Midterm

Janette Avelar

5/2/2022

Instructions:

This exam will be graded out of 50 points. So, please choose your own combination of problems to add up to 50 possible points. Extra credit up to 60 total points is allowed.

Questions selected:
* Question 1 (10)
* Question 2 (10)
* Question 3 (20)
* Question 4 (20)

Question 1 (10 pts)

Students in 613 can be divided into three groups: those who own 1 computer, those who own 2 computers, and those who own 3 or more computers. The level of stats-related anxiety for five of those with 1 computer is: 4, 3, 4, 2, 3 (on a 5-point scale); the level of anxiety for five of those with 2 computers is: 4, 3, 3, 2, 3 (on the same scale); finally, the level of anxiety for five of those with 3 or more computers is: 2, 3, 2, 2, 3.

Write down the GLM equation to predict stats-related anxiety based on this grouping variable. Solve for the parameter vector (β) and the error vector (ϵ) by hand (showing all of your work). (Hint: for help on calculating the inverse of X, see Lecture 1, pp. 4.)

Step 1: Write down the GLM equation Y = XB + e where:
* Y = stats related anxiety
* X = type of computer owned
* B = predicted anxiety when the type of computer owned is our reference group
* e = error

In other words,

$$Anxiety = PredAnxiety * CompType + error$$

Step 2: Specify our matrices and solve for B. Y is the observed values of stats related anxiety:

$$Y = \begin{bmatrix} 4 \\ 3 \\ 4 \\ 2 \\ 3 \\ 4 \\ 3 \\ 3 \\ 2 \\ 3 \\ 2 \\ 3 \\ 2 \\ 2 \\ 3 \end{bmatrix}$$

X is dummy coded values for computer type where the reference group is students who own 1 computer.

$$XDesignMatrix = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

To solve for B, we need to multiply the inverse of X by Y.

I know, according to the matrix multiplication rules, that the inverse of X multiplied by itself will give us the identity matrix. And further, that in order to get that value it must be a 3x15 matrix. I'll begin by transposing to get a 3x15 matrix. I've shifted each row above into a column to transpose:

$$X' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

I also know, according to the rules of matrix multiplication, that multiplying X and X' will make me multiply and sum so that I get a 3x3 matrix with 5's, rather than 1's, along the diagonal. So I'd need the resulting sum to equal 1, rather than 5. I can do this by dividing each 1 in the transposed matrix by 5.

$$X^{-}1 = \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix}$$

Now I can solve for B by multiplying $X^{-}1 * Y =$

$$B = \begin{bmatrix} (1/5*4)+(1/5*3)+(1/5*4)+(1/5*2)+(1/5*3)+(0*4)+(0*3)+(0*3)+(0*2)+(0*3)+(0*2)+(0*3) \\ (0*4)+(0*3)+(0*4)+(0*2)+(0*3)+(1/5*4)+(1/5*3)+(1/5*3)+(1/5*2)+(1/5*3)+(0*2)+(0*3) \\ (0*4)+(0*3)+(0*4)+(0*2)+(0*3)+(0*4)+(0*3)+(0*3)+(0*2)+(0*3)+(1/5*2)+(1/5*3)+(1/5 \end{bmatrix}$$

==>

$$B = \begin{bmatrix} (.8+.6+.8+.4+.6)=3.2 \\ (.8+.6+.6+.4+.6)=3 \\ (.4+.6+.4+.4+.6)=2.4 \end{bmatrix}$$

