Final Exam

Tyler Reed

12/6/2021

```
# Packages required for Chapter 1
library(knitr)
library(rstanarm)
library(readr)
library(tidyverse)
library(kableExtra)
library(tidymodels)
library(corrr)
library(viridis)
library(ggthemes)
library(patchwork)
library(car)
library(leaps)
library(glmnet)
library(parsnip)
library(probably)
library(dplyr)
library(rsample)
library(modeldata)
data("lending_club")
# library(MASS)
TipData <- as_tibble(read_csv("~/Downloads/RStudio Files STA631/STA631-Exams_Lessons-rstudio-export/exa
```

You are allowed to use your book and the internet to complete this exam. You are also able to use other written material, but please indicate a reference if you used an idea from another source. You do not have to site if you use online documentation or generally available code. If you are in doubt, site your source.

You may NOT talk to anyone in person or over any type of media about this exam. You may not ask questions of any other human. You may not help anyone else.

Honor Pledge I affirm that I did not give or receive any unauthorized help on this exam, and that all work is my own. retype the pledge:

I affirm that I did not give or receive any unauthorized help on this exam, and that all work is my own.

you can type your name below to indicate agreement with a date:

Signature: Tyler Reed

Problem 1

A student collected data from a restaurant where she was a waitress [@Dahlquist2011]. The student was interested in learning under what conditions a waitress can expect the largest tips—for example: At dinner

time or late at night? From younger or older patrons? From patrons receiving free meals? From patrons drinking alcohol? From patrons tipping with cash or credit? And should tip amount be measured as total dollar amount or as a percentage? Data can be found in `TipData.csv`. Here is a quick description of the variables collected:

- ‘Day‘ = day of the week HIST
- ‘Meal‘ = time of day (Lunch, Dinner, Late Night)
- ‘Payment‘ = how bill was paid (Credit, Cash, Credit with Cash tip)
- ‘Party‘ = number of people in the party
- ‘Age‘ = age category of person paying the bill (Yadult, Middle, SenCit)
- ‘GiftCard‘ = was gift card used?
- ‘Comps‘ = was part of the meal complimentary?
- ‘Alcohol‘ = was alcohol purchased?
- ‘Bday‘ = was a free birthday meal or treat given?
- ‘Bill‘ = total size of the bill
- ‘W.tip‘ = total amount paid (bill plus tip)
- ‘Tip‘ = amount of the tip
- ‘Tip.Percentage‘ = proportion of the bill represented by the tip

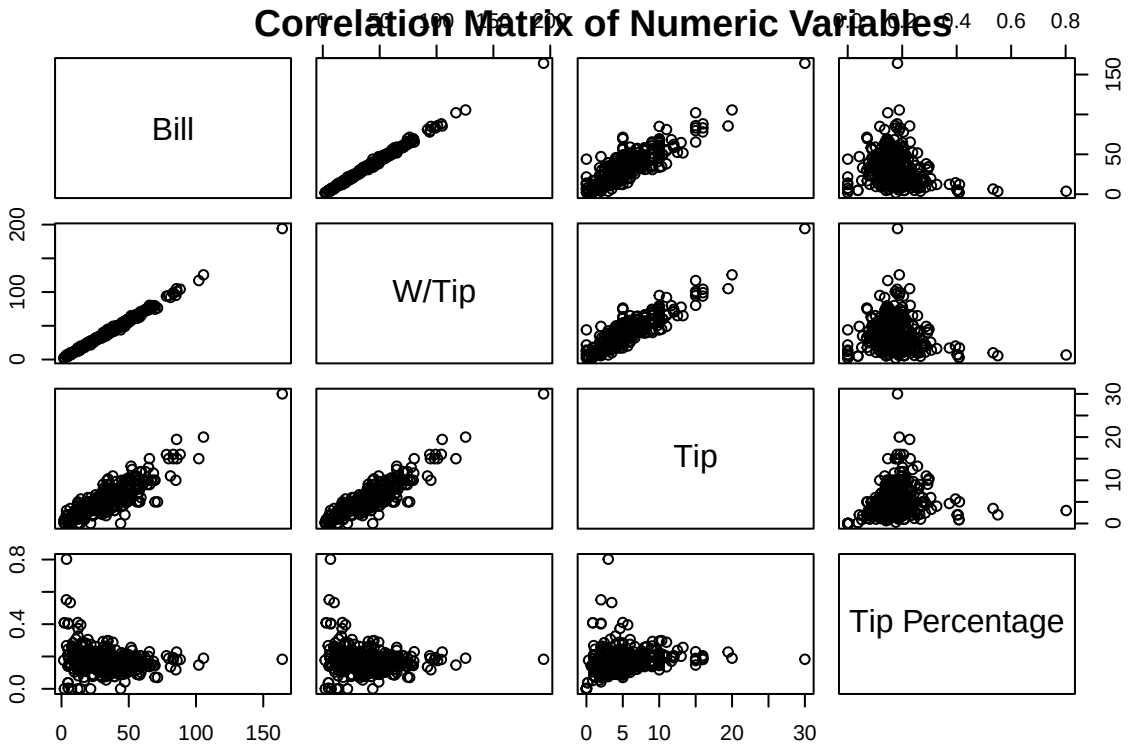
A. (10 points) Create graphics and data summaries that explore your data, keeping in mind the research questions the student waitress is trying to answer. Also give a paragraph describing what you have learned about the data based on your exploratory data analysis.

```
# summary
summary(TipData)
```

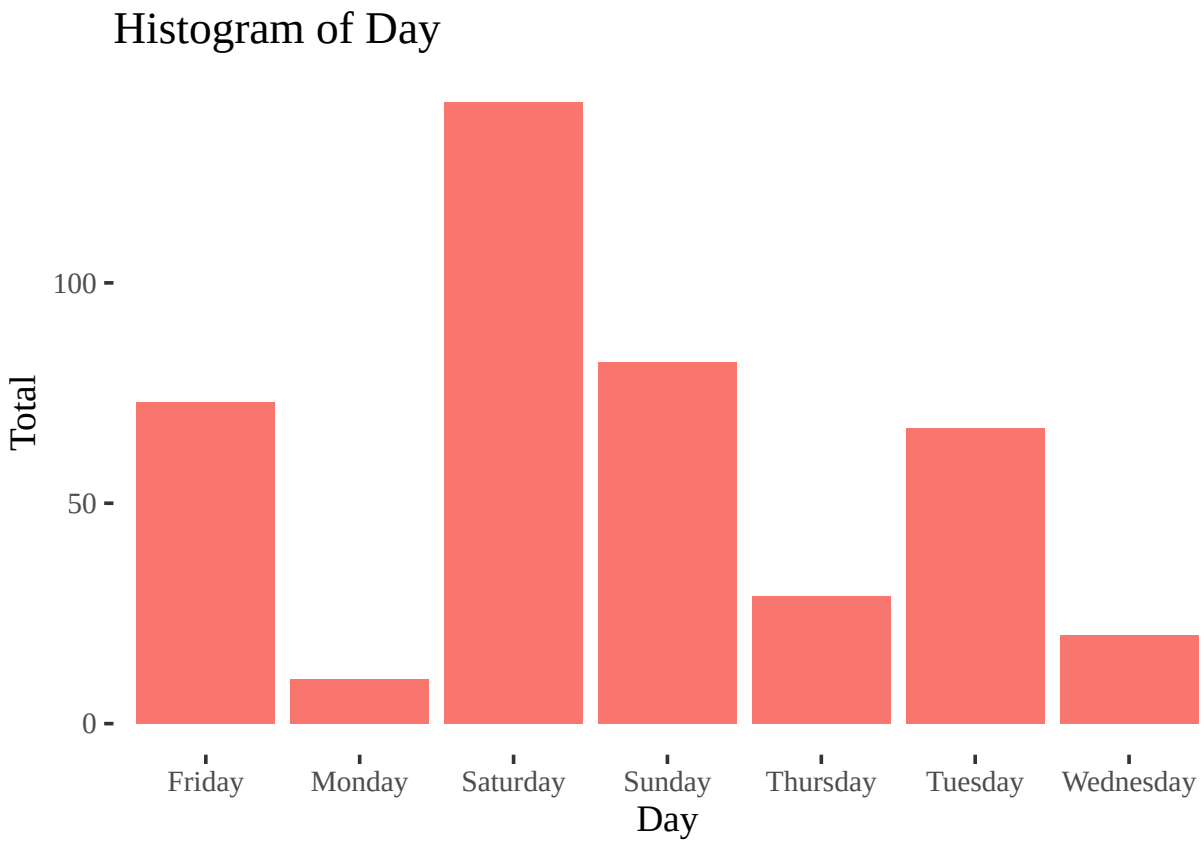
```
     Day                Meal              Payment              Party      
 Length:422         Length:422         Length:422         Min.   :1.000  
 Class :character   Class :character   Class :character   1st Qu.:1.000  
 Mode  :character   Mode  :character   Mode  :character   Median :2.000  
                                                          Mean   :2.349  
                                                          3rd Qu.:3.000  
                                                          Max.   :9.000  
                                                          NA's   :253    
     Age              GiftCard            Comps              Alcohol         
 Length:422         Length:422         Length:422         Length:422        
 Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character  
                                                                            
                                                                            
                                                                            
     Bday                Bill             W/Tip              Tip        
 Length:422         Min.   :  1.70   Min.   :  2.00   Min.   : 0.000  
 Class :character   1st Qu.: 18.00   1st Qu.: 21.68   1st Qu.: 3.000  
 Mode  :character   Median : 29.27   Median : 33.80   Median : 4.925  
                    Mean   : 31.65   Mean   : 36.81   Mean   : 5.166  
                    3rd Qu.: 42.16   3rd Qu.: 48.00   3rd Qu.: 6.175  
                    Max.   :164.13   Max.   :194.13   Max.   :30.000  

 Tip Percentage   
 Min.   :0.0000   
 1st Qu.:0.1356   
```

```
 Median :0.1662
 Mean   :0.1728
 3rd Qu.:0.2023
 Max.   :0.8021
```

