CAP 5768: Final Project, part 1

Due on Canvas by Friday, April 24, 2020 at 11:59pm

Place name here: Yousef Al-Kafif

Preliminary instructions

All analyses must be performed in R using the **`tidyverse`** and **`glmnet`** packages discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of assignment

In this assignment you will be using the dataset **`GlobalAncestry.csv`**, which is available on Canvas. You will be analyzing genetic data from 242 humans sampled across the world from six ancestries. The first column in each dataset, labeled **`ancestry`**, takes the following values:

`African`	San and Yoruban individuals from sub-Saharan Africa
`European`	Italian and Russian individuals from Europe
`EastAsian`	Chinese and Japanese individuals from East Asia
`Oceanian`	Melanesian and Papuan individuals from Oceania
`NativeAmerican`	Pima and Mayan individuals from the Americas
`Mexican`	Mexican individuals from the Americas
`Unknown1`	Unknown ancestry
`Unknown2`	Unknown ancestry
`Unknown3`	Unknown ancestry
`Unknown4`	Unknown ancestry
`Unknown5`	Unknown ancestry

The **`GlobalAncestry.csv`** is a large dataset with genetic data for individuals 242 at 8916 genomic locations. As we discussed in our introductory lecture for this course, each individual will have a value of **`0`**, **`1`**, or **`2`** at each of these genomic locations, indicating "genotype" that the individual has at this location.

Training a lasso penalized multinomial regression classifier

The goal is to train a multinomial regression classifier to predict K=5 ancestries (**`African`**, **`European`**, **`EastAsian`**, **`Oceanian`**, and **`NativeAmerican`**). The training dataset will consist only of individuals with **`African`**, **`European`**, **`EastAsian`**, **`Oceanian`**, and **`NativeAmerican`** ancestries, and the best classifier will be determined by lasso-penalized multinomial regression and 10-fold cross validation. You will consider 500 tuning parameter values (λ), taking values between 0.001 and 1000 evenly on a base-10 logarithmic scale, as we have highlighted several times in class. You will then choose the classifier that is the simplest classifier that is within 1 standard error of the best classifier.

Predicting ancestry of individuals with unknown ancestry

You will then use this classifier to predict the ancestries of the five unknown individuals (**`Unknown1`**, **`Unknown2`**, **`Unknown3`**, **`Unknown4`**, and **`Unknown5`**) based on their genetics.

Predicting ancestry proportions of individuals with Mexican ancestry

You will also use predicted class probabilities to estimate the fraction of ancestry that each individual of Mexican descent has from each of the five continental ancestries used to train the classifier. You will then use violin plots to visualize the distributions of these probabilities across the set of individuals of Mexican ancestry, and hypothesize about the historical reasons for the ancestry distributions you observe.

Instructions for loading GlobalAncestry dataset into your RStudio Cloud environment

Recall that to upload a file to RStudio Cloud, you first must download the **`GlobalAncestry.csv`** file to your computer. Once the file is downloaded, within the "Files" panel of the RStudio Cloud environment, click "Upload" and browse to the appropriate directory on your computer to upload the **`GlobalAncestry.csv`** file.

The **`GlobalAncestry.csv`** file can be loaded using the **`read_csv()`** function of the **`readr`** package that comes loaded with **`tidyverse`**, and assigned to an object called **`GlobalAncestry`** as

```
GlobalAncestry <- read_csv("GlobalAncestry.csv")
```

If you are having trouble loading the file, then refer back to the video lecture on Linear Regression where this was demonstrated in class.

Note about using `glmnet` for classification

When using **`glmnet`**, you will ***not*** need to recode classes as values **`1`**, **`2`**, **`3`**, etc. We only performed this recoding in class to illustrate the connection with using linear regression applied to a response with values **`0`** and **`1`**, as linear regression *requires* a quantitative response. Therefore, do ***not*** recode the **`ancestry`** values in the dataset, and simply use the values as is.

Questions and problems

1. [10%] Load the **`GlobalAncestry.csv`** dataset, and split and store the dataset into three separate datasets: training dataset, test dataset of unknown ancestries, and test dataset of Mexican ancestry. That is, create the following three datasets:

1. Training data frame called **`train`**, which only includes observations with **`ancestry`** values **`African`**, **`European`**, **`EastAsian`**, **`Oceanian`**, and **`NativeAmerican`**.
2. Test data frame called **test**, which only includes observations with **`ancestry`** values **`Unknown1`**, **`Unknown2`**, **`Unknown3`**, **`Unknown4`**, and **`Unknown5`**.
3. Test data frame called **`testmex`**, which only includes observations with **`ancestry`** value **`Mexican`**.

Provide code below:

- Creating variable containing list of ancestry names, for use in filter.

```
targetTrainAncestors <- c("African", "European", "EastAsian", "Oceanian",
"NativeAmerican")

TestTarget <- c("Unknown1", "Unknown2", "Unknown3", "Unknown4",
"Unknown5")
```

- Creating 3 dataframes.

```
GA.train <- GlobalAncestry %>%
filter(ancestry %in% targetTrainAncestors)

GA.test <- GlobalAncestry %>%
+ filter(ancestry %in% TestTarget)

GA.testmex <- GlobalAncestry %>%
+ filter(ancestry == "Mexican")
```

2. [20%] Apply **`glmnet`** to the training dataset **`train`** from Question 1, to train a multinomial regression classifier with a lasso penalty across 500 tuning parameter (λ) values, taking values between 0.001 and 1000 evenly on a base-10 logarithmic scale. The response will be **`ancestry`**, and the input features will be the values at the set of 8916 genomic locations. Train this lasso-penalized multinomial regression model across the 500 tuning parameter values, and plot the regression coefficients for each of the *K*=5 classes as a function of log(λ). Based on these results, does it appear that regularization and feature selection is working? Explain your answer.

Note: The multinomial regression model will have a distinct set of regression coefficients for each of the *K*=5 classes, and so you must provide five graphs. You can access each of the five graphs using the back and forward arrows under the "Plots" subpanel of RStudio. You will also not need to plot a legend for each line, as there are simply too many potential lines on each graph (up to 8916 lines) to make a legend feasible.

Provide code below:

Creating input and output:-

```
Y <- GA.train %>%
+ select(ancestry)%>%
+ as.matrix()

 X <- GA.train %>%
+ select(-c(ancestry))%>%
+ as.matrix()
```

Creating lambda variable for values between 0.001 and 1000, across 500 tuning parameter values:

```
lambdas <- 10^seq(-3, 3, length.out = 500)
```

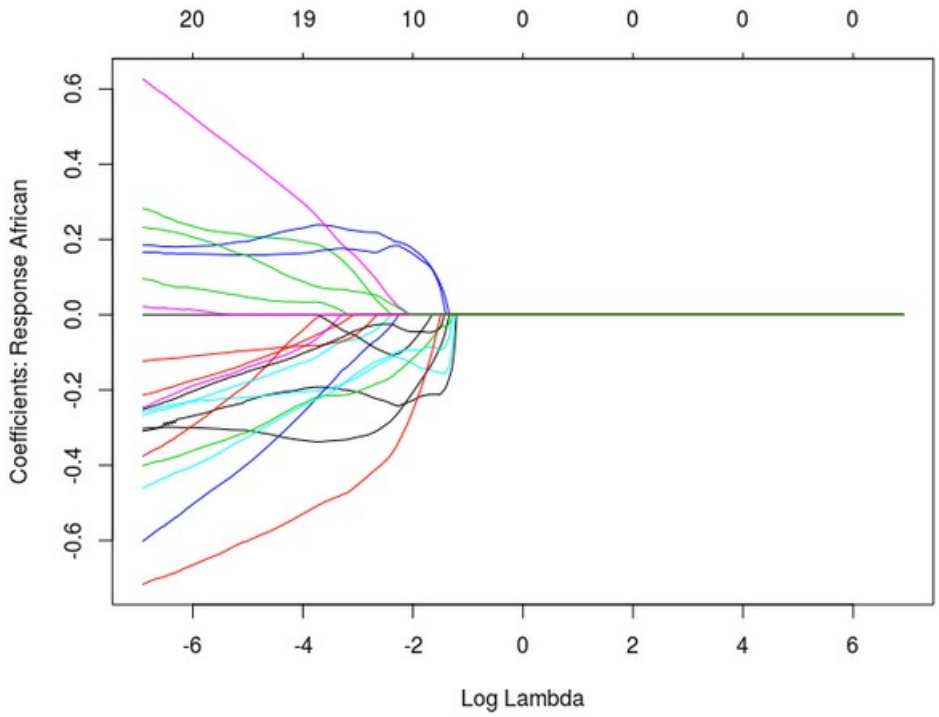
Fitting:-

```
lasso.fit <- glmnet(X, Y, family = "multinomial", alpha = 1, lambda = lambdas)
```

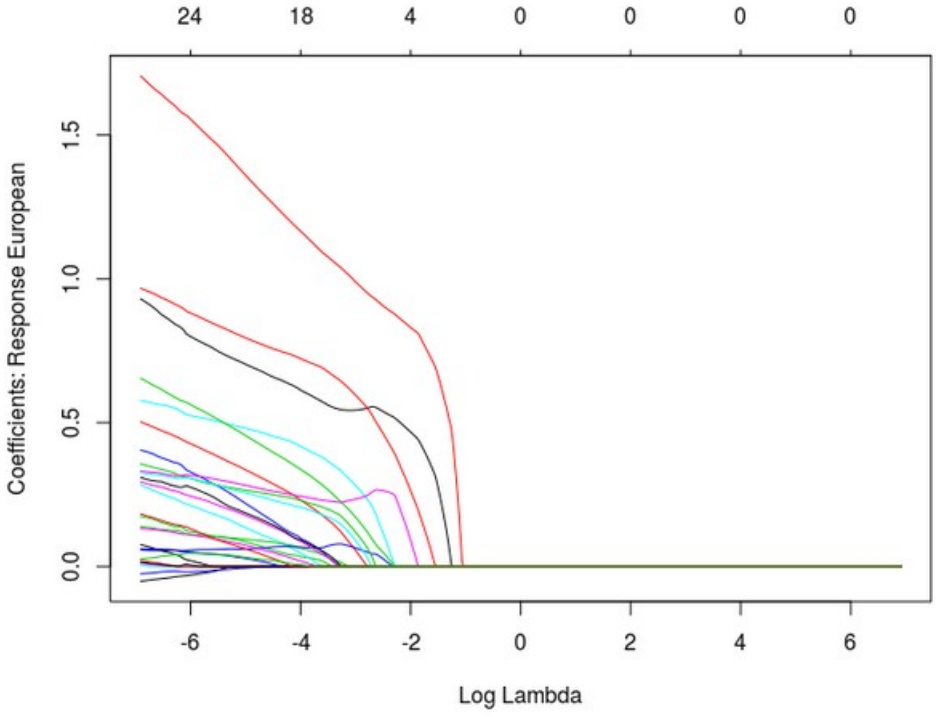
Plotting:

```
plot(lasso.fit, xvar ="lambda")
```

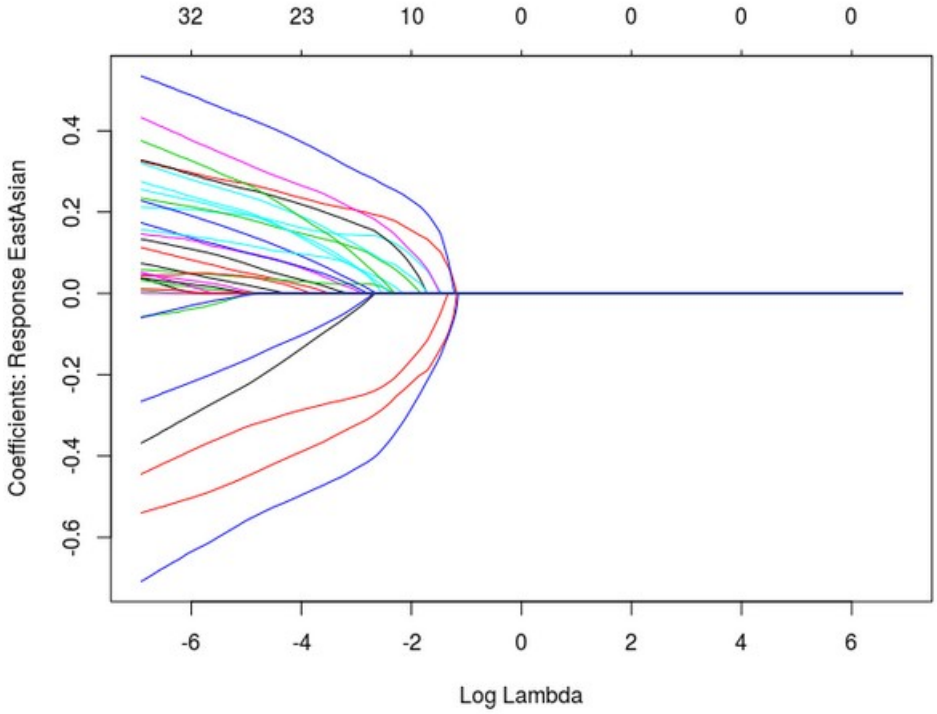
Provide figure for African regression coefficients below:



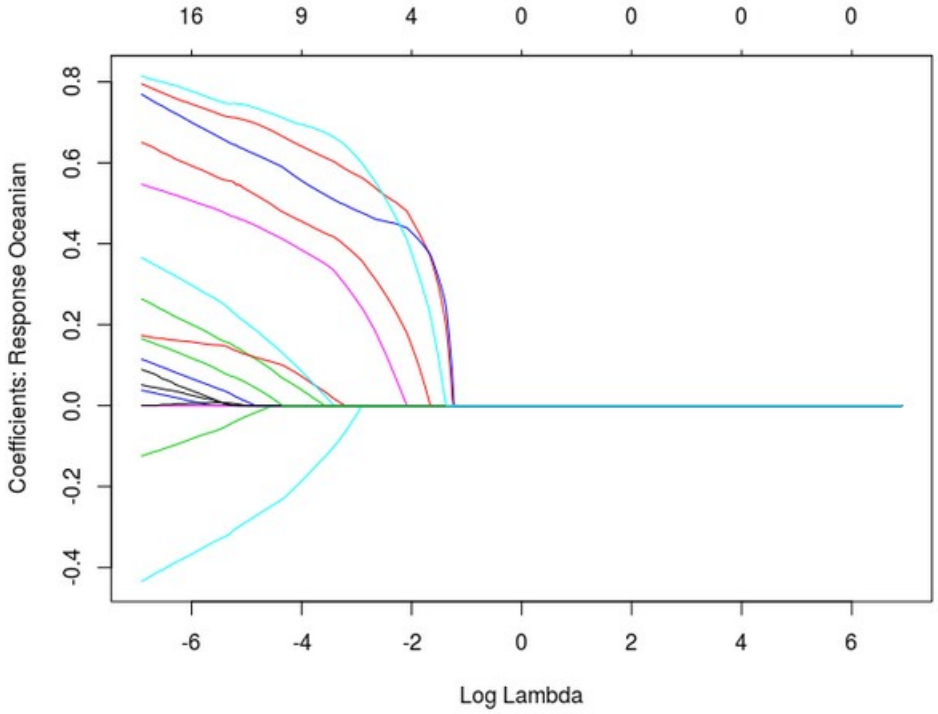
Provide figure for European regression coefficients below:



