, which is 0.0388.

Problem 1.27

```r
muscle <- read_delim('data/CH01PR27.txt', delim = " ", col_names = F)
```

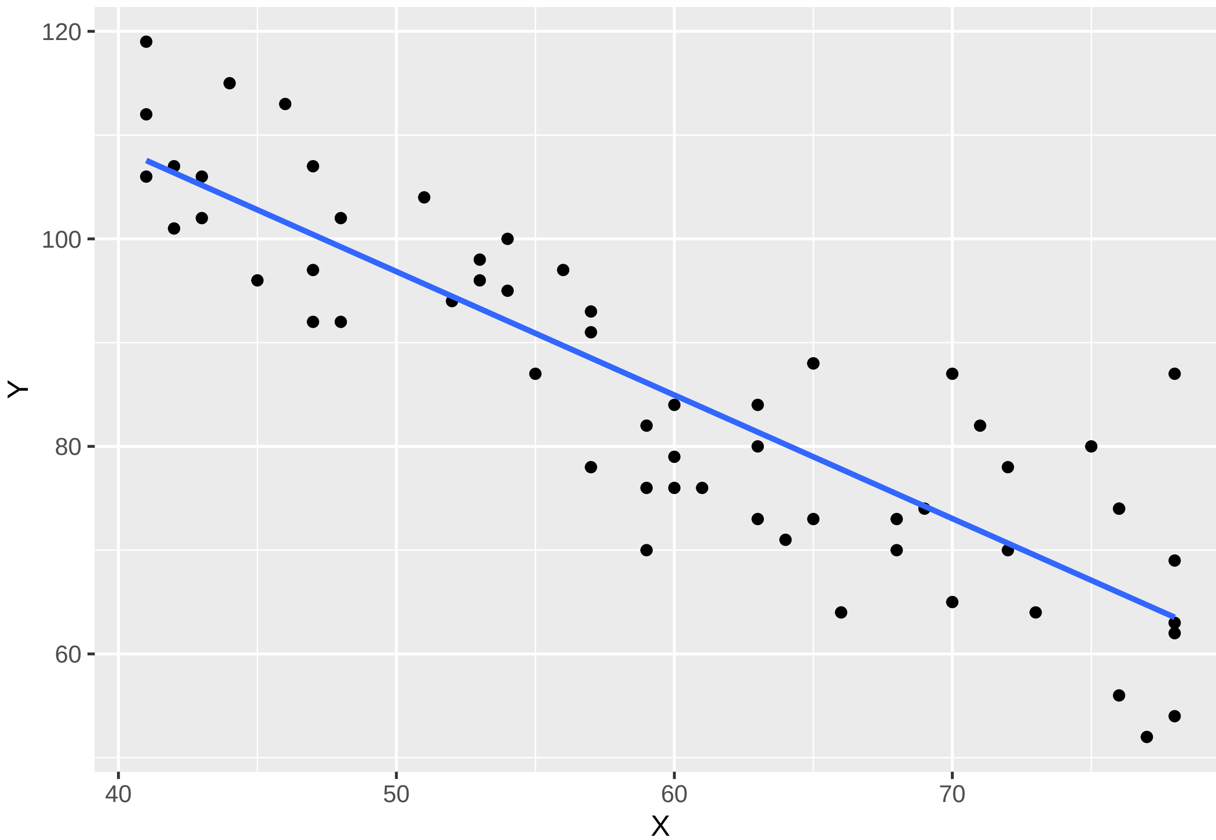
```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character()
## )
```

```r
muscle <- lapply(muscle, as.numeric)
muscle <- as_tibble(muscle)
muscle <- muscle %>%
  transmute(X = X2, Y = X1)

#a
beta_1_m <- cov(muscle$X, muscle$Y) / var(muscle$X)
beta_0_m <- mean(muscle$Y) - beta_1_m*mean(muscle$X)

muscle %>%
  ggplot(aes(X, Y)) +
  geom_point() +
  geom_smooth(method = lm, se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



This appears to be a pretty accurate fit. There is a clear downward, linear relationship between muscle mass and age.

$Y_i = 156.3466 - 1.1900 * x_i$

b

1) The point estimate for the difference in muscle mass of woman differing by one year is exactly Beta1, which is 1.1900 pounds.

2) The point estimate for the mean muscle mass of a woman aged 60 years is $156.3466 - 1.1900 * 60 = 84.9468$ lbs.

3) For the eighth case, Age = 48 and muscle mass = 112. Predicted muscle mass = $156.3466 - 1.1900 * 48 = 99.2268$, so $112 - 99.2268 = 12.7732$

4)

```
muscle <- muscle %>%
  mutate(y_hat = beta_0_m + beta_1_m*X) %>%
  mutate(resid = Y - y_hat)

s_squared <- sum(muscle$resid^2)/58
```

An unbiased estimator for simga squared = 66.8008.