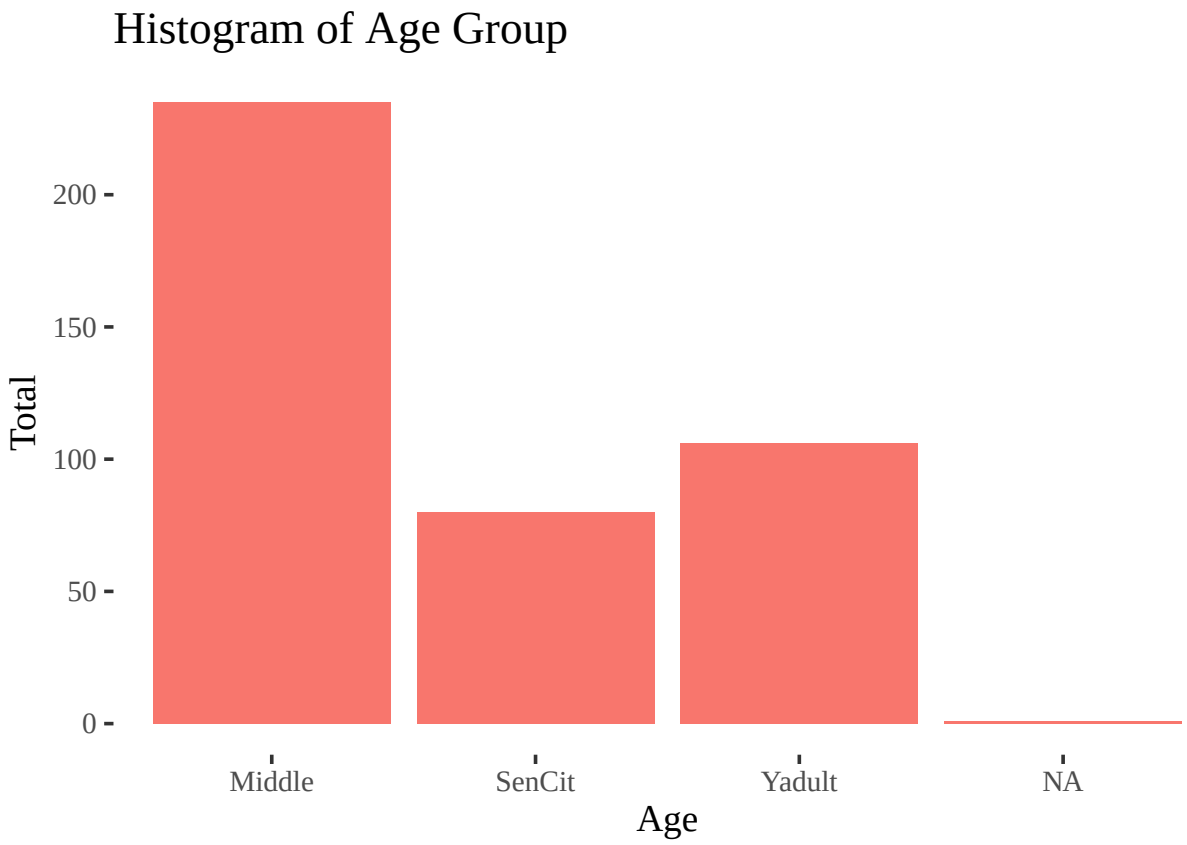
```r
# cor matrix for numeric variables
Tip_Data_num <- TipData %>%
                      select(Bill, "W/Tip", Tip, `Tip Percentage`)
pairs(Tip_Data_num)
title("Correlation Matrix of Numeric Variables")
```



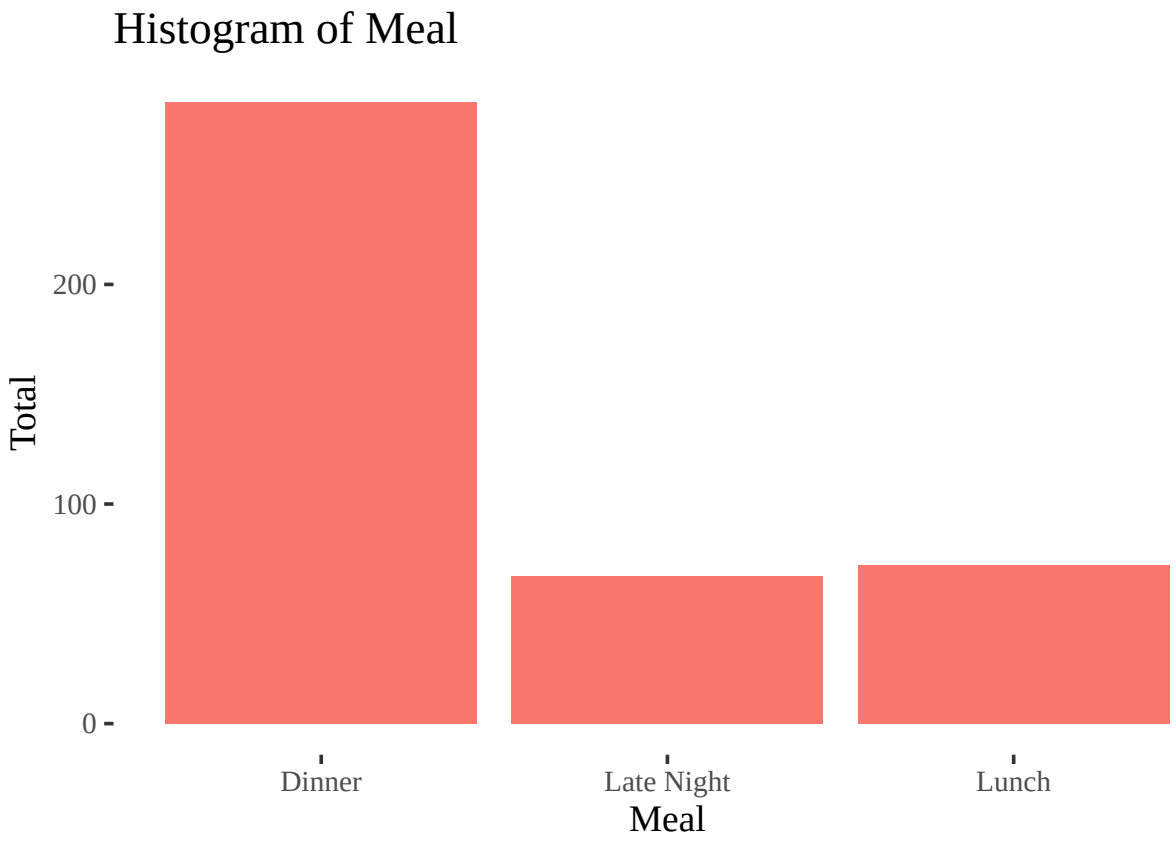

```r
# non-numeric variables
TipData %>%
  ggplot(aes(x=Day, fill="Red")) +
  geom_histogram(stat = "Count") +
  labs(x = "Day",
       y = "Total",
       title = "Histogram of Day") +
  theme_tufte(base_size = 14) +
  theme(legend.position = "none")
```



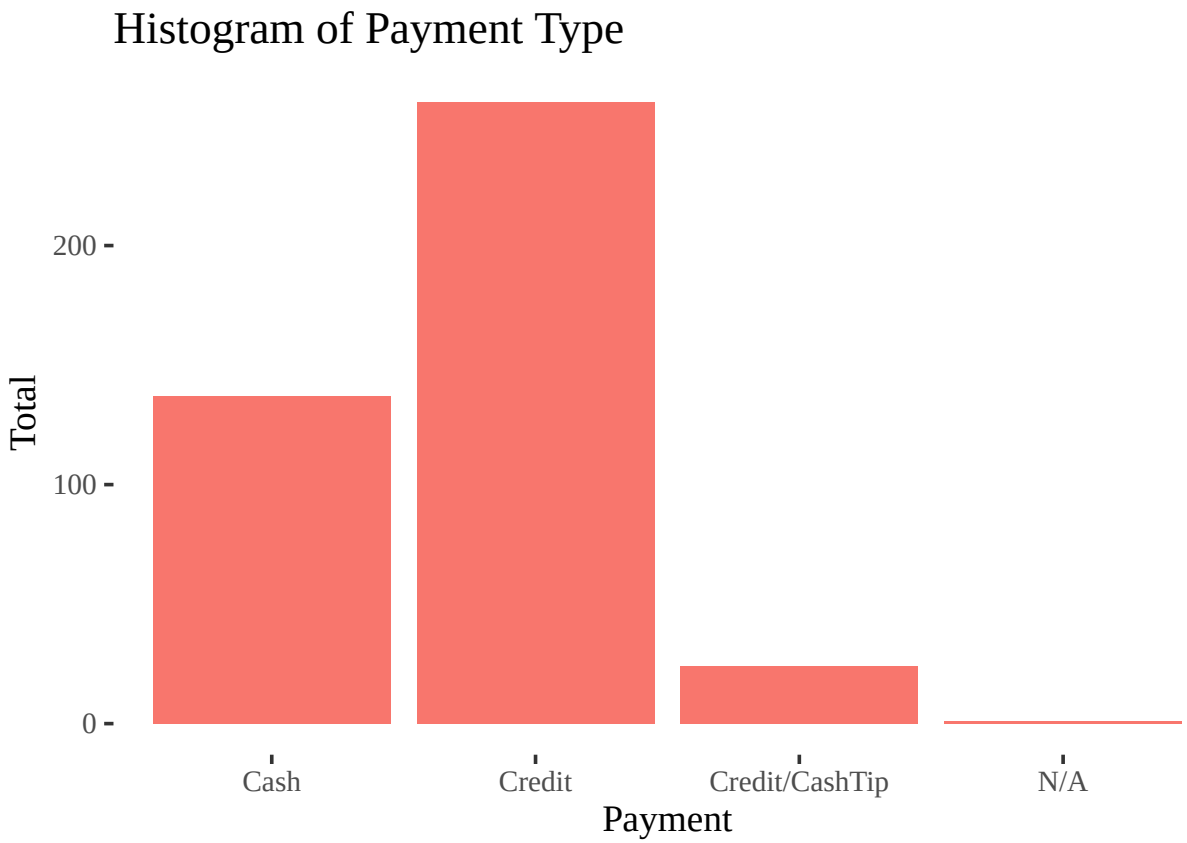

```
TipData %>%
  ggplot(aes(x=Age, fill = "Red")) +
  geom_histogram(stat = "Count") +
  labs(x = "Age",
       y = "Total",
       title = "Histogram of Age Group") +
  theme_tufte(base_size = 14) +
  theme(legend.position = "none")
```




```
TipData %>%
  ggplot(aes(x=Meal, fill = "Red")) +
  geom_histogram(stat = "Count") +
  labs(x = "Meal",
       y = "Total",
       title = "Histogram of Meal") +
  theme_tufte(base_size = 14) +
  theme(legend.position = "none")
```




```
TipData %>%
  ggplot(aes(x=Payment, fill = "Red")) +
  geom_histogram(stat = "Count") +
  labs(x = "Payment",
       y = "Total",
       title = "Histogram of Payment Type") +
  theme_tufte(base_size = 14) +
  theme(legend.position = "none")
```




```
# clean data by removing `Party` and NAs

TipData_clean <- TipData %>%
                        select(-Party) %>%
                        drop_na()

correlate(TipData_clean$Tip, TipData_clean$`Tip Percentage`)
```

```
# A tibble: 1 x 2
  term      x
  <chr> <dbl>
1 x     0.159
```

Summary Findings

Qualitative

`Party` has 253 NAs, which is more than half data length. `Day` is skewed heavily towards the weekend, with Saturdays being the busiest. `Age` the highest count is among Middle. `Meal` is outweighed by Dinner with Dinner having more than double the count of the other two meals combined. `Payment` displays a vast majority of users paying with Credit.