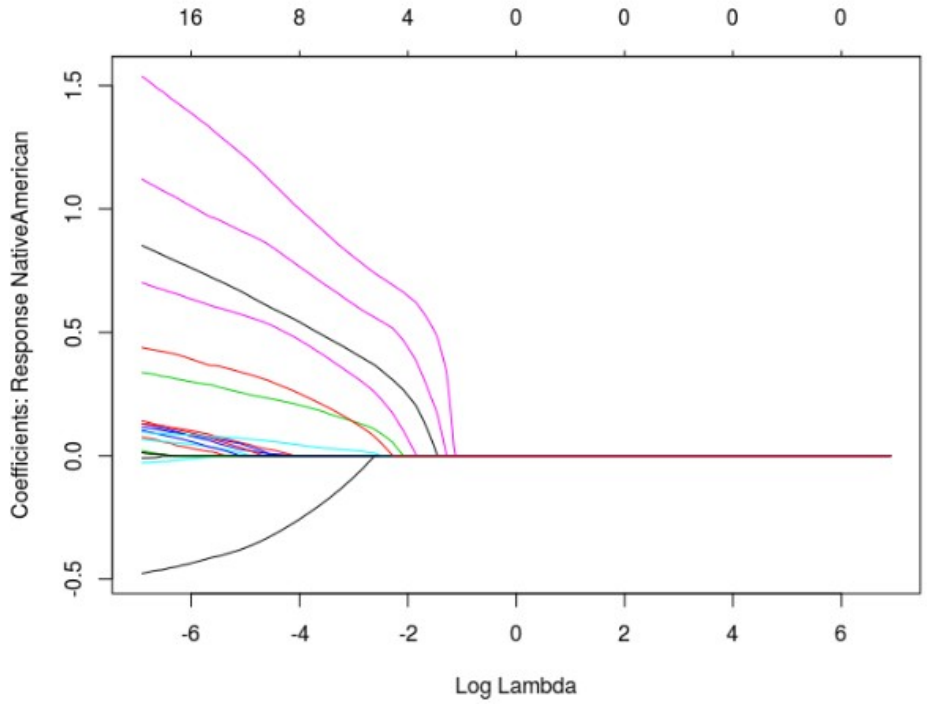
Provide figure for East Asian regression coefficients below:



Provide figure for Oceanian regression coefficients below:



Provide figure for Native American regression coefficients below:



Provide answer to question below:

Yes. This is shown by the graph displaying how the features go towards 0. As a lasso multinomial regression would shrink correlated features to 0. The lasso also might remove features altogether as the prediction is not based on those, this is evidenced in all the graphs but most notable in the NativeAmerican results. Some features are set to 0 (i.e. the gene was not relevant to the prediction at all) and this is an indicator of feature selection.

3. [15%] Apply **`glmnet`** to the training dataset **`train`** from Question 1, to perform 10-fold cross validation for a multinomial regression classifier with a lasso penalty across 500 tuning parameter (λ) values, taking values between 0.001 and 1000 evenly on a base-10 logarithmic scale. The response will be the genetic ancestry, and the input features will be the values at the set of 8916 genomic locations. Train this lasso-penalized multinomial regression model across the 500 tuning parameter values, and the cross validation error as a function of log(λ). What is the best tuning parameter value, and what is the tuning parameter value associated with the simplest model that is within 1 standard error of the best model.

Provide code below:

Creating input and output:-

```
Y <- GA.train %>%
+ select(ancestry)%>%
+ as.matrix()

 X <- GA.train %>%
+ select(-c(ancestry))%>%
+ as.matrix()
```

Creating lambda variable for values between 0.001 and 1000, across 500 tuning parameter values:

```
lambdas <- 10^seq(-3, 3, length.out = 500)
```

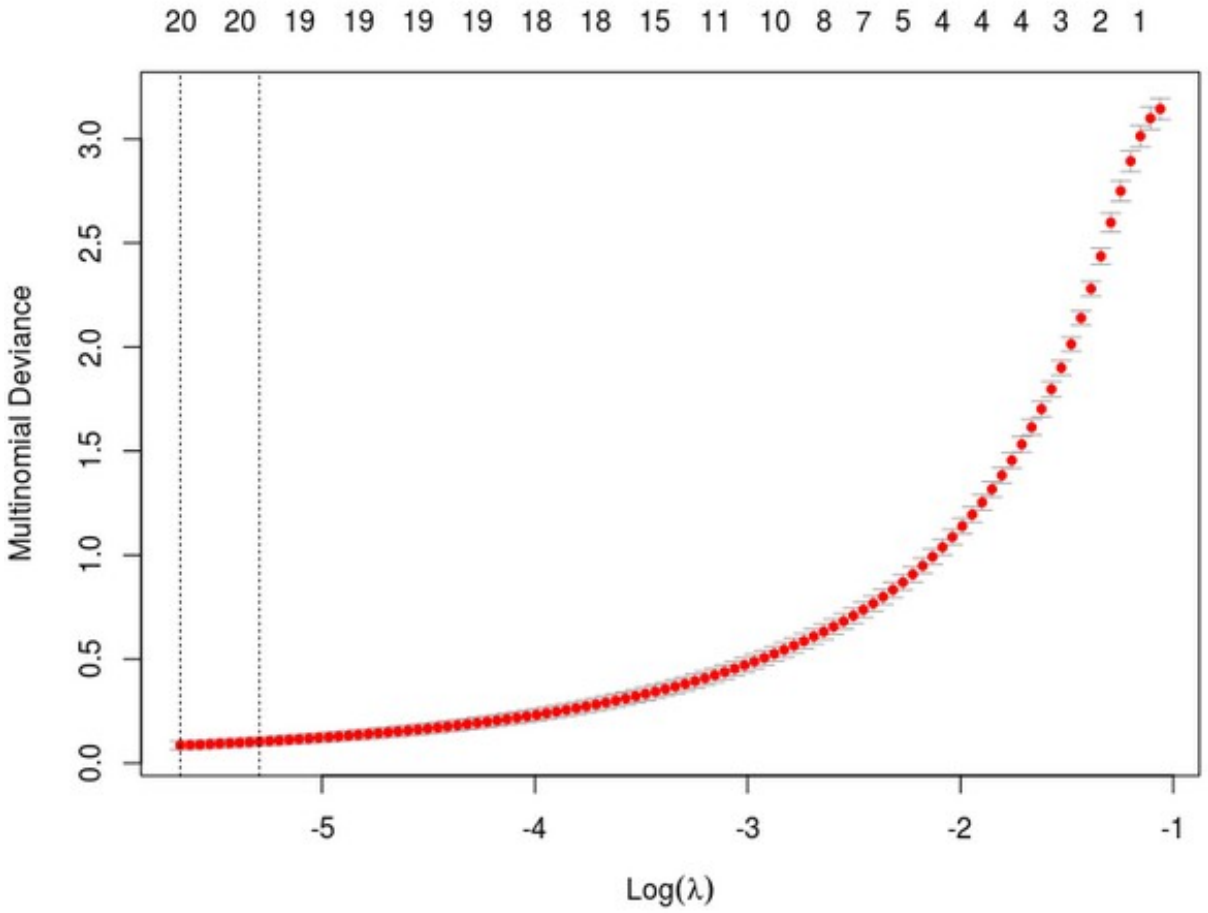
Fitting (10fold) :

```
lasso.cv <- cv.glmnet( X , Y, family = “multinomial”, alpha = 1, lamda = lamdas, nfolds = 10)
```

Plotting:

```
plot(lasso.cv)
```

Provide figure below:



Provide answer to questions below:

What is the best tuning parameter value, and what is the tuning parameter value associated with the simplest model that is within 1 standard error of the best model.

Best fit model is with 20 features – this is indicated by the first dotted asymptote from the left. Thus our simplest model that is within one standard error from the best fit model is the next dotted asymptote which coincides with a tuning parameter value of roughly -5.25. The best tuning parameter value coincides with the best fit model and that is at roughly -5.8. This best tuning parameter value can be found exactly via " lasso.cv$lambda.min" (I.e. YourRegressionObject$lamda.min) and in this case returns a value of 0.003461179.

Note that the tuning parameter value for the simplest model within a standard error of the best model is found via "lasso.cv$lambda.1se"

4. [15%] Apply **`glmnet`** to the training dataset **`train`** from Question 1, to train a multinomial regression classifier with a lasso penalty using the tuning parameter (λ) that is associated with the simplest model within 1 standard error of the best from Question 3. Make predictions of the classes for the training dataset using this fitted model, and add these predictions to the training dataset, storing the new training dataset with predictions as a new object called **`train.withPreds`**. Create a confusion matrix and estimate classification accuracy for the training dataset.

Provide code below:

Training multinomial regression classifier with lasso penalty via glmnet:

```
lasso.1se <- glmnet(X, Y, family = "multinomial", alpha = 1, lamda = lasso.cv$lambda.1se)
```

Predictions:

```
predClasses <- predict(lasso.1se, X, type = "class", s = lasso.cv$lambda.1se)
```

Adding predictions to training dataset :

```
 train.withPreds <- GA.train %>%
+ mutate(pred = c(predClasses))
```

Confusion matrix

```
train.withPreds %>%
+ select(ancestry, pred)%>%
+ table()
```

Class accuracy

```
train.withPreds %>%
+ summarize(classAccuracy = mean(pred == ancestry))
```

Provide confusion matrix below:

```
                pred
ancestry         African EastAsian European NativeAmerican Oceanian
  African             25         0        0              0        0
  EastAsian            0        61        0              0        0
  European             0         0       36              0        0
  Mexican              0         0        0              0        0
  NativeAmerican       0         0        0             34        0
  Oceanian             0         0        0              0       27
  Unknown1             0         0        0              0        0
  Unknown2             0         0        0              0        0
  Unknown3             0         0        0              0        0
  Unknown4             0         0        0              0        0
  Unknown5             0         0        0              0        0
```

Provide accuracy estimate below:

```
> train.withPreds %>%
+ summarize(classAccuracy = mean(pred == ancestry))
  classAccuracy
1             1
```

5. [15%] Given the trained model from Question 4, apply **`glmnet`** to the test dataset **`test`** from Question 1, to predict the ancestries of the five observations from the test dataset. Report the estimated ancestries for each of the five individuals.

Provide code below:

Creating test datasets inputs as a matrix.

```
X.test <- GA.test %>%
+ select(-c(ancestry))%>%
+ as.matrix()
```

Predicting given test datasets inputs (predicting as a probability, i.e. estimated ancestries) using trained model from question 4.

```
predUnknowns <- predict(lasso.1se, X.test , type = "response", s = lasso.cv$lambda.1se)
```

```
> predUnknowns
, , 1

         African   EastAsian    European NativeAmerican    Oceanian
[1,] 0.003016359 0.988790446 0.003433556    0.001929502 0.002830137
[2,] 0.907415814 0.018385761 0.050121725    0.006231935 0.017844764
[3,] 0.032388497 0.098412212 0.745842126    0.077426717 0.045930448
[4,] 0.002657196 0.008334832 0.002905816    0.984056052 0.002046104
[5,] 0.003121991 0.004860195 0.003516758    0.003667914 0.984833141
```

Fill in the following predictions:

Ancestry	Predicted ancestry
Unknown1	EastAsian
Unknown2	African
Unknown3	European
Unknown4	NativeAmerican
Unknown5	Oceanian

6. [25%] Given the trained model from Question 4, apply **`glmnet`** to the Mexican-ancestry test dataset **`testmex`** from Question 1, to predict the probabilities of each of the *K*=5 ancestries for each individual in the **`testmex`** dataset. The reported probability for each class can be used as a proxy for the fraction of an individual's ancestry deriving from the ancestry associated with that class.

Visualize the distributions of the probabilities of the *K*=5 classes across the Mexican-ancestry individuals in the **`testmex`** dataset using violin plots. Specifically, using **`ggplot`**, add a new violin plot layer for each of the *K*=5 ancestries (*i.e.*, add five layers) to display the distributions of class probability for each of the ancestries. Fill the violin plots with the following colors based on ancestries:

Ancestry	Fill color
`African`	**`orange`**
`European`	**`blue`**
`EastAsian`	**`pink`**
`Oceanian`	**`green`**
`NativeAmerican`	**`purple`**

Do the distributions of ancestry proportions make sense based on knowledge of history? Explain your answer.

Provide code below:

Creating test datasets inputs as a matrix.

```
X.mexican <- GA.testmex %>%
+ select(-c(ancestry))%>%
+ as.matrix()
```

Predicting given test datasets inputs (predicting as a probability of each ancestry) using Q4s trained glmnet model. :-

```
predMexicans <- predict(lasso.1se, X.mexican , type = "response", s = lasso.cv$lambda.1se)
> predMexicans
, , 1

           African   EastAsian    European NativeAmerican     Oceanian
 [1,] 0.0012446254 0.004711831 0.952004833   0.0416405190 0.0003981911
 [2,] 0.0809174235 0.302572769 0.382559061   0.1092627971 0.1246879486
 [3,] 0.0052586943 0.003578083 0.948012602   0.0410217520 0.0021288690
 [4,] 0.0260497418 0.076119502 0.184631946   0.6962261933 0.0169726167
 [5,] 0.0362858846 0.139012876 0.263766908   0.5375802915 0.0233540391
 [6,] 0.0026619600 0.021280703 0.029061987   0.9430028472 0.0039925028
 [7,] 0.0280827094 0.067102352 0.358654136   0.5243831203 0.0217776818
 [8,] 0.0662160302 0.047920164 0.275599329   0.5926172931 0.0176471833
 [9,] 0.0083500483 0.107184356 0.642246628   0.2338927353 0.0083262315
[10,] 0.0478964624 0.136275654 0.549956574   0.1876991888 0.0781721209
```

