mpg_data

Vijay Rohin Periaiah

July 31, 2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of
 3 multivalued discrete and 5 continuous attributes. (Quinlan, 1993)

# @author: Vijay Rohin Periaiah

#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
#Mean imputation is used to impute the missing values in horsepower.
#By using "pairwise.complete.obs" (correlation matrix), we can find the correlation of the datas
et,
#which ignores the rows with missing values, from that matrix we can infer that cylinders and di
splacement are highly correlated to horsepower.
#Thus we find the mean of the horsepower values for the combination of cylinder and displacement
 for the missing values.

# To read given train data csv file
train_data <- read.csv('data/mpg.csv')

# To get the column names of the dataset
colnames(train_data)
```

```
##  [1] "X"  "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9"
```

```
# To rewrite the column names properly
colnames(train_data) <- c("s_no", "mpg", "cylinders", "displacement", "horsepower", "weight", "a
cceleration", "model_year", "origin", "car_name")

# To verify columns, which are of numeric type
sapply(train_data, is.numeric)
```

```
##         s_no          mpg    cylinders displacement   horsepower
##         TRUE         TRUE         TRUE         TRUE        FALSE
##       weight acceleration   model_year       origin     car_name
##         TRUE         TRUE         TRUE         TRUE        FALSE
```

```
# To update with NA in horsepower column for non-numeric values
train_data$horsepower = suppressWarnings(as.numeric(as.character(train_data$horsepower)))

# To determine the rows having NA values in horsepower column
train_data[is.na(train_data$horsepower),]
```

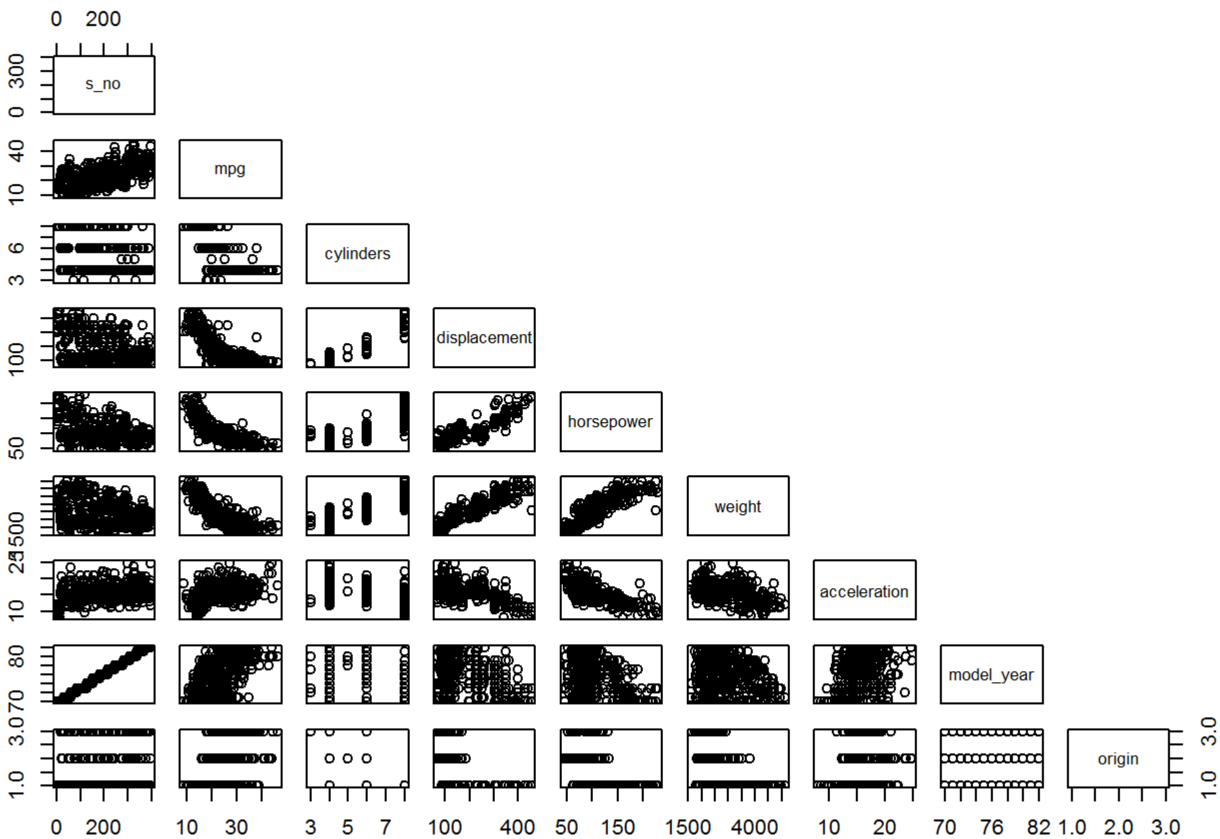
```
##     s_no  mpg cylinders displacement horsepower weight acceleration
## 33    33 25.0         4           98         NA   2046         19.0
## 127  127 21.0         6          200         NA   2875         17.0
## 331  331 40.9         4           85         NA   1835         17.3
## 337  337 23.6         4          140         NA   2905         14.3
## 355  355 34.5         4          100         NA   2320         15.8
## 375  375 23.0         4          151         NA   3035         20.5
##     model_year origin             car_name
## 33          71      1           ford pinto
## 127         74      1        ford maverick
## 331         80      2 renault lecar deluxe
## 337         80      1   ford mustang cobra
## 355         81      2         renault 18i
## 375         82      1       amc concord dl
```