Upon visual inspection of the binary variables, none seem to have any issues unless the `Comps` was the resoponse variable for a logistic regression. Then the 9 values of "Yes" may not be enough for accurate parameter estimation.

Quantitative

The correlation matrix displays little of note. The variables behave as expected with `Tip`, `Bill`, and `W/Tip` all being strongly positively correlated. However,`Tip.Percentage` is quite constant across the all sizes of `Bill` except for when the `Bill` is between 0 and 10 dollars, roughly. Investigating the relationships between `Bill` and `Tip.Percentage` and `Bill` and `Tip` seems to be the most intriguing.

B. (5 points) Write a paragraph discussing any issues of measurement in the data.

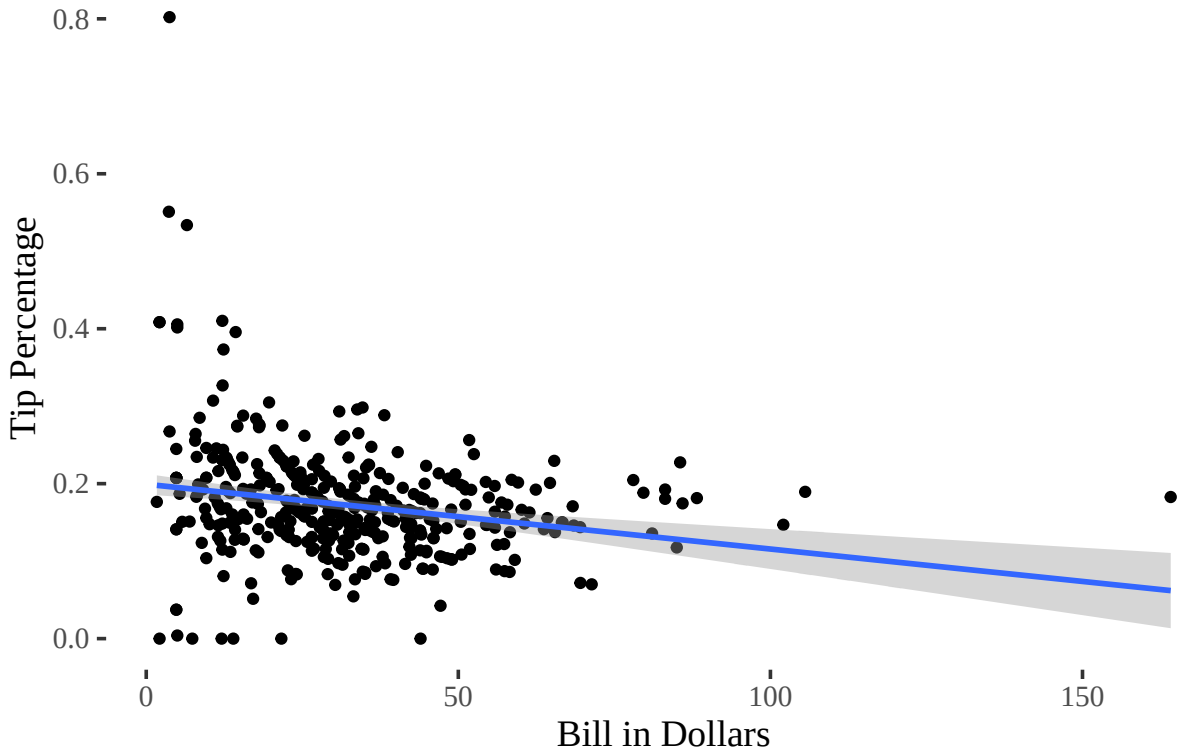
The main issue is that `Party` has 253 NAs, which is more than half data length. Dropping `Party` should be considered. As stated above, if `Comps` was the resoponse variable for a logistic regression. Then the 9 values of "Yes" may not be enough for accurate parameter estimation.

C. (5 points) Fit a linear regression with a single quantitative predictor, you can pick the predictor and one of tip or tip.percentage as the response, use the same response you select for all problems below. Graph the data along with the fitted line. Interpret the estimated parameters and their uncertainties. (You can write just one sentence interpreting each parameter in the model.)

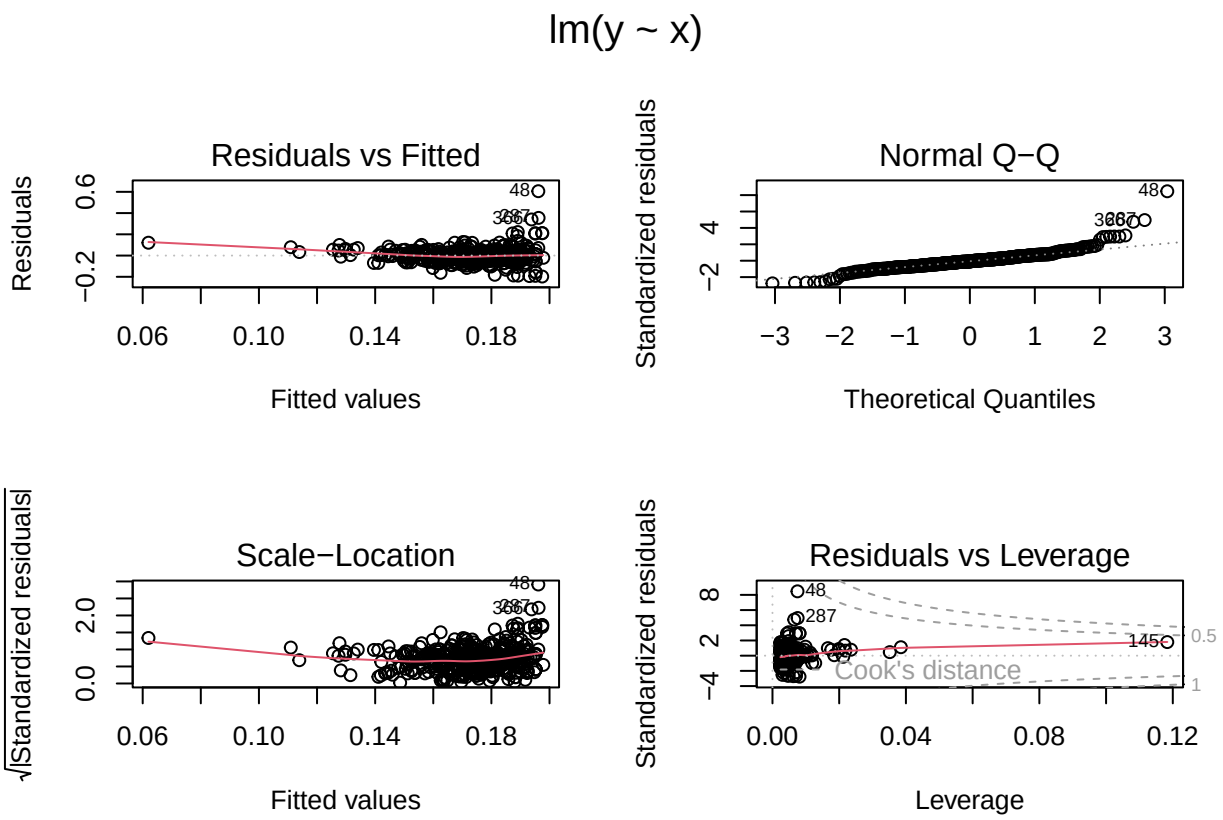
```
x <- TipData_clean$Bill
y <- TipData_clean$`Tip Percentage`

fit_1 <- lm(y ~ x, data=TipData_clean)
summary(fit_1)
```

```
Call:
lm(formula = y ~ x, data = TipData_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.19748 -0.03908 -0.00569  0.03019  0.60599

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1992741  0.0068030  29.292  < 2e-16 ***
x           -0.0008364  0.0001846  -4.531 7.66e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07184 on 419 degrees of freedom
Multiple R-squared:  0.04671,   Adjusted R-squared:  0.04444
F-statistic: 20.53 on 1 and 419 DF,  p-value: 7.657e-06
```

```
TipData_clean %>%
  ggplot(aes(x=Bill, y=`Tip Percentage`)) +
  geom_point() +
  labs(x = "Bill in Dollars",
       y = "Tip Percentage",
       title = "Scatter plot of Bill and Tip Percentage with Best Fit Line") +
  theme_tufte(base_size = 14) +
  theme(legend.position = "none") +
  geom_smooth(method = "lm")
```




```
par(mfrow = c(2, 2), oma = c(0, 0, 2, 0)) -> opar # derived from https://www.stat.auckland.ac.nz/~ihaka,
plot(fit_1)
```



At an intercept of 0.199, when the bill is 0 dollars, the tip percentage will be 0.199%, on average.

For each additional dollar increase for the bill, on average, the tip percentage will decrease by a factor of -0.0008364.

D. (5 points) Fit a linear regression with two predictors and an interaction. The model should make sense; that is, there should be a good applied reason for fitting it. Explain each of the estimated parameters and their uncertainties, using one sentence for each parameter.

```
x1 <- TipData_clean$Bill
x2 <- TipData_clean$Alcohol
y <- TipData_clean$`Tip Percentage`

fit_2 <- lm(y ~ x1 + x2 + x1*x2, data=TipData_clean)
summary(fit_2)
```

```
Call:
lm(formula = y ~ x1 + x2 + x1 * x2, data = TipData_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.20271 -0.04009 -0.00519  0.03290  0.60097

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.2047861  0.0082018  24.969  < 2e-16 ***
x1          -0.0009680  0.0002482  -3.900 0.000112 ***
```

```
x2Yes       -0.0203320  0.0154859  -1.313 0.189927
x1:x2Yes     0.0004155  0.0003887   1.069 0.285735
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07186 on 417 degrees of freedom
Multiple R-squared:  0.05068,   Adjusted R-squared:  0.04385
F-statistic:  7.42 on 3 and 417 DF,  p-value: 7.483e-05
```

At an intercept of 0.204, when the bill is 0 dollars, the tip percentage will be 0.204%, on average.