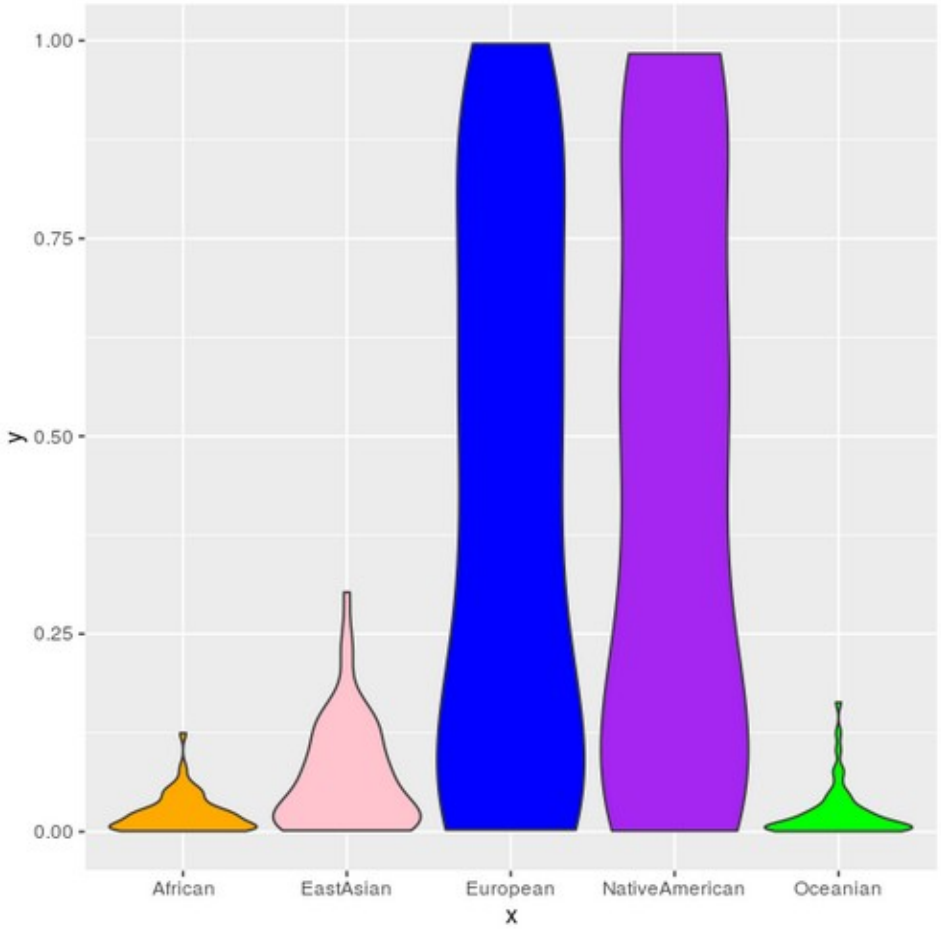
//only displaying first 10

Converting predictions into data frame for compatibility with ggplot :
predMexicansDF <- as.data.frame(predMexicans)

Plotting :-

```
ggplot( data = predMexicansDF, aes(y = "Probability"))+
+ geom_violin(mapping = aes(x= "African", y= African.1), fill = "orange") +
+ geom_violin(mapping = aes(x= "European", y= European.1), fill = "blue") +
+ geom_violin(mapping = aes(x= "EastAsian", y= EastAsian.1), fill = "pink") +
+ geom_violin(mapping = aes(x= "Oceanian", y= Oceanian.1), fill = "green") +
+ geom_violin(mapping = aes(x= "NativeAmerican", y= NativeAmerican.1), fill = "purple")
```

Provide figure below:



Provide answer to questions below:

Yes, the distributions of ancestries make sense based on past world history. As Mexico was ruled by spain between the 1600s until Mexicos independence on September 27th of 1821, this will leave a European footprint down the line that we see being prevalent now with our genetic data of 54 Mexicans. The Native American genetics can be explained by remembering that Mexico used to encompass a significant part of North America, important areas to note but are not limited to is New Mexico, Arizona, and Nevada. These places are known for Native American people and influences still current to this day. So it may be fair to conclude that the Mexican people and Native American people were mixing or perhaps were one and the same before the loss of land to the US. This may explain why their genetic makeup is similar. Regarding the East Asian similarities, this could be explained through the trade route "Manila Galleons" which was used by Spanish trading ships throughout their reign in Mexico. This route went from the Mexican city of Acapulco to Manila, the capital city of phillipines. This trading did not only consist of goods but also of slaves which may be why there is a slight genetic similarity shown in our data. Oceania and Africa show very little similarity as Mexico has had no significant influence from their regions.

CAP 5768: Homework 3

Due on Canvas by Friday, April 10, 2020 at 11:59pm

Place name here:

Yousef Al-Kafif

Preliminary instructions

All analyses must be performed in R using **`tidyverse`** and other libraries discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of assignment

In this assignment you will be analyzing the **`College`** dataset that comes with the **`ISLR`** package. This dataset has information on 18 features for 777 US colleges obtained from the 1995 issue of US News and World Reports. The columns in the dataset are:

Name	Description
`Private`	A factor with levels **`No`** and **`Yes`** including private or public university
`Apps`	Number of applications received
`Accept`	Number of applications accepted
`Enroll`	Number of new students enrolled
`Top10perc`	Percent new students from top 10% of high school class
`Top25perc`	Percent new students from top 25% of high school class
`F.Undergrad`	Number of full-time undergraduates
`P.Undergrad`	Number of part-time undergraduates
`Outstate`	Out-of-state tuition
`Room.Board`	Room and board costs
`Books`	Estimated book costs
`Personal`	Estimated personal spending
`PhD`	Percent of faculty with a Ph.D.
`Terminal`	Percent of faculty with a terminal degree
`S.F.Ratio`	Student/faculty ratio
`perc.alumni`	Percent alumni who donate
`Expend`	Instructional expenditure per student
`Grad.Rate`	Graduation rate

Questions and problems

1. [6%] Recode the binary feature **`Private`** with values **`0`** and **`1`** in place of **`No`** and **`Yes`** and store in a new data frame called **`College.recoded`**.

Provide code below:

```
> College.recoded <- College %>%

+ mutate(Private = ifelse(Private == "Yes", 1, 0))
```

2. [10%] Fit a multiple linear regression model to predict private school status with the 17 other features. Which feature is most important in this model, and what evidence tells you that?

Provide code below:

```
> lmMultiple.fit <- lm(formula = Private ~ Apps + Accept + Enroll +
Top10perc + Top25perc + F.Undergrad + P.Undergrad + Outstate +
Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio +
perc.alumni + Expend + Grad.Rate, data = College.recoded)

> summary(lmMultiple.fit)


Call:
lm(formula = Private ~ Apps + Accept + Enroll + Top10perc + Top25per
c +
    F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
    Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend +
    Grad.Rate, data = College.recoded)

Residuals:
     Min       1Q   Median       3Q      Max
-1.05640 -0.15757  0.02107  0.16986  1.42289

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.799e-01  1.018e-01   8.640  < 2e-16 ***
Apps        -3.371e-05  9.401e-06  -3.586 0.000358 ***
Accept       4.259e-05  1.835e-05   2.320 0.020582 *
Enroll      -1.058e-05  4.928e-05  -0.215 0.830024
Top10perc    2.178e-03  1.530e-03   1.424 0.154973
Top25perc   -2.001e-04  1.178e-03  -0.170 0.865099
F.Undergrad -2.919e-05  8.496e-06  -3.435 0.000624 ***
P.Undergrad -9.345e-06  8.398e-06  -1.113 0.266212
Outstate     4.370e-05  4.787e-06   9.128  < 2e-16 ***
Room.Board   3.677e-05  1.262e-05   2.913 0.003685 **
Books        6.055e-05  6.223e-05   0.973 0.330844
Personal     3.199e-07  1.648e-05   0.019 0.984517
PhD         -4.052e-03  1.205e-03  -3.362 0.000812 ***
Terminal    -4.012e-03  1.323e-03  -3.032 0.002510 **
S.F.Ratio   -1.472e-02  3.358e-03  -4.384 1.33e-05 ***
perc.alumni  2.689e-03  1.067e-03   2.520 0.011931 *
Expend      -5.457e-06  3.303e-06  -1.652 0.098894 .
Grad.Rate    1.540e-03  7.726e-04   1.993 0.046605 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.272 on 759 degrees of freedom
Multiple R-squared:  0.6358,    Adjusted R-squared:  0.6276
F-statistic: 77.94 on 17 and 759 DF,  p-value: < 2.2e-16
```

Provide answer to question below:

- The feature with the lowest p-value closest to 0 has the highest relation to the prediction. This feature is “Outstate” as shown both on the multiple linear regression fit

3. [5%] Fit a simple linear regression model to predict private school status based on the most important feature from Question 2. Is this feature still important in this model, and what evidence tells you that?

Provide code below:

```
> lmOutstate.fit <- lm(formula = Private ~ Outstate, data = College.
recoded)
> summary(lmOutstate.fit)

Call:
lm(formula = Private ~ Outstate, data = College.recoded)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0511 -0.3506  0.1008  0.3004  0.7688

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.790e-02  3.711e-02   2.369   0.0181 *
Outstate    6.123e-05  3.317e-06  18.460   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3717 on 775 degrees of freedom
Multiple R-squared:  0.3054,    Adjusted R-squared:  0.3045
F-statistic: 340.8 on 1 and 775 DF,  p-value: < 2.2e-16
```

Provide answer to question below:

- This simple linear regression fit using “Outstate” as the sole input feature shows that it still retains its low p-value. It also has a decent R – squared value (the lower the worse). Although its F-statistic is a little lacking as the more extreme positive values indicate a higher relation, a value of 340 is not that extreme. But overall the p-value and r-squared values indicate that it is an important feature to the model.

4. [10%] Visualize the simple linear regression model from Question 3 using a scatter plot and the fitted linear model using the appropriate arguments in **`geom_smooth()`**.

Provide code below:

Simple Linear Regression model visualization w/ scatter plot:-

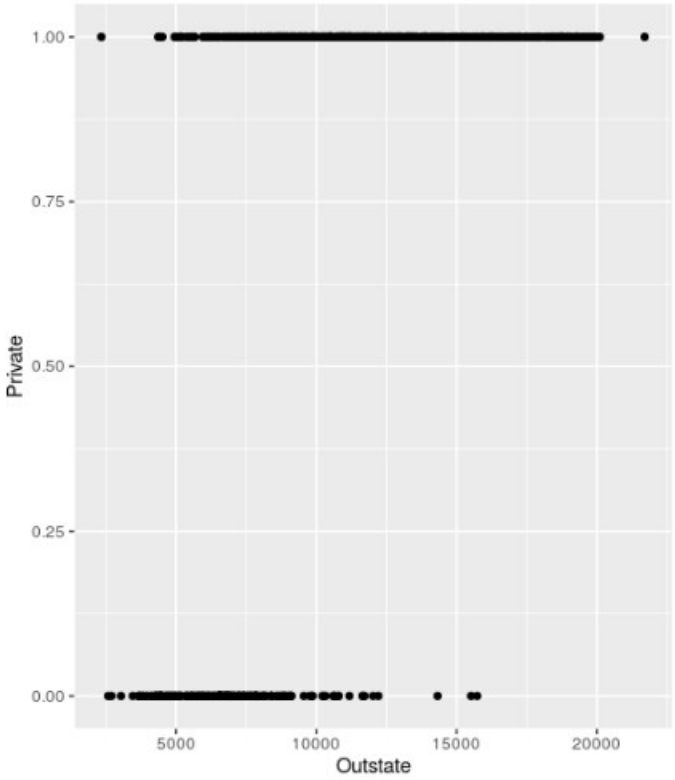
```
ggplot(data = lmOutstate.fit)+
+ geom_point( mapping = aes(x = Outstate, y = Private))
```

Fitted linear model visualization using appropriate arguments w/ geom_smooth():-

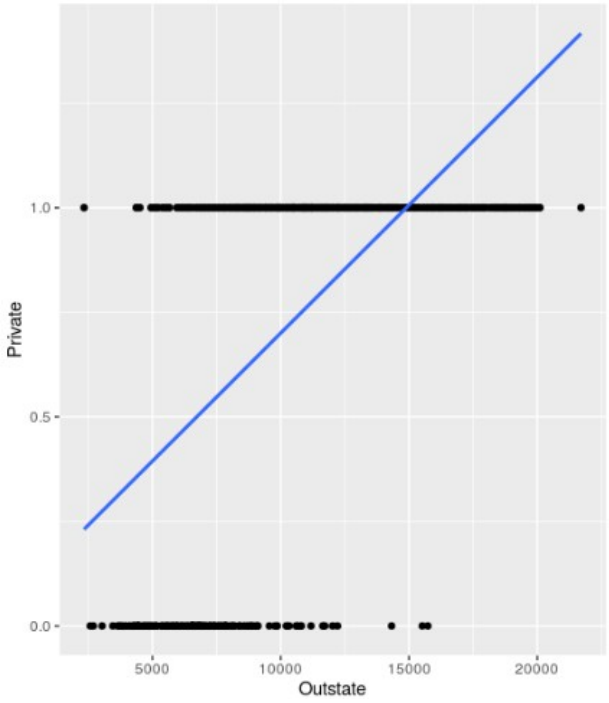
```
ggplot(data=College.recoded ,mapping=aes(x=Outstate,y=Private)) +
+     geom_point() +
+     geom_smooth(method="lm", se=FALSE)
```

Provide figure below:

Simple Linear model scatter plot



Fitted Linear graph w/ geom_smooth()



5. [10%] Make predictions of the classes for the training dataset using your simple linear regression model from Question 3, and add these predictions to the data frame **`College.recoded`** that you created in Question 1. Create a confusion matrix and estimate classification accuracy for the training dataset.

Provide code below:

Creating predictions :-

```
> lmOutstate.predicts <- predict(lmOutstate.fit, type = "response")
```

Adding predictions to recoded dataframe :-

```
College.recoded.withProbs <- College.recoded %>%
+ mutate(probs = lmOutstate.predicts, pred = ifelse(probs > 0.5, 1, 0))
```

Creating Confusion Matrix :-

```
Cmatrix5 <- College.recoded.withProbs %>%
+ select(Private, pred) %>%
+ table()
> Cmatrix5
       pred
Private   0   1
      0 111 101
      1  38 527
```

Provide confusion matrix below:

```
> Cmatrix5
       pred
Private   0   1
      0 111 101
      1  38 527
```

Provide accuracy estimate below:

```
> College.recoded.withProbs %>%
+ summarize(accuracy = mean(pred == Private))
   accuracy
1 0.8211068
```

6. [12%] Perform the same operations as in Question 5, except use the multiple linear regression model from Question 2. Has the classifier improved in performance on the training data compared to the results from Question 5? Explain why you conclude this, and provide a reason as to why this model did or did not improve upon the training error from Question 5.

Provide code below:

Creating Predictions :-

```
lmMultiple.predicts <- predict(lmMultiple.fit, type = "response")
```

Adding to recoded data frame :-

```
College.recoded.withProbs.MultipleLR <- College.recoded %>%
+ mutate(probs = lmMultiple.predicts, pred = ifelse(probs > 0.5, 1, 0))
```

Creating confusion matrix :-

```
Cmatrix6 <- College.recoded.withProbs.MultipleLR %>%
+ select(Private, pred)%>%
+ table()
```

Provide confusion matrix below:

```
> Cmatrix6
       pred
Private   0   1
      0 178  34
      1  13 552
```

Provide accuracy estimate below:

```
> College.recoded.withProbs.MultipleLR %>%
+ summarize(accuracy = mean(pred == Private))
   accuracy
1 0.9395109
```

Provide answer to questions below:

The linear regression model establishes the relationship between only 2 variables, the dependent variable i.e. prediction (private) and the independent variable i.e. input feature (outstate). Although this 'outstate' feature proved to be important in the prediction model, the multiple regression model demonstrated a better accuracy. This is because the 'outstate'

feature isn't the only feature that may be important, so computing the prediction while using the other 17 features as well yielded a better prediction. This is common as it is rare that a dependent variable can be deduced from only one variable.

7. [10%] Fit a multiple logistic regression model to predict private school status with the 17 other features. Which feature is most important in this model, and what evidence tells you that?

