# Applied Data Mining: Homework #5

Due on October 24 2017

*Instructor: Hasan Kurban*

**Keith Hickman**

October 24, 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?
The dataset appears to be several performance and characteristic metrics of several makes and models of cars, mostly older models. The variables are both continuous and discrete, of float, decimal, and string types. Use cases could include predicting mpg by the different variables using machine learning algorithms, or detecting linear correlation between 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

   The first two variables, MPG and Cylinders appear to be actually discrete, as they take on only a few numeric values. We can treat these as categories later on. The next four variables displacement, horsepower, weight, and acceleration are continuous. MPG, which is likely the target variable, is non-normally distributed and slightly right-skewed. Other variables are either normal, roughly normal, or non-normal and slightly skewed to the right. I decided to drop the last two variables, as they were mostly missing, they likely wouldn't add any predictive value, and if they added any analytic or descriptive value, we could add them back in after the fact. . . .

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

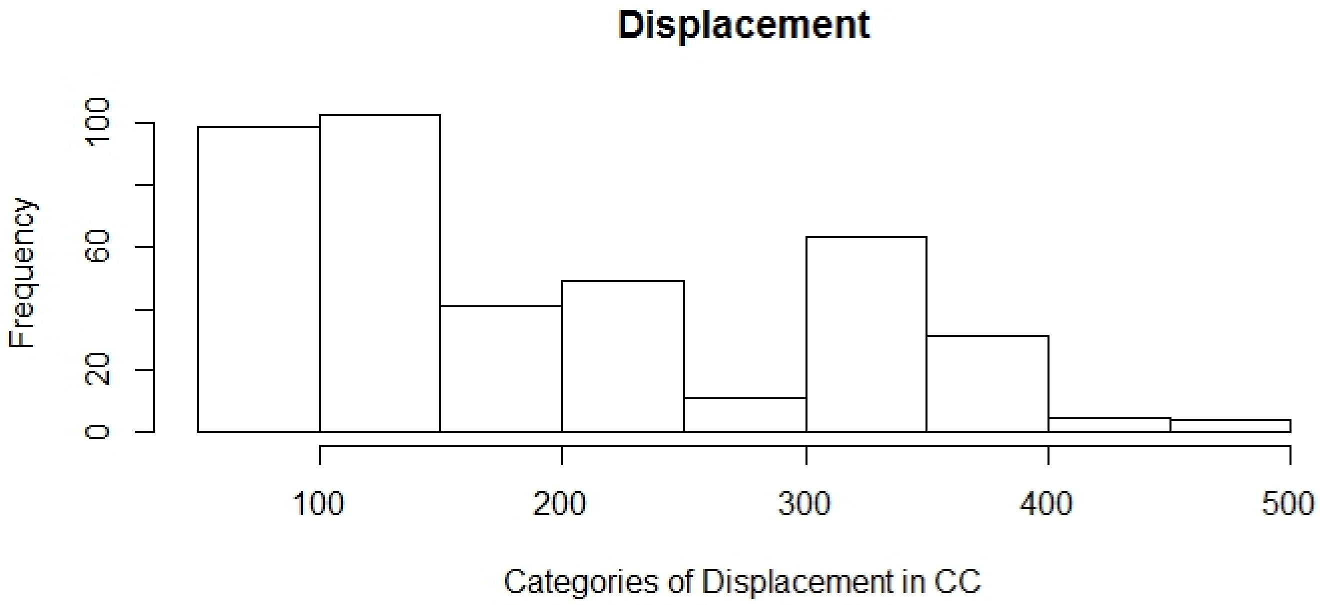
```
library(data.table)
library(car)
library(ggplot2)

??fread
cars <- fread("cars.csv", header=TRUE)
cars <- cars[,c(0:7,9)]
head(cars)
summary(cars)

```

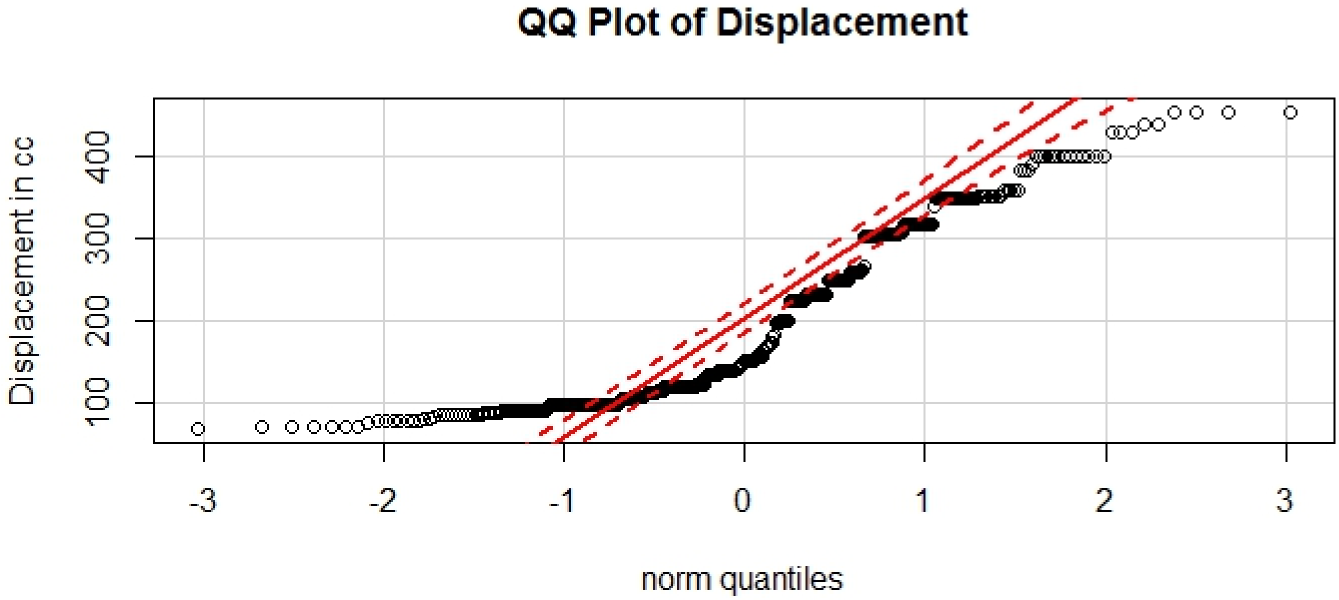
Problem 2 continued on next page. . .

```
##hist(cars$acceleration)
##hist(cars$horsepower)
##hist(cars$weight)
##hist(cars$mpg)
##hist(cars$cylinders)

hist(cars$displacement, main="Displacement", xlab="Categories of Displacement
    in CC")

qqPlot(cars$displacement, main="QQ Plot of Displacement", ylab="Displacement
    in cc")

boxplot(cars$weight)
```