==>

$$B = \begin{bmatrix} 3.2 \\ 3 \\ 2.4 \end{bmatrix}$$

Step 3: Solve for e. To solve for e, we need to solve the following formula:

$$e = Y - XB$$

We'll begin with XB:

$$XB = \begin{bmatrix} (1*3.2)+(0*3)+(0*2.4)=3.2 \\ (1*3.2)+(0*3)+(0*2.4)=3.2 \\ (1*3.2)+(0*3)+(0*2.4)=3.2 \\ (1*3.2)+(0*3)+(0*2.4)=3.2 \\ (1*3.2)+(0*3)+(0*2.4)=3.2 \\ (0*3.2)+(1*3)+(0*2.4)=3 \\ (0*3.2)+(1*3)+(0*2.4)=3 \\ (0*3.2)+(1*3)+(0*2.4)=3 \\ (0*3.2)+(1*3)+(0*2.4)=3 \\ (0*3.2)+(1*3)+(0*2.4)=3 \\ (0*3.2)+(0*3)+(1*2.4)=2.4 \\ (0*3.2)+(0*3)+(1*2.4)=2.4 \\ (0*3.2)+(0*3)+(1*2.4)=2.4 \\ (0*3.2)+(0*3)+(1*2.4)=2.4 \\ (0*3.2)+(0*3)+(1*2.4)=2.4 \end{bmatrix}$$

==>

$$XB = \begin{bmatrix} 3.2 \\ 3.2 \\ 3.2 \\ 3.2 \\ 3.2 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 2.4 \\ 2.4 \\ 2.4 \\ 2.4 \\ 2.4 \end{bmatrix}$$

Now we can subtract XB from Y:

$$
Y - XB = \begin{bmatrix} (4-3.2) = .8 \\ (3-3.2) = -.2 \\ (4-3.2) = .8 \\ (2-3.2) = -1.2 \\ (3-3.2) = -.2 \\ (4-3) = 1 \\ (3-3) = 0 \\ (3-3) = 0 \\ (2-3) = -1 \\ (3-3) = 0 \\ (2-2.4) = -.4 \\ (3-2.4) = .6 \\ (2-2.4) = -.4 \\ (2-2.4) = -.4 \\ (3-2.4) = .6 \end{bmatrix}
$$

Our final matrix for e is:

$$
e = \begin{bmatrix} .8 \\ -.2 \\ .8 \\ -1.2 \\ -.2 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \\ -.4 \\ .6 \\ -.4 \\ -.4 \\ .6 \end{bmatrix}
$$

```
X <- matrix(data = c(1, 0, 0,
                     1, 0, 0,
                     1, 0, 0,
                     1, 0, 0,
                     1, 0, 0,
                     0, 1, 0,
                     0, 1, 0,
                     0, 1, 0,
                     0, 1, 0,
                     0, 1, 0,
                     0, 0, 1,
                     0, 0, 1,
                     0, 0, 1,
                     0, 0, 1,
                     0, 0, 1),
            nrow = 15, ncol = 3, byrow = TRUE)
```

```r
Y <- matrix(data = c(4, 3, 4, 2, 3,
                     4, 3, 3, 2, 3,
                     2, 3, 2, 2, 3),
            nrow = 15, ncol = 1, byrow = TRUE)

MASS::ginv(X) #confirms my X' was correct
```

Step 4: Check our work using R.

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]  0.2  0.2  0.2  0.2  0.2  0.0  0.0  0.0  0.0   0.0   0.0   0.0   0.0   0.0
## [2,]  0.0  0.0  0.0  0.0  0.0  0.2  0.2  0.2  0.2   0.2   0.0   0.0   0.0   0.0
## [3,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0   0.0   0.2   0.2   0.2   0.2
##      [,15]
## [1,]   0.0
## [2,]   0.0
## [3,]   0.2
```