Adjusting for Alcohol, for each additional dollar increase for the bill, on average, the tip percentage will decrease by a factor of -0.0009680.

Adjusting for the bill amount, tip percentage from patrons buying alcohol will be 0.0203% less than patrons not purchasing alcohol.

For patrons purchasing alcohol, the effect of `Bill` on `Tip Percentage` is -0.00968 + (0.000416*1) = -0.009264. For two patrons purchasing alcohol, we expect a patron paying one dollar more on the bill to have 0.009264 tip percentage less than those patrons paying less on their bill.

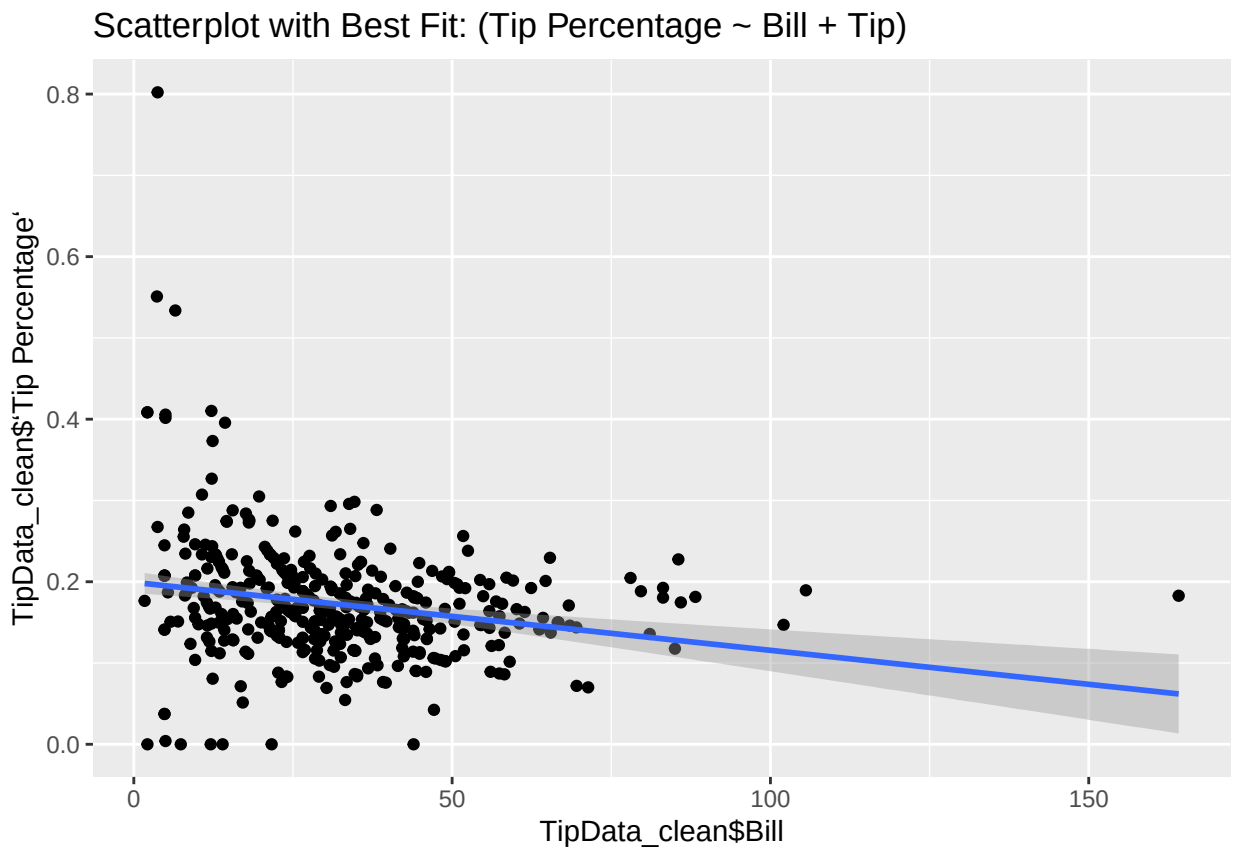
Note: the model itself is significant, but neither the interaction term or `Alcohol` variable are with their standard errors almost the size of the point estimates.

E. (5 points) Fit a linear regression with multiple predictors and get diagnostic plots. List the assumptions of the model and explain, in one sentence each, if these are reasonable here for this model.

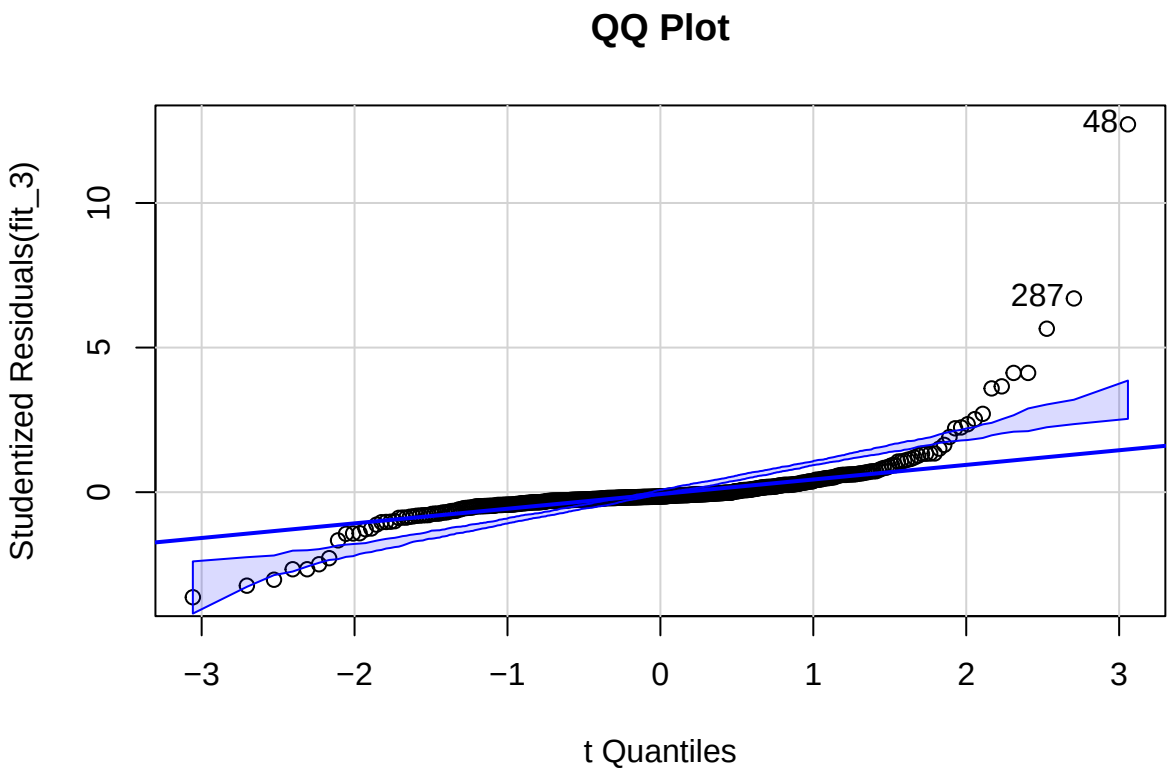
```
x1 <- TipData_clean$Bill
x2 <- TipData_clean$Tip
y <- TipData_clean$`Tip Percentage`

fit_3 <- lm(y ~ x1 + x2, data=TipData_clean)
summary(fit_3)
```

```
Call:
lm(formula = y ~ x1 + x2, data = TipData_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17928 -0.01459 -0.00760  0.00813  0.54029

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1911728  0.0047764   40.02   <2e-16 ***
x1          -0.0055597  0.0002601  -21.37   <2e-16 ***
x2           0.0304901  0.0014575   20.92   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05027 on 418 degrees of freedom
Multiple R-squared:  0.5343,    Adjusted R-squared:  0.5321
F-statistic: 239.8 on 2 and 418 DF,  p-value: < 2.2e-16
```

```
qplot(x = TipData_clean$Bill, y = TipData_clean$`Tip Percentage`, data = TipData_clean) +
  geom_smooth(method = "lm") +
  labs(title = "Scatterplot with Best Fit: (Tip Percentage ~ Bill + Tip)")
```




```
# DIAGNOSTIC PLOTS: adapted from https://www.statmethods.net/stats/rdiagnostics.html

# Normality of Residuals
# qq plot for studentized resid
qqPlot(fit_3, main="QQ Plot")
```

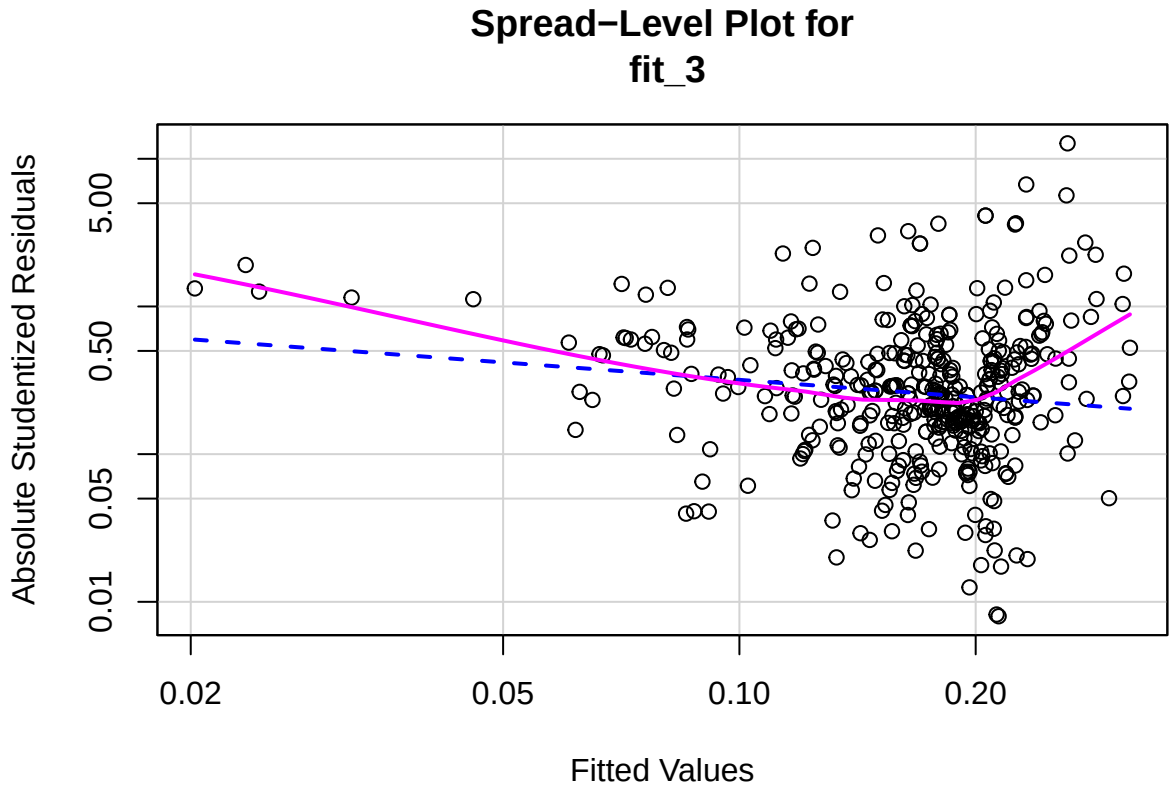



```
[1]  48 287
```

```
# Evaluate homoscedasticity
# non-constant error variance test
ncvTest(fit_3)
```

