Applied Data Mining: Homework #5

Due on Oct 22 2017

Instructor: Hasan Kurban

Naimesh Chaudhari

October 22, 2017

In this homework, you will work on some of basic tasks of data mining: data preprocessing, exploratory data analysis and predictive model construction. Click here to download Auto MPG Data Set (auto-mpg.data-original). Load the data set into R and answer the following questions.

Problem 1

Discussion of Data [20 pt]

Briefly describe this data set–what is its purpose? How should it be used? What are the kinds of data it's using?
This dataset captures various properties of a car such as a year, origin, acceleration, cylinders, weight, mpg, etc. We can use this information to see if we can predict the mpg of a car based on a few of these parameters. It is using numeric or categorical variables.

Problem 2

Data Visualization and Summarization [50 pt]

1. Observe the statistical properties of the data using "summary" function and briefly discuss each variable.

Discussion of Variables

MPG seems to be a continuous variable that ranges between 9 and 46. There are six instances where there are missing values for this variable. The mean and median seem to be within the same range.

Cylinders seem to be a categorical variable from range (3 to 8). Most of the data is between 4,6,and 8 cylinders.

Displacement, horsepower, weight , acceleration all are continious with fairly close means and medians and a diverse range.

Model Year and Origin are categorical variables.

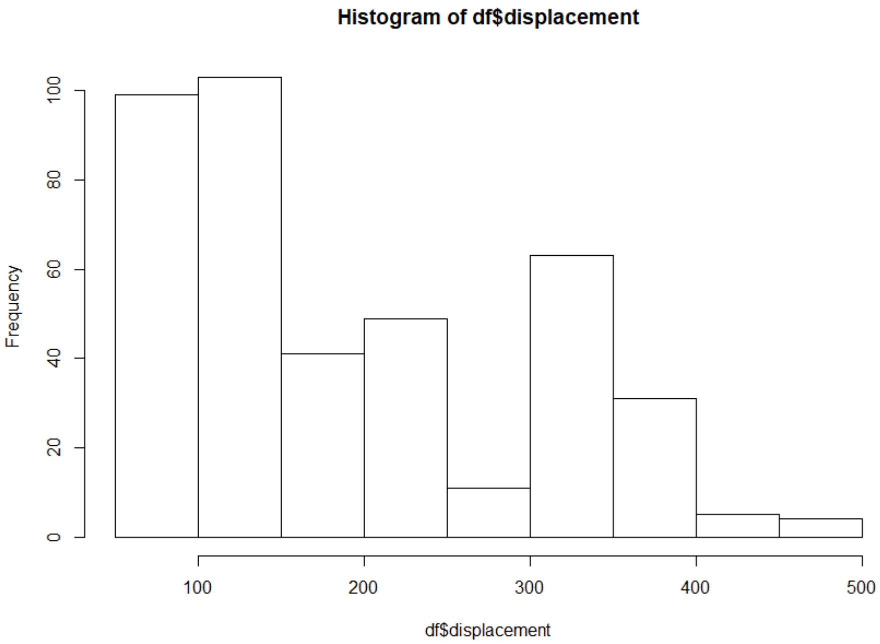
2. Create a histogram and Q-Q plot of variable "displacement". Using the plots, explain whether or not variable "displacement" follows a normal distribution. Discuss the plots (Use ggplot2 and car packages to make histogram and Q-Q plot figures).

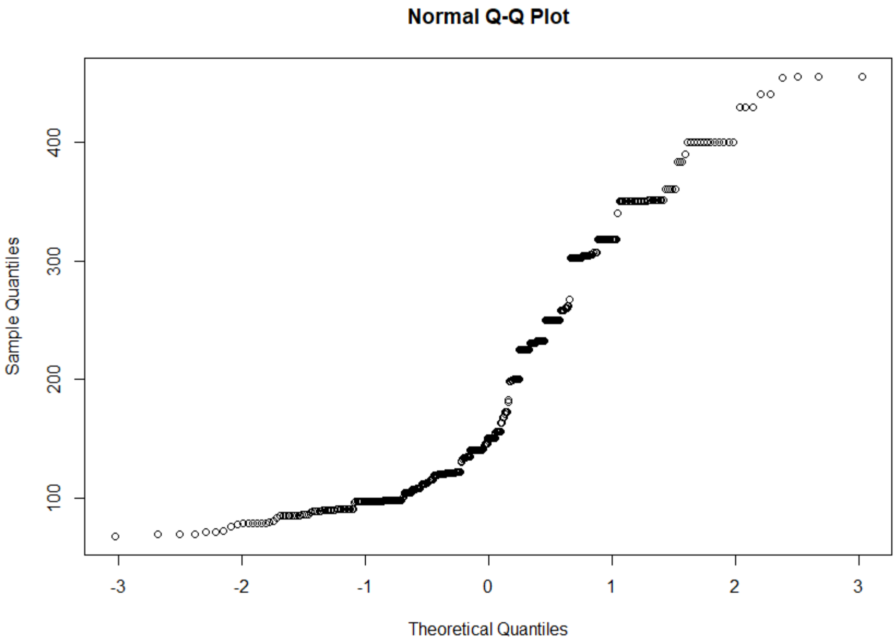
R Code

Listing 1: Sample R Script With Highlighting

```r
df <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/
    auto-mpg/auto-mpg.data-original")
columns <- c('mpg','cylinders','displacement','horsepower','weight','
    acceleration','modelyear','origin','car name')
colnames(df) <- columns
summary(df)
hist(df$displacement)
qqnorm(df$displacement) #Not straight, seems to be skewed
```

Histogram and Q-Q plot Figures





Discussion of Plots

Both the histogram and Q-Q plot suggest that this data is not normal. Generally for the data to be normal the histogram would have close to a bell curve, and the Q-Q plot would look like a straight line.