```r
B <- ginv(X)%*%Y
B #confirms my B was correct
```

```
##      [,1]
## [1,]  3.2
## [2,]  3.0
## [3,]  2.4
```

```r
e <- Y - (X%*%B)
e #confirms my e was correct
```

```
##       [,1]
##  [1,]  0.8
##  [2,] -0.2
##  [3,]  0.8
##  [4,] -1.2
##  [5,] -0.2
##  [6,]  1.0
##  [7,]  0.0
##  [8,]  0.0
##  [9,] -1.0
## [10,]  0.0
## [11,] -0.4
## [12,]  0.6
## [13,] -0.4
## [14,] -0.4
## [15,]  0.6
```

Question 2 (10 pts)

It turns out that the more statistics courses you have taken, the more computers you have destroyed out of frustration. Ii would like to know the equation that relates the two variables so that I can predict the number of computers future students are likely to have broken based on how many stats courses they have taken.

These two variables are stored in a datafile called `First_day_survey.csv`: `nStatsCourses` and `nCompBreak`. Using R code, run a regression predicting `nCompBreak` from `nStatsCourses`. Be sure to calculate the Bs (unstandardized coefficients), the SS-regression, the SS-error, the df-regression, the df-error, the F value for the regression model, and the p-value associated with that F value.

Step 1: Specify X, Y, and Parameters Our first step is to specify X and Y, along with our parameter estimates.

```
#import data file
q2data <- import(here("data", "First_day_survey.csv")) %>%
  janitor::clean_names()
#our Y = q2y or number of computers broken
comp_break <- matrix(data = q2data$n_comp_break,
                     nrow = 250, ncol = 1, byrow = TRUE)
#our X = stats_exp or number of stats courses taken
stats_exp <- matrix(data = c(rep(1, length(q2data$n_stats_courses)), q2data$n_stats_courses),
                    nrow = 250, ncol = 2, byrow = FALSE)
#we also need to find our parameters
q2b <- ginv(stats_exp) %*% comp_break
#we get a 2x1 matrix where [1,1] = intercept and [2,1] = slope
q2int <- q2b[1,1]
q2slope <- q2b[2,1]
#we also need an error term for our formula
q2e <- comp_break - (stats_exp%*%q2b)
```

Step 2: Define necessary terms Calculate the following terms:
* SS-regression
* SS-error
* df-regression
* df-error

```
#sum of squared total
sst <- sum((comp_break-mean(comp_break))^2)
#sum of squared error
sse <- sum(q2e^2)
#sum of squared regression
ssr <- sst - sse
```

With the code above, we've determined our SS-total is 2028.64 and SS-error is 2021.78. These two values were used to calculate SS-regression, which is 6.86.

We also know that df-regression is equal to the number of predictor variables (k) in our model, which means df-regression = 1. And df-error is equal to the number of our sample size - k - 1, which means df-error = 248.

Step 3: F-value & p-value Calculate F-value and associated p-value.

Our formula for the F statistic is $F = \frac{MS_{regression}}{MS_{Error}}$, so we need to begin by calculating MS-regression and MS-error.

```
#assign values to df-regression and df-error for our formulas
dfr <- 1
```

```r
dfe <- 248
#find MS-regression = ssr/dfr
msr <- ssr/dfr
#find MS-error = sse/dfr
mse <- sse/dfe
#F = msr/mse
f <- msr/mse
```

Our F-statistic = 0.003391903. Now we can get an associated p-value:

```r
pf(f, dfr, dfe, lower.tail = FALSE)
```

```
## [1] 0.3599472
```

Our resulting p-value is 0.3599472.

Step 4: Check our work Now we can check our work using `lm()` and `anova()`.