```
# To get numeric values from the train data set only
train_data <- train_data[, sapply(train_data, is.numeric)]

# To construct correlation matrix and determine their relationships
cor(train_data, use = 'pairwise.complete.obs', method = 'pearson')
```

```
##                    s_no        mpg  cylinders displacement horsepower
## s_no          1.0000000  0.5851312 -0.3630399   -0.3869756 -0.4229012
## mpg           0.5851312  1.0000000 -0.7753963   -0.8042028 -0.7784268
## cylinders    -0.3630399 -0.7753963  1.0000000    0.9507214  0.8429834
## displacement -0.3869756 -0.8042028  0.9507214    1.0000000  0.8972570
## horsepower   -0.4229012 -0.7784268  0.8429834    0.8972570  1.0000000
## weight       -0.3188685 -0.8317409  0.8960168    0.9328241  0.8645377
## acceleration  0.2876343  0.4202889 -0.5054195   -0.5436841 -0.6891955
## model_year    0.9968000  0.5792671 -0.3487458   -0.3701642 -0.4163615
## origin        0.1997020  0.5634504 -0.5625433   -0.6094094 -0.4551715
##                  weight acceleration model_year     origin
## s_no         -0.3188685    0.2876343  0.9968000  0.1997020
## mpg          -0.8317409    0.4202889  0.5792671  0.5634504
## cylinders     0.8960168   -0.5054195 -0.3487458 -0.5625433
## displacement  0.9328241   -0.5436841 -0.3701642 -0.6094094
## horsepower    0.8645377   -0.6891955 -0.4163615 -0.4551715
## weight        1.0000000   -0.4174573 -0.3065643 -0.5810239
## acceleration -0.4174573    1.0000000  0.2881370  0.2058730
## model_year   -0.3065643    0.2881370  1.0000000  0.1806622
## origin       -0.5810239    0.2058730  0.1806622  1.0000000
```

```
# On examining the above matrix (correlation), we can infer that horsepower, cylinders and displ
acement are definitely correlated
# Horsepower is restricted to certain ranges for few different values in cylinder and displaceme
nt, whereas weight is varied a lot.
# We have imputed the missing values with the mean of horsepower for that certain range.
# There were 6 missing values in horsepower, so the mean of horsepower is computed based on it's
 corresponding cylinders and displacement
# Cylinder 6 and Displacement = 200, horsepower (mean imputed value) = 86
# Cylinder 4 and Displacement = 98, horsepower (mean imputed value) = 72
# Cylinder 4 and Displacement = 85, horsepower (mean imputed value) = 65
# Cylinder 4 and Displacement = 140, horsepower (mean imputed value) = 85
# Cylinder 4 and Displacement = 151, horsepower (mean imputed value) = 89
# Cylinder 4 and Displacement = 100, horsepower (mean imputed value) = 83
# For this above last entry alone mean of horsepower with displacement = 101 and cylinders = 4 i
s taken,
# since there are no rows to calculate mean horsepower for displacement = 100 and cylinders = 4.

train_data <- within(train_data, horsepower[displacement == 200 & cylinders == 6] <- 86)
train_data <- within(train_data, horsepower[displacement == 98 & cylinders == 4] <- 72)
train_data <- within(train_data, horsepower[displacement == 85 & cylinders == 4] <- 65)
train_data <- within(train_data, horsepower[displacement == 140 & cylinders == 4] <- 85)
train_data <- within(train_data, horsepower[displacement == 151 & cylinders == 4] <- 89)
train_data <- within(train_data, horsepower[displacement == 100 & cylinders == 4] <- 83)

#mpg vs cylinders:
#  There lies a negative correlation between mpg and cylinders, since it follows a step-wise pat
tern where mpg varies within a certain range for each value of cylinders and slopes downwards fr
om left to right.

#mpg vs displacement:
#  There lies a negative correlation between mpg and displacement, where the pattern slopes down
wards from left to right.

#mpg vs horsepower:
#  There lies a negative correlation between mpg and horsepower, since the pattern shows a downw
ard sloping from left to right.

#mpg vs weight:
#  There lies a negative correlation between mpg and weight similar to the above two comparison
s, where the pattern slopes downwards from left to right.

#mpg vs acceleration:
#  From the plot, we infer that there is a slight little positive correlation between mpg and ac
celeration, since the pattern slopes upwards from left to right.

#mpg vs model_year:
#  There a lies a positive correlation between mp and model year.

#mpg vs origin:
#  There lies a slight little positive correlation between mpg and origin, where horizontal patt
ern (step-wise pattern) exits.