3. Box plots provide some key properties of a continuous variable. Create a box plot of variable “weight”

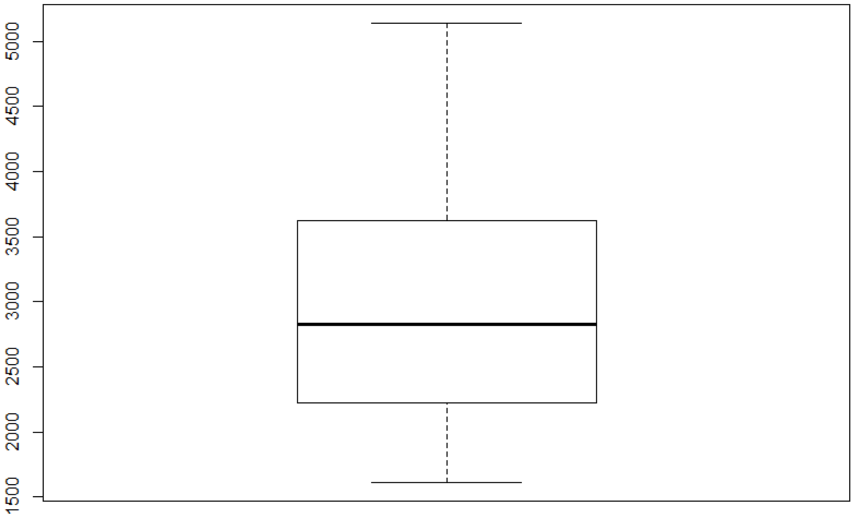
and discuss the the distribution of values, i.e., skewed. Discuss variable “weight” using the box plot (Use ggplot2 package).

R Code

Listing 2: Sample R Script With Highlighting

```
boxplot(df$weight)
```

Box plot Figure



Discussion of Plots

The data for weight primairly resided between 2300 and 3500, the mean in between this data seems to be slightly left skeded.

4. Make a set of box plots (conditional box plot) to observe how the distribution of “mpg” variable looks with the “origin” variable. Do not forget to convert the “origin” variable to a factor variable. In additional to the box plot, create a violin plot of “mpg” variable against “origin” variable. Discuss the plots.

```
data[,8] < - as.factor(data[,8])
```

Similarly, you should also convert other categorical variables into factor variables (Variables 2 and 7).