```r
q2mod <- lm(q2data$n_comp_break ~ q2data$n_stats_courses, data = q2data)
anova(q2mod) #all my numbers matched up.
```

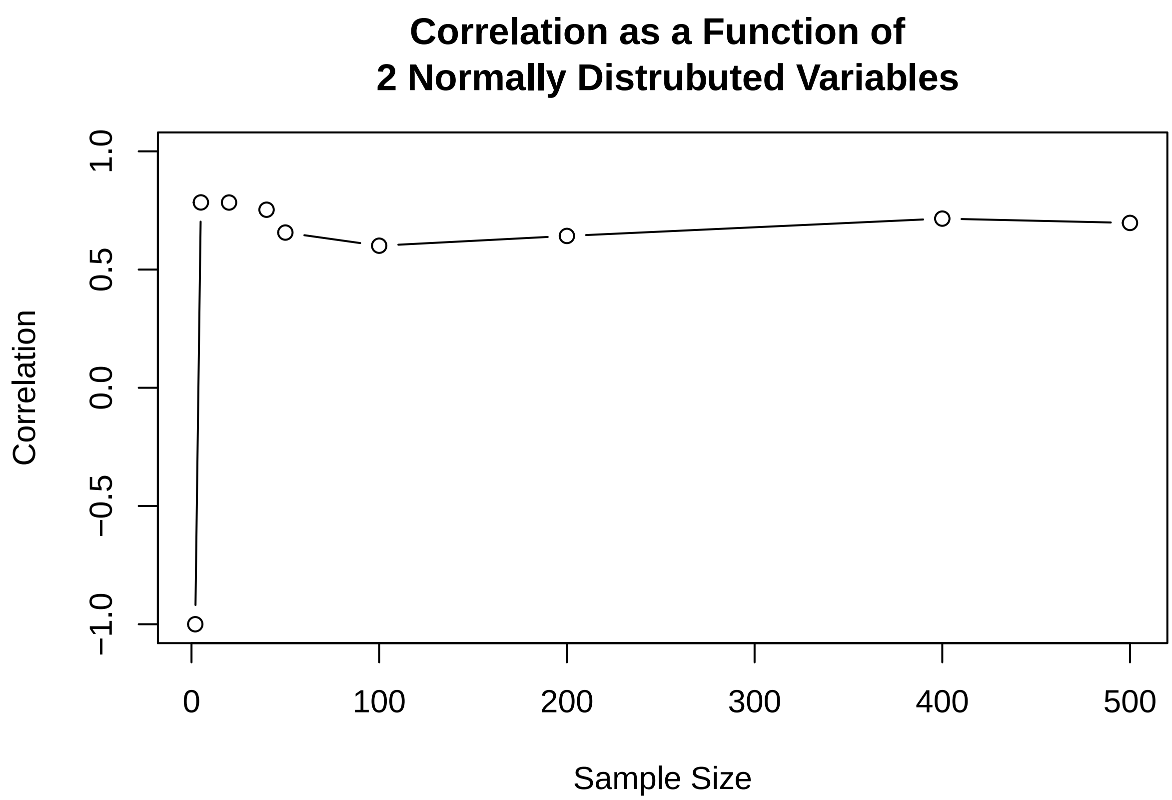
```
## Analysis of Variance Table
##
## Response: q2data$n_comp_break
##                         Df  Sum Sq Mean Sq F value Pr(>F)
## q2data$n_stats_courses   1    6.86  6.8577  0.8412 0.3599
## Residuals              248 2021.78  8.1524
```

Question 3 (20 points)

Run a simulation demonstrating the effect of sample size on correlation. Use a range of sample sizes from N=2 to N=500. For each different sample size, draw a sample from a random normal distribution (use the `rnorm()` function). Let that sample be your DV. Then create an IV that is equal to your DV plus some random noise (again using `rnorm()`). If the SD of your random noise is the same as the SD of your DV, the "true" correlation between the DV and IV should be 0.5. Calculate what the correlation is between your IV and DV for each sample size. Plot the correlations as a function of sample size.

```r
set.seed(12345)
# Set iterations
iter <- 1000
# Set sample sizes
sample_sizes = c(2, 5, 20, 40, 50, 100, 200, 400, 500)
cors <- rep(NA_real_, 9)

# Pull 1000x over 12 sample sizes, create DV and IV from each, save correlation
for (i in 1:length(sample_sizes)) { #looping over each sample size
  # Create vectors to save samples
  DV      <- rnorm(n = sample_sizes[i], mean = 50, sd = 20) # Get sample of specified sample
  IV      <- DV + rnorm(n = sample_sizes[i], mean = 10, sd = 20)
  cors[i] <- cor(DV, IV, use = "pairwise.complete.obs")
```