Provide code below:

```
glm.mult.fit <- glm(Private ~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad + Outstate + Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend + Grad.Rate, College.recoded, family = "binomial")
```

```
> summary(glm.mult.fit)

Call:
glm(formula = Private ~ Apps + Accept + Enroll + Top10perc +
    Top25perc + F.Undergrad + P.Undergrad + Outstate + Room.Board +
    Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni +
    Expend + Grad.Rate, family = "binomial", data = College.recoded)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7673  -0.0318   0.0502   0.1717   4.2070

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.574e-02  1.860e+00  -0.014  0.98896
Apps        -5.138e-04  2.284e-04  -2.249  0.02452 *
Accept       9.328e-05  4.382e-04   0.213  0.83144
Enroll       1.331e-03  8.487e-04   1.568  0.11687
Top10perc    8.451e-03  2.841e-02   0.297  0.76614
Top25perc    7.305e-03  1.895e-02   0.385  0.69993
F.Undergrad -4.168e-04  1.472e-04  -2.832  0.00462 **
P.Undergrad  1.836e-05  1.348e-04   0.136  0.89164
Outstate     6.822e-04  1.099e-04   6.207  5.4e-10 ***
Room.Board   1.901e-04  2.575e-04   0.738  0.46053
Books        2.059e-03  1.318e-03   1.562  0.11837
Personal    -3.283e-04  2.700e-04  -1.216  0.22395
PhD         -6.027e-02  2.665e-02  -2.262  0.02371 *
Terminal    -3.590e-02  2.580e-02  -1.392  0.16402
S.F.Ratio   -8.461e-02  6.076e-02  -1.393  0.16372
perc.alumni  4.782e-02  2.097e-02   2.280  0.02260 *
Expend       2.077e-04  1.207e-04   1.721  0.08529 .
Grad.Rate    1.634e-02  1.171e-02   1.395  0.16294
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 910.75  on 776  degrees of freedom
Residual deviance: 239.50  on 759  degrees of freedom
AIC: 275.5

Number of Fisher Scoring iterations: 8
```

Provide answer to question below:

- The most important feature is Outstate. This is because it has the lowest Pr value of them all.

8. [5%] Fit a simple logistic regression model to predict private school status based on the most important feature from Question 7. Is this feature still important in this model, and what evidence tells you that?

Provide code below:

```
glm.LINEAR.fit <- glm(Private ~ Outstate, College.recoded, family = "binomial")
```

//note: I realize that 'linear' is the incorrect term to use here as linear != simple. But it was too late to change.

```
> summary(glm.LINEAR.fit)

Call:
glm(formula = Private ~ Outstate, family = "binomial", data = Colleg
e.recoded)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2535  -0.5405   0.2167   0.5774   2.4935

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.522e+00  4.094e-01  -11.04   <2e-16 ***
Outstate     6.235e-04  4.957e-05   12.58   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 910.75  on 776  degrees of freedom
Residual deviance: 574.89  on 775  degrees of freedom
AIC: 578.89

Number of Fisher Scoring iterations: 6
```

Provide answer to question below:

Yes, this feature is still important. The evidence is that the Pr value is very low and close to 0.

9. [10%] Visualize the simple logistic regression model from Question 8 using a scatter plot and the fitted logistic model using the appropriate arguments in **`geom_smooth()`**.

Provide code below:

Simple logistic regression model visualization w/ scatter plot:-

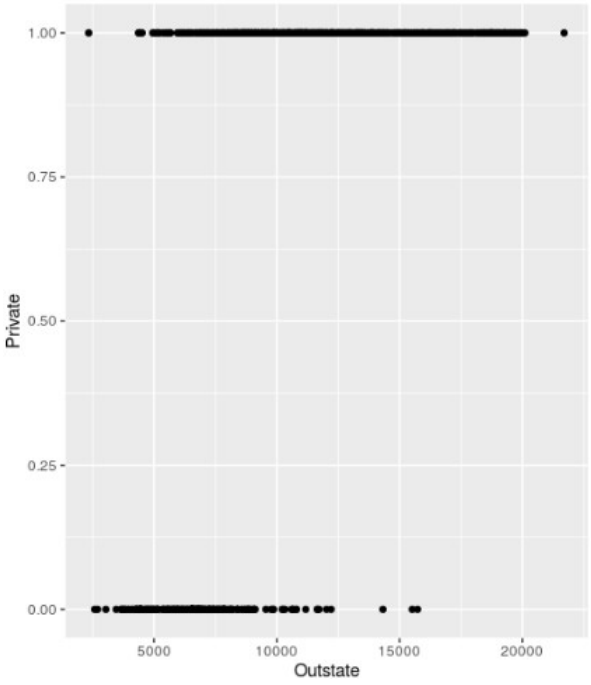
```
> ggplot(data = glm.LINEAR.fit)+
+ geom_point( mapping = aes(x = Outstate, y = Private))
```

Fitted logistic model visualization using appropriate arguments w/ geom_smooth():-

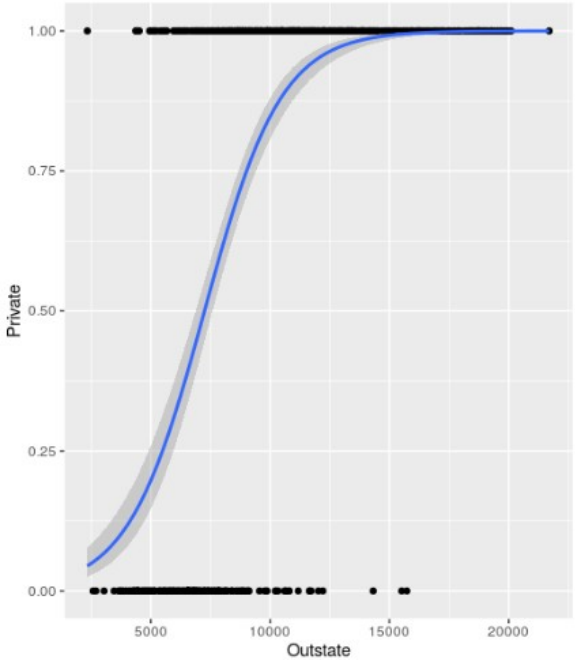
```
ggplot(data=College.recoded ,mapping=aes(x=Outstate,y=Private)) +
+  geom_point() +
+   geom_smooth(method="glm", method.args = c(family = "binomial"))
```

Provide figure below:

Visualizing simple logistic regression model using scatter plot:



Fitted logistic regression model using geom_smooth() w/ appropriate arguments:-



10. [10%] Make predictions of the classes for the training dataset using your simple logistic regression model from Question 8, and add these predictions to the data frame **`College.recoded`** that you created in Question 1. Create a confusion matrix and estimate classification accuracy for the training dataset.

Provide code below:

Creating predictions:-

```
> LogR.predicts <- predict(glm.LINEAR.fit, type = "response")
```

Adding to data frame:-

```
> College.recoded.withProbs.LogRsimple <- College.recoded %>%
+ mutate(probs = LogR.predicts, pred = ifelse(probs > 0.5, 1, 0))
```

Creating Confusion Matrix :-

```
Cmatrix10logRsimple <- College.recoded.withProbs.LogRsimple %>%
+ select(Private, pred)%>%
+  table()
```

Provide confusion matrix below:

```
> Cmatrix10logRsimple
       pred
Private   0   1
      0 140  72
      1  53 512
```

Provide accuracy estimate below:

```
> College.recoded.withProbs.LogRsimple %>%
+ summarize(classification = mean(pred == Private))
  classification
1      0.8391248
```

11. [12%] Perform the same operations as in Question 10, except use the multiple logistic regression model from Question 7. Has the classifier improved in training accuracy compared to the results of the multiple linear regression model from Question 6? Explain why you conclude this, and provide a reason as to why this model did or did not improve upon the training error from Question 6.

Provide code below:

Creating predictions :-

```
LogRMultiple.predicts <- predict(glm.mult.fit, type = "response")
```

Adding to data frame :-

```
College.recoded.withProbs.LogRMultiple <- College.recoded %>%
+ mutate(probs = LogRMultiple.predicts, pred = ifelse(probs > 0.5, 1, 0))
```

Creating Confusion Matrix :-

```
Cmatrix11logRmulti <- College.recoded.withProbs.LogRMultiple %>%
+ select(Private, pred)%>%
+ table()
```

Provide confusion matrix below:

```
> Cmatrix11logRmulti
       pred
Private   0   1
      0 191  21
      1  22 543
```

Provide accuracy estimate below:

```
> College.recoded.withProbs.LogRMultiple %>%
+ summarize(accuracy = mean(pred == Private))
   accuracy
1 0.9446589
```

Provide answer to questions below:

Multiple regression models tend to work better than simple regression models as a dependent variable is highly unlikely to depend on one independent variable, even if the importance of that IV is high as seen in the case here.

A valid reason that can be seen here is that although the Outstate feature is important, there are other features that are important as well such as the F.Undergrad, PhD, perc.alumni, and Apps features. This importance is shown in the multiple logistic regression summary.

So because of this, the multiple logistic regression models yields a higher accuracy as it also takes into account the lesser important but nonetheless important features in its prediction.

CAP 5768: Midterm Exam

Due on Canvas by Wednesday, March 18, 2020 at 11:59pm

Place name here: Yousef Al-Kafif

Preliminary instructions

All analyses must be performed in R using the **`tidyverse`** package that we discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of assignment

In this assignment you will be analyzing the **`abalone.csv`** dataset, which is available on Canvas. Abalone are shellfish, and the age of abalone is typically determined by cutting open the shell, staining it, and counting the number of rings under a microscope, which is a laborious task. Here, we explore a number of physical measurements about abalone, and in the end attempt to make a model to be able to predict abalone age (*i.e.*, its number of rings) from features other than rings. The columns in the dataset are:

Name	Description
`Sex`	M (male), F (female), and I (infant)
`Length`	Longest shell measurement
`Diameter`	Perpendicular to length
`Height`	With meat in shell
`WholeWeight`	Whole abalone
`ShuckedWeight`	Weight of meat
`VisceraWeight`	Gut weight (after bleeding)
`ShellWeight`	After being dried
`Rings`	Number of rings (+1.5 gives age in years)

Instructions for loading abalone dataset into your RStudio Cloud environment

Recall that to upload a file to RStudio Cloud, you first must download the **`abalone.csv`** file to your computer. Once the file is downloaded, within the "Files" panel of the RStudio Cloud environment, click "Upload" and browse to the appropriate directory on your computer to upload the **`abalone.csv`** file.

The **`abalone.csv`** file can be loaded using the **`read_csv()`** function of the **`readr`** package that comes loaded with **`tidyverse`**, and assigned to an object called **`abalone`** as

```
abalone <- read_csv("abalone.csv")
```

If you are having trouble loading the file, then refer back to the video lecture on Linear Regression where this was demonstrated in class.

Questions and problems

1. [10%] How many abalone observations are there in the dataset? How many features are in the dataset? How many of these features are categorical?

Provide answer to question below:

- There are 4,177 observations of which contains 9 different features. 2 of these features are categorical, this would be the Sex (M or F or I) and Rings (an integer value i.e. a whole number). Although 'Rings' is an integer value, in this particular case it could represent a category as they are a characteristic that distinguishes observations into types based on how many rings they have which represents their age, and this feature does not dictate any value of importance other than separating into categories. E.g. if an integer type feature in a dataset were to represent colors, and a lower number represents darker colors and so forth, this feature would be treated as a continuous variable and not a categorical.

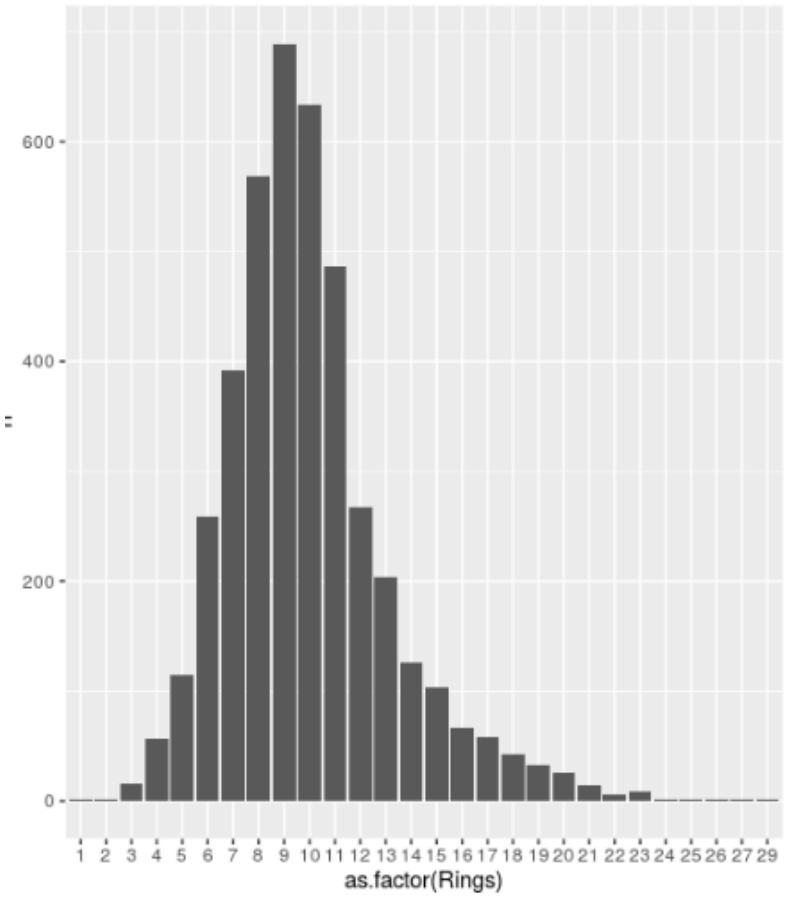
2. [10%] Use a bar plot to show the number of abalone observed for each age (ring number). What appears to be the most prevalent number of rings?

Note: Remember to convert the integer-valued feature **`Rings`** to a factor using the **`as.factor()`** function prior to plotting to explicitly tell R that each integer value for the feature **`Rings`** is a separate category.

Provide code below:

```
abalone %>%
    count(Rings) %>%
    ggplot(mapping = aes(x = as.factor(Rings), y = n)) +
    geom_col()
```

Provide figure below:



Provide answer to question below:

- Based on the bar plot, the most prevalent number of rings is 9.

3. [15%] Create a new categorical feature called adult (non-infant) to distinguish between adult and infant abalone, and use a stacked bar plot to visualize the relative proportions of adult abalone and infant abalone for each age (ring number). Do the distributions of these relative proportions of adult and infant abalone make sense based on the ring number?

Note: Remember to convert the integer-valued feature **`Rings`** to a factor using the **`as.factor()`** function prior to plotting, to explicitly tell R that each integer value for the feature **`Rings`** is a separate category.