# To remove car name and acquire numeric data from the train dataset
```

```
train_data <- train_data[, sapply(train_data, is.numeric)]

# To construct the pair plots for all the given variables in the train dataset except car name
pairs(train_data, upper.panel = NULL)
```



```
#mpg vs cylinders:
#  There lies a negative correlation between mpg and cylinders, since it follows a step-wise pat
tern where mpg varies within a certain range for each value of cylinders and slopes downwards fr
om left to right.

#mpg vs displacement:
#  There lies a negative correlation between mpg and displacement, where the pattern slopes down
wards from left to right.

#mpg vs horsepower:
#  There lies a negative correlation between mpg and horsepower, since the pattern shows a downw
ard sloping from left to right.

#mpg vs weight:
#  There lies a negative correlation between mpg and weight similar to the above two comparison
s, where the pattern slopes downwards from left to right.

#mpg vs acceleration:
#  From the plot, we infer that there is a slight little positive correlation between mpg and ac
celeration, since the pattern slopes upwards from left to right.

#mpg vs model_year:
#  There a lies a positive correlation between mp and model year.

#mpg vs origin:
#  There lies a slight little positive correlation between mpg and origin, where horizontal patt
ern (step-wise pattern) exits.


#On observing the plots and correlation matrix, we can infer that weight, cylinders, displacemen
t and horsepower is strongly correlated.
#Therefore, all of the variables cannot be considered together as they might give rise to multic
ollinearity problem.
#Weight seems to have a very strong correlation with mpg. So its is a strong candidate.
#Also, acceleration doesn't seem to impact mpg much. Therefore, acceleration can be ignored.
#Model year also shows some positive correlation, so it can be considered as well.

#In order to confirm our choices, we can run a initial set of pairwise regression which will com
firm the choices.

#Therefore, it is proposed that, weight, model year and origin will explain mpg well.

# To construct correlation matrix and determine their relationships (correlation) after mean imp
utation for missing values
cor(train_data, method = 'pearson')
```

```
##                    s_no        mpg  cylinders displacement horsepower
## s_no          1.0000000  0.5851312 -0.3630399   -0.3869756 -0.4215314
## mpg           0.5851312  1.0000000 -0.7753963   -0.8042028 -0.7795976
## cylinders    -0.3630399 -0.7753963  1.0000000    0.9507214  0.8439529
## displacement -0.3869756 -0.8042028  0.9507214    1.0000000  0.8993374
## horsepower   -0.4215314 -0.7795976  0.8439529    0.8993374  1.0000000
## weight       -0.3188685 -0.8317409  0.8960168    0.9328241  0.8656229
## acceleration  0.2876343  0.4202889 -0.5054195   -0.5436841 -0.6833752
## model_year    0.9968000  0.5792671 -0.3487458   -0.3701642 -0.4155291
## origin        0.1997020  0.5634504 -0.5625433   -0.6094094 -0.4554978
##                  weight acceleration model_year     origin
## s_no         -0.3188685    0.2876343  0.9968000  0.1997020
## mpg          -0.8317409    0.4202889  0.5792671  0.5634504
## cylinders     0.8960168   -0.5054195 -0.3487458 -0.5625433
## displacement  0.9328241   -0.5436841 -0.3701642 -0.6094094
## horsepower    0.8656229   -0.6833752 -0.4155291 -0.4554978
## weight        1.0000000   -0.4174573 -0.3065643 -0.5810239
## acceleration -0.4174573    1.0000000  0.2881370  0.2058730
## model_year   -0.3065643    0.2881370  1.0000000  0.1806622
## origin       -0.5810239    0.2058730  0.1806622  1.0000000
```

```
# From the correlation matrix and pair plots we can conclude that cylinders, displacement, horse
power and weight (All 4 variables) are highly correlated.
# This explicitly states the presence of multicollinearity problem, if all the 4 variables are i
ncluded as predictors.

# Linear model 1 - All available variables vs (against) mpg (miles per gallon)
lim_1 <- lm(mpg ~ cylinders + displacement + horsepower + weight +
              acceleration + model_year + origin, data = train_data)
summary(lim_1)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year + origin, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.593 -2.168 -0.149  1.864 12.993
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.686e+01  4.605e+00  -3.662 0.000285 ***
## cylinders    -4.443e-01  3.219e-01  -1.380 0.168246
## displacement  2.035e-02  7.544e-03   2.697 0.007298 **
## horsepower   -1.947e-02  1.364e-02  -1.427 0.154327
## weight       -6.554e-03  6.432e-04 -10.189  < 2e-16 ***
## acceleration  6.945e-02  9.649e-02   0.720 0.472105
## model_year    7.498e-01  5.050e-02  14.846  < 2e-16 ***
## origin        1.456e+00  2.756e-01   5.282 2.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.332 on 390 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8183
## F-statistic: 256.4 on 7 and 390 DF,  p-value: < 2.2e-16
```

```
# For the below created linear models from 2 - 7, we have considered only the correlated predict
ors.
# These shows that one of the predictors becomes insignificant or R-squared value decreases.