```
}
plot(x = sample_sizes, y = cors,
     type = "b",
     ylim = c(-1, 1),
     main = "Correlation as a Function of \n 2 Normally Distrubuted Variables",
     xlab = "Sample Size",
     ylab = "Correlation")
```



Question 4 (20 points)

Using the survey data from the first day of class (First_day_survey.csv), I would like to test whether the type of computer you use (3 categorical groups: Mac (0), PC (1), or other (2); variable=`mac_pc`) interacts with the number of stats classes you've taken (continuous variable = `n_stats_courses`) to predict your continuous comfort with statistics (`comfort`): Does the effect of previous stats experience on comfort differ as a function of what kind of computer you use?

Center the appropriate variables, create the interaction term, and run a hierarchical regression predicting `comfort` from the two main effects (`mac_pc` and `n_stats_courses`) and their interaction. ***Regardless of whether or not they are significant****, plot the simple slopes of stats experience on comfort for each type of computer (using a computer) and test the significance of at least one of them. Report all of the parameters and the R2 change test, and interpret your findings.*

Step 1: Centering First we'll center the predictor variables.

```r
#we'll center the continuous predictor; not the categorical one
q4 <- q4 %>%
  mutate(stats_ctr = n_stats_courses - mean(n_stats_courses, na.rm = TRUE),
  #also factor mac_pc while we're here
  mac_pc = factor(mac_pc,
                  levels = c("0", "1", "2"),
                  labels = c("Mac", "PC", "Other")))
#check our work
round(mean(q4$stats_ctr), 5) #mean is 0, we can be assured it's centered
```

```
## [1] 0
```

Step 2: Run regression Now we'll create the interaction term and run regression predicting `comfort`. A plot of the main effects is included below, with simple slopes color-coded for each group of computer users: Mac, PC, and Other.

```r
#make sure mac_pc is coded as a factor
levels(q4$mac_pc) #3 levels = Mac, PC, Other
```

```
## [1] "Mac"   "PC"    "Other"
```

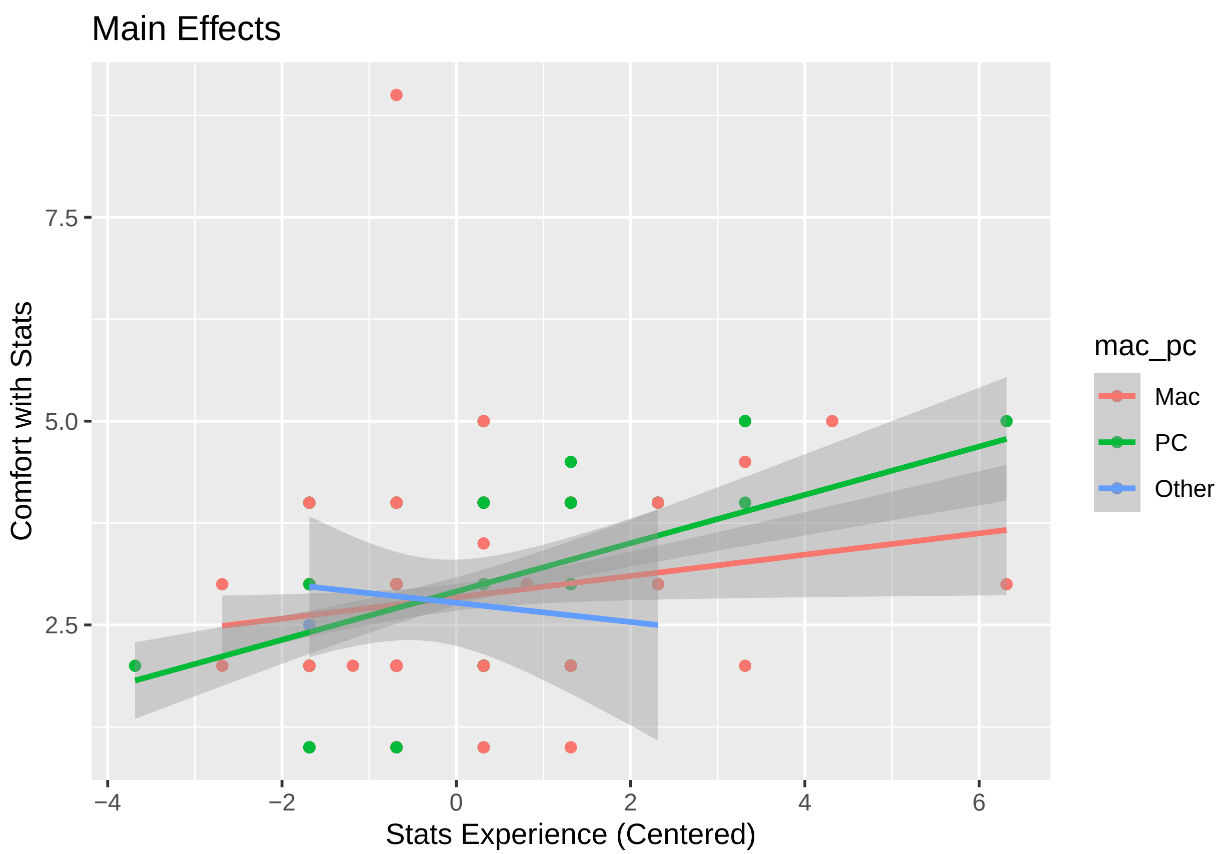
```r
comp_stats_model <- lm(comfort ~ mac_pc*stats_ctr, data = q4)
summary(comp_stats_model)
```