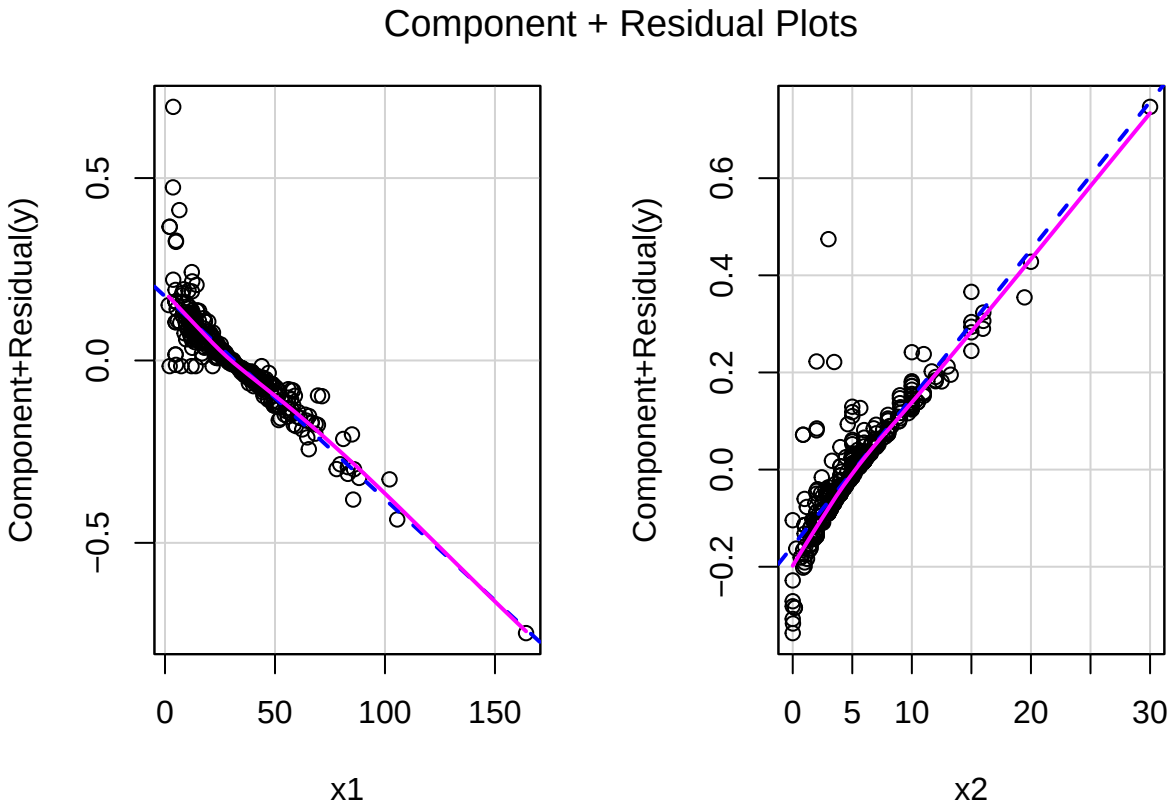
```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 90.57465, Df = 1, p = < 2.22e-16
```

```
# plot studentized residuals vs. fitted values
spreadLevelPlot(fit_3)
```




```
Suggested power transformation:  1.392885
```

```
# Evaluate Nonlinearity
# component + residual plot
crPlots(fit_3)
```




```
# Evaluate Collinearity
vif(fit_3) # variance inflation factors
```

```
      x1       x2
4.055107 4.055107
```

```
sqrt(vif(fit_3)) > 2 # problem?
```

```
  x1   x2
TRUE TRUE
```

Multiple Linear Regression Assumptions

Linearity may not hold. Most of the data lies within a linear relationship except for the ends of the range as seen on the scatterplot

Multivariate Normality does not seem to hold with the QQ-plot as the tails are quite a ways off the fit line with an S-like shape.

Multicollinearity is possible here with a VIF above 2. A transformation may remedy the issue.

F. (10 points) Fit two different linear regressions, each with multiple predictors. Both models should make sense; that is, there should be good applied reasons for fitting them. Compare the fits using five-fold cross validation, and in one sentence discuss what you found.

```r
x1a <- TipData_clean$Bill
x2a <- TipData_clean$Alcohol
y_a <- TipData_clean$`Tip Percentage`

fit_a <- lm(y_a ~ x1a + x2a, data=TipData_clean)
summary(fit_2)
```

```
Call:
lm(formula = y ~ x1 + x2 + x1 * x2, data = TipData_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.20271 -0.04009 -0.00519  0.03290  0.60097

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.2047861  0.0082018  24.969  < 2e-16 ***
x1          -0.0009680  0.0002482  -3.900 0.000112 ***
x2Yes       -0.0203320  0.0154859  -1.313 0.189927
x1:x2Yes     0.0004155  0.0003887   1.069 0.285735
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07186 on 417 degrees of freedom
Multiple R-squared:  0.05068,   Adjusted R-squared:  0.04385
F-statistic:  7.42 on 3 and 417 DF,  p-value: 7.483e-05
```

```r
x1b <- TipData_clean$Bill
x2b <- TipData_clean$Bday
y_b <- TipData_clean$`Tip Percentage`

fit_b <- lm(y_b ~ x1b + x2b, data=TipData_clean)
summary(fit_3)
```

```
Call:
lm(formula = y ~ x1 + x2, data = TipData_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17928 -0.01459 -0.00760  0.00813  0.54029

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1911728  0.0047764   40.02   <2e-16 ***
x1          -0.0055597  0.0002601  -21.37   <2e-16 ***
x2           0.0304901  0.0014575   20.92   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05027 on 418 degrees of freedom
Multiple R-squared:  0.5343,    Adjusted R-squared:  0.5321
F-statistic: 239.8 on 2 and 418 DF,  p-value: < 2.2e-16
```

```
# 5-fold CV fit_a

library(boot)
set.seed(1)
cv.error.5 <- rep(0, 5)
for (i in 1:5) {
  glm.fit <- glm(y_a ~ x1a + x2a, data = TipData_clean)
  cv.error.5[i] <- cv.glm(TipData_clean, glm.fit, K = 5)$delta[1]
}
print("fit_a MSE vector")
```

```
[1] "fit_a MSE vector"
```

```
print(cv.error.5, str(mean(cv.error.5)))
```

```
 num 0.00567
[1] 0.005588153 0.005721600 0.005704039 0.005641945 0.005709180
```

```
# 5-fold CV fit_b

library(boot)
set.seed(1)
cv.error.5 <- rep(0, 5)
for (i in 1:5) {
  glm.fit <- glm(y_b ~ x1b + x2b, data = TipData_clean)
  cv.error.5[i] <- cv.glm(TipData_clean, glm.fit, K = 5)$delta[1]
}
print("fit_b MSE vector")
```

```
[1] "fit_b MSE vector"
```

```
print(cv.error.5, str(mean(cv.error.5)))
```

```
 num 0.0057
[1] 0.005633451 0.005713442 0.005708260 0.005660931 0.005763895
```

The 5-fold cross-validation shows the two models being almost identical in average MSE error across each fold with fit_a having an average of 0.00567 and fit_b, 0.0057. This makes sense as `Alcohol` and `Bday` have comparable effects on `Bill`. Also, the adjusted R-square values are close with 0.04385 for fit_a and 0.04882 for fit_b.

G. (10 points) Divide the data into training and test. Fit a linear regression model with no interactions and using all available predictors that make sense to use. Use ordinary least squares to fit the model with the training data, and then estimate MSE using the test data.

```
set.seed(1)
split1<- sample(c(rep(0, 0.7 * nrow(TipData_clean)), rep(1, 0.3 * nrow(TipData_clean)))) # adapted from

TipData_train<- TipData_clean %>%
                 slice(-c(421)) %>%
                 mutate(split1 = split1)
```

```
train <- TipData_train[split1 == 0, ]
test <- TipData_train[split1 == 1, ]

x1 <- train$Bill
x2 <- train$Tip
x3 <- train$Alcohol
y <- train$`Tip Percentage`

fit_5 <- lm(y ~ x1 + x2 + x3, data=train)
summary(fit_5)
```

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.18128 -0.01725 -0.00758  0.00876  0.53659

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1936657  0.0059543  32.525   <2e-16 ***
x1          -0.0057897  0.0003312 -17.481   <2e-16 ***
x2           0.0311780  0.0018016  17.306   <2e-16 ***
x3Yes        0.0050897  0.0068616   0.742    0.459
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05307 on 290 degrees of freedom
Multiple R-squared:  0.5321,    Adjusted R-squared:  0.5273
F-statistic: 109.9 on 3 and 290 DF,  p-value: < 2.2e-16
```

