HW2_Cantelmo

Robert G. Cantelmo

March 14, 2018

Problem 1

Section 1: Matrix Form

```
sprinters<-read.csv("sprinters.csv")
#In R, Create a matrix X comprised of three columns: a column of ones, a column made of the variable ye
sprinters$ones <-c (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
X <- matrix(data=c(sprinters$ones, sprinters$year, sprinters$women), nrow = 42, ncol=3, byrow=FALSE)
#Create a matrix y comprised of a single column, made up of the variable finish.
Y <- matrix(data=c(sprinters$finish), nrow = 42, ncol=1)
#Compute the following using R's matrix commands (note that you will need to use the matrix multiplicat

b <- (solve(t(X)%*%X)%*%t(X)%*%Y)
summary(b)
```

```
##        V1
##  Min.   :-0.01261
##  1st Qu.: 0.54010
##  Median : 1.09281
##  Mean   :12.01341
##  3rd Qu.:18.02642
##  Max.   :34.96004
```

Section 2: Fitting a linear model

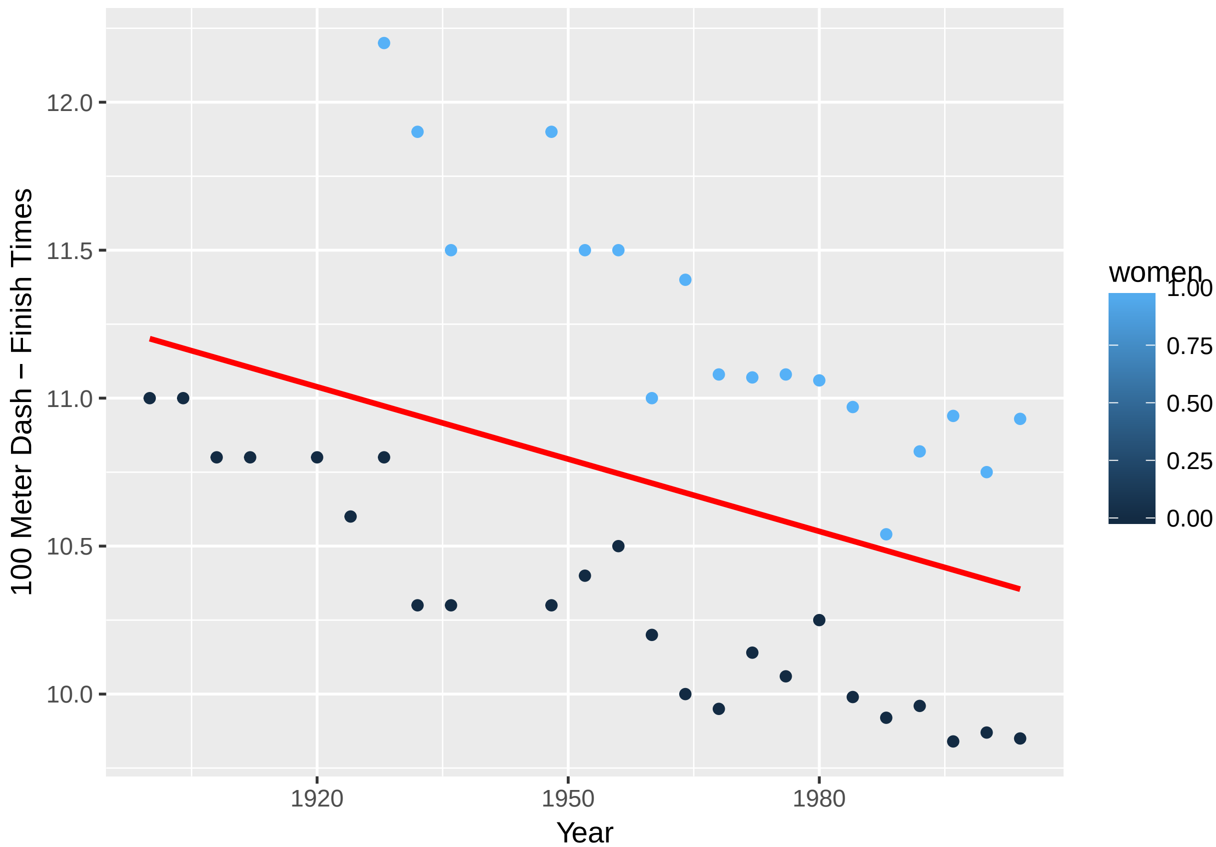
```
#Using the function lm, run a regression of finish on year and women.
#Compare the results the calculation you did in Section 1.
lm_finish <- lm(finish ~ year + women, data=sprinters)
summary(lm_finish)
```

```
##
## Call:
## lm(formula = finish ~ year + women, data = sprinters)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44623 -0.10170  0.02093  0.11094  0.45724
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.960037   1.964903   17.79  < 2e-16 ***
## year        -0.012609   0.001005  -12.54 2.89e-15 ***
## women        1.092812   0.059502   18.37  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1852 on 39 degrees of freedom
## Multiple R-squared:  0.9125, Adjusted R-squared:  0.9081
## F-statistic: 203.5 on 2 and 39 DF,  p-value: < 2.2e-16
```

```r
#The coefficients are the same as the matrix results!

#Make a nice plot summarizing this regression. On a single graph, plot the data and the regression line

ggplot(sprinters, aes(x=year, y=finish))+geom_point(aes(color=women))+labs(y = "100 Meter Dash - Finish
```




```r
#Rerun the regression, adding an interaction between women and year.
lm_finish_interact <- lm(finish ~ year * women, data=sprinters)
summary(lm_finish_interact)
```

```
##
## Call:
## lm(formula = finish ~ year * women, data = sprinters)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37579 -0.05460  0.00738  0.08276  0.32234
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 31.826453   2.128910  14.950  < 2e-16 ***
## year        -0.011006   0.001089 -10.104 2.56e-12 ***
## women       12.520596   4.076141   3.072  0.00392 **
## year:women  -0.005817   0.002074  -2.804  0.00791 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1707 on 38 degrees of freedom
## Multiple R-squared:  0.9275, Adjusted R-squared:  0.9218
## F-statistic: 162.1 on 3 and 38 DF,  p-value: < 2.2e-16
```

```
#Redo the plot with a new fit, one for each level of women.
ggplot(sprinters, aes(x=year, y=finish, col))+geom_point(aes(color=women))+labs(y = "100 Meter Dash - Fi
```

Section 3: Predicted Values

```
#Suppose that an Olympics had been held in 2001. Use the predict function to calculate the expected fin
MenOLY2001  <-predict(lm_finish, newdata = data_frame(year=2001, women=0), interval = "confidence", leve
summary(MenOLY2001)
```

```
##       fit             lwr             upr
##  Min.   :9.729   Min.   :9.608   Min.   :9.851
##  1st Qu.:9.729   1st Qu.:9.608   1st Qu.:9.851
##  Median :9.729   Median :9.608   Median :9.851
##  Mean   :9.729   Mean   :9.608   Mean   :9.851
##  3rd Qu.:9.729   3rd Qu.:9.608   3rd Qu.:9.851
##  Max.   :9.729   Max.   :9.608   Max.   :9.851
```

```
#9.729

WomenOLY2001  <-predict(lm_finish, newdata = data_frame(year=2001, women=1), interval = "confidence", le
summary(WomenOLY2001)
```

```
##       fit             lwr             upr
##  Min.   :10.82   Min.   :10.71   Min.   :10.93
```

```
##  1st Qu.:10.82   1st Qu.:10.71   1st Qu.:10.93  
##  Median :10.82   Median :10.71   Median :10.93  
##  Mean   :10.82   Mean   :10.71   Mean   :10.93  
##  3rd Qu.:10.82   3rd Qu.:10.71   3rd Qu.:10.93  
##  Max.   :10.82   Max.   :10.71   Max.   :10.93  
```

```
#10.82

#The authors of the Nature article were interested in predicting the finishing times for the 2156 Olymp

MenOLY2156  <-predict(lm_finish, newdata = data_frame(year=2156, women=0), interval = "confidence", leve
summary(MenOLY2156)
```

```
##       fit             lwr             upr       
##  Min.   :7.775   Min.   :7.358   Min.   :8.192  
##  1st Qu.:7.775   1st Qu.:7.358   1st Qu.:8.192  
##  Median :7.775   Median :7.358   Median :8.192  
##  Mean   :7.775   Mean   :7.358   Mean   :8.192  
##  3rd Qu.:7.775   3rd Qu.:7.358   3rd Qu.:8.192  
##  Max.   :7.775   Max.   :7.358   Max.   :8.192  
```

```
#7.775

WomenOLY2156  <-predict(lm_finish, newdata = data_frame(year=2156, women=1), interval = "confidence", le
summary(WomenOLY2156)
```

```
##       fit             lwr             upr       
##  Min.   :8.868   Min.   :8.477   Min.   :9.259  
##  1st Qu.:8.868   1st Qu.:8.477   1st Qu.:9.259  
##  Median :8.868   Median :8.477   Median :9.259  
##  Mean   :8.868   Mean   :8.477   Mean   :9.259  
##  3rd Qu.:8.868   3rd Qu.:8.477   3rd Qu.:9.259  
##  Max.   :8.868   Max.   :8.477   Max.   :9.259  
```

```
#8.868

#Do you trust the model's predictions? Is there reason to trust the 2001 prediction more than the 2156 p

#I do not trust the model's predictions because it the predicted data assume the trend will be unbroken
```

Problem 2

```
library("tidyverse")
```

```
## Warning: package 'tidyverse' was built under R version 3.4.3

## -- Attaching packages --------------------------------- tidyverse 1.2.1 --

## v tibble  1.4.2     v purrr   0.2.4
## v tidyr   0.8.0     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.4.3

## Warning: package 'tidyr' was built under R version 3.4.3

## Warning: package 'readr' was built under R version 3.4.3

## Warning: package 'purrr' was built under R version 3.4.3

## Warning: package 'forcats' was built under R version 3.4.3
```

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

```
anscombe2 <- anscombe %>%
    mutate(obs = row_number()) %>%
    gather(variable_dataset, value, - obs) %>%
    separate(variable_dataset, c("variable", "dataset"), sep = 1L) %>%
    spread(variable, value) %>%
    arrange(dataset, obs)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.3
```

Section 4: Looking at your data beyond summary statistics

```
#For each dataset: calculate the mean and standard deviations of x and y, and correlation between x and
x1 <- filter(anscombe2, dataset==1)
mean(x1$x)
```

```
## [1] 9
```

```
mean(x1$y)
```

```
## [1] 7.500909
```

```
# x-mean: 9
# y-mean: 7.501
sd(x1$x)
```

```
## [1] 3.316625
```

```
sd(x1$y)
```

```
## [1] 2.031568
```

```
# x-mean: 3.317
# y-mean: 2.032
cor(x1$x,x1$y)
```

```
## [1] 0.8164205
```

```
#0.816

x2<- filter(anscombe2, dataset==2)
mean(x2$x)
```

```
## [1] 9
```

```
mean(x2$y)
```

```
## [1] 7.500909
```

```
# x-mean: 9
# y-mean: 7.501
sd(x2$x)
```

```
## [1] 3.316625
```

```
sd(x2$y)
```

```
## [1] 2.031657
```

```r
# x-mean: 3.317
# y-mean: 2.032
cor(x2$x,x2$y)
```

```
## [1] 0.8162365
```

```r
#0.8162

x3 <- filter(anscombe2, dataset==3)
mean(x3$x)
```

```
## [1] 9
```

```r
mean(x3$y)
```

```
## [1] 7.5
```

```r
# x-mean: 9
# y-mean: 7.5
sd(x3$x)
```

```
## [1] 3.316625
```

```r
sd(x3$y)
```

```
## [1] 2.030424
```

```r
# x-mean: 3.317
# y-mean: 2.030
cor(x3$x,x3$y)
```

```
## [1] 0.8162867
```

```r