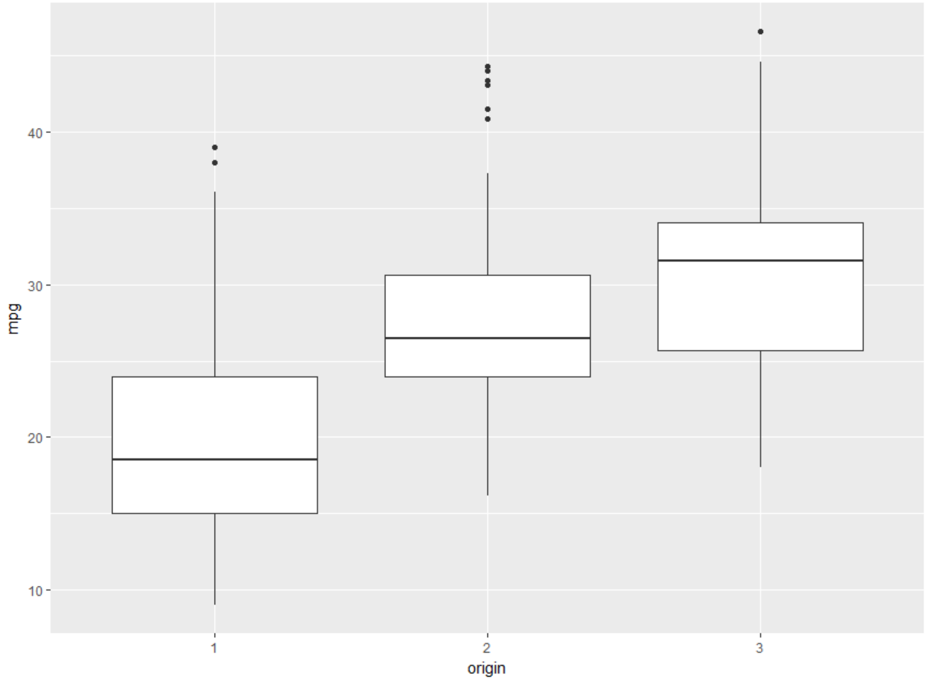
R Code

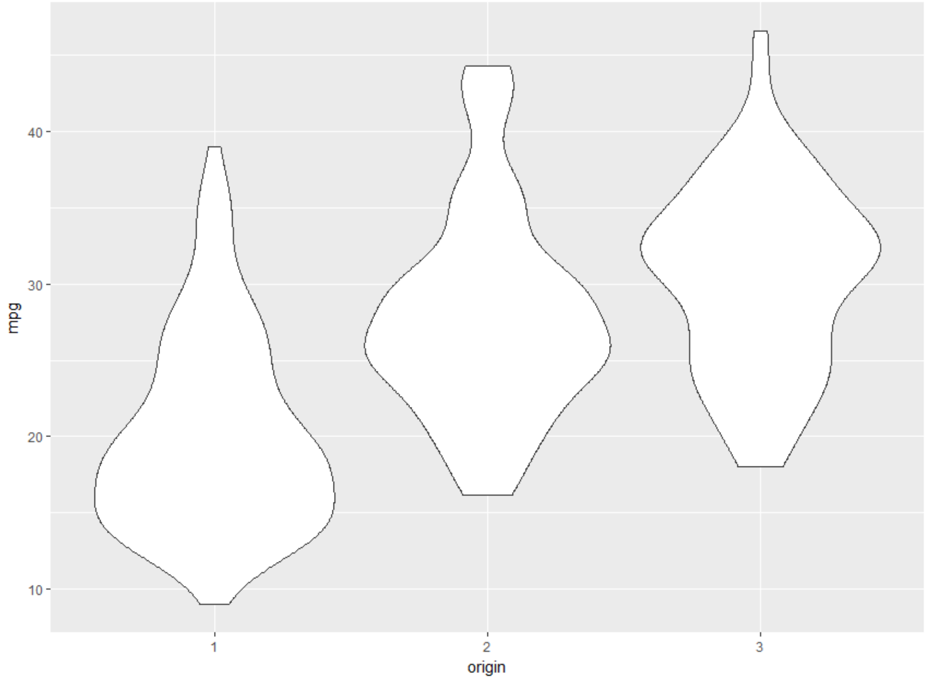
Listing 3: Sample R Script With Highlighting

```r
df$origin <- as.factor(df$origin)
df$cylinders <- as.factor(df$cylinders)
df$modelyear <- as.factor(df$modelyear)
boxplot(df$mpg, df$origin)
ggplot(df, aes(x = origin, y= mpg)) + geom_boxplot()
ggplot(df, aes(x=origin, y=mpg)) + geom_violin()
```

Problem 2 continued on next page. . .

Figures





Discussion of Plots

In both graphs we can see as origin gets larger so does the mean and median of the mpg.

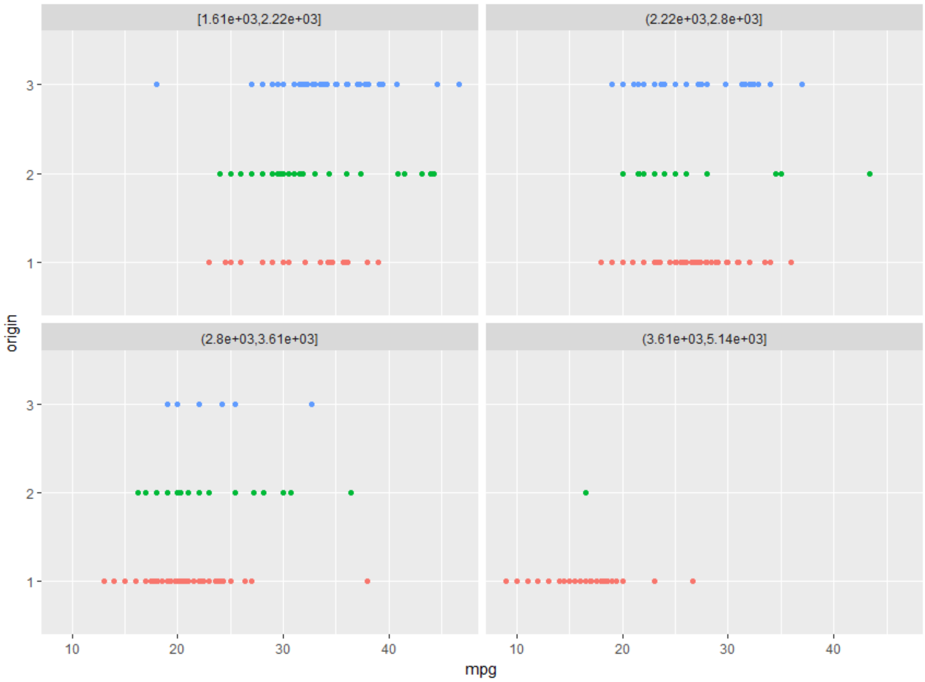
5. Discretize "weight" variable as shown in the textbook, page 203. Observe the behavior of variable "mpg" conditioned by "weight" and "origin" variables over a conditional plot. Discuss the plot.

R Code

Listing 4: Sample R Script With Highlighting

```
disgraph <- filter(df,!is.na(mpg)) %>% mutate(weight=cut(weight, quantile(
    weight,c(0,0.25,.5,.75,1)), include.lowest=TRUE))
ggplot(disgraph,aes(x=mpg,y=origin, color=origin)) + geom_point() +  facet_
    wrap(~ weight) + guides(color=FALSE)
```

Figure



Discussion of Plot

This graph further supports heigher weight cards have a larger mpg requirement.

Problem 3

Handling Missing Values [50 pt]

In this question, you will replace the missing data using different techniques.

1. How many entries are in the data set? There are 406 entries

Listing 5: Sample R Script With Highlighting

```
nrow(df)    #406
```

2. How many unknown or missing data are in the data set? There are 14 missing values

Listing 6: Sample R Script With Highlighting

```
table(is.na(df))    #14
```

3. Use “manyNAs” function to report the rows in the data that have certain number of unknowns ($nORp = 0.1$). Rows 11,12,13,14,15,18,39,40,134,338,344,362,368,383 contain NA’s

Listing 7: Sample R Script With Highlighting

```
manyNAs(df, nORp = 0.1)
```

4. Filling in the Unknowns with the Most Frequent Values:

   (a) Replace missing values of variable “horsepower” and “mpg” variables using “centralImputation” function. Explain how this function fills in the unknown values.