```
# MSE
(ols_mse <- mean((y - predict.lm(fit_5, test)) ^ 2))
```

```
[1] 0.002778002
```

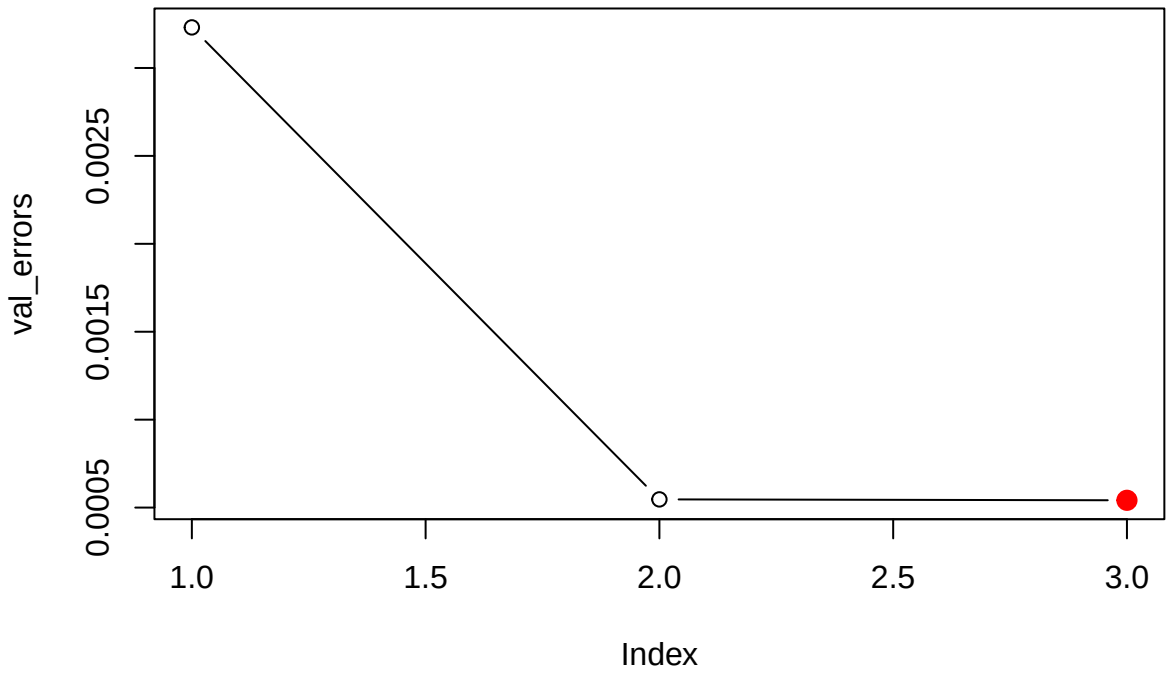
H. (20 points) Using the training data and cross-validation select two methods that we talked about in Chapter 6 of ISL, to fit a model. Then estimate MSE using the test data. Which of the three methods OLS, and the two used here seem to fit the data best?

```
# detach(package:MASS, unload = TRUE)
TipData_train <- TipData_train %>%
                        select(`Tip Percentage`, Bill, Tip, Alcohol)

train_t <- TipData_train[1:4] %>%
    sample_frac(0.75)

test_t <- TipData_train[1:4] %>%
    setdiff(train_t)

regfit_best_train <- regsubsets(`Tip Percentage`~., data=train_t, nvmax = 3)
```

```
test_mat <- model.matrix(`Tip Percentage`~., data=test_t)

val_errors <- rep(NA,3)

for(i in 1:3){
  coefi <- coef(regfit_best_train, id = i)
  pred <- test_mat[,names(coefi)]%*%coefi
  val_errors[i] <- mean((test_t$`Tip Percentage`-pred)^2)
}

min <- which.min(val_errors)
min
```

```
[1] 3
```

```
plot(val_errors, type = 'b')
points(min, val_errors[min][1], col = "red", cex = 2, pch = 20)
```




```
k <- 5
set.seed(1)

folds <- sample(1:k, nrow(train_t), replace = TRUE)
cv_errors <- matrix(NA, k, 3, dimnames = list(NULL, paste(1:3)))
```

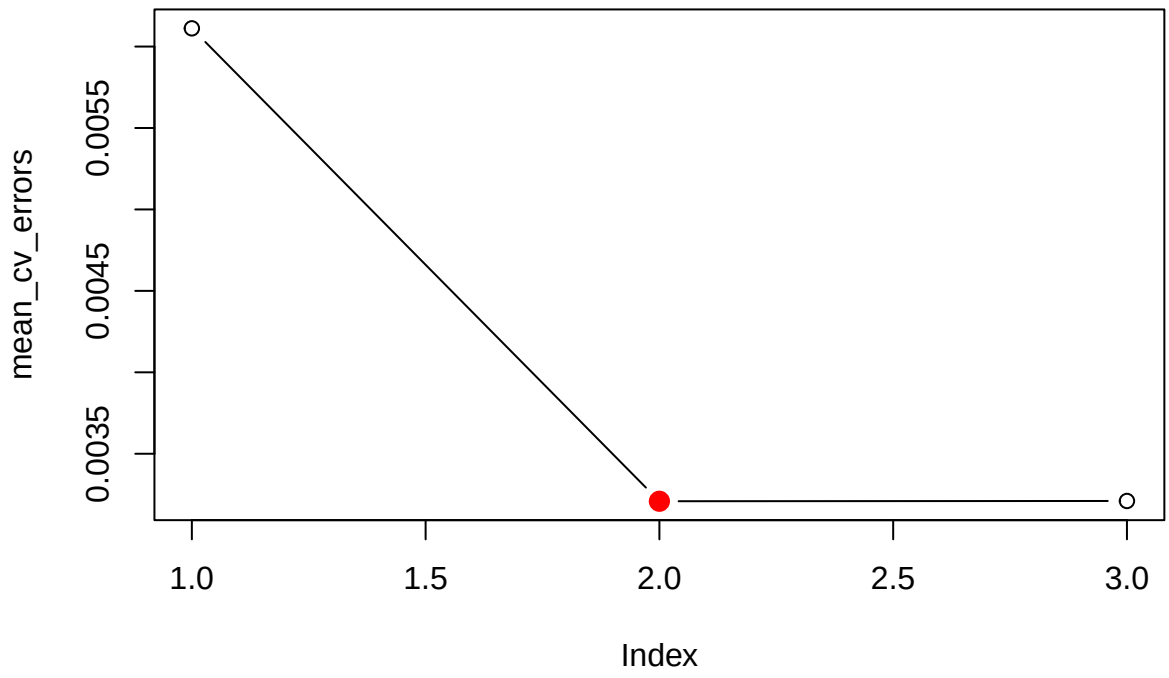
```
predict.subregsubsets <- function(object, newdata, id){
    form <- as.formula(object$call[[2]])
    mat <- model.matrix(form, newdata)
    coefi <- coef(object, id=id)
    xvars <- names(coefi)
    mat[,xvars]%*%coefi
}

for(j in 1:k) {
  best_fit <- regsubsets(`Tip Percentage`~., data = train_t[folds !=j, 1:4], nvmax = 3)
  for(i in 1:3) {
    pred = predict.subregsubsets(best_fit, train_t[folds==j, 1:4], id=i)
    cv_errors[j,i] <- mean((train_t$`Tip Percentage`[folds==j]-pred)^2)
  }
}

(mean_cv_errors <- apply(cv_errors, 2, mean))
```

```
          1           2           3
0.006111775 0.003208553 0.003210180
```

```
(min <- which.min(mean_cv_errors))
```

```
2
2
```

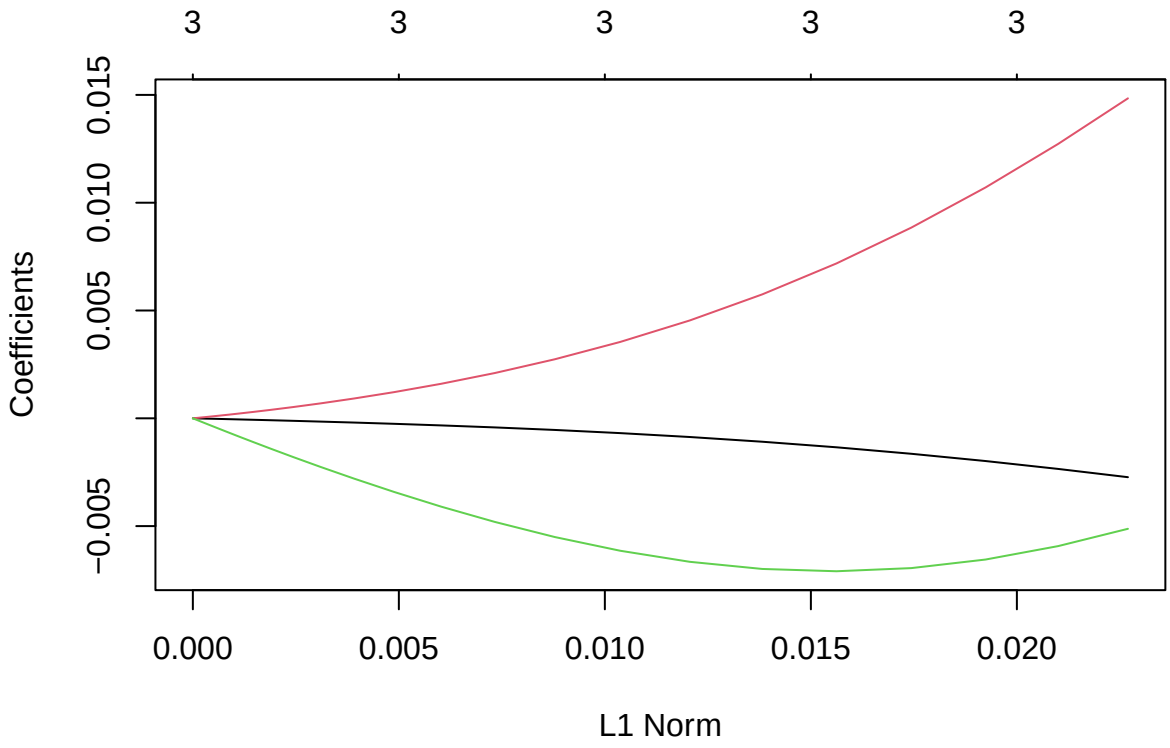
```
plot(mean_cv_errors, type='b')
points(min, mean_cv_errors[min][1], col = 'red', cex = 2, pch = 20)
```




```
reg_best <- regsubsets(`Tip Percentage`~., data= TipData_train[1:4], nvmax = 2)
coef(reg_best, 2)
```