Problem 2 continued on next page. . .

### Histogram and Q-Q plot Figures





### Discussion of Plots

The distribution of the displacement variable is clearly non-normal, as evidenced by both the QQ plot and the Histogram. According to the Histogram, the distribution is right-skewed, with most values falling in the 75-200 range, with another cluster of values at the 300 range. The QQ plot shows that there are quite a few values at the tail ends of the distribution.

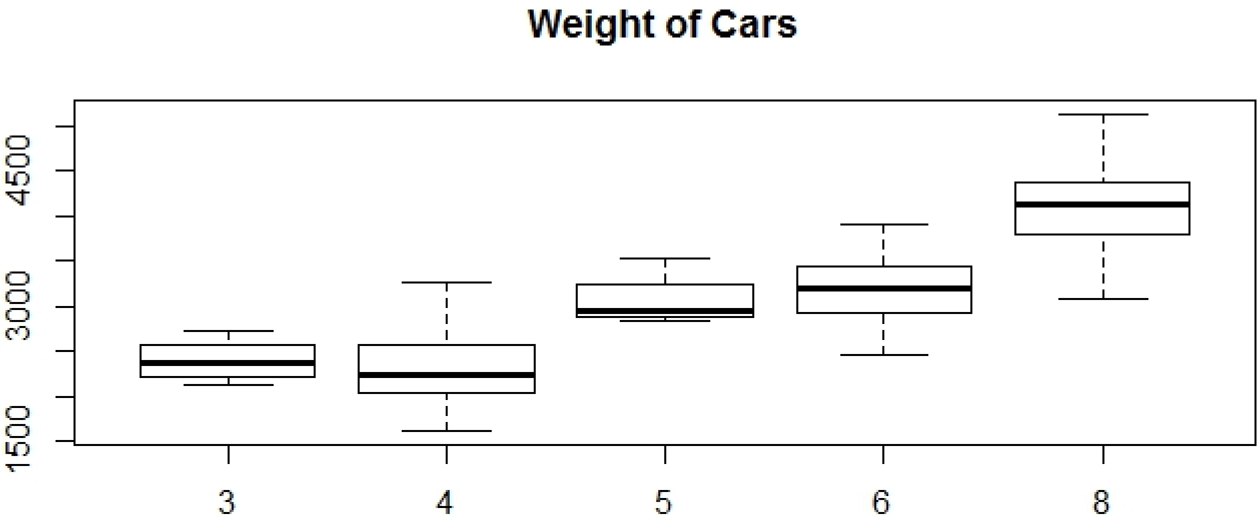
3. Box plots provide some key properties of a continuous variable. Create a box plot of variable "weight" and discuss the the distribution of values, i.e., skewed. Discuss variable "weight" using the box plot (Use ggplot2 package).

## R Code

Listing 2: Sample R Script With Highlighting

```
boxplot(weight ~ cylinders, cars, main="Weight of Cars")

##hist(cars$weight, main="Weight of Cars", xlab="Weight in lbs", ylab="Count")
##ggplot(cars,aes(x=origin, y=weight) + geom boxplot)
```

## Box plot Figure



## Discussion of Plots

The weights variable is a non-normal distribution and is right-skewed, indicating that most of the values fall into categories of cars that weigh less than 3500 pounds. Additionally, there don't appear to be any outliers. An examination of the boxplots comparing the weight variable across (the randomly-selected variable) the cylinder variable indicates that the cars which are 4, 6, or 8 cylinders are normally distributed, while cars that have either 3 or 5 cylinders are right skewed and non-normal. . . .