Listing 8: Sample R Script With Highlighting

```
dfclean <- centralImputation(df)
```

Discussion

The centralImputation fills in all unknowns in a data set using a statis centrality. Default for categorical variables: the mode, for numeric variables: median

5. Filling in the Unknown Values by Exploring Correlations:

   (a) First, reload the original data that contains the missing data . Observe the correlations among the continuous variables (Variables 1,3,4,5,6) and report the correlation matrix (use symnum function). Discuss the results.

Listing 9: Sample R Script With Highlighting

```
symnum(cor(df[,c(1,3,4,5,6)],use="complete.obs"))
```

Discussion and Correlation Matrix

```
mpg          1
displacement + 1
horsepower   , + 1
weight       + * + 1
acceleration . . , . 1
attr(,"legend")
[1] 0   0.3 . 0.6 , 0.8 + 0.9 * 0.95 B 1

                    mpg displacement horsepower     weight acceleration
mpg           1.0000000   -0.7920771 -0.7609257 -0.8239854    0.4104485
displacement -0.7920771    1.0000000  0.8967035  0.9324747   -0.5579836
horsepower   -0.7609257    0.8967035  1.0000000  0.8643686   -0.6944147
weight       -0.8239854    0.9324747  0.8643686  1.0000000   -0.4300858
acceleration  0.4104485   -0.5579836 -0.6944147 -0.4300858    1.0000000
```

Displacement weight and Displacement horsepwer have a good posetive correleation.

(b) What variable has the highest linear correlation with "horsepower" variable. Fit a linear model to fill in unknown values of "horsepower" via this variable. Report the new values of unknown data.

Listing 10: Sample R Script With Highlighting

```
dfcor <- df[-manyNAs(df, nORp = 0.1), ]
lm(horsepower ~ displacement, data = dfcor)    #40.31 & 0.33
fillhp <- function(hp) ifelse(is.na(hp),NA,40.31 + 0.33 * hp)
df[is.na(df$horsepower), "horsepower"] <- sapply(df[is.na(df$horsepower),
    "displacement"], fillhp)
```

Results

Displacement has the highest correlation with horsepower.

```
 displacement horsepower
11          133     115.00
12          350     165.00
13          351     153.00
14          383     175.00
15          360     175.00
18          302     140.00
39           98      72.65
40           97      48.00
134         200     106.31
338          85      68.36
344         140      86.51
362         100      73.31
368         121     110.00
383         151      90.14
```

6. Filling in the Unknown Values by Exploring Similarities between Cases:

(a) First, reload the original data that contains the missing data. Replace missing values of the data set using "knnImputation()" function. Explain how this function replaces the missing values (Pick, $k = 5$, $meth$ = "median"). The clean data obtained after using "knnImputation()" function will be the data set that must be used to answer the rest of the problems.

Listing 11: Sample R Script With Highlighting

```
df<- knnImputation(df, k = 5, meth = "median")
```

Discussion

```
  mpg cylinders displacement horsepower weight acceleration modelyear origin
11  21.0         4          133        115   3090          17.5        70      2
12  14.0         8          350        165   4142          11.5        70      1
13  14.0         8          351        153   4034          11.0        70      1
```

```
14  14.0         8          383         175   4166         10.5         70       1
15  14.0         8          360         175   3850         11.0         70       1
18  14.0         8          302         140   3353          8.0         70       1
39  25.0         4           98          83   2046         19.0         71       1
40  26.0         4           97          48   1978         20.0         71       2
134 21.0         6          200          88   2875         17.0         74       1
338 40.9         4           85          62   1835         17.3         80       2
344 23.6         4          140          88   2905         14.3         80       1
362 34.5         4          100          74   2320         15.8         81       2
368 35.0         4          121         110   2800         15.4         81       2
383 23.0         4          151          88   3035         20.5         82       1
```

It used k number of nerest neighbord to fill in the median value.

Problem 4

Use the clean data set from question 3.6.*a* to answer problem 4. Remove the last variable, car name, from the clean data before answering this question.

Obtaining Predictive Models (Linear Regression and Regression Trees) [50 pt]

1. Simple linear regression:

   (a) Obtain a linear model using variable “weight” to predict “mpg” variable. Use summary() function to explain the model and discuss output of summary() function, i.e., what does “Adjusted R-squared” show?, what are the coefficients?, etc. Is this a good model?

R Code

Listing 12: Sample R Script With Highlighting

```r
df <- df[,-9]
lms <- lm(mpg ~ weight, data = df)
summary(lms)
```

Discussion of Output of Summary() Function and Results

```
R^2 shows th proportion of variance in the data that is explained
by the model closer to 1 the better.
Coefficients:
1- Estimate: coefficient value for each variable
2- Std. Error: an estimate of the variability of these coefficients.
3- t value and Pr(> |t|): to statistically check the importance
of each coefficient
```