# Linear model 2 - weight & horsepower (correlated) vs (against) mpg (miles per gallon)
# Here we can see that horsepower becomes insignificant
lim_2 <- lm(mpg ~ weight + horsepower + model_year + origin, data = train_data)
summary(lim_2)
```

```
##
## Call:
## lm(formula = mpg ~ weight + horsepower + model_year + origin,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9108 -2.1020 -0.1491  1.6756 13.1811
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.666e+01  4.097e+00  -4.067 5.76e-05 ***
## weight      -5.610e-03  4.403e-04 -12.742  < 2e-16 ***
## horsepower  -1.067e-02  9.327e-03  -1.144    0.253
## model_year   7.376e-01  5.041e-02  14.632  < 2e-16 ***
## origin       1.203e+00  2.597e-01   4.633 4.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.352 on 393 degrees of freedom
## Multiple R-squared:  0.818,  Adjusted R-squared:  0.8161
## F-statistic: 441.5 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 3 - weight & cylinders (correlated) vs (against) mpg (miles per gallon)
# Here we can see that cylinders becomes insignificant
lim_3 <- lm(mpg ~ weight + cylinders + model_year + origin, data = train_data)
summary(lim_3)
```

```
##
## Call:
## lm(formula = mpg ~ weight + cylinders + model_year + origin,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9701 -2.1311 -0.0412  1.7403 13.2119
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.800e+01  4.044e+00  -4.450 1.12e-05 ***
## weight      -6.079e-03  4.583e-04 -13.263  < 2e-16 ***
## cylinders    3.314e-02  2.282e-01   0.145    0.885
## model_year   7.571e-01  4.863e-02  15.568  < 2e-16 ***
## origin       1.171e+00  2.599e-01   4.504 8.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.357 on 393 degrees of freedom
## Multiple R-squared:  0.8174, Adjusted R-squared:  0.8155
## F-statistic: 439.7 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 4 - weight & displacement (correlated) vs (against) mpg (miles per gallon)
# Here we can see that displacement becomes insignificant
lim_4 <- lm(mpg ~ weight + displacement + model_year + origin, data = train_data)
summary(lim_4)
```

```
##
## Call:
## lm(formula = mpg ~ weight + displacement + model_year + origin,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8460 -2.1231  0.0029  1.8091 13.1498
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -18.544887   3.984405  -4.654 4.45e-06 ***
## weight        -0.006690   0.000555 -12.053  < 2e-16 ***
## displacement   0.006403   0.004750   1.348    0.178
## model_year     0.772590   0.049350  15.655  < 2e-16 ***
## origin         1.250741   0.265037   4.719 3.30e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.35 on 393 degrees of freedom
## Multiple R-squared:  0.8182, Adjusted R-squared:  0.8163
## F-statistic: 442.2 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 5 - displacement & cylinders (correlated) vs (against) mpg (miles per gallon)
# Here we can see that cylinders becomes insignificant
lim_5 <- lm(mpg ~ displacement + cylinders + model_year + origin, data = train_data)
summary(lim_5)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + cylinders + model_year + origin,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.248  -2.378  -0.269   2.057  13.749
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -22.154842   4.684673  -4.729 3.15e-06 ***
## displacement  -0.033784   0.006401  -5.278 2.17e-07 ***
## cylinders     -0.684486   0.372372  -1.838   0.0668 .
## model_year     0.706771   0.057145  12.368  < 2e-16 ***
## origin         1.408970   0.309381   4.554 7.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.903 on 393 degrees of freedom
## Multiple R-squared:  0.7531, Adjusted R-squared:  0.7506
## F-statistic: 299.7 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 6 - displacement & horsepower (correlated) vs (against) mpg (miles per gallon)
# Here we can see that R-squared value is decreased
lim_6 <- lm(mpg ~ displacement + horsepower + model_year, data = train_data)
summary(lim_6)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + horsepower + model_year, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0404 -2.6652 -0.2661  2.2609 14.7932
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -15.131007   4.801820  -3.151  0.00175 **
## displacement  -0.040845   0.004376  -9.335  < 2e-16 ***
## horsepower    -0.032765   0.012182  -2.689  0.00746 **
## model_year     0.657253   0.059317  11.080  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.975 on 394 degrees of freedom
## Multiple R-squared:  0.7433, Adjusted R-squared:  0.7414
## F-statistic: 380.3 on 3 and 394 DF,  p-value: < 2.2e-16
```

```
# Linear model 7 - cylinders & horsepower (correlated) vs (against) mpg (miles per gallon)
# Here we can see that R-squared value is decreased
lim_7 <- lm(mpg ~ cylinders + horsepower + model_year, data = train_data)
summary(lim_7)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + horsepower + model_year, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3428 -2.8534 -0.1872  2.3512 15.2959
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.45705    4.87955  -1.938   0.0533 .
## cylinders   -1.88518    0.22215  -8.486 4.39e-16 ***
## horsepower  -0.06227    0.01018  -6.117 2.31e-09 ***
## model_year   0.65436    0.06027  10.857  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.039 on 394 degrees of freedom
## Multiple R-squared:  0.735,  Adjusted R-squared:  0.733
## F-statistic: 364.3 on 3 and 394 DF,  p-value: < 2.2e-16
```

```
# To determine the most strongest predictor among the above correlated variables,
# Individually we run, each one of the correlated variable with other given variables like accel
eration, model year & origin.
# Finally, we can conclude that acceleration turns out to be insignificant almost all the times.