Provide code below:

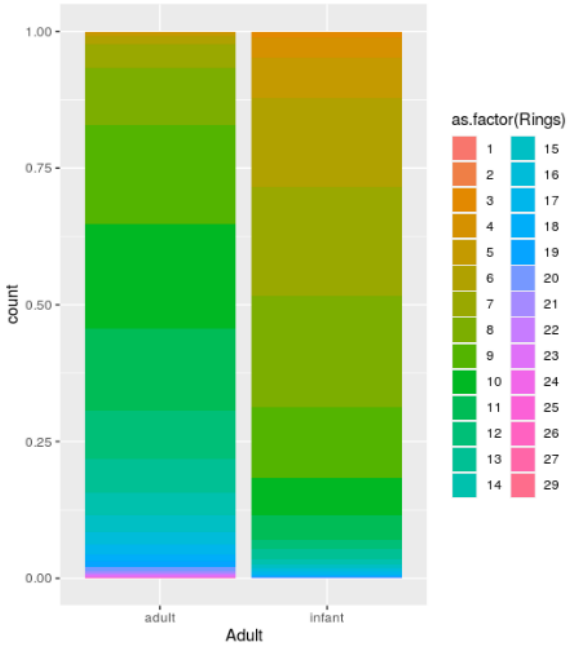
Creating adult variable:

```
abalone3 <- abalone %>%
+     mutate(Adult=case_when(
+         (Sex == "M") | (Sex == "F") ~ "adult",
+         (Sex == "I") ~ "infant"
+     ))
```

Plotting:

```
ggplot(data = abalone3) +
   geom_bar(mapping = aes(x = Adult, fill = as.factor(Rings)),
            position = "fill")
```

Provide figure below:



Provide answer to questions below:

The distribution seems to correlate with adults having more rings and infants having less. But the data may be jumbled as there seems to be a low amount of adults having rings in the range of 6-5 which could arguably be considered an infant based on the data (the infants seem to be considered between the age of 1-10).

The infants have a significant amount of which has rings/age between 7-10 which could arguably be considered adults based on the data (adults seem to be considered between the age/rings of 6-29).

Overall, it seems that there was no hard cutoff value between adults and infants based on rings and that has obscured the data and caused the data to make less sense than it could.

4. [15%] Create a new categorical feature called adult (non-infant) to distinguish between adult and infant abalone. Use box plots with appropriate notches to examine whether the median numbers of rings are different between adult and infant abalone. Are the median number of rings for adult abalone smaller, larger, or roughly the same as for infant abalone?

Provide code below:

Creating dataframe with adult and infant category :-

```
abalone4 <- mutate(abalone, adult = (Sex == 'M') | (Sex == 'F'), infant = (Sex == 'I'))
```

Creating AdultsOnly & InfantsOnly dataframe:-

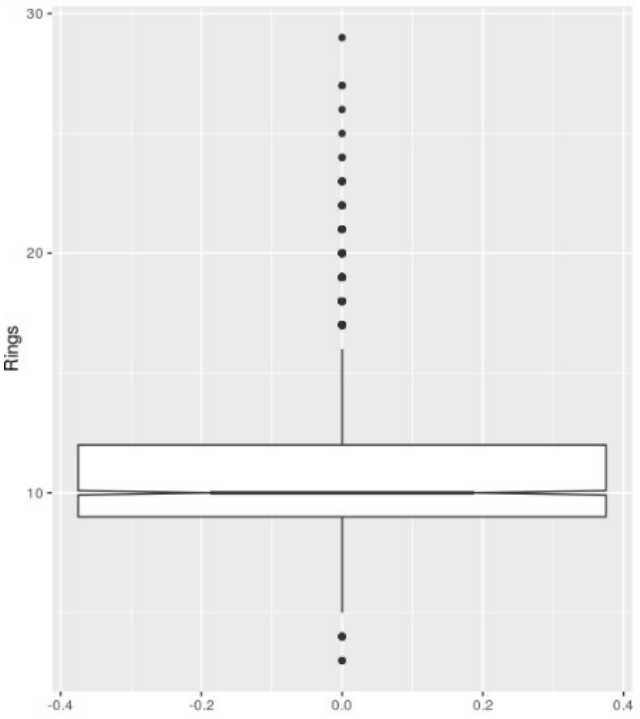
```
AdultsOnly <- abalone4 %>%
+  filter(adult == "TRUE")

InfantsOnly <- abalone4 %>%
+ filter(infant == "TRUE")
```

Plotting AdultsOnly and InfantsOnly :-

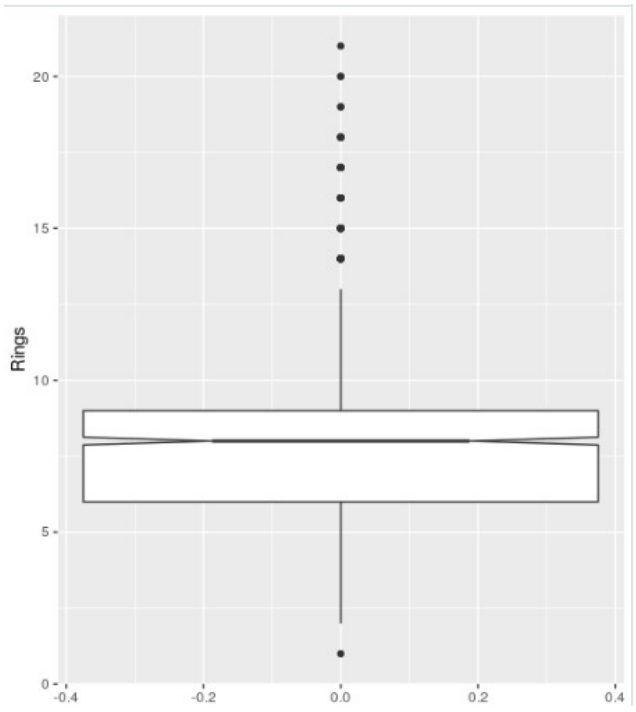
```
ggplot(data=AdultsOnly, mapping = aes(y=Rings)) +
+ geom_boxplot( notch = TRUE)


ggplot(data=InfantsOnly, mapping = aes(y=Rings)) +
+ geom_boxplot( notch = TRUE)
```

Provide figure below:

Adults



Infants



Provide answer to questions below:

The median number of rings for adult abalones are higher than infant abalones. With adults having a median of about 10 and infants having a median of about 6.

5. [15%] Considering only adult (non-infant) abalone, use box plots with appropriate notches to examine whether the median numbers of rings are different between male and female abalone. Are the median number of rings for male abalone smaller, larger, or roughly the same as for female abalone?

Provide code below:

Creating smaller dataframes:-

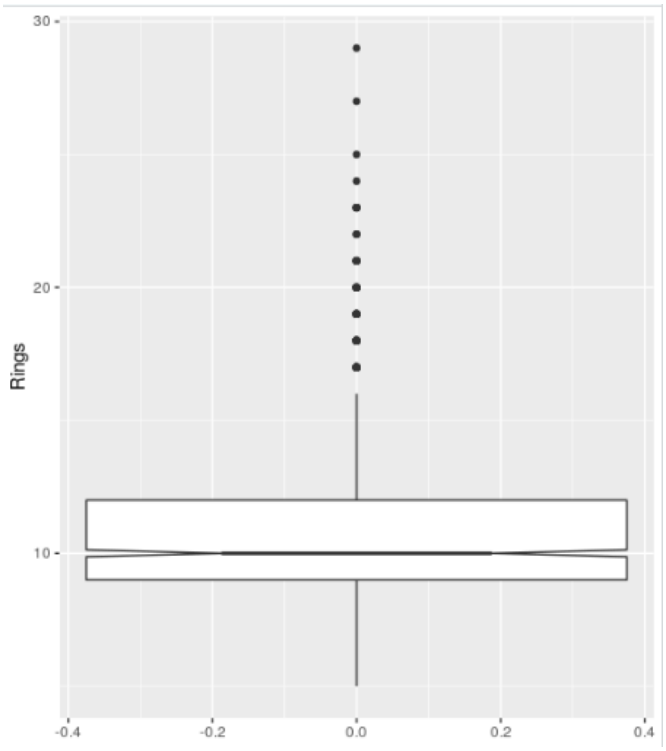
```
AdultsFemaleOnly <- abalone %>%
+ filter(Sex == "F")

AdultsMaleOnly <- abalone %>%
+  filter(Sex == "M")
```

Creating Plot:-

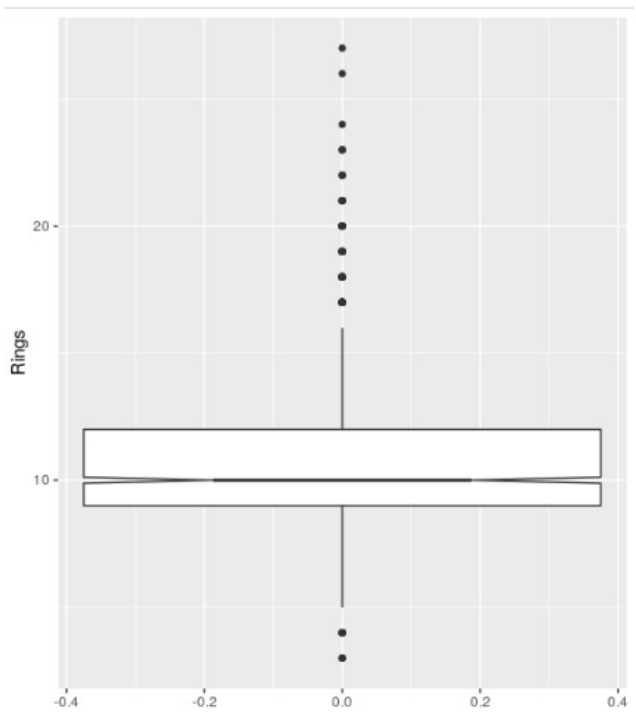
```
ggplot(data=AdultsFemaleOnly, mapping = aes(y=Rings)) +
+ geom_boxplot( notch = TRUE)

ggplot(data=AdultsMaleOnly, mapping = aes(y=Rings)) +
+  geom_boxplot( notch = TRUE)
```

Provide figure below:

Females



Males



Provide answer to questions below:

The median number of rings for male abalone are roughly the same as female abalones.

6. [15%] Considering only adult abalone, use box plots with appropriate notches to examine whether the median proportion of an abalone's weight being meat is different between male and female abalone. The proportion of an abalone's weight being meat is the abalone's meat weight divided by its entire weight. Are the median proportion of an abalone's weight being meat for male abalone smaller, larger, or roughly the same as for female abalone?

Provide code below:

Creating Dataframe with Adult females and one for Adult males and Infants separated :-

```
AdultsMaleOnly <- abalone %>%
+  filter(Sex == "M")

AdultsFemaleOnly <- abalone %>%
+ filter(Sex == "F")
```

Adding meat weight variable :-

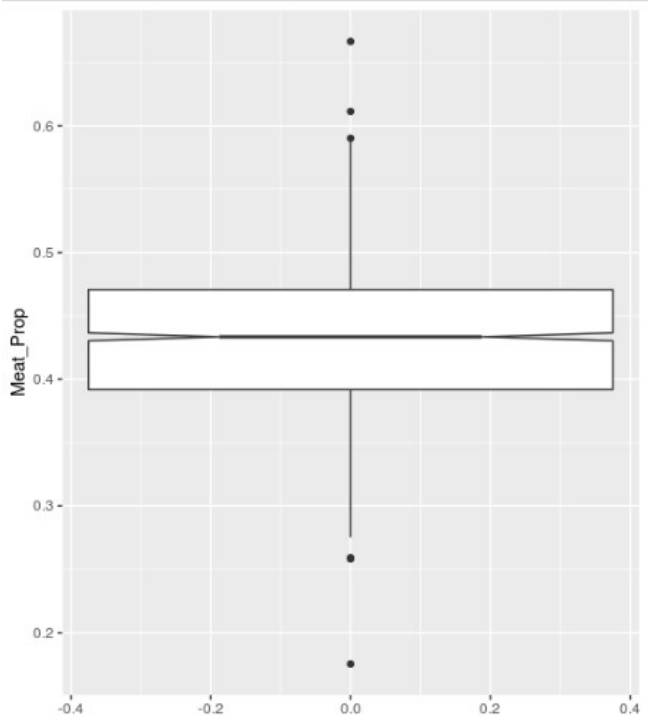
```
AdultsFemaleOnly <- mutate(AdultsFemaleOnly, Meat_Prop = ShuckedWeight/WholeWeight )

AdultsMaleOnly <- mutate(AdultsMaleOnly, Meat_Prop = ShuckedWeight/WholeWeight )
```

Plotting :-

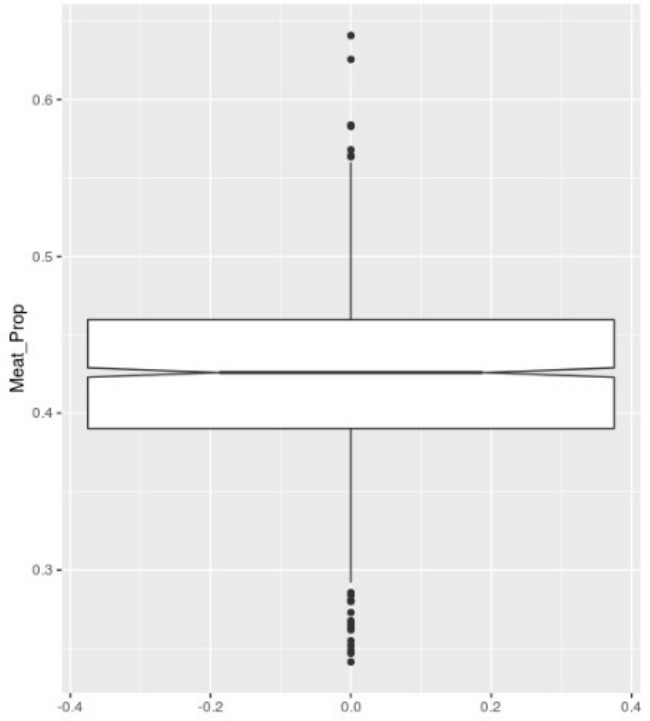
```
ggplot(data= AdultsMaleOnly, mapping = aes(y= Meat_Prop)) +
+  geom_boxplot( notch = TRUE)

ggplot(data= AdultsFemaleOnly, mapping = aes(y= Meat_Prop)) +
+  geom_boxplot( notch = TRUE)
```

Provide figure below:

Males Meat Proportions



Females Meat Proportion



Provide answer to questions below:

The meat proportions of both female and male adults appear to be roughly the same, with the male adults tending to be very slightly heavier and with less outliers.

7. [15%] Using multiple linear regression, fit a linear model to predict the response **`Rings`** from input features **`Length`**, **`Diameter`**, **`Height`**, **`WholeWeight`**, **`VisceraWeight`**, and **`ShellWeight`**. Based on the fitted model, is at least one of the input features related to the response? If so, then what evidence and statistical test answers this question?