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])
```

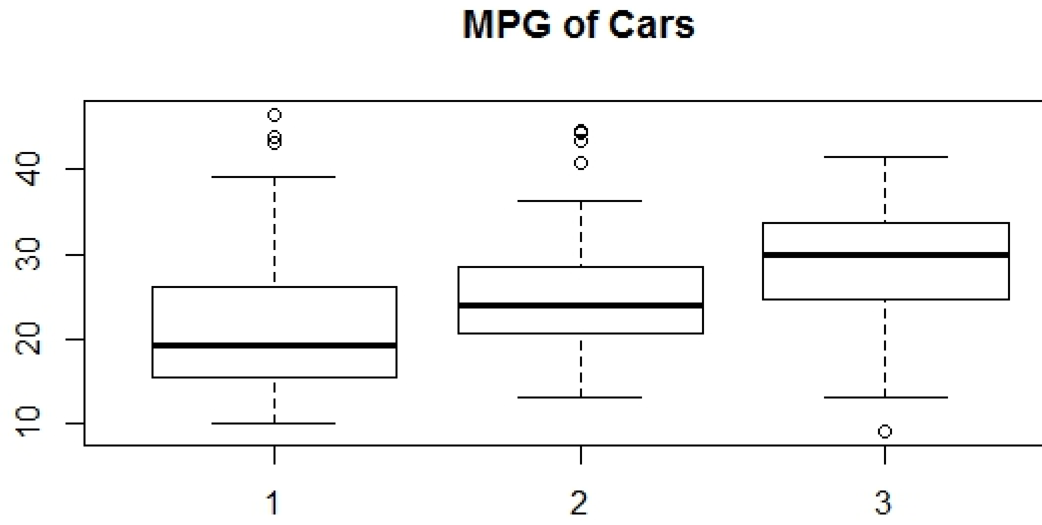
Similarly, you should also convert other categorical variables into factor variables (Variables 2 and 7).

## R Code

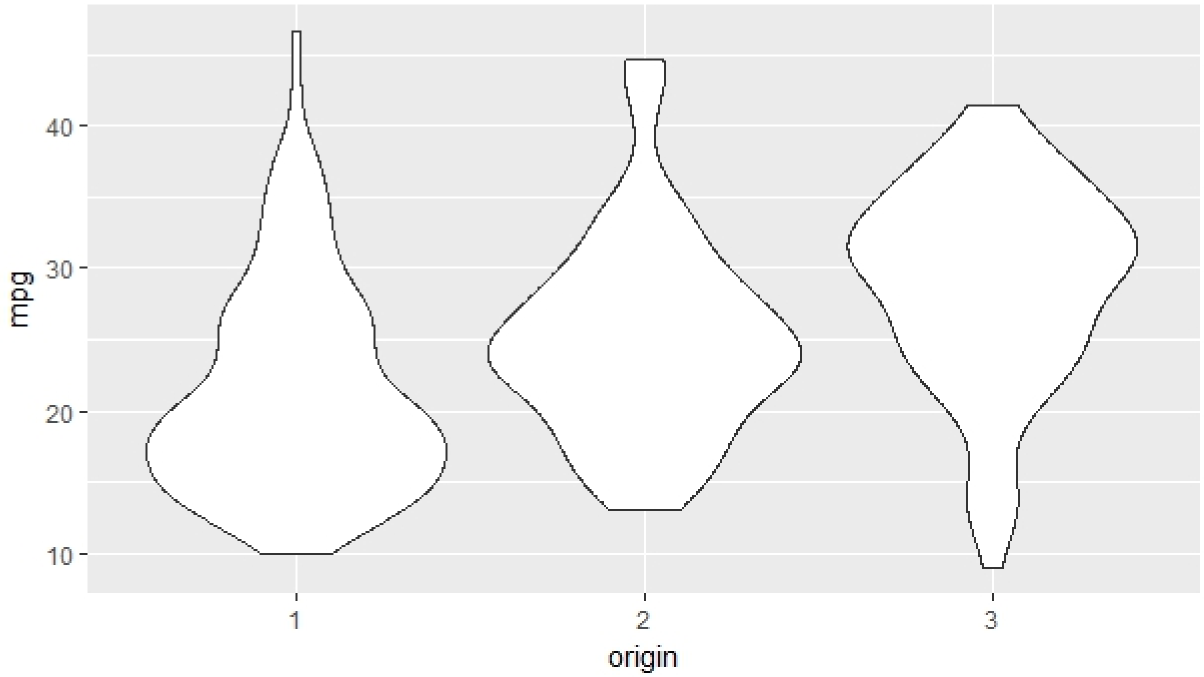
Listing 3: Sample R Script With Highlighting

```
head(cars)
cars$origin <- as.factor(unlist(cars$origin))
cars$cylinders <- as.factor(unlist(cars$cylinders))
cars$modelyear <- as.factor(unlist(cars$modelyear))
cars$newweight <- cut(cars$weight,5,c("very light", "light", "medium", "heavy", "very heavy"))

boxplot(mpg ~ origin, cars, main="MPG of Cars")