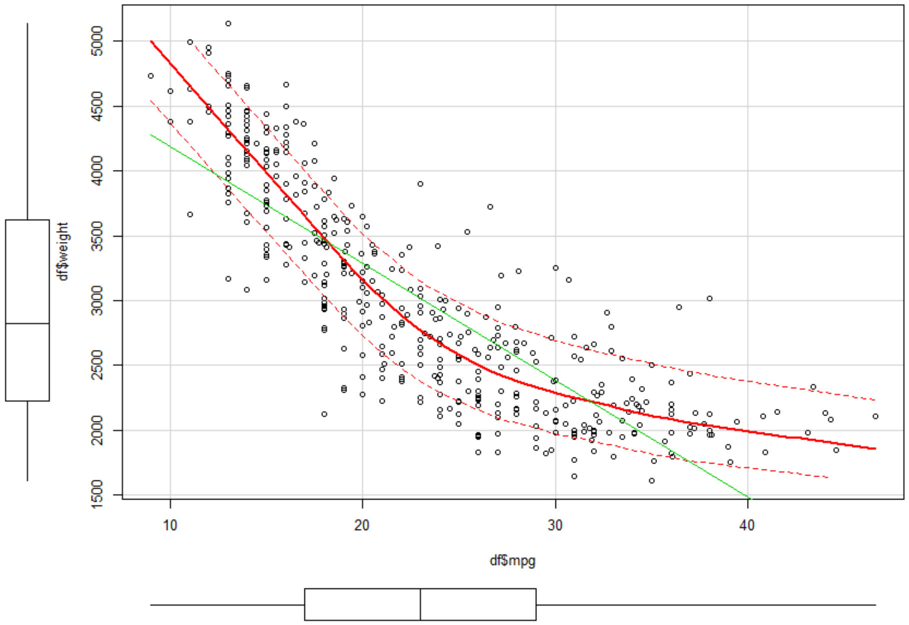
   (b) Plot the data points in a scatter plot (mpg vs. weight) and show the model from problem 4.*a* on this scatter plot? Is the correlation between the variables positive or negative? The correlation is definately negative.

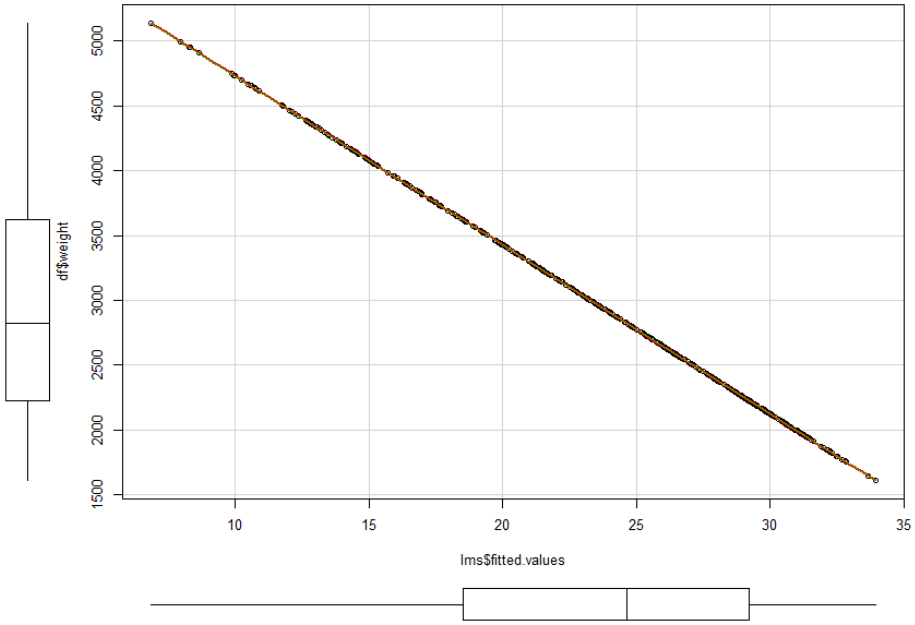
R Code

Listing 13: Sample R Script With Highlighting

```
scatterplot(df$mpg, df$weight)
scatterplot(lms$fitted.values, df$weight)
```

Figure





2. Multivariate linear regression:

(a) Train a linear model that predicts variable "mpg" using all other variables (variables 2-8). Use summary() function to explain the model and discuss output of summary() function.

R Code

Listing 14: Sample R Script With Highlighting

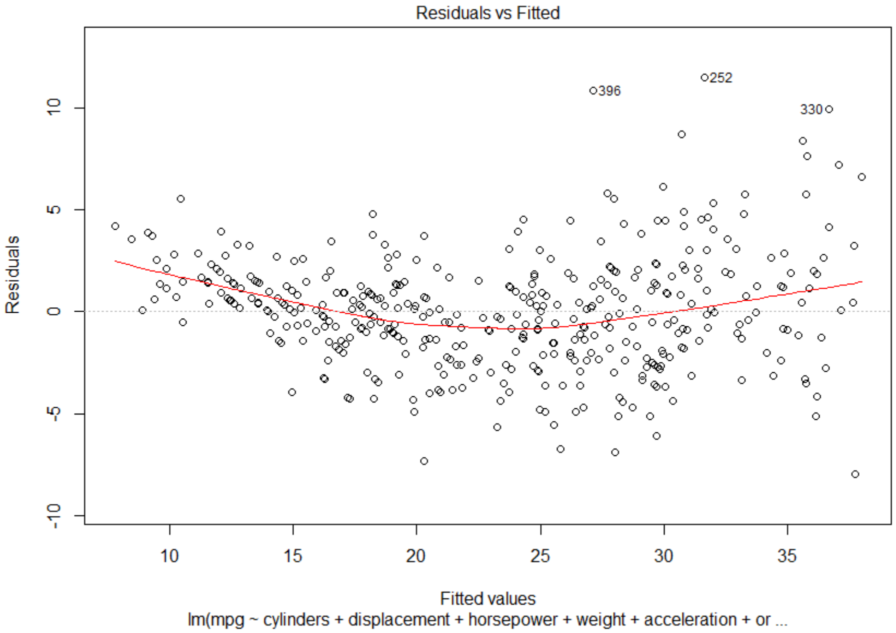
```r

lmm <- lm(mpg ~
            cylinders
          + displacement
          + horsepower
          + weight
          + acceleration
          + origin
          + modelyear , data= df)
summary(lmm)
plot(lmm)
```