```
(Intercept)        Bill         Tip
 0.19118364 -0.00555899  0.03048625
```

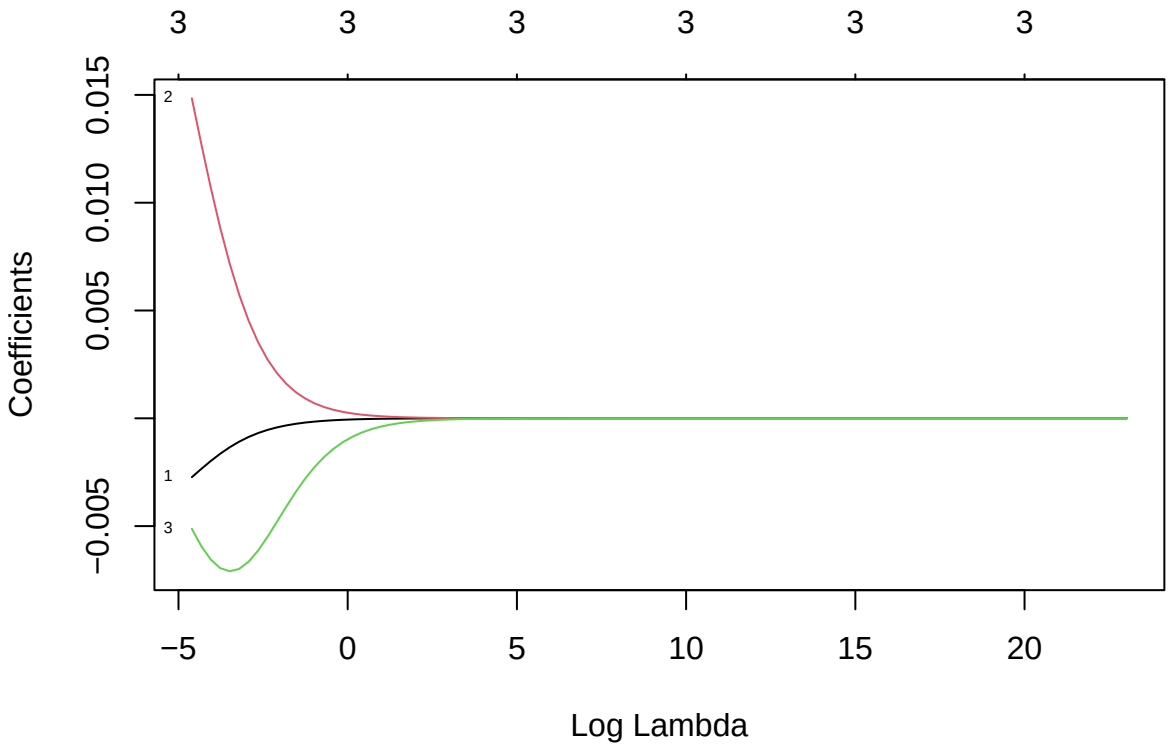
```
# ridge reg
x <- model.matrix(`Tip Percentage`~., TipData_train)[,-1]
y <- TipData_train$`Tip Percentage`

grid <- 10^seq(10, -2, length = 100)
ridge_mod <- glmnet(x, y, alpha = 0, lambda=grid)

plot(ridge_mod)
```




```
dim(coef(ridge_mod))
```

```
[1]   4 100
```

```
 # Draw plot of coefficients

plot(ridge_mod, xvar = "lambda", label = TRUE)
```




```
ridge_mod$lambda[50] #Display 50th lambda value
```

```
[1] 11497.57
```

```
coef(ridge_mod)[,50] # Display coefficients associated with 50th lambda value
```

```
  (Intercept)          Bill           Tip    AlcoholYes
 1.729213e-01 -5.340767e-09  2.204286e-08 -9.244847e-08
```

```
sqrt(sum(coef(ridge_mod)[-1,50]^2)) # Calculate l2 norm
```

```
[1] 9.518998e-08
```

```
predict(ridge_mod, s = 50, type = "coefficients")[1:4,]
```

```
  (Intercept)          Bill           Tip    AlcoholYes
 1.729409e-01 -1.251019e-06  5.170027e-06 -2.163257e-05
```

```
x_train <- model.matrix(`Tip Percentage`~., train_t)[,-1]
x_test <- model.matrix(`Tip Percentage`~., test_t)[,-1]
y_train <- train_t$`Tip Percentage`
y_test <- test_t$`Tip Percentage`
```

```
ridge_mod = glmnet(x_train, y_train, alpha=0, lambda = grid, thresh = 1e-12)
ridge_pred = predict(ridge_mod, s = 4, newx = x_test)
(ridge_mse <- mean((ridge_pred - y_test)^2))
```

```
[1] 0.003247498
```

```
# compare to OLS
ols_mse
```

```
[1] 0.002778002
```

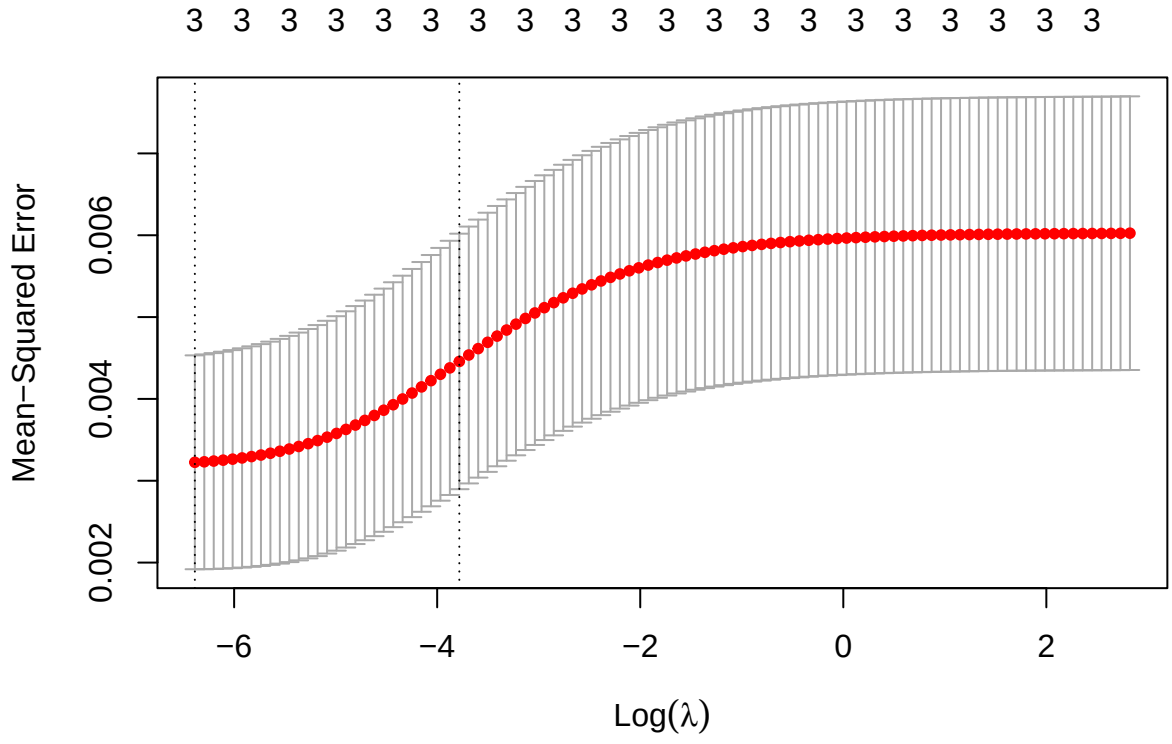
```
(ridge_mse <- mean((ridge_pred - y_test)^2))
```

```
[1] 0.003247498
```

```
# 5-fold cv
set.seed(1)
cv.out <- cv.glmnet(x_train, y_train, alpha = 0) # Fit ridge regression model on training data
bestlam <- cv.out$lambda.min  # Select lambda that minimizes training MSE
bestlam
```

```
[1] 0.001683495
```

```
plot(cv.out)
```



```
ridge_pred <- predict(ridge_mod, s = bestlam, newx = x_test) # Use best lambda to predict test data
mean((ridge_pred - y_test)^2) # Calculate test MSE
```

```
[1] 0.001052628
```

```
out <- glmnet(x, y, alpha = 0) # Fit ridge regression model on full dataset
predict(out, type = "coefficients", s = bestlam)[2:4,] # Display coefficients using lambda chosen by CV
```

```
        Bill          Tip   AlcoholYes
-0.004746170  0.025929806  0.000367881
```

The ridge regression method on the two predictor model with `Tip` and `Bill` has sightly outperformed the OLS method on the same model with MSEs of 0.002580452 and 0.002778002, respectively.

I. (10 points) Write a few sentences that give any overall conclusions and study limitations.

The model does not seem robust enough for prediction. An R^2 value of under 0.05 and two highly correlated variables being `Bill` and `Tip` perform poorly to explain the overall variation in `Tip Percentage`. Do to the many qualitative variables, a logistic regression may be a more suitable approach.

Problem 2

Data for Medical School Admissions is in `MedGPA.csv`, taken from undergraduates from a small liberal arts school over several years. We are interested in student attributes that are associated with higher acceptance rates. This problem uses Logistic Regression.

- ‘Accept‘ = accepted (A) into medical school or denied (D)
- ‘Acceptance‘ = accepted (1) into medical school or denied (0)
- ‘Sex‘ = male (M) or female (F)
- ‘BCPM‘ = GPA in natural sciences and mathematics
- ‘GPA‘ = overall GPA
- ‘VR‘ = verbal reasoning subscale score of the MCAT
- ‘PS‘ = physical sciences subscale score of the MCAT
- ‘WS‘ = writing samples subscale score of the MCAT
- ‘BS‘ = biological sciences subscale score of the MCAT
- ‘MCAT‘ = MCAT total score
- ‘Apps‘ = number of schools applied to

Be sure to interpret model coefficients and associated tests of significance or confidence intervals when answering the following questions. You will be graded on the clarity of your explanations.

```
MedGPA <- read_csv("~/Downloads/RStudio Files STA631/STA631-Exams_Lessons-rstudio-export/exam/MedGPA.csv
```

A. (10 points) Compare the relative effects of improving your MCAT score versus improving your GPA on your odds of being accepted to medical school.

```
# clean data
MedGPA <- as_tibble(MedGPA)
sapply(MedGPA, function(x) sum(is.na(x))) # adapted from https://www.r-bloggers.com/2015/09/how-to-perf
```

```
    Accept Acceptance        Sex       BCPM        GPA         VR         PS
         0          0          0          0          0          0          0
        WS         BS       MCAT       Apps
         1          0          0          0
```

```r
MedGPA_cln <- MedGPA[-54,-1] # removing row 54, only row with NA
MedGPA_cln$MCAT <- as.integer(MedGPA_cln$MCAT)

# fit model
model <- glm(Acceptance~ MCAT + GPA, family = binomial, data=MedGPA_cln)
summary(model)
```

```
Call:
glm(formula = Acceptance ~ MCAT + GPA, family = binomial, data = MedGPA_cln)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7127  -0.8198   0.3414   0.7765   1.9925

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.3510     6.4694  -3.455 0.000551 ***
MCAT          0.1642     0.1034   1.588 0.112211
GPA           4.6736     1.6424   2.846 0.004432 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.192  on 53  degrees of freedom
Residual deviance: 54.011  on 51  degrees of freedom
AIC: 60.011

Number of Fisher Scoring iterations: 5
```

Comparing Effects

Model:

$$log\left(\frac{\hat{\pi}_{yes}}{\hat{\pi}_{no}}\right) = -22.351 + 0.164(MCAT) + 4.6736(GPA)$$

At a $\hat{\beta}_{MCAT} = 0.1642$ and controlling for GPA, the estimated odds of being accepted into med school rather than not is $exp[0.1642] = 1.17845$ times higher for an increase in 1 point on the MCAT. However, `MCAT` is not significant at α=0.05.

B. (10 points) After controlling for MCAT and GPA, is the number of applications related to odds of getting into medical school?