# Linear model 8 - cylinders & other given variables vs (against) mpg (miles per gallon)
lim_8 <- lm(mpg ~ cylinders + acceleration + model_year + origin, data = train_data)
summary(lim_8)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + acceleration + model_year + origin,
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0228  -2.4199  -0.3349   2.2771  13.7451
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -23.12528    4.98836  -4.636 4.85e-06 ***
## cylinders     -2.47883    0.16825 -14.733  < 2e-16 ***
## acceleration   0.01533    0.08653   0.177    0.859
## model_year     0.74906    0.05906  12.682  < 2e-16 ***
## origin         1.89870    0.30757   6.173 1.67e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.039 on 393 degrees of freedom
## Multiple R-squared:  0.7356, Adjusted R-squared:  0.7329
## F-statistic: 273.4 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 9 - displacement & other given variables vs (against) mpg (miles per gallon)
lim_9 <- lm(mpg ~ displacement + acceleration + model_year + origin, data = train_data)
summary(lim_9)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + acceleration + model_year +
##     origin, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2059  -2.1700  -0.3217   1.9518  14.3179
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -21.859448   4.798314  -4.556 6.98e-06 ***
## displacement  -0.046587   0.002902 -16.056  < 2e-16 ***
## acceleration  -0.119292   0.086801  -1.374     0.17
## model_year     0.713230   0.057545  12.394  < 2e-16 ***
## origin         1.290365   0.314544   4.102 4.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.911 on 393 degrees of freedom
## Multiple R-squared:  0.7522, Adjusted R-squared:  0.7497
## F-statistic: 298.2 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 10 - horsepower & other given variables vs (against) mpg (miles per gallon)
lim_10 <- lm(mpg ~ horsepower + acceleration + model_year + origin, data = train_data)
summary(lim_10)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + acceleration + model_year + origin,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9047 -2.5232 -0.4744  2.2734 12.9389
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9.206827   5.181676  -1.777   0.0764 .
## horsepower   -0.133134   0.008096 -16.445  < 2e-16 ***
## acceleration -0.468050   0.097850  -4.783 2.44e-06 ***
## model_year    0.659465   0.057802  11.409  < 2e-16 ***
## origin        2.380842   0.275915   8.629  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.873 on 393 degrees of freedom
## Multiple R-squared:  0.7569, Adjusted R-squared:  0.7544
## F-statistic: 305.9 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
# Linear model 11 - weight & other given variables vs (against) mpg (miles per gallon)
lim_11 <- lm(mpg ~ weight + acceleration + model_year + origin, data = train_data)
summary(lim_11)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + model_year + origin,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8230 -2.1282 -0.0342  1.7693 13.1528
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.858e+01  4.012e+00  -4.630 4.97e-06 ***
## weight       -5.930e-03  2.672e-04 -22.192  < 2e-16 ***
## acceleration  7.195e-02  6.842e-02   1.052    0.294
## model_year    7.464e-01  4.865e-02  15.341  < 2e-16 ***
## origin        1.180e+00  2.581e-01   4.573 6.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.353 on 393 degrees of freedom
## Multiple R-squared:  0.8179, Adjusted R-squared:  0.816
## F-statistic: 441.2 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
#Residuals:
#The linear model summary provides various information about the Residuals.
#Residuals = Actual value - Predicted Value (from model).
#Gives us the difference between actual value and predicted value using the model.
#To analyse whether the residuals are symmetrically distributed about the mean, we can use minim
um, median, third quartile and max values.
#We except to have the median close to zero (0) with first and third quartiles symmetrical about
 the mean.
#In the model chosen the median is quite close to zero, however the residuals are not as symmetr
ic as we would like them to be. The max values indicate that some large values exists.

#Coefficients
#The coefficents are the constants that follows the amount of change in the predictor variable,
 causing a unit of change in the response variable
#All the predictors and intercept is highly significant in our regression.

#Estimates
#The estimates of the intercept term, each predictor variables' coefficent betas are provided by
 the column estimates in our summary.
#The expected value of Y given all X are equal to zero is presented by the intercept term.
#In our regression the intercept has a value of -16.02, coefficient for weight is -0.006195 whic
h means weight neagatively impacts mpg although the impact is very small.
#For a 0.6 unit increase in weight, mpg reduces by 1 unit. Model year positively impacts mpg.
#The coefficient for model year is 0.7412 which means for a for a newer model manufactured every
 7 months there is a 1 unit increase in mpg.
#The coefficient of origin is 1.07 which means that if the manufacturing process is done in more
 than 1 place, the mpg increases by 1 unit.

#Standard Error
#The estimate of the standard deviation of the coefficents is given by "Standard Error", which i
s used in the measurement of precision of the estimated coefficent.
#In our regression all of the coefficients have very small standard errors except intercept.