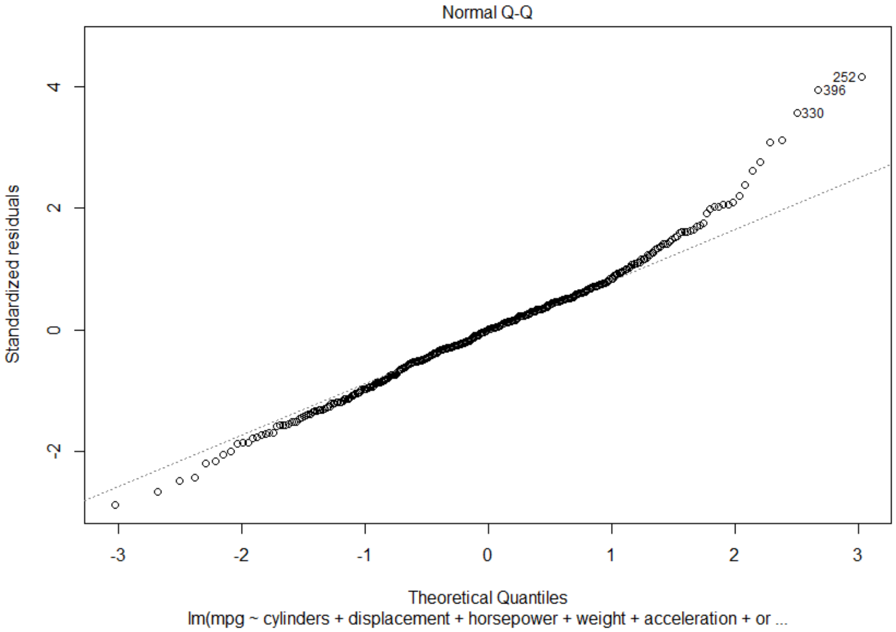
Discussion of Output of Summary() Function and Results

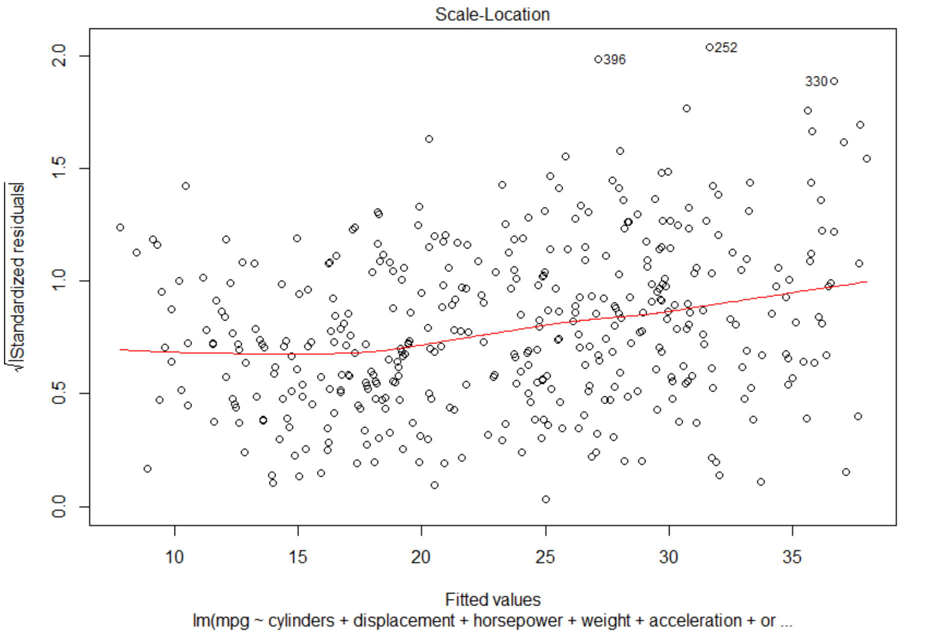
```
 This model seems to have a much better fit. An adjusted R^2 of  0.8666
```

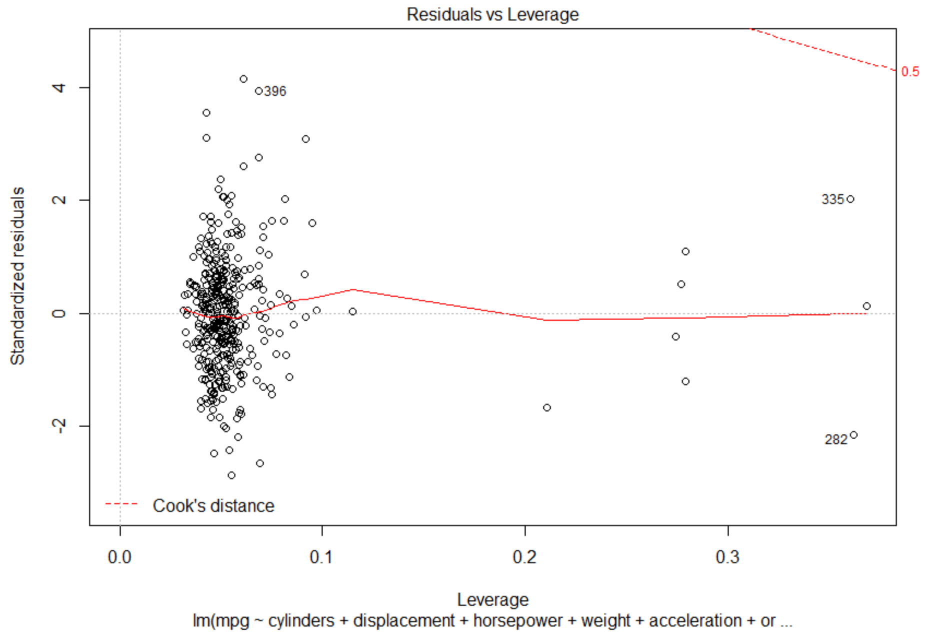
(b) Use plot() function to understand performance of the model. Insert the four figures below and discuss the plots below that.