##hist(cars$weight, main="Weight of Cars", xlab="Weight in lbs", ylab="Count")
ggplot(cars,aes(x=origin, y=mpg)) + geom_violin()
```

### Figures



Place the figures here.



### Discussion of Plots

The boxplot and violin plots illustrate a likely correlation between origin and mpg - cars made in the US tend to have an mpg beween 15-25 with a mean slightly below 20. Cars made in Europe (2) are clustered at 25-30, and cars made in Japan tend to have the highest mpg ratings at 30+. Both plots provide similar information about distribution, however the boxplot also indicates that the distribution of US-made is more spread out, and includes some outliers which we need to investigate.Surprisingly, two of the top four highest-rated mpg cars are US-made. . . .

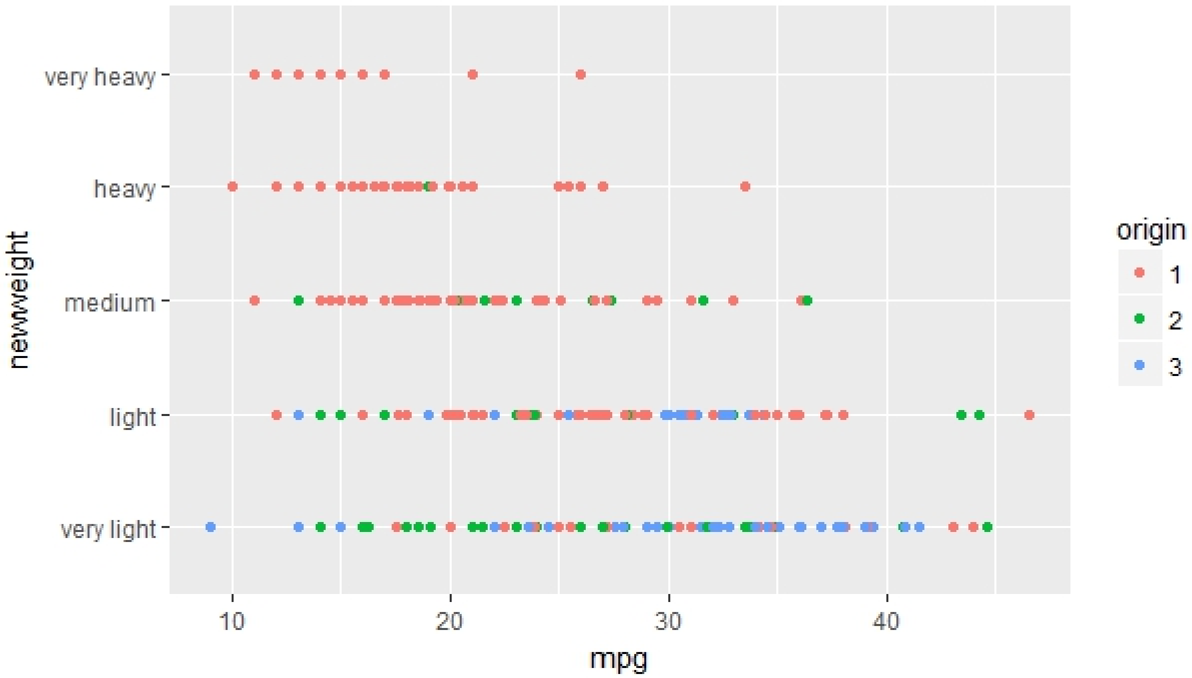
5. Discretize “weight” variable as shown in the textbook, page 203. Observe the behavior of variable ”mpg” conditioned by “weight” and “origin” variables over a conditional plot. Discuss the plot.

## R Code

Listing 4: Sample R Script With Highlighting

```
install.packages("forcats")
library(forcats)
library(dplyr)

cars2graph <- filter(cars,!is.na(weight)) %>%
  mutate(weight2=cut(weight, quantile(weight,c(0,.25,.5,.75,1))))
ggplot(cars,aes(x=mpg, y=newweight, color=origin)) + geom_point()
```

## Figure



## Discussion of Plot

Initially, I attempted to discretize the variables with 5 breakpoints, and got uneven bins. Though there is some information contained in this type of distribution, that unevenness is not optimal for determining correlation or other tasks. In this plot, there is a very clear relationship between mpg and weight, as well as between origin and those two variables. Though there is a clear relationship, there are quite a few observations that fall outside of our relationship that I would address by adjusting the number of bins. . . .

# 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 observations of 11 variables. ...

Listing 5: Sample R Script With Highlighting

```
summary(cars)
```

2. How many unknown or missing data are in the data set? There are two variables with missing values: mpg and horsepower, with 8 and 6 NA values, respectively. ...

Listing 6: Sample R Script With Highlighting

```
summary(cars)
```

3. Use "manyNAs" function to report the rows in the data that have certain number of unknowns ($nORp = 0.1$). There aren't any rows with more than 10 percent NA values, but there are 15 rows with 9 percent NA values. ...

Listing 7: Sample R Script With Highlighting

```

carsCI <- centralImputation(cars)
## carsCI
cars[12]
??centralImputation
```

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

```

carsCI <- centralImputation(cars)
## carsCI
cars[12]
??centralImputation
```

### Discussion

The central imputation function fills in missing values with the statistic of centrality for our "mpg" column, here median as the column is a continuous variable. After comparing the imputed values to values in complete rows, the imputed values don't seem to make sense, and may be higher than they should otherwise be. E.g. row 12, should have an mpg of between 17-19, but is imputed in my variable as 23, which is significantly higher than it should be and would cause issues for us down the road. ...

Problem 3 continued on next page...

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

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

## Discussion and Correlation Matrix

There are several interesting correlations between “weight”, “horsepower”, and “displacement” . . .

   (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

```
summary(cars)
 install.packages("corrplot")
 library(corrplot)
cm <- cor(cars[,c(1,3,4,5,6)],use="complete.obs")
 corrplot(cm)

 cor(cars$mpg, cars$horsepower)

lm(displacement ~ horsepower, data = cars)

 fillhorsepower <- function(hp) ifelse(is.na(hp),NA,-60.59 + 2.44 * hp)
 cars[is.na(cars$horsepower), "horsepower"] <- sapply(cars[is.na(cars$
    horsepower), "displacement"], fillhorsepower)
```

## Results

The displacement variable has the highest linear correlation with the horsepower variable. The new values of missing data are between 46 and 48 mpg, which is in line with similar cases. . . .

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

```
 library(DMwR)

 carsknn <- knnImputation(cars, k=5, meth="median")
summary(carsknn)
```

### Discussion

By specifying method as “median”, the knnImputation function fills in cases with NA values by using the median value of each continuous variable (or mode of categorical), as opposed to a weighted average. This function replaces the NA values in each row. . . .

# 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