#t Value
#We use t statistics to perform hypothesis testing on the estimates of the coefficents. It is us
ed in the measurement of how many standard deviations away from zero.
#(estimator - parameter) / estimated standard error of estimator.
#Null hypothesis H0: Beta = 0
#t value = (hat)beta / se(hat(beta))
#***se -> standard error
#This enables us to find whether Y is related to X.
#We reject null hypothesis, if the modulus of t statistic is greater than (>) calculated critica
l value.
#For our regression model, based on the t values we can confirm that the coefficients are signif
icant i.e we can reject the null hypothesis at both 5% and 1% level of significance.


#|pr > t|
#  p value: (|pr > t|):
#  p value denotes the probability of observing the particular t value.
#p value is defined as as the lowest significance level at which null hypothesis can be rejecte
d.
#p value must as close to zero as possible, lesser p value is better.
```

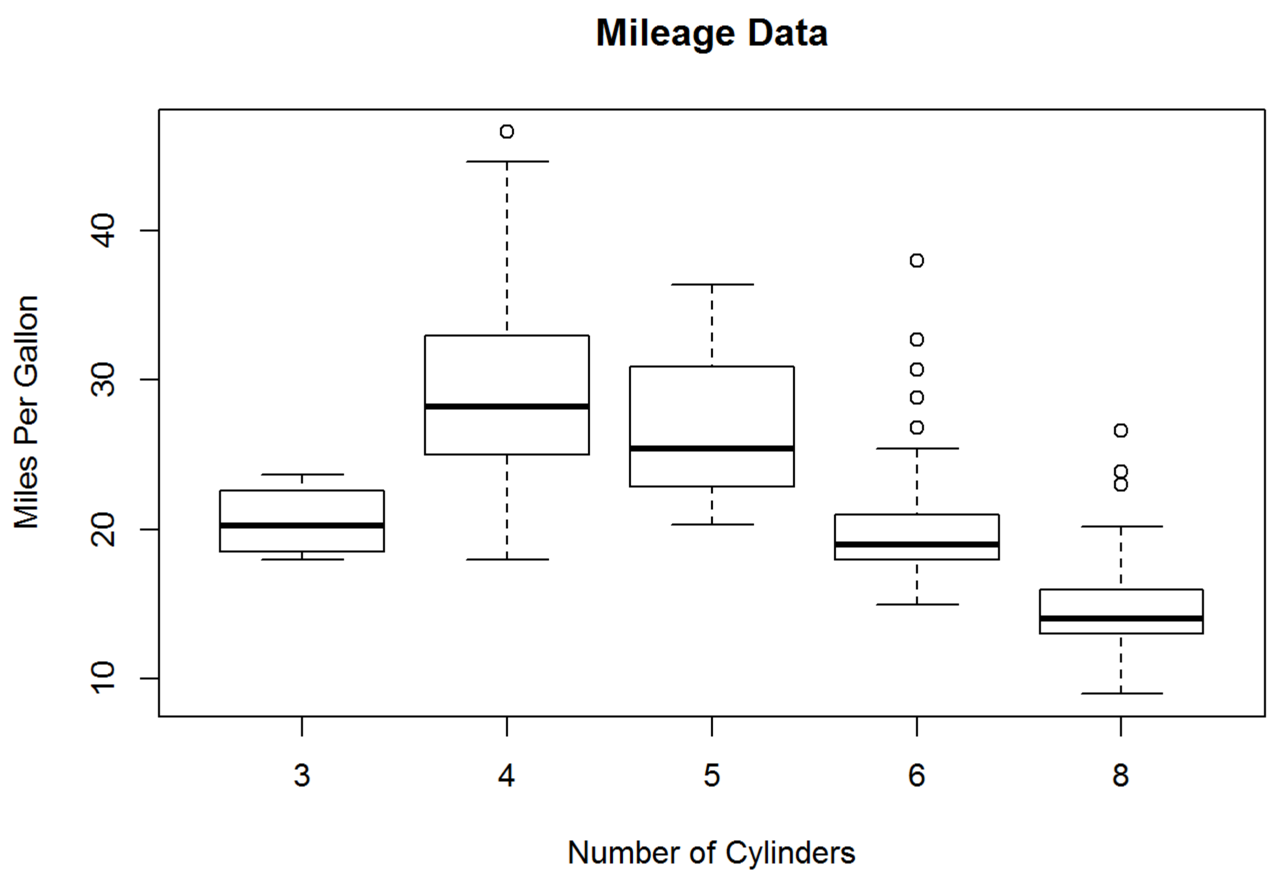
```
#The p-values are extremely close to zero for all the coefficients in our model. The three stars
 beside the p value indicates that they are highly significant.

#Residual Standard Error
#It is te square root of mean square error (mse^-2).
#It is the sd of the residuals of regression and a measure of quality of regression line's fit.
#Lesser RSE value is better.
#The residual standard error in or regression is 3.457 which is smaller when compared to our oth
er models.

#R-Squared
#Total Sum of Squares (TSS) = Explained Sum of Squares (ESS) + Residual Sum of Squares (RSS)
#The total variation of actual Y values about sample mean is called Total Sum of Squares (TSS)
#The variation of estimated Y value about their mean ( variations explained by regressions ) is
 called Explained Sum of Squares (ESS).
#The unexplained variation of Y about the regression line is called Residual Sum of Squares (RS
S).
#The R-squared value for our regression is 0.8149 or 81.49% which the high. So the predictors su
fficiently explain the dependent variable mpg.