Figure


Residuals vs Fitted
Residuals
10
5
0
-5
-10
396
252
330
10
15
20
25
30
35
Fitted values
lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + or ...


Normal Q-Q
Standardized residuals
4
2
0
-2
252
396
330
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + or ...





Discussion of the Plots

The data seems to fit fairly well. The qq plot suggests normal data, residual does not seem to show a pattern.

(c) Find the variable that least contributes to the reduction of the fitting error of the model (use anova()). Then, use update() function to remove that variable from the model. How much of the variance is explained by this new model? Answer here . . .

R Code

Listing 15: Sample R Script With Highlighting

```
anova(lmm)
lmm2 <- update(lmm, .~. -acceleration)
summary(lmm2)
```

(d) Use step() function to optimize the model obtained in question 4.*a*. and call this optimized model, “final.lm”

R Code

Listing 16: Sample R Script With Highlighting

```
final.lm <- step(lmm)
```

3. Regression Trees:

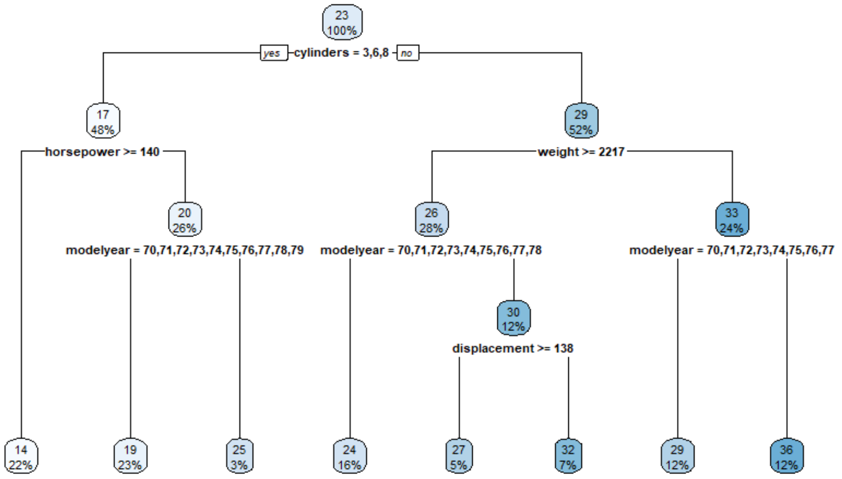
(a) Train a regression tree to predict “mpg” variable using all other variables (variables 2-8). Visualize the tree. Call this model, “final.tree”.

R Code

Listing 17: Sample R Script With Highlighting

```
final.tree <- rpart(mpg ~
                      cylinders
                    + displacement
                    + horsepower
                    + weight
                    + acceleration
                    + origin
                    + modelyear , data= df)
rpart.plot::rpart.plot(final.tree)
```

Regression Tree Figure



Problem 5

Model Evaluation and Selection [50 pt]

Use the clean data set from question 3.6.*a* to answer problem 5. Remove the last variable, car name, from the clean data before answering this question. predict() function takes a model and a test data and retrieves the corresponding model predictions. In this question, you will use predict() function to compare final.lm and final.tree. Answer the questions below:

1. Calculate mean absolute error (MAE) for final.lm and final.tree. Which one is better? LM performed better.

R Code

Listing 18: Sample R Script With Highlighting

```
final.lm.predict <- predict(final.lm, df)
final.tree.predict <- predict(final.tree, df)
mae.mpg.lm <- mean(abs(final.lm.predict - df[["mpg"]])) #2.147
mae.mpg.rt <- mean(abs(final.tree.predict - df[["mpg"]])) #2.229
```

2. Calculate mean squared error (MSE) for final.lm and final.tree. Which one is better? LM performed better.

Problem 5 continued on next page. . .

R Code

Listing 19: Sample R Script With Highlighting

```
mse.mpg.lm <- mean((final.lm.predict - df[["mpg"]])^2) #8.140
mse.mpg.rt <- mean((final.tree.predict - df[["mpg"]])^2) #9.055
```