```
??lm
carsknn
carslm <- lm(mpg ~ weight, carsknn)
summary(carslm)

plot(carslm)

plot(x=carsknn$mpg, y=carsknn$weight)
```

### Discussion of Output of Summary() Function and Results

The residuals represent the difference between the observed values and the values the model predicted. The coefficients appear to be normally distributed, which at least indicates a consistent error. The r-squared and adjusted r-squared indicate a relationship between our predictor and target variable. A value closer to 1 indicates that the relationship explains most of the variance. Here, we have an r-squared value of .39, which is not a very strong relationship.

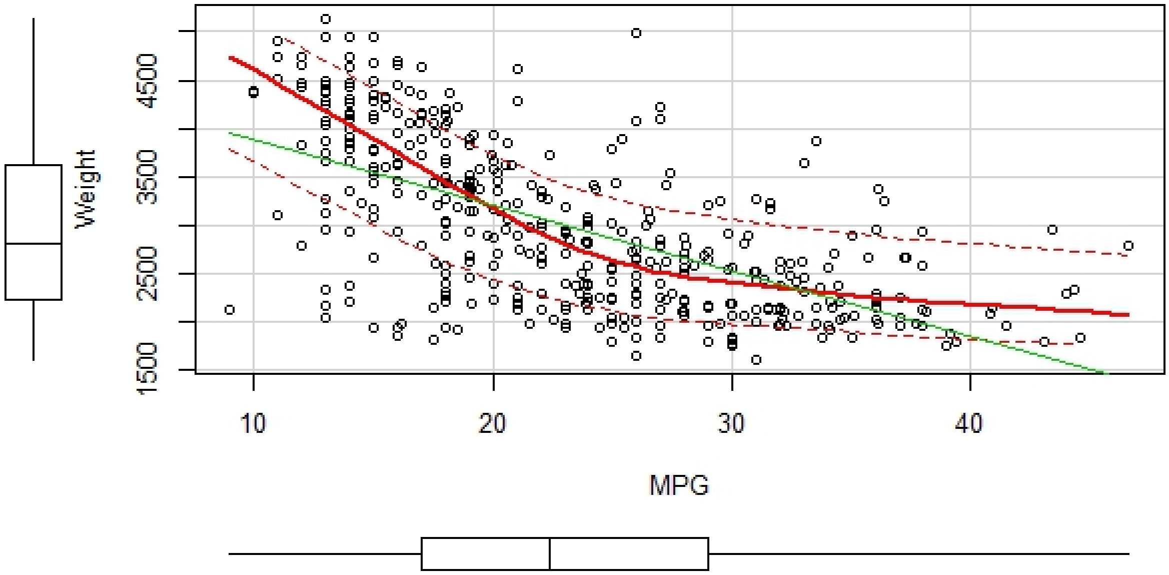
(b) Plot the data points in a scatter plot (mpg vs. weight) and show the model from problem 4.*a* on this scatter plot? Is the correlation between the variables positive or negative? There is a negative correlation of -.62 between the two variables. This result makes sense, because as the mpg increases, the weight decreases, and vice versa. . . .

### R Code

Listing 13: Sample R Script With Highlighting

```
scatterplot(x = carsknn$mpg, y=carsknn$weight)
scatterplot(lm)
cor(carsknn$mpg,carsknn$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

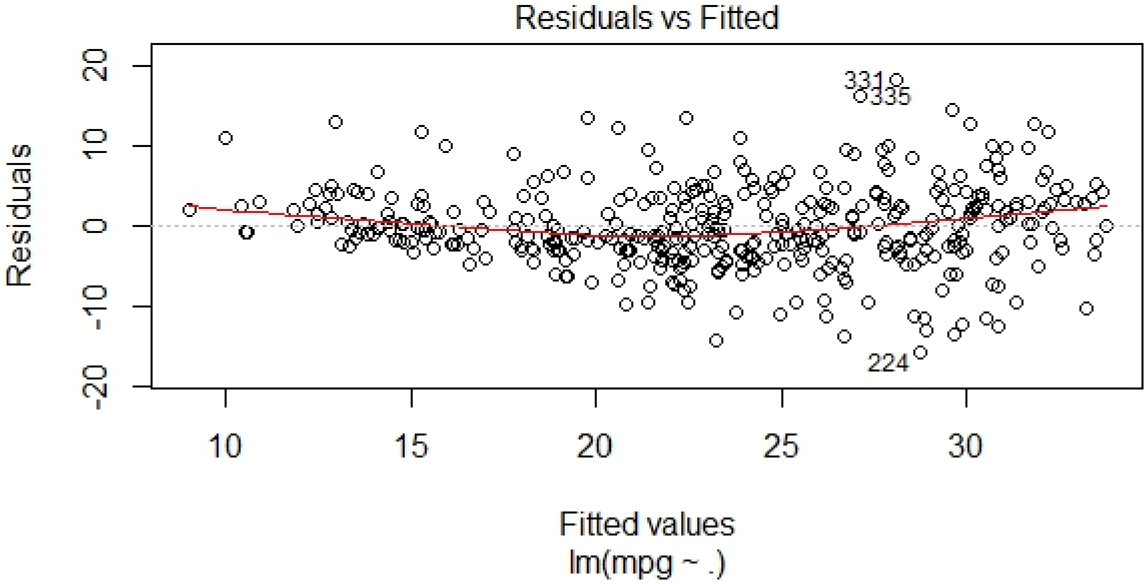
```
lm.mpg <- lm(mpg ~ ., data=carsknn)
## lm.mpg.weight <- lm(mpg ~ weight, data = carsknn)
summary(lm.mpg)
## summary(lm.mpg.weight)
plot(lm.mpg)

anova(lm.mpg)
```