#Adjusted - R-square
#As name denotes its the R-square value adjusted for the degrees of freedom, which corrects the
 model if many predictors (variables) are included in it. The Adjusted R squared value is 0.813
3.

# Since acceleration is insignificant almost all times, we can ignore them.
# Among all the correlated predictors (cylinders, displacement, horsepower and weight), the most
 significant is weight.
# The most significant and simplest model that illustrates mpg in a explanatory fashion contains
 weight, model year and origin (little effect on mpg)

# Linear model 12 - weight, model year and origin vs (against) mpg (miles per gallon)
lim_12 <- lm(mpg ~ weight + model_year + origin, data = train_data)
summary(lim_12)
```

```
##
## Call:
## lm(formula = mpg ~ weight + model_year + origin, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0019  -2.0996  -0.0485   1.7371  13.2227
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.788e+01  3.958e+00  -4.518 8.27e-06 ***
## weight      -6.023e-03  2.523e-04 -23.873  < 2e-16 ***
## model_year   7.559e-01  4.781e-02  15.808  < 2e-16 ***
## origin       1.166e+00  2.578e-01   4.524 8.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.353 on 394 degrees of freedom
## Multiple R-squared:  0.8174, Adjusted R-squared:  0.816
## F-statistic: 587.7 on 3 and 394 DF,  p-value: < 2.2e-16
```

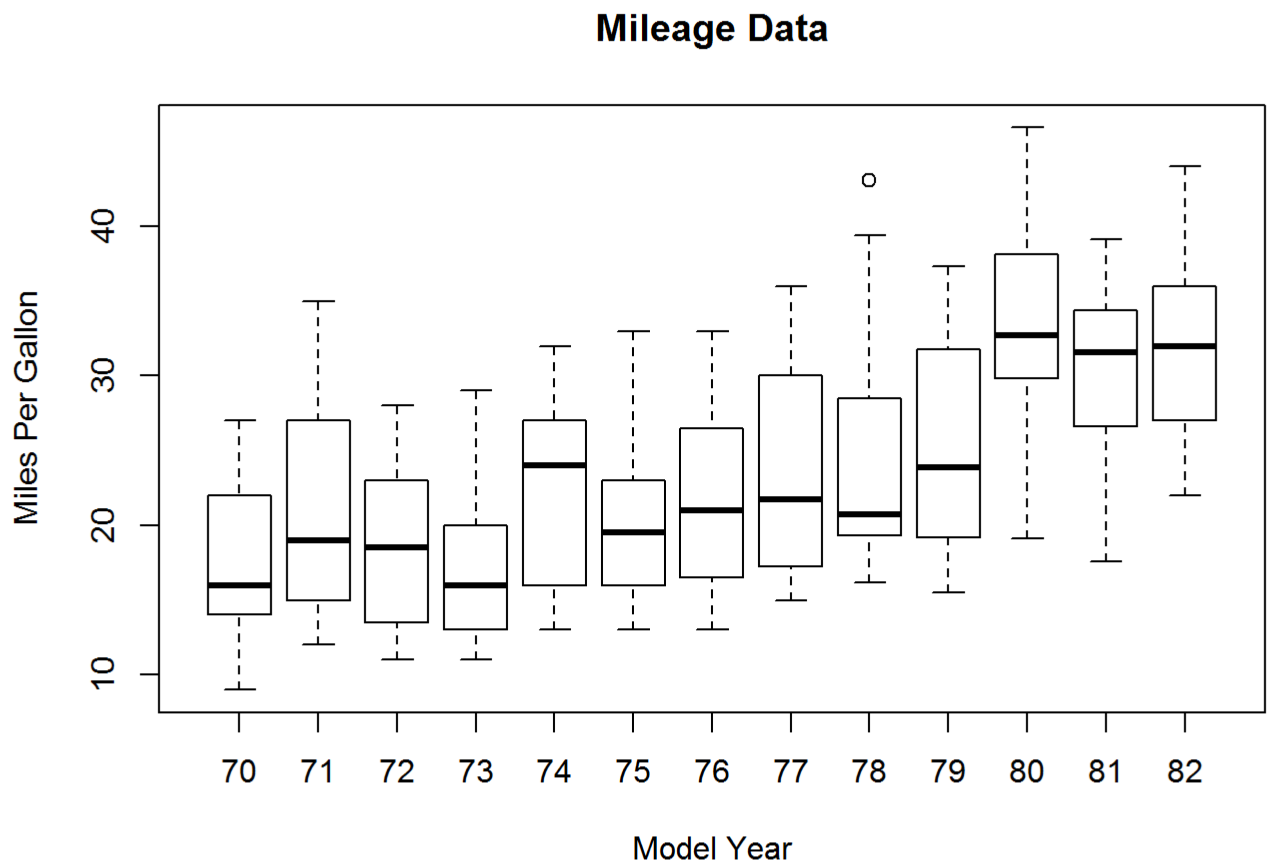
```
# Linear model 13 - weight, model year and ratio between them along with origin vs (against) mpg
 (miles per gallon)
lim_13 <- lm(mpg ~ weight + weight:model_year + model_year+ origin, data = train_data)
summary(lim_13)
```