```
##
## Call:
## lm(formula = comfort ~ mac_pc * stats_ctr, data = q4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0111 -0.7494  0.1198  0.3814  6.2506
##
## Coefficients:
##                        Estimate Std. Error t value            Pr(>|t|)
## (Intercept)             2.83917    0.07802  36.391 <0.0000000000000002 ***
## mac_pcPC                0.07157    0.12456   0.575              0.5661
## mac_pcOther            -0.06694    0.28213  -0.237              0.8127
## stats_ctr               0.13081    0.05897   2.218              0.0275 *
## mac_pcPC:stats_ctr      0.16552    0.08820   1.877              0.0618 .
## mac_pcOther:stats_ctr -0.24846    0.26552  -0.936              0.3503
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9362 on 244 degrees of freedom
## Multiple R-squared:  0.09628,    Adjusted R-squared:  0.07777
## F-statistic: 5.199 on 5 and 244 DF,  p-value: 0.0001488
```

```r
#plot main effects
ggplot(data = q4, aes(stats_ctr, comfort, group = mac_pc, color=mac_pc)) +
  geom_point() +
  geom_smooth(method="lm") +
```

```r
  labs(x = "Stats Experience (Centered)",
       y = "Comfort with Stats",
       title = "Main Effects")
```

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



Step 3: Test simple slopes Now we can test our simple slopes, beginning by writing out our full regression formula:

$$\hat{Y}comfort = \beta_0 + \beta_1 stats + \beta_2 comp + \beta_3 * \beta_1 stats * \beta_2 comp$$

Our full formula helps inform our 3 decomposed subsequent formulas for Mac(0), PC(1), and Other(2).

Mac formula:

$$\hat{Y}comfort = \beta_0 + \beta_1 stats$$

=>

$$\hat{Y}comfort = 2.83917 + 0.13081 * stats$$

PC formula:

$$\hat{Y}comfort = \beta_0 + \beta_1 stats + \beta_2 comp + \beta_3 * \beta_1 * stats * \beta_2 comp$$

=>

$$\hat{Y}comfort = 2.83917 + 0.13081 * stats + 0.07157 * 1 + 0.16552 * 0.13081 * stats * 0.07157$$

=>

$$\hat{Y}comfort = 2.91074 + 0.13081 * stats + 0.00154961 * stats$$

=>

$$\hat{Y}comfort = 2.91074 + 0.1323596 * stats$$

Other formula:

$$\hat{Y}comfort = \beta_0 + \beta_1 stats - 0.06694 + \beta_3 * \beta_1 stats * -0.06694$$

=>

$$\hat{Y}comfort = 2.83917 + 0.13081 * stats - 0.06694 - 0.24846 * \beta_1 * stats * -0.06694$$

=>

$$\hat{Y}comfort = 2.77223 + 0.13081 * stats - 0.01663191 * stats$$

=>

$$\hat{Y}comfort = 2.77223 + 0.1141781 * stats$$