Provide code below:

lm.fit <- lm(formula = Rings ~ Length + Diameter + Height + WholeWeight + VisceraWeight + ShellWeight, data = abalone)

```
> summary(lm.fit)

Call:
lm(formula = Rings ~ Length + Diameter + Height + WholeWeight +
    VisceraWeight + ShellWeight, data = abalone)

Residuals:
     Min       1Q   Median       3Q      Max
-13.2528  -1.4527  -0.4955   0.8849  16.4185

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.6744     0.2863  12.833  < 2e-16 ***
Length         -5.1832     1.9456  -2.664  0.00775 **
Diameter       14.3178     2.3927   5.984 2.36e-09 ***
Height         13.1757     1.6550   7.961 2.18e-15 ***
WholeWeight    -5.6705     0.4379 -12.949  < 2e-16 ***
VisceraWeight  -1.2027     1.3433  -0.895  0.37067
ShellWeight    26.0171     0.9493  27.406  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.372 on 4170 degrees of freedom
Multiple R-squared:  0.4593,    Adjusted R-squared:  0.4586
F-statistic: 590.4 on 6 and 4170 DF,  p-value: < 2.2e-16
```

Provide answer to questions below:

```
> summary(lm.fit)

Call:
lm(formula = Rings ~ Length + Diameter + Height + WholeWeight +
    VisceraWeight + ShellWeight, data = abalone)

Residuals:
     Min       1Q   Median       3Q      Max
-13.2528  -1.4527  -0.4955   0.8849  16.4185

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.6744     0.2863  12.833  < 2e-16 ***
Length         -5.1832     1.9456  -2.664  0.00775 **
Diameter       14.3178     2.3927   5.984 2.36e-09 ***
Height         13.1757     1.6550   7.961 2.18e-15 ***
WholeWeight    -5.6705     0.4379 -12.949  < 2e-16 ***
VisceraWeight  -1.2027     1.3433  -0.895  0.37067
ShellWeight    26.0171     0.9493  27.406  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.372 on 4170 degrees of freedom
Multiple R-squared:  0.4593,    Adjusted R-squared:  0.4586
F-statistic: 590.4 on 6 and 4170 DF,  p-value: < 2.2e-16
```

The lower the p-value, the higher the relation as it gets closer to 0. Thus it seems that all the input features apart from VisceraWeight are related to the response. Especially the WholeWeight.

The statistical test is the F-Statistic (the more extreme positive values, the better) and the evidence could also be provided by singling out features through simple linear regression like so and noting the F-Statistic, P-value (closer to 0 – the better), and R-Squared values (the higher – the better).

```
> lm.fit <- lm(formula = Rings ~ WholeWeight , data = abalone)
> summary(lm.fit)

Call:
lm(formula = Rings ~ WholeWeight, data = abalone)

Residuals:
    Min      1Q  Median      3Q     Max
-6.2693 -1.7518 -0.6874  1.0177 15.7029

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.98924    0.08244   84.78   <2e-16 ***
WholeWeight  3.55291    0.08562   41.50   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.713 on 4175 degrees of freedom
Multiple R-squared:  0.292,     Adjusted R-squared:  0.2919
F-statistic:  1722 on 1 and 4175 DF,  p-value: < 2.2e-16
```

This simple linear regression using the WholeWeight displays that at least one input feature (the WholeWeight in this case) is related to the response. As shown by the low p-value and the moderately good R-squared value (the lower the worse).

8. [5%] Based on your linear model fit from Question 7, are there any features not related to the response? If so, then what evidence and statistical test answers this question?

Provide answer to questions below:

The VisceraWeight input feature is not related to the response as its p-value was significantly higher than the other input features p-values and also it is not as close to 0 as the other features as shown below.

The F-Statistic test, p-value, and R-squared value can prove this.

Multiple Linear Regression:

```
> summary(lm.fit)

Call:
lm(formula = Rings ~ Length + Diameter + Height + WholeWeight +
    VisceraWeight + ShellWeight, data = abalone)

Residuals:
     Min       1Q   Median       3Q      Max
-13.2528  -1.4527  -0.4955   0.8849  16.4185

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.6744     0.2863  12.833  < 2e-16 ***
Length         -5.1832     1.9456  -2.664  0.00775 **
Diameter       14.3178     2.3927   5.984 2.36e-09 ***
Height         13.1757     1.6550   7.961 2.18e-15 ***
WholeWeight    -5.6705     0.4379 -12.949  < 2e-16 ***
VisceraWeight  -1.2027     1.3433  -0.895  0.37067
ShellWeight    26.0171     0.9493  27.406  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.372 on 4170 degrees of freedom
Multiple R-squared:  0.4593,    Adjusted R-squared:  0.4586
F-statistic: 590.4 on 6 and 4170 DF,  p-value: < 2.2e-16
```

Simple linear regression with only VisceraWeight :

```
> lm.fit <- lm(formula = Rings ~ VisceraWeight, data = abalone)
> summary(lm.fit)

Call:
lm(formula = Rings ~ VisceraWeight, data = abalone)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5200 -1.7622 -0.7097  1.0310 16.9782

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.25743    0.08307   87.37   <2e-16 ***
VisceraWeight 14.81923    0.39322   37.69   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.785 on 4175 degrees of freedom
Multiple R-squared:  0.2538,    Adjusted R-squared:  0.2537
F-statistic:  1420 on 1 and 4175 DF,  p-value: < 2.2e-16
```

The evidence is the extremely low R-squared value.

CAP 5768: Homework Assignment 2

Due on Canvas by Thursday, February 27, 2020 at 11:59pm

Place name here:
- Yousef Al-Kafif

Preliminary instructions

All analyses must be performed in R using the **`tidyverse`** package that we discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of assignment

In this assignment you will be analyzing the **`flights`** data frame that we extensively discussed in class, which has information on 19 features for 336,776 flights that left New York City during 2013. The purpose of this assignment is to become more familiar with data transformations and exploratory data analysis, requiring you to think of solutions to questions. You can obtain the **`flights`** data frame by installing and loading the **`nycflights13`** library.

Questions and problems

1. [25%] Using box plots with appropriate notches, is the median distance between airports for canceled flights shorter, longer, or roughly the same as for non-canceled flights? Provide an explanation for the result you found.

Provide code below:

For cancelled flights :-

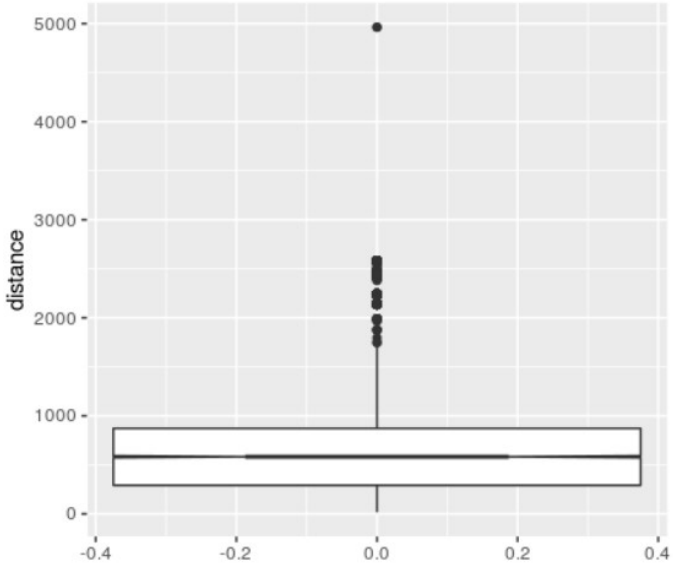
```
> cancelled <- flights %>%
+ filter (is.na(dep_delay), is.na(arr_delay))

> ggplot(data=cancelled, mapping = aes(y=distance)) +

+ geom_boxplot( notch = TRUE)
```

For not cancelled flights :-

```
> not_cancelled <- flights %>%
+ filter (!is.na(dep_delay), !is.na(arr_delay))

> ggplot(data=not_cancelled, mapping = aes(y=distance)) +
+  geom_boxplot( notch = TRUE)
```

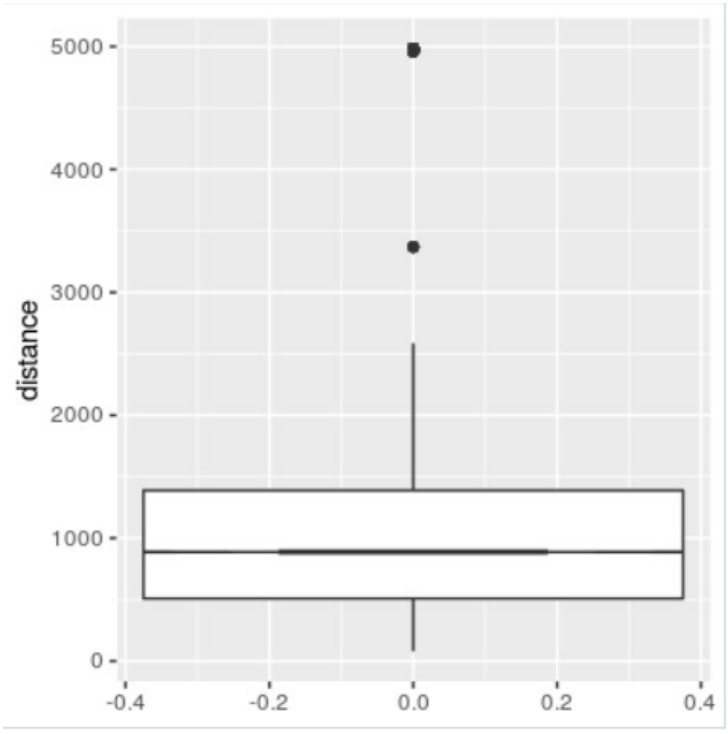
Provide figure below:

CANCELLED FLIGHTS BOX PLOT :-



Median = ~600

NOT CANCELLED FLIGHTS BOX PLOT :-



Median = ~900

Provide answer to questions below:

The median for non-cancelled flights is higher at about 900 whilst the cancelled flights median is at about 600. This may be because an airline is aiming at inconveniencing the least amount of passengers, so they would rather cancel smaller flights which tend to be shorter distances rather than the larger flights that tend to run longer distances. The medians are the bold lines i.e. the 50^{th} percentile.

2. [40%] Do canceled flights tend to occur more often in certain months? That is, compared to other months, are there certain months with a large proportion of their flights canceled? Provide an explanation for the result you found. To answer this question, generate a bar plot with month of year on the *x*-axis and proportion of that month's flights that are canceled on the *y*-axis.

Note: Unlike a typical bar plot, you will need to compute and provide the values on the *y*-axis. You can use the function **`geom_col()`** to generate a bar plot for which you provide the appropriate *x*- and *y*-axis features. In addition, like for bar plots, **`geom_col()`** will expect that the feature on the *x*-axis is categorical. To explicitly tell R that each integer value for the feature **`month`** is a category, use the code **`as.factor(month)`** to convert the month feature into a categorical variable taking 12 values (1, 2, …, 12).

Provide code below:

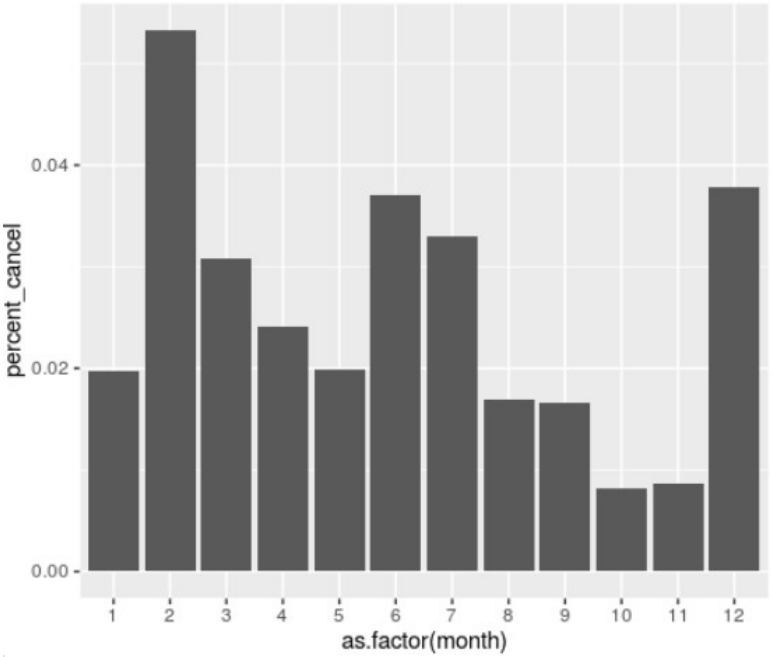
Creating ratio of cancelled flights per month dataframe :-

```
> ratioCancelPerMonth <- flights %>%
+     group_by(month) %>%
+     summarise( cancelled = sum(is.na(dep_delay)),
+                not_cancelled=sum(!is.na(dep_time)),

                Percent_cancel = sum(cancelled/ not_cancelled) )
```

Plotting :-

```
> ggplot(data=ratioCancelPerMonth, mapping = aes(x=as.factor(month),
y=percent_cancel)) +
+  geom_col()
```

Provide figure below:




```
> ratioCancelPerMonth
# A tibble: 12 x 4
   month cancelled not_cancelled percent_cancel
   <int>     <int>         <int>          <dbl>
 1     1       521         26483         0.0197
 2     2      1261         23690         0.0532
 3     3       861         27973         0.0308
 4     4       668         27662         0.0241
 5     5       563         28233         0.0199
 6     6      1009         27234         0.0370
 7     7       940         28485         0.0330
 8     8       486         28841         0.0169
 9     9       452         27122         0.0167
10    10       236         28653         0.00824
11    11       233         27035         0.00862
12    12      1025         27110         0.0378
```

Provide answer to questions below:

The data illustrates that there is a significantly larger proportion of cancelled flights in the months of February, June, July, and December. December and February cancellations can be attributed to winter storms as that is a common occurrence in NYC, more specifically there was a historically cold winter hitting the states at that year. As described here : https://en.wikipedia.org/wiki/2013%E2%80%9314_North_American_winter . June and July cancellations could be caused by the increase of international flights as tourism increases in the summer, along with these international flights come higher security measures and this could lead to more cancellations.

3. [35%] Is there a relationship between average distance between airports for flights flown on each of the 365 days of the year and the standard deviation of the distances between airports for flights flown on each of those days? Provide an explanation for the result you found. Generate a scatter plot to examine this question, and add a fitted line with confidence intervals through the scatter plot using the **'lm'** method of the **geom_smooth()** function.