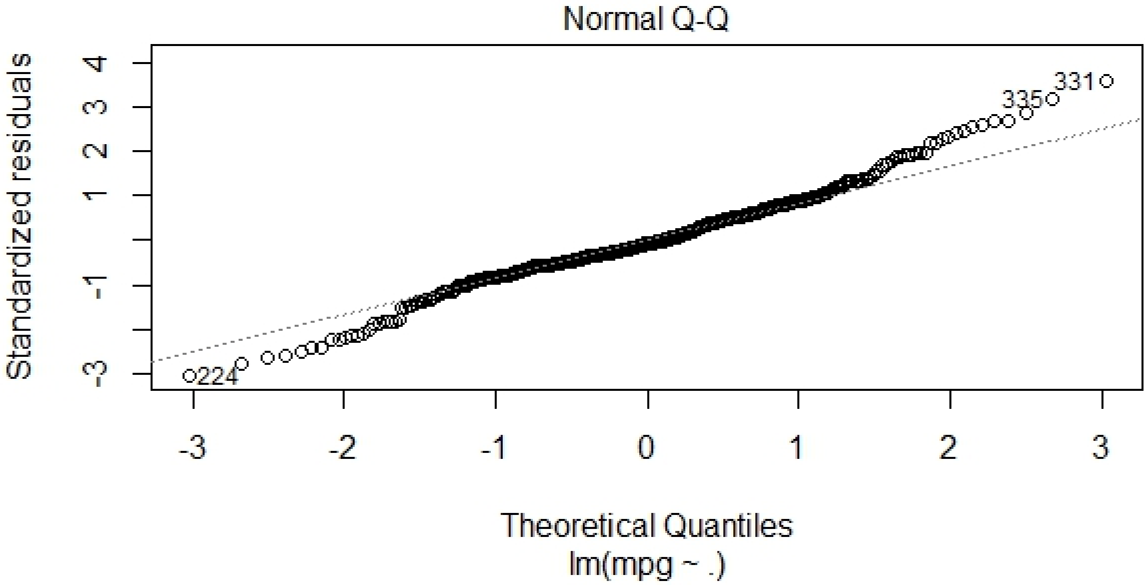
## Discussion of Output of Summary() Function and Results

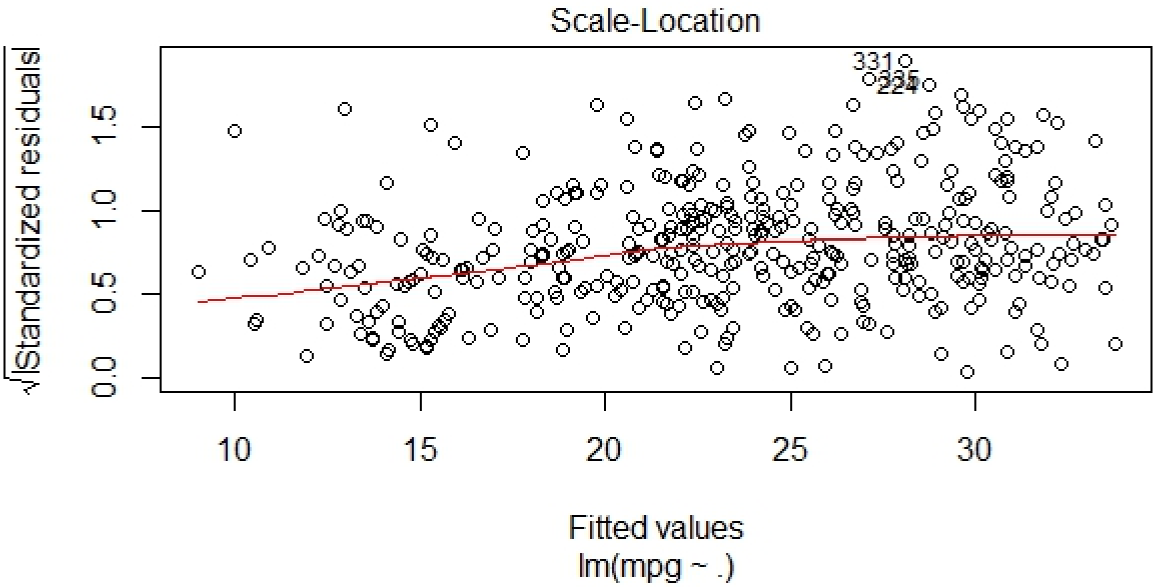
Our multi-variate linear model predicts mpg better than the "weight" variable alone by a significant amount.

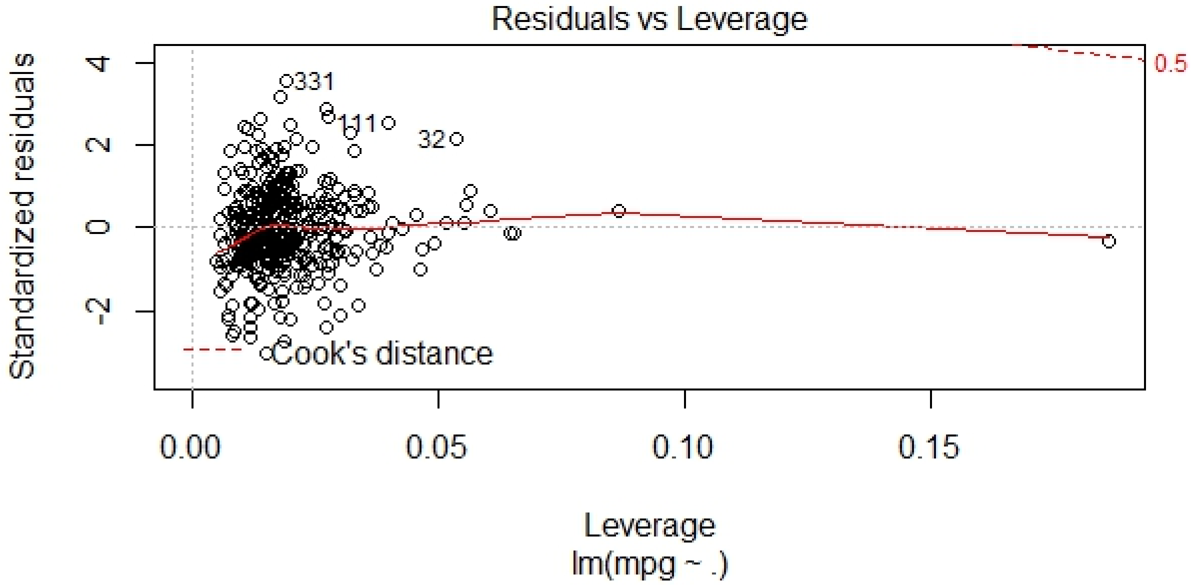
   (b) Use plot() function to understand performance of the model. Insert the four figures below and discuss the plots below that.