```r
#First, run significance tests:
#subset data
pc_data = subset(q4, mac_pc == "PC")
mac_data = subset(q4, mac_pc == "Mac")
other_data = subset(q4, mac_pc == "Other")
#run models for each group separately to test simple effects
pc_model = lm(comfort ~ stats_ctr, data = pc_data) # PC users
anova(pc_model) # PC summary
```

```
## Analysis of Variance Table
##
## Response: comfort
##           Df Sum Sq Mean Sq F value     Pr(>F)
## stats_ctr  1 17.888 17.8876  25.136 0.00000262 ***
## Residuals 91 64.758  0.7116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mac_model = lm(comfort ~ stats_ctr, data = mac_data) # Mac users
anova(mac_model) # Mac summary
```

```
## Analysis of Variance Table
##
## Response: comfort
##            Df  Sum Sq Mean Sq F value  Pr(>F)
## stats_ctr   1   4.313  4.3132   4.328 0.03929 *
## Residuals 142 141.513  0.9966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
other_model = lm(comfort ~ stats_ctr, data = other_data) # Other users
anova(other_model) # Other summary
```

```
## Analysis of Variance Table
##
## Response: comfort
##           Df Sum Sq Mean Sq F value Pr(>F)
## stats_ctr  1 0.1810 0.18100  0.2624 0.6186
## Residuals 11 7.5882 0.68984
```

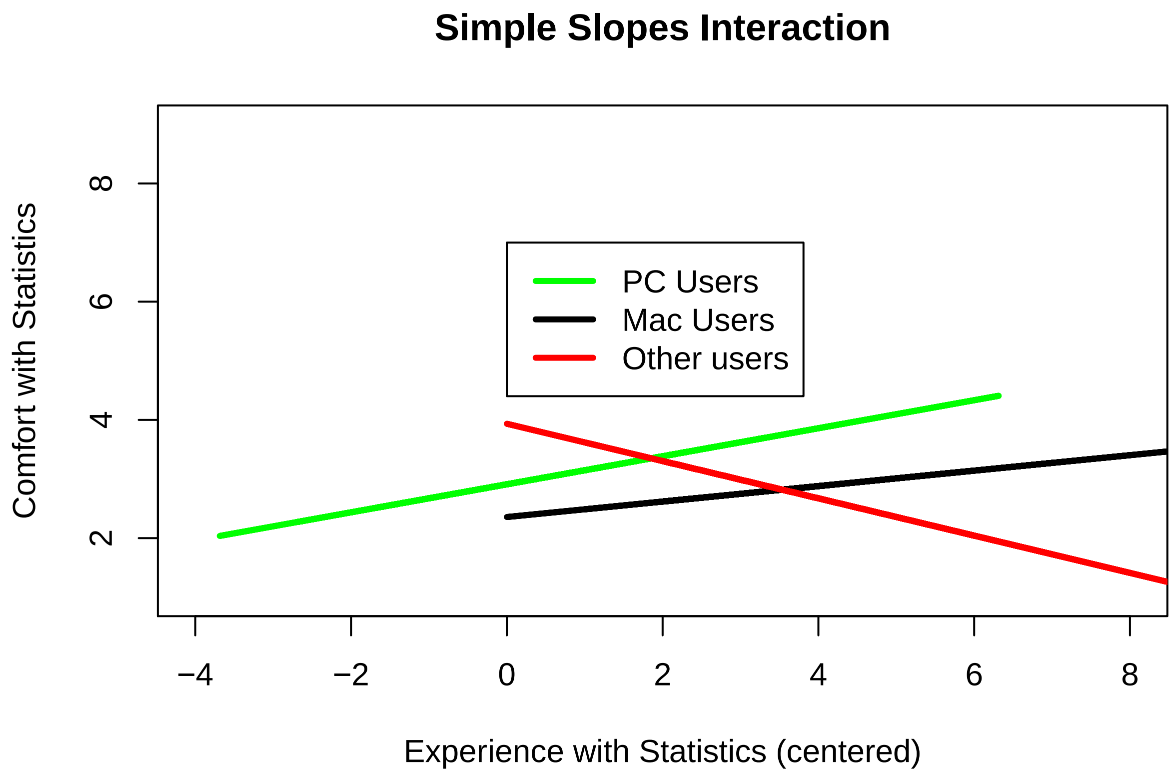
```r
#Also need adjusted R^2 for each of these
#Adjusted R^2 = SS/SS_Total
pc_r2 <- 17.888/(17.888+64.758)
mac_r2 <- 4.313/(4.313+141.513)
other_r2 <- 0.1810/(0.1810+7.5882)