Provide code below:

Creating Dataframe with Avg Distance and STD for each of the 365 days (using my not_cancelled dataframe from question 1) :-

- Note : I imported the 'lubridate' library to use a 'date' object which is optimal for continuous graphs through time. The ymd() function is from this library.

```
AvgDistanceAndSD <- not_cancelled %>%
     + mutate(date = ymd(paste(year, month, day))) %>%
     + group_by(year, month, day) %>%
     + summarise(AvgDist = mean(distance), date = first(date),
     STD = sd(distance))
```

Creating plots :-

Plot relating Average distance to STD:-

```
ggplot(data=AvgDistanceAndSD, mapping = aes(x=AvgDist, y=STD)) +
+ geom_point() +
+ geom_smooth(method = "lm" )
```

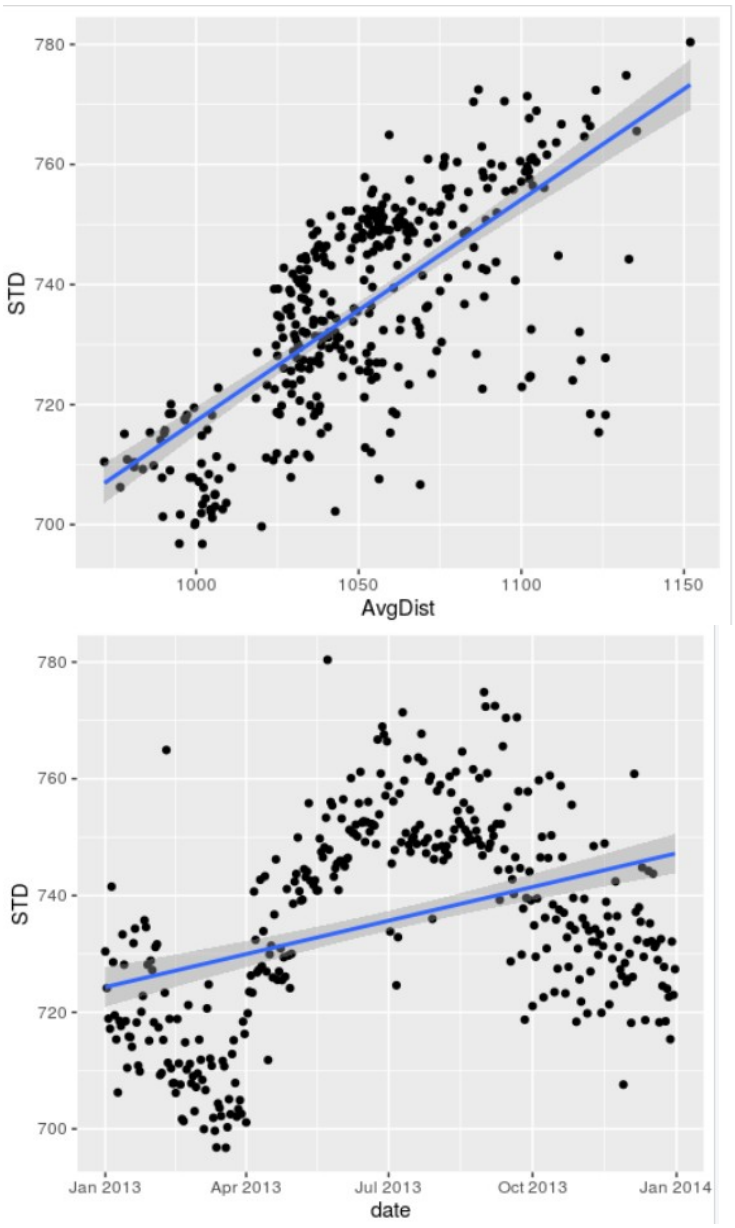
Plot relating average distance to 365 days:-

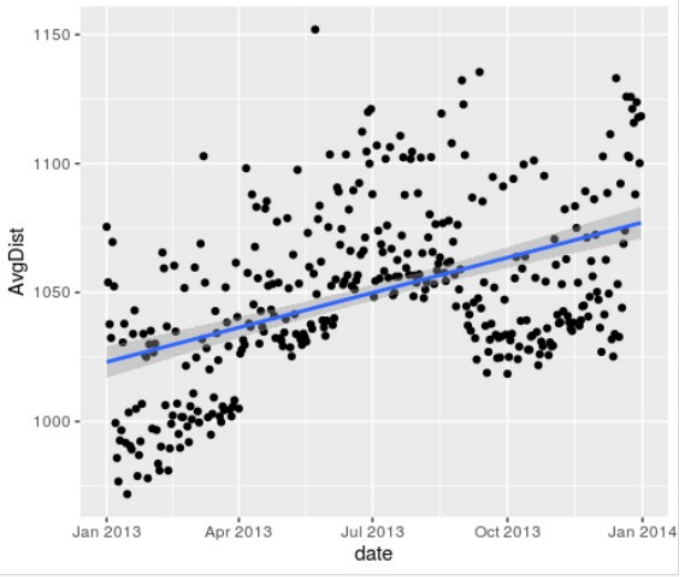
```
ggplot(data=AvgDistanceAndSD, mapping = aes(x=date, y=AvgDist))
+
+     geom_point() +
+     geom_smooth(method = "lm" )
```

Plot relating STD to 365 days :-

```
> ggplot(data=AvgDistanceAndSD, mapping = aes(x=date, y=STD)) +
+ geom_point() +
+ geom_smooth(method = "lm" )
```

Provide figure below:





Provide answer to questions below:

Yes, there appears to be a positive relationship between STD and average distance throughout the 365 days in this particular case, although in terms of theory the two have no direct relationship. This positive relationship is because standard deviation measures the amount of variability/dispersion of a set of data from the mean. This dispersion is illustrated in the space between the points on the average distance vs time graph, as the graphs points are getting more dispersed as time goes by – as a consequence the standard deviation is also increasing, especially in the middle of the year where the dispersion spiked.

CAP 5768: Homework Assignment 1

Due on Canvas by Monday, February 10, 2020 at 11:59pm

Place name here:

YOUSEF AL-KAFIF

Preliminary instructions

All analyses will must be performed in R using the **`tidyverse`** package that we discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of assignment

In this assignment you will be analyzing the **`diamonds`** data frame that comes with **`tidyverse`**, which contains 53,930 observations on 10 features related to diamonds. A large portion of this assignment is to get to know R better, and so you will need to use resources such as the **`ggplot2`** cheat sheet on Canvas and the help menus to learn how to make certain types of plots.

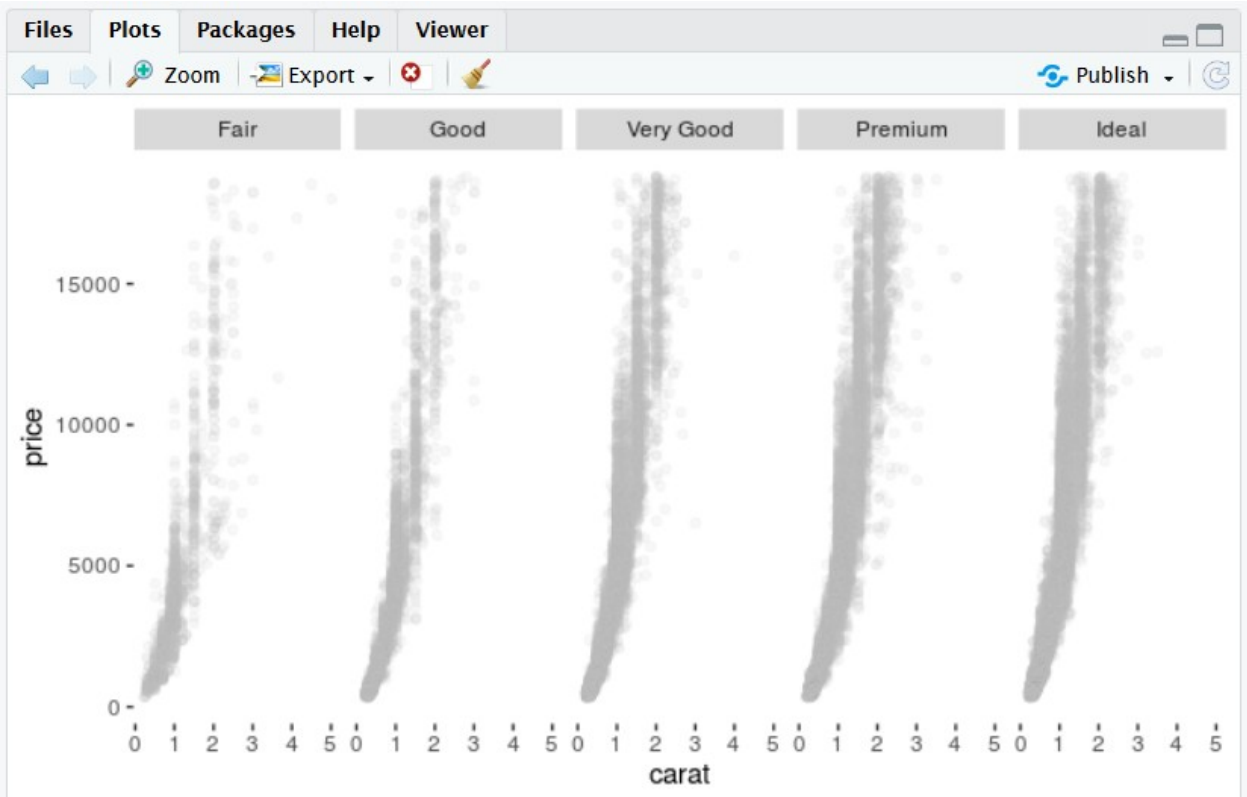
Questions and problems

1. [10%] Generate a jittered scatter plot of the price of a diamond as a function of diamond weight that is broken into five sub-panels across two rows, where each sub-panel represents a different cut quality. Depict the points as the color gray, with a transparency alpha value of 0.1. Moreover, set the background to be white rather than gray that is used by default.

Provide code below:

```
> ggplot(data = diamonds) +
+     geom_jitter(mapping = aes(x = carat, y = price), color = "gray", alpha =
0.1) +
+     facet_wrap(~ cut, ncol = 5, nrow = 2) +
+     theme(panel.background = element_rect(fill = "white", color = "white"))
>
```

Provide figure below:



2. [5%] What is a rug plot, and what geometric layer (function) in **`ggplot2`** can be used to generate one?

A rug plot can be seen as a 1D scatter plot or a histogram with zero-width bins. It is not a completely different plot but a 1D display 'layer' that you could add to existing plots to highlight certain information.

Can be done so using the geom_rug() function which operates the same as the other geometric layer functions (same parameters).

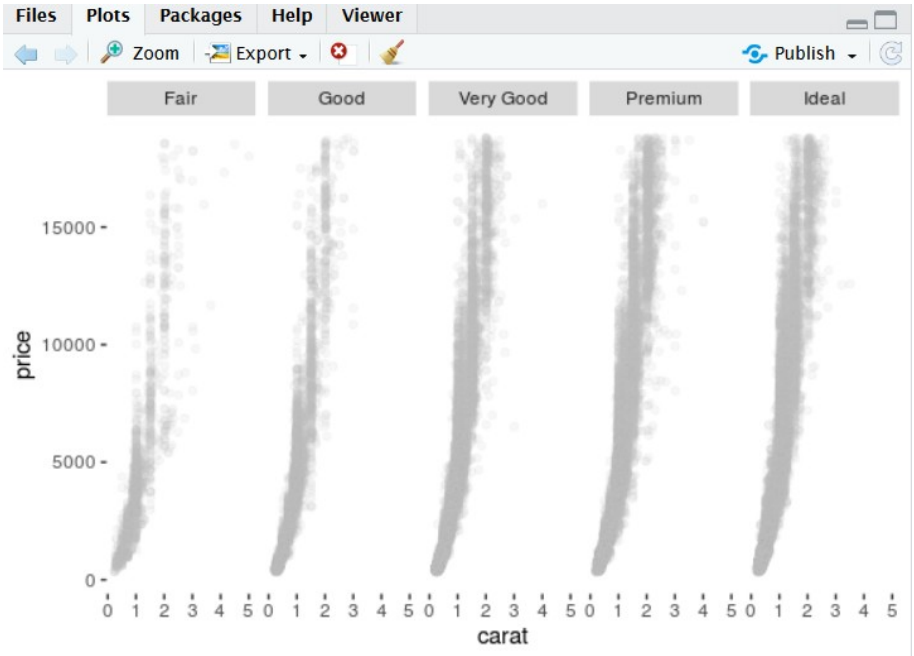
3. [10%] Add a rug plot to the figure from question 1.

Provide code below:

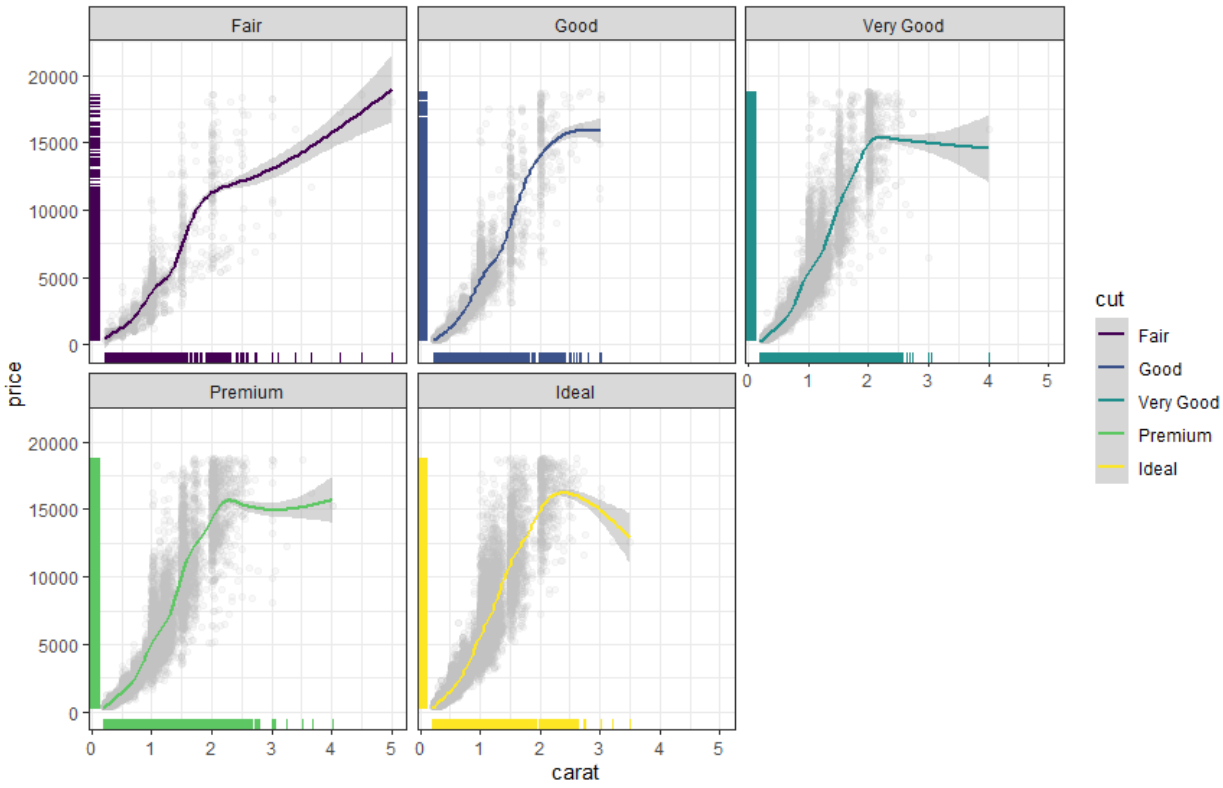
```
> q1g <- ggplot(data = diamonds) +
+     geom_jitter(mapping = aes(x = carat, y = price), color = "gray", alpha = 0.1) +
+     facet_wrap(~ cut, ncol = 5, nrow = 2) +
+     theme(panel.background = element_rect(fill = "white", color = "white"))
> q1g + geom_rug()
>
```

//q1g was the original plot

Provide figure below:



4. [20%] Provide the code to generate the following plot, where the bands around the fitted colored lines are 95% confidence intervals.