3. Calculate normalized mean squared error (NMSE) for final.lm and final.tree. Which one is better? LM performed better.

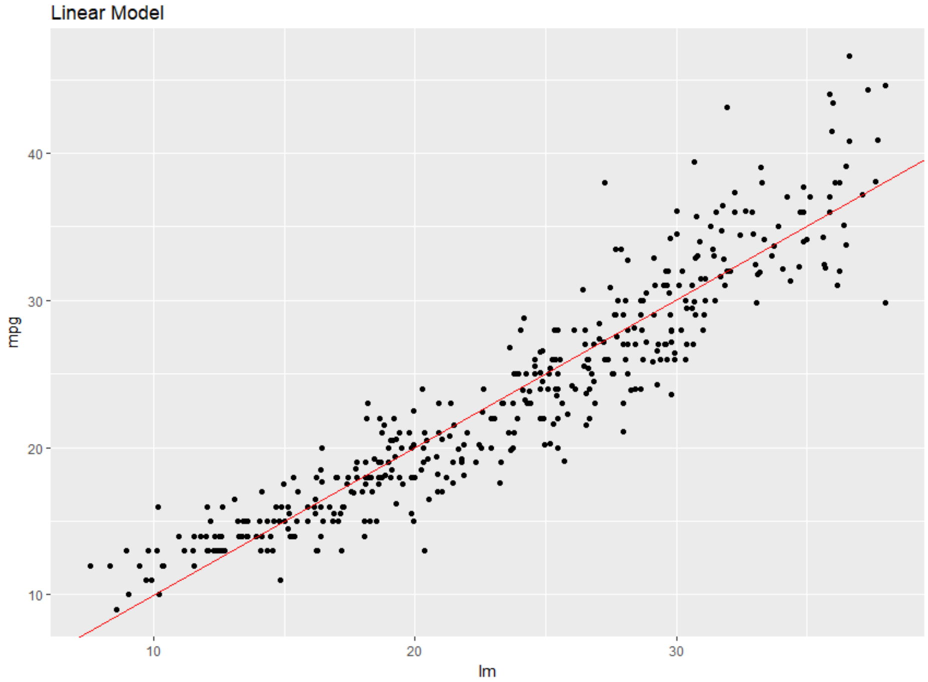
R Code

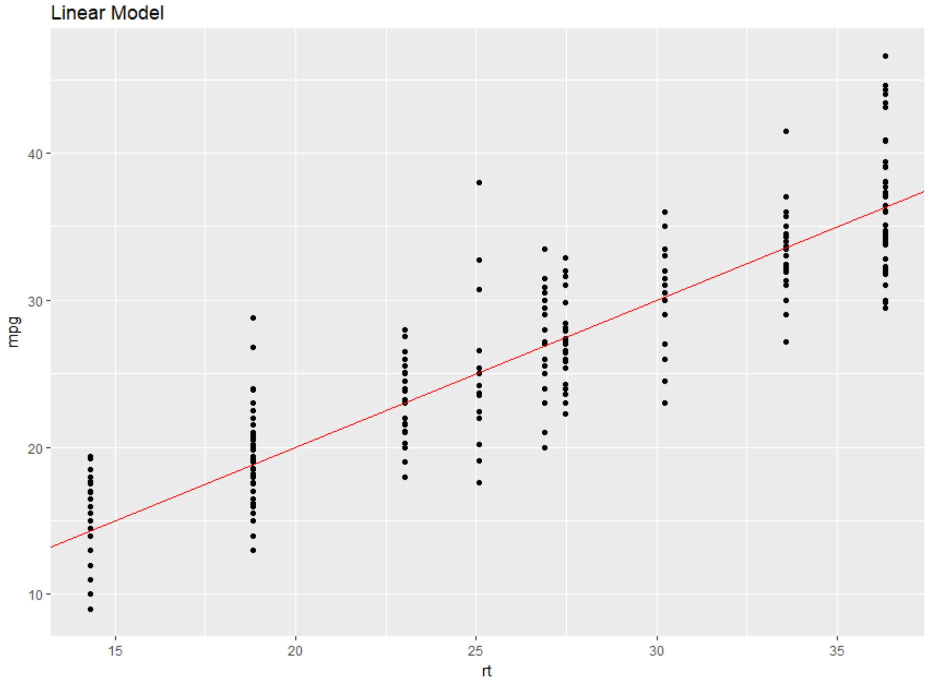
Listing 20: Sample R Script With Highlighting

```
(nmse.lm <- mean((final.lm.predict-df[['mpg']])^2)/  #Normalized MSE (NMSE)
  mean((mean(df[['mpg']])-df[['mpg']])^2))

(nmse.rt <- mean((final.tree.predict-df[['mpg']])^2)/  #Normalized MSE (NMSE)
    mean((mean(df[['mpg']])-df[['mpg']])^2))
```

4. Observe the errors for final.lm and final.tree via scatter plots. See figure 4.11, texbook, page 227. Discuss the plots, i.e., did the models perform well?

Error Scatter Plots





e.

R Code

Listing 21: Sample R Script With Highlighting

```
dg <- data.frame(lm=final.lm.predict,
                 rt=final.tree.predict,
                 mpg=df[["mpg"]])
ggplot(dg,aes(x=lm,y=mpg)) +
  geom_point() + geom_abline(slope=1,intercept=0,color="red") +
  ggtitle("Linear Model")

ggplot(dg,aes(x=rt,y=mpg)) +
  geom_point() + geom_abline(slope=1,intercept=0,color="red") +
  ggtitle("Linear Model")
```

Discussion of the Error Plots

Both plots seem to have preformed well. The error percent is not too far off from the modeled regressi

5. Cross Validation is a technique to measure performance of models over unseen data. In this question, use performanceEstimation() function in R to make use of cross validation technique and compare several models. Take the performanceEstimation() code from the textbook, page 228 and edit it for your data – Fit one linear and three regression tree models with the given parameters. Only tune the parameters of the "EstimationsTask as follows":

- EstimationsTask: metric= "mse" , method=CV(nReps=3,nFolds=5)

R Code

Listing 22: Sample R Script With Highlighting

```
res <- performanceEstimation(
  PredTask(mpg ~ ., dfclean[,1:8], "mpg"),
  c(Workflow(learner="lm",pre="knnImp",post="onlyPos"),
    workflowVariants(learner="rpartXse",learner.pars=list(se=c(0,0.5,1)))),
  EstimationTask(metrics="mse",method=CV(nReps=3,nFolds=5))
)

summary(res)
plot(res)
```

Answer the question below:

(a) Explain the code.

Discussion of performanceEstimation() Function

```
predictive task to compare: one linear model against 3 regression trees
 knnImp to replace missing values
onlyPos : to make negative predictions to 0. Since we know, that is not possible
```

(b) Compare the models? Which model performed best? The linear model performed the best.