**Figure**


Residuals vs Fitted
331
335
224
Residuals
20
10
0
-10
-20
10
15
20
25
30
Fitted values
lm(mpg ~ .)


Normal Q-Q
331
335
224
Standardized residuals
4
3
2
1
-1
-3
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(mpg ~ .)





## Discussion of the Plots

The linear model may not have captured the non-linear relationship indicated by the first plot of Residuals vs. Fitted. There is a slight negative parabolic curve to the data shown in the plot. The Normal QQ plot of residual errors looks slightly skewed at the tails, though the overall distribution is close enough to normal. The Scale-location or Spread Location plot looks close to normal, as the spread of residuals looks roughly the same near 10 as it does near 50, with the exception of a few outliers. Finally, in the last plot, there are no values outside of the dashed line, Cook's distance, that would materially alter the regression analysis, though there are several outliers. In conclusion, though not optimal, this model does a mediocre to sufficient job of fitting the data. For the amount of effort we put into creating the model (one line of R plus some munging), the return on investment is decent.

(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

```
%% You provide code here %%
```

(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

```
anova(lm.mpg)
carsclean <- carsknn[,c(1:6,8)]
final.lm <- lm(mpg ~ .,data=carsclean)
summary(final.lm)
```

3. Regression Trees:

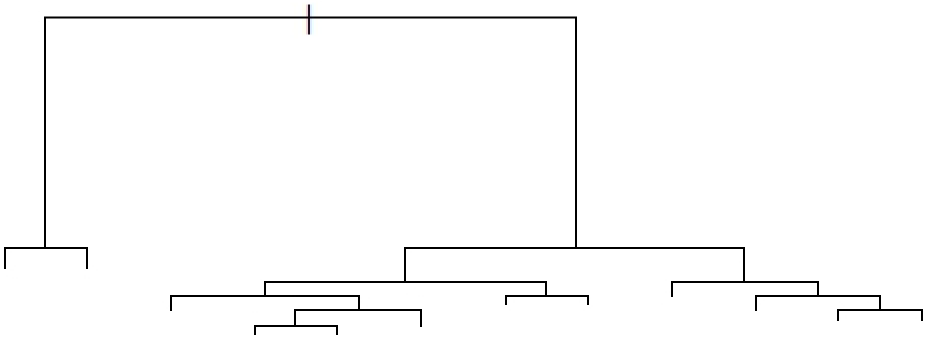
(a) Train a regression tree to predict “mpg” variable using all other variables (variables 2-8). Visualize the tree. Call this model, “final.tree”.

## R Code

Listing 17: Sample R Script With Highlighting

```
require(rpart)
require(rpart.plot)

final.tree <- rpart(mpg ~ .,data=carsclean)
summary(final.tree)
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? The tree prediction model has an MAE of 3.91 vs. 4.48 for the linear model....

### R Code

Listing 18: Sample R Script With Highlighting

```
treepredict <- predict(final.tree, carsclean)
lmpredict <- predict(final.lm, carsclean)
summary(treepredict)
summary(lmpredict)

mae.treepredict <- mean(abs(treepredict - carsclean[["mpg"]]))
mae.lmpredict <- mean(abs(lmpredict - carsclean[["mpg"]]))
```

2. Calculate mean squared error (MSE ) for final.lm and final.tree. Which one is better? Again, the tree model wins with a lower MSE of 2.57 vs. 3.48 for the linear model. ...

## R Code

Listing 19: Sample R Script With Highlighting

```r
treepredict <- predict(final.tree, carsclean)
lmpredict <- predict(final.lm, carsclean)
summary(treepredict)
summary(lmpredict)