Provide code below:

// Had to add layer ontop otherwise the gray points in the background would not have been displayed.

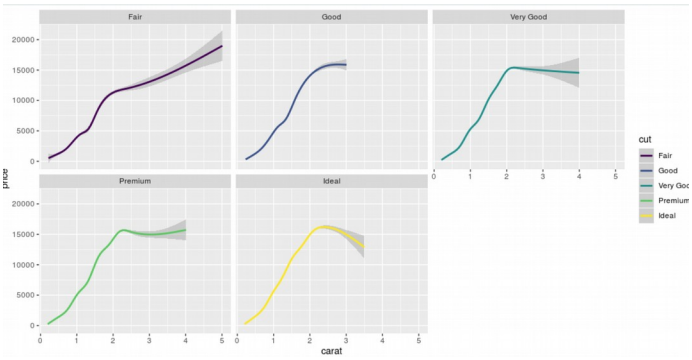
```
> q4g2 <- q1g
> q4g2 + geom_smooth(mapping = aes(x = carat, y = price, color = cut)) +
+ facet_wrap(~ cut, ncol = 3, nrow = 2)
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
>
```

5. [5%] From your solution to question 4, does there appear to be a relationship between diamond price and diamond weight? If there is a relationship, then what is it?

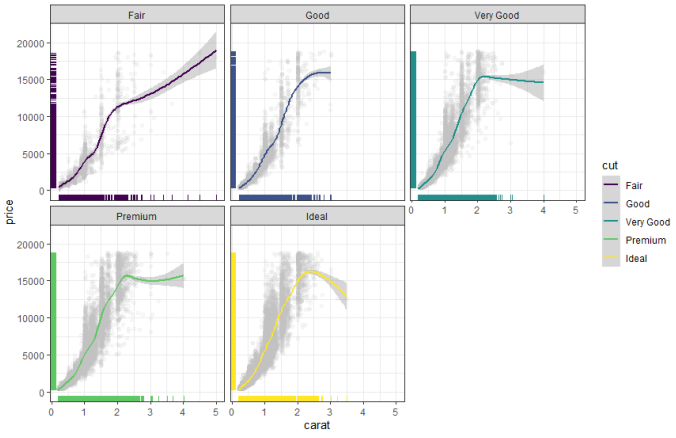
Yes, the major trend is the higher the weight, the higher the price. But it seems that this only holds true all the time for fair cuts as the degree of this correlation wavers after a certain weight (~2) for the other types of cuts.

6. [5%] In the plot from question 4, why are the confidence intervals much narrower for diamonds weighing less than three carats than for diamonds weighing greater than three carats?

It is narrower because the model is more 'confident' that the price will fall in that range given the weight, and this range is smaller/narrower because of our data base. Our database consisted mostly of objects (diamonds) weighing less than 3 carats as can be seen with the 2nd figure below, thus it has more material to use from that range and thus its predictions would be more confident hence the narrower confidence interval. The confidence interval gets wider as the weight passes 3 carats as the database has little information regarding those weights.



<-confidence interval only



<- confidence + points

7. [5%] What is a violin plot, and what geometric layer (function) in **`ggplot2`** can be used to generate one?

A violin plot is basically a box plot with the important added benefit of displaying kernel probability density of the data, this reveals the entire distribution of the data and is integral when dealing with multimodal data (distribution with more than one peak).

Can be done so using the geom_violin() function which operates the same as the other geometric layer functions (same parameters).

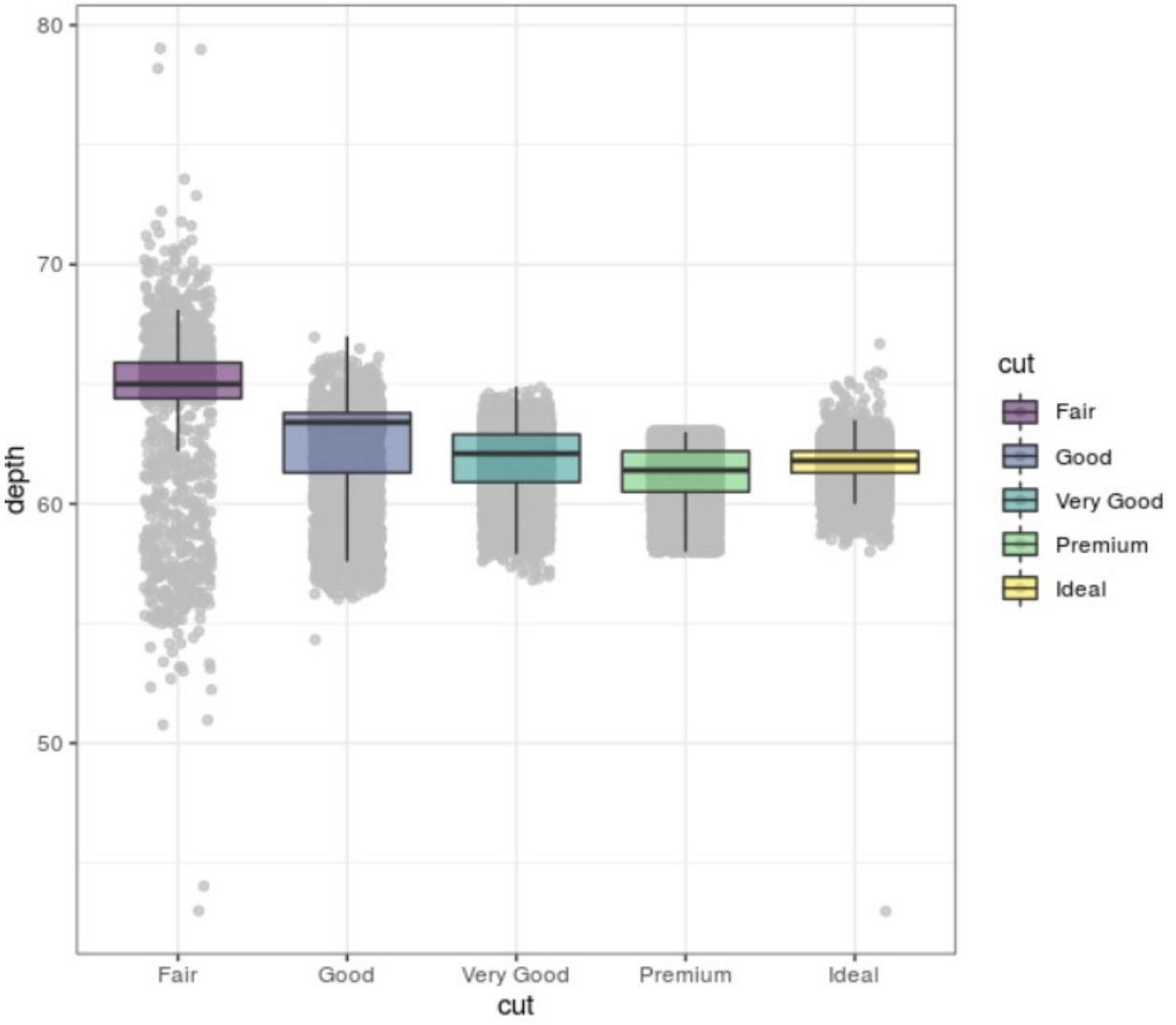
8. [5%] How are violin plots different from box plots?

They display the kernel probability density of the data as well as all the features of a box plot.

9. [10%] Box plots can be flexible, in that it is possible add and remove features. One feature that we can remove is the inclusion of outliers, such that only the box and whiskers are plotted. This can be done by assigning the argument **`outlier.shape`** to **`NA`** (a not assigned value). With this in mind, make a plot with the following code.

```
ggplot(data = diamonds,
       mapping = aes(x = cut, y = depth, fill = cut)) +
     geom_jitter(width = 0.2, color = "gray", alpha = 0.75) +
     geom_boxplot(alpha = 0.5, outlier.shape = NA) +
     theme_bw()
```

Provide figure below:



Based on this figure, what is the purpose of the following code, since the *x*-axis is categorical?

```
geom_jitter(width = 0.2, color = "gray", alpha = 0.75)
```

Provide your answer below:

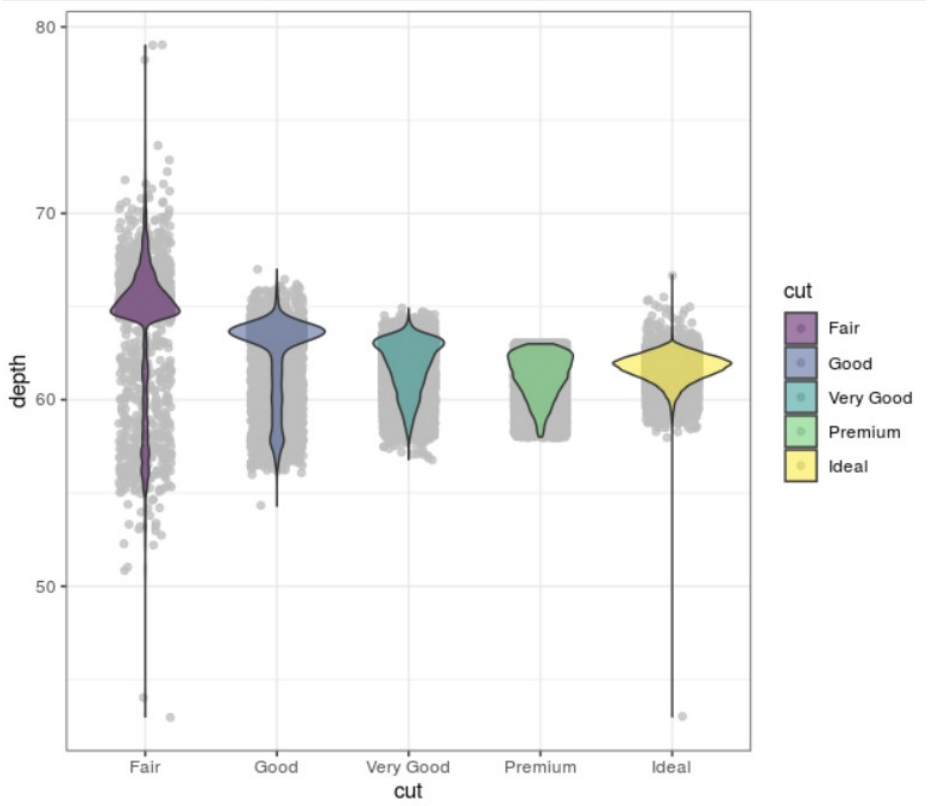
The purpose is to illustrate the data points that produced the box plot in order to give a better understanding of the data. The width is the space surrounding the box of which the data points will be within, the color is the color of the data points, alpha is the level of transparency.

10. [10%] Replace the box plots with violin plots in your figure from question 9, giving them the same level of transparency as the box plots.

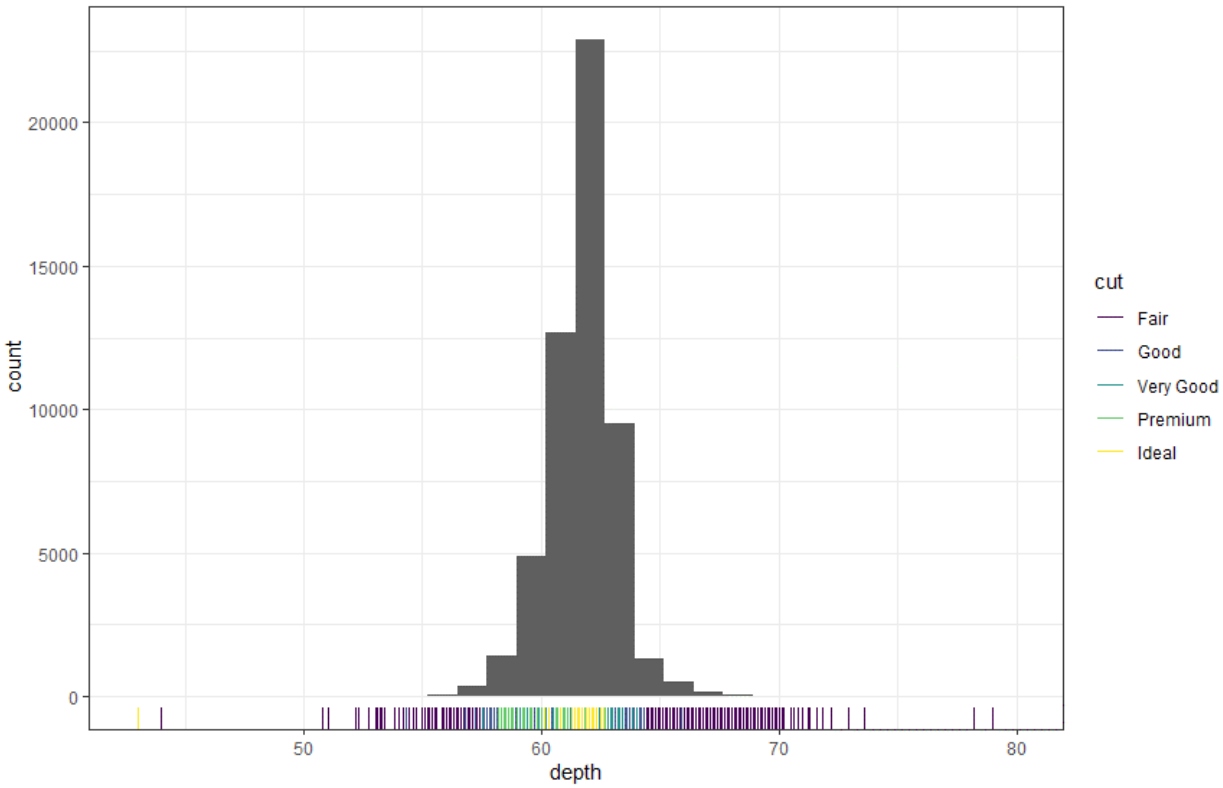
Provide code below:

```
> ggplot(data = diamonds,
+        mapping = aes(x = cut, y = depth, fill = cut)) +
+     geom_jitter(width = 0.2, color = "gray", alpha = 0.75) +
+     geom_violin(alpha = 0.5, outlier.shape = NA) +
+     theme_bw()
Warning: Ignoring unknown parameters: outlier.shape
```

Provide figure below:



11. [15%] Provide the code to generate the following plot.



Provide code below:

I attempted to use group_by() to get the counts but I kept receiving this:

```
> by_cut <- group_by(cut)
Error in UseMethod("group_by_") :
  no applicable method for 'group_by_' applied to an object of class "
function"
```

So then I tried to transform to create a new table with an 11th added column being the count and then to plot it but received this:

```
> newD <- transform(diamonds, count=ave(as.numeric(cut), FUN=length
> ggplot(data = newD, mapping = aes(x = depth, y = count, color = c
t)) +
+     geom_histogram()
Error: stat_bin() must not be used with a y aesthetic.
>
```