```
##
## Call:
## lm(formula = mpg ~ weight + weight:model_year + model_year +
##     origin, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.957 -1.885 -0.118  1.630 12.156
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.094e+02  1.262e+01  -8.664  < 2e-16 ***
## weight             2.671e-02  4.325e-03   6.177 1.64e-09 ***
## model_year         1.988e+00  1.686e-01  11.792  < 2e-16 ***
## origin             9.164e-01  2.433e-01   3.766 0.000191 ***
## weight:model_year -4.404e-04  5.810e-05  -7.580 2.50e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.136 on 393 degrees of freedom
## Multiple R-squared:  0.8406, Adjusted R-squared:  0.839
## F-statistic: 518.3 on 4 and 393 DF,  p-value: < 2.2e-16
```

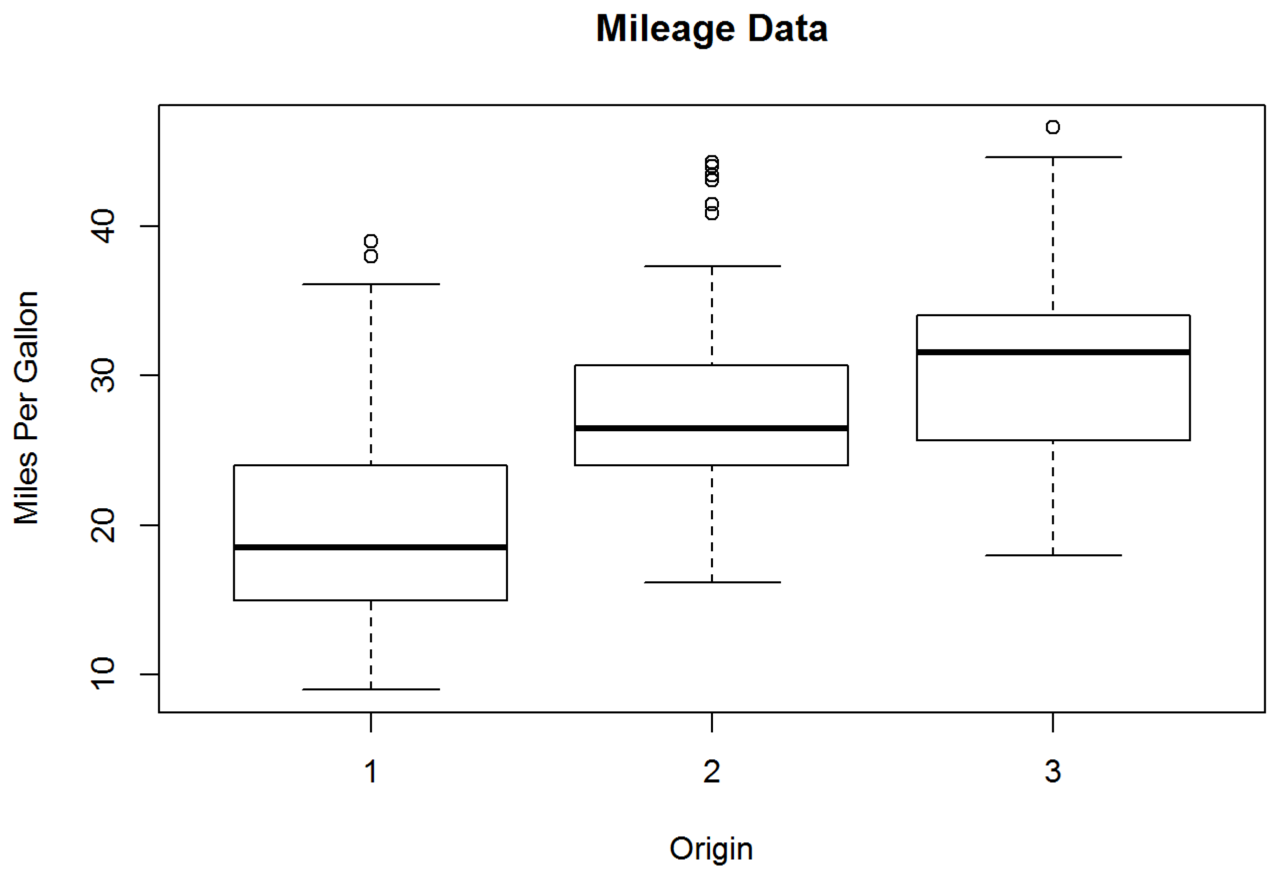
```
#When compared with our previous significant linear model, the adjusted R square value increased
 from 0.816 to 0.839 due to the inclusion of interaction term of weight to model year.

# Box plots are drawn for 3 multivariate discrete attributes against mpg
boxplot(mpg ~ cylinders, data = train_data, xlab = "Number of Cylinders",ylab = "Miles Per Gallo
n", main = "Mileage Data")
```