mse.treepredict <- mean(treepredict - carsclean[["mpg"]])^2
mse.lmpredict <- mean(lmpredict - carsclean[["mpg"]])^2
```

3. Calculate normalized mean squared error (NMSE ) for final.lm and final.tree. Which one is better? Answer here. The tree model wins again - 2.42 vs. 5.75 NMSE . . .

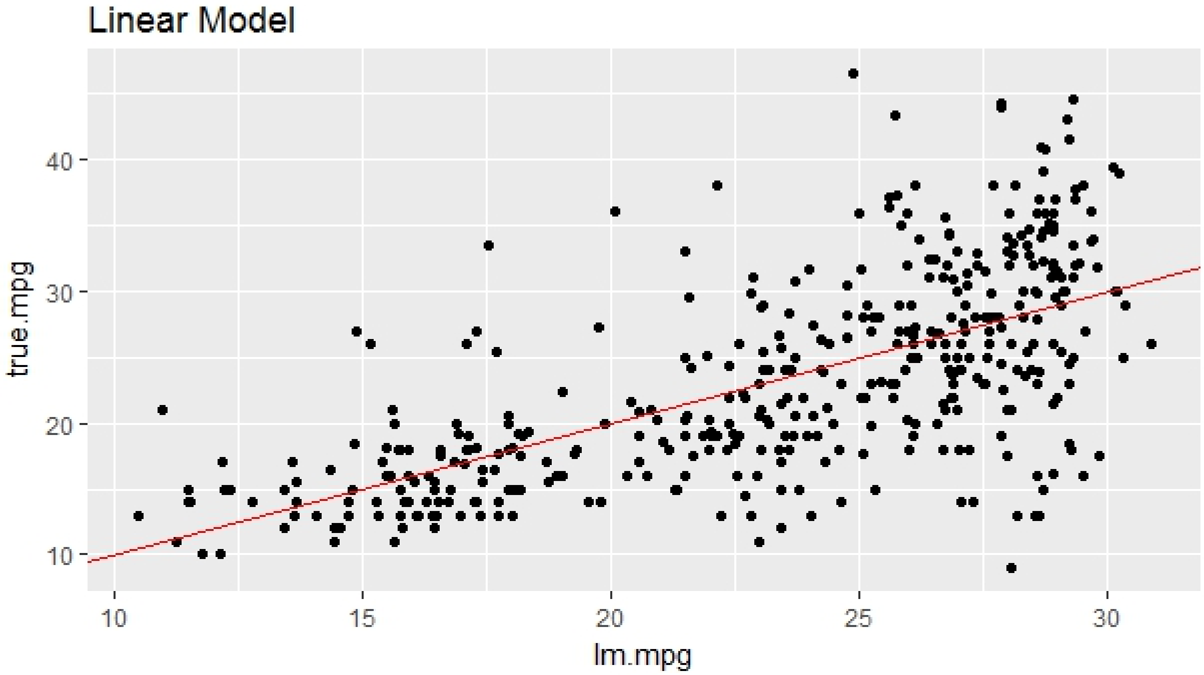
## R Code

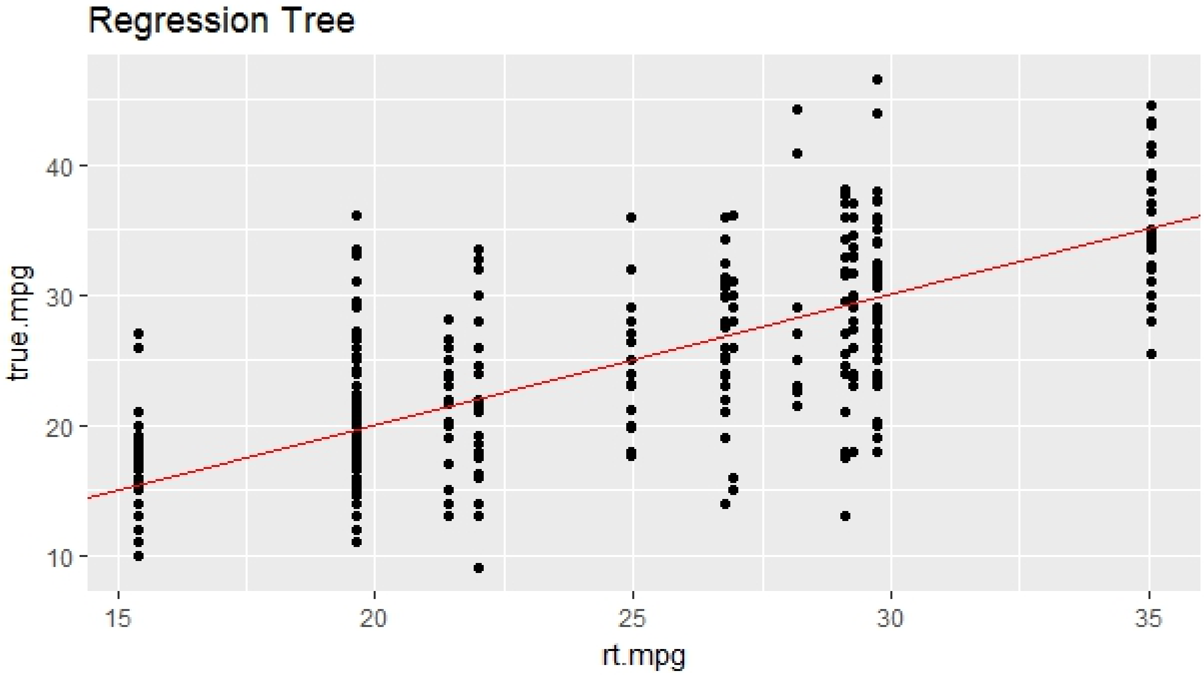
Listing 20: Sample R Script With Highlighting

```r
treepredict <- predict(final.tree, carsclean)
lmpredict <- predict(final.lm, carsclean)
summary(treepredict)
summary(lmpredict)

mse.treepredict <- mean(treepredict - carsclean[["mpg"]])^2
mse.lmpredict <- mean(lmpredict - carsclean[["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





### R Code

Listing 21: Sample R Script With Highlighting

```
require(ggplot2)

carframe <- data.frame(lm.mpg=lmpredict,
                          rt.mpg=treepredict,
                          true.mpg=carsclean[["mpg"]])

ggplot(carframe,aes(x=lm.mpg,y=true.mpg)) +
  geom_point() + geom_abline(slope=1,intercept=0,color="red") +
  ggtitle("Linear Model")

ggplot(carframe,aes(x=rt.mpg,y=true.mpg)) +
  geom_point() + geom_abline(slope=1,intercept=0,color="red") +
  ggtitle("Regression Tree")
```

### Discussion of the Error Plots

The error plots appear to confirm what we already know, which is that the regression tree performed better. The values were less spread out over the regression tree plot, but both models only fit the data with mediocre results. The linear model was more accurate at predicting lower-valued mpg values, though that was likely because 1) there were fewer values, and 2) the other variables were likely less distributed for cars (observations) with lower mpgs (target variable.) ...

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

```
%% You provide code here %%
```

Answer the question below:

(a) This function (according to the documentation) provides bootstrap estimates of the performance of a predictive task, e.g. our linear model or regression tree.

### Discussion of performanceEstimation() Function

Answer here ...

(b) Compare the models? Which model performed best? The final model that performed the best appears to be the linear model with 5 reps of 10 folds. ...