```r
# fit model
model <- glm(Acceptance~ MCAT + GPA + Apps, family = binomial, data=MedGPA_cln)
summary(model)
```

```
Call:
```

```
glm(formula = Acceptance ~ MCAT + GPA + Apps, family = binomial,
    data = MedGPA_cln)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6948  -0.8506   0.3102   0.8001   1.8235

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -23.67191    7.03781  -3.364  0.00077 ***
MCAT          0.17263    0.10554   1.636  0.10189
GPA           4.85827    1.69523   2.866  0.00416 **
Apps          0.04371    0.07618   0.574  0.56611
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.192  on 53  degrees of freedom
Residual deviance: 53.680  on 50  degrees of freedom
AIC: 61.68

Number of Fisher Scoring iterations: 5
```

Applications Effect

Model:

$$log\left(\frac{\hat{\pi}_{yes}}{\hat{\pi}_{no}}\right) = -23.67191 + 0.17263(MCAT) + 4.848(GPA) + 0.04371(Apps)$$

At a $\hat{\beta}_{Apps} = 0.04371$ and controlling for `MCAT` and `GPA`, the estimated odds of being accepted into med school rather than not is $exp[0.04371] = 1.0468$ times higher for an increase in 1 additional application. This is quite close to 1 which may have no practical significance to the likelihood of being accepted into medical school. Also, `Apps` is not significant at α=0.05.

C. (10 points) Is there any evidence that the effect of MCAT total score or GPA differs for males and females? Remember to explain your reasoning, and discuss any tests or confidence intervals you used.

```
# fit model
model <- glm(Acceptance~ MCAT + Sex, family = binomial, data=MedGPA_cln)
summary(model)
```

```
Call:
glm(formula = Acceptance ~ MCAT + Sex, family = binomial, data = MedGPA_cln)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9415  -0.9843   0.5072   1.0168   1.8321

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.91891    3.37756  -2.641  0.00828 **
MCAT         0.26597    0.09474   2.807  0.00500 **
```

```
SexM        -1.06378    0.63379  -1.678  0.09326 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.192  on 53  degrees of freedom
Residual deviance: 61.700  on 51  degrees of freedom
AIC: 67.7

Number of Fisher Scoring iterations: 4
```

```
# fit model
model <- glm(Acceptance~ GPA + Sex, family = binomial, data=MedGPA_cln)
summary(model)
```

```
Call:
glm(formula = Acceptance ~ GPA + Sex, family = binomial, data = MedGPA_cln)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8157  -0.9827   0.3229   0.7343   2.4583

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -21.0060     6.4360  -3.264 0.001099 **
GPA           6.1147     1.8378   3.327 0.000878 ***
SexM         -1.1657     0.7187  -1.622 0.104814
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.192  on 53  degrees of freedom
Residual deviance: 53.932  on 51  degrees of freedom
AIC: 59.932

Number of Fisher Scoring iterations: 5
```

Comparison by Sex

No meaningful difference is apparent when `Sex` is considered in comparing the effects of `MCAT` and `GPA` on the likelihood of being accepted to medical school.

Both `GPA` and `MCAT` coefficients are insignificant at an $\alpha = 0.05$. Also, if it were significant, the change in likelihood of acceptance would be that males are 0.311 times more likely of being accepted over females, which in practical terms, is less of a likelihood since the factor is less than 1.

D. (10 points) Build a logistic regression model with `GPA`, `MCAT`, `Apps` and `Sex` predictors, do not include interactions for this problem, if you have a predictor that is completely multicollinear with other predictors you can drop it. Write out your estimated model.

```r
# fit model
model <- glm(Acceptance~ MCAT + GPA + Apps + Sex, family = binomial, data=MedGPA_cln)
summary(model)
```

```
Call:
glm(formula = Acceptance ~ MCAT + GPA + Apps + Sex, family = binomial,
    data = MedGPA_cln)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0082  -0.8329   0.2507   0.6588   2.1264

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -24.93144    7.40251  -3.368 0.000757 ***
MCAT          0.18507    0.10917   1.695 0.090045 .
GPA           5.28313    1.89251   2.792 0.005245 **
Apps          0.03308    0.07476   0.442 0.658139
SexM         -1.23776    0.73282  -1.689 0.091213 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.192  on 53  degrees of freedom
Residual deviance: 50.589  on 49  degrees of freedom
AIC: 60.589

Number of Fisher Scoring iterations: 5
```

```r
# test for multicollinearity
cor(MedGPA_cln[, c("MCAT", "GPA", "Apps")])
```

```
           MCAT        GPA       Apps
MCAT  1.0000000  0.4438267 -0.1782781
GPA   0.4438267  1.0000000 -0.1687736
Apps -0.1782781 -0.1687736  1.0000000
```

Final Model

Model:

$$log\left(\frac{\hat{\pi}_{yes}}{\hat{\pi}_{no}}\right) = -24.93144 + 0.18507(MCAT) + 5.28313(GPA) + 0.03308(Apps) - 1.23776(Sex)$$

E. (20 points) Get two different confusion matrices with different cut-off probabilities, use 0.50 for the cut-off for one confusion matrix and then try a different cut-off value as well. Give the sensitivity, specificity and percent correct for both confusion matrixes and compare the two results. Discuss the trade-off between sensitivity and specificity when choosing a cut-off value.

```r
set.seed(2005)
# tidy MedGPA_cln
df_final <- MedGPA_cln[, c("MCAT", "GPA", "Apps", "Sex", "Acceptance")]
df_final <- df_final %>%
mutate(Acceptance_bool = df_final$Acceptance,
Acceptance = MedGPA_cln$Acceptance)

multiple_logi_df_final<- logistic_reg(mode = "classification") %>%
  set_engine('glm') %>%
  fit(data = df_final, as.factor(df_final$Acceptance_bool) ~ MCAT + GPA + Sex + Apps)
multiple_logi_df_final %>% tidy()
```

```
# A tibble: 5 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) -24.9       7.40      -3.37  0.000757
2 MCAT          0.185     0.109      1.70  0.0900
3 GPA           5.28      1.89       2.79  0.00524
4 SexM         -1.24      0.733     -1.69  0.0912
5 Apps          0.0331    0.0748     0.442 0.658
```

```r
kableExtra::kable(x = broom::tidy(multiple_logi_df_final), format = "pipe")
```

term	estimate	std.error	statistic	p.value
(Intercept)	-24.9314377	7.4025139	-3.367969	0.0007572
MCAT	0.1850654	0.1091727	1.695163	0.0900446
GPA	5.2831259	1.8925069	2.791602	0.0052448
SexM	-1.2377550	0.7328192	-1.689032	0.0912133
Apps	0.0330782	0.0747558	0.442484	0.6581390

```r
df_final <- df_final %>%
  mutate(Acceptance = str_trim(ifelse(df_final$Acceptance == 1, "Accept", "Deny")))

train_df <- df_final %>%
    sample_frac(0.75)

test_df <- df_final %>%
    setdiff(train_df)

predictions <- multiple_logi_df_final %>%
  predict(test_df, type = "prob")


df_test_pred <- bind_cols(predictions, test_df)

hard_pred_0.5 <- df_test_pred %>%
  mutate(.pred = make_two_class_pred(df_test_pred$.pred_0, levels(as_factor(Acceptance_bool)), threshol
  select(Acceptance_bool, contains(".pred"))
```

```r
(hard_pred_0.5 <- hard_pred_0.5 %>%
  count(.truth = Acceptance_bool, .pred))
```

```
# A tibble: 3 x 3
  .truth      .pred     n
   <dbl> <clss_prd> <int>
1      0          0     4
2      1          0     3
3      1          1     7
```

```r
# threshold = 0.91
hard_pred_0.91 <- df_test_pred %>%
  mutate(.pred = make_two_class_pred(df_test_pred$.pred_0, levels(as_factor(Acceptance_bool)), threshol
  select(Acceptance_bool, contains(".pred"))

(hard_pred_0.91<- hard_pred_0.91 %>%
  count(.truth = Acceptance_bool, .pred))
```

```
# A tibble: 3 x 3
  .truth      .pred     n
   <dbl> <clss_prd> <int>
1      0          0     1
2      0          1     3
3      1          1    10
```

```r
# at  0.5
tibble("At 0.5" = c("Truth: Accept", "Truth: Deny", "Percentages"), "Predicted: Accept" =c(7,0, "50%"),
```

```
# A tibble: 3 x 5
  `At 0.5`      `Predicted: Accept` `Predicted: Deny` Percentages `Sens/Spec`
  <chr>         <chr>               <chr>             <chr>       <chr>
1 Truth: Accept 7                   3                 71.14%      "0.7"
2 Truth: Deny   0                   4                 28.57%      "1"
3 Percentages   50%                 50%               100%        ""
```

```r
# 0.75
tibble("At 0.91" = c("Truth: Accept", "Truth: Deny", "Percentages"), "Predicted: Accept" =c(10,3, "92.8
```

```
# A tibble: 3 x 5
  `At 0.91`     `Predicted: Accept` `Predicted: Deny` Percentages `Sens/Spec`
  <chr>         <chr>               <chr>             <chr>       <chr>
1 Truth: Accept 10                  0                 71.14%      "1"
2 Truth: Deny   3                   1                 28.57%      "0.25"
3 Percentages   92.86%              7.14%             100%        ""
```

At 50% threshold, sensitivity is at 0.7, while specificity is at 1. At 91% threshold, sensitivity increases to 1 and specificity decreases all the way to .25.. The increased sensitivity at the 91% threshold is a great improvement, but at the cost of predicting only 1 out of 4 denials from med school. It may be a better trade-off to use an in-between threshold of 75 to keep specificity high. I think you'd rather have all the applicants who were accepted by predicted to be accepted again than some be predicted to be accepted when they really weren't qualified to. Retention may be impacted, as well as performance of the pool of students.