#0.816

x4<- filter(anscombe2, dataset==4)
mean(x4$x)
```

```
## [1] 9
```

```r
mean(x4$y)
```

```
## [1] 7.500909
```

```r
# x-mean: 9
# y-mean: 7.501
sd(x4$x)
```

```
## [1] 3.316625
```

```r
sd(x4$y)
```

```
## [1] 2.030579
```

```r
# x-mean: 3.316
# y-mean: 2.031
cor(x4$x,x4$y)
```

```
## [1] 0.8165214
```

```r
#0.817
```

```r
#Run a linear regression between x and y for each dataset.
lm_x1 <- lm(y ~ x, data=x1)
summary(lm_x1)
```

```
##
## Call:
## lm(formula = y ~ x, data = x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## x             0.5001     0.1179   4.241  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```r
lm_x2 <- lm(y ~ x, data=x2)
summary(lm_x2)
```

```
##
## Call:
## lm(formula = y ~ x, data = x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x              0.500      0.118   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```r
lm_x3 <- lm(y ~ x, data=x3)
summary(lm_x3)
```

```
##
## Call:
## lm(formula = y ~ x, data = x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x             0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

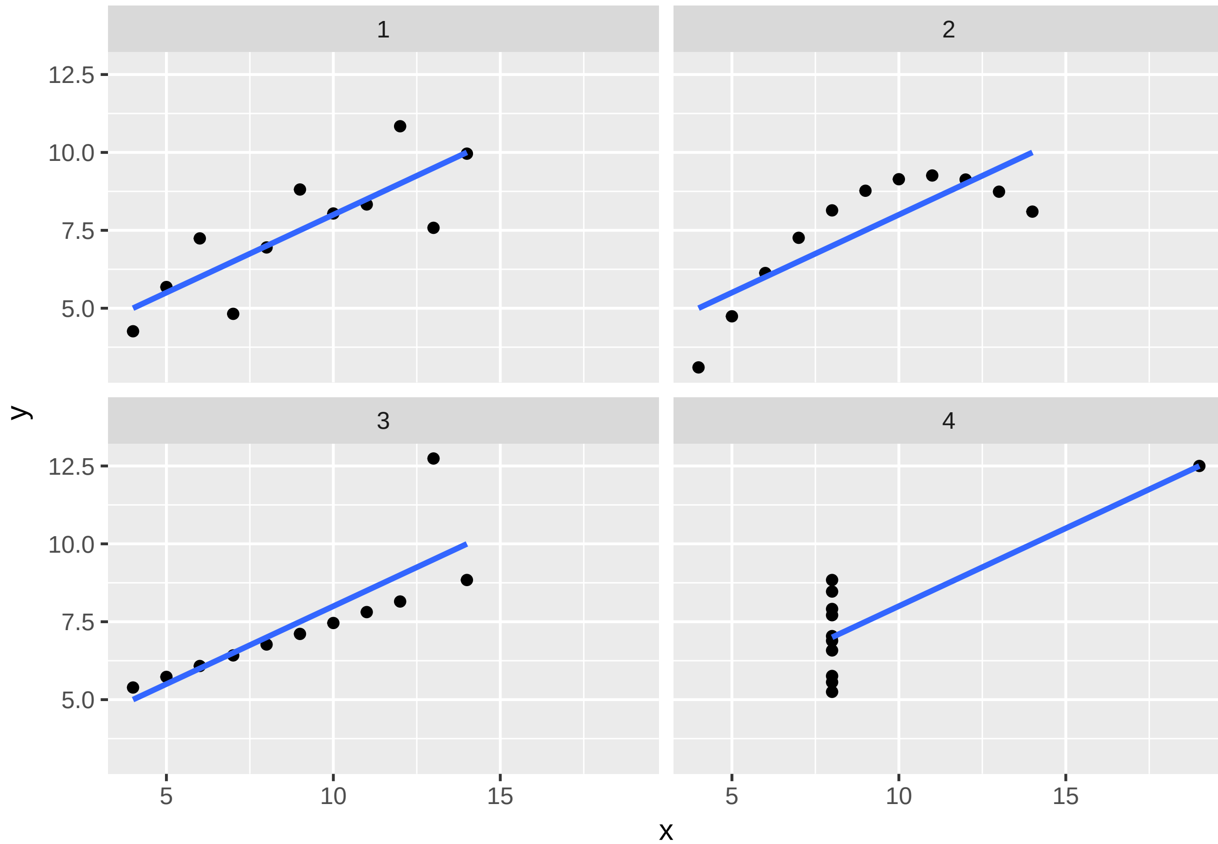
```
lm_x4 <- lm(y ~ x, data=x4)
summary(lm_x4)
```

```
##
## Call:
## lm(formula = y ~ x, data = x4)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## x             0.4999     0.1178   4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

```
#How similar do you think that these datasets will look?

#It is difficult to determine how the data will look based on the information provided. The mean, stand

#Create a scatter plot of each dataset and its linear regression fit. Hint: you can do this easily with
ggplot(data=anscombe2, aes(x=x, y=y)) + geom_point()+ stat_smooth(method = "lm", se = FALSE)+ facet_wra
```




```
#How do we make sense of these plots?
#Plots 1&3 are the best approximation of a linear relationship. The first plot essentially has no outli
```

Problem 3

Section 5: Research Project

```
#Robert and Tessa to discuss project with Sergio separately this week.
```