#now we can build models to plot
#assign coefficients to build models
b0 = comp_stats_model$coefficients[1] #intercept
b1 = comp_stats_model$coefficients[4] #predicted change with 1 SD stats course and mac as our ref group
b2_pc = comp_stats_model$coefficients[2] #slope for pc users
b2_other = comp_stats_model$coefficients[3] #slope for other users
b3_pc = comp_stats_model$coefficients[5] #interaction for PC users
b3_other = comp_stats_model$coefficients[6] #interaction for other users
#coefficient check:
comp_stats_model$coefficients
```

```
##          (Intercept)             mac_pcPC          mac_pcOther
##           2.83917313           0.07157209          -0.06693784
##            stats_ctr    mac_pcPC:stats_ctr mac_pcOther:stats_ctr
##           0.13081331           0.16552062          -0.24846037
```

```r
#simple slope models
#at mean stats experience
pc_ss = (b0 + b2_pc) + (b2_pc*q4$stats_ctr + b3_pc*q4$stats_ctr)
mac_ss = b0 + b1*(q4$stats_ctr)
other_ss = (b0 + b2_other) + (b2_other*q4$stats_ctr + b3_other*q4$stats_ctr)
#plot
plot(q4$stats_ctr,pc_ss,type="l", col="green", lwd=3,
     xlab="Experience with Statistics (centered)",
     ylab="Comfort with Statistics",
     xlim = c(-4, 8),
     ylim = c(1, 9),
     main = "Simple Slopes Interaction")
lines(q4$n_stats_courses,mac_ss,type="l", col="black", lwd=3)
lines(q4$n_stats_courses,other_ss,type="l", col="red", lwd=3)
describe(q4$stats_ctr)
```

```
##    vars   n mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 250    0 1.37   0.31   -0.12 1.48 -3.69 6.31    10 1.11     2.91 0.09
```

```r
legend(0, 7, c("PC Users", "Mac Users", "Other users"),
       lty=c(1,1,1),
       lwd=c(3,3,3),
       col=c("green","black","red")
)
```



The plot above shows the decomposed simple slope interactions for our model.

Step 4: Report your findings A hierarchical regression analysis was run to examine the relationship between previous experience in statistics courses and type of computer (Mac, PC, or other) on students' comfort levels with statistics. Our analysis yielded a significant interaction, $F(5, 244) = 5.199, p < 0.001$ but the model explained less than 1% of the variance, adjusted Rˆ2 = 0.078.

Follow-up simple slopes significance tests were run to determine the significance of experience for Mac users, PC users, and users of other computer types. For PC users, previous experience was significant, contributing an additional 1.37 points of comfort for each standard deviation of experience, $F(1, 91) = 25.136, p < .001$ and the model explained 21.6% of the variance with an adjusted Rˆ2 = 0.2164. For Mac users, there was no significant interaction with previous experience, $F(1, 142) = 4.328, p = 0.393, adjusted R^2 = 0.0296$. For other computer users there was similarly no interaction with previous experience, $F(1,11) = 0.262, p = 0.619, adjusted Rˆ2 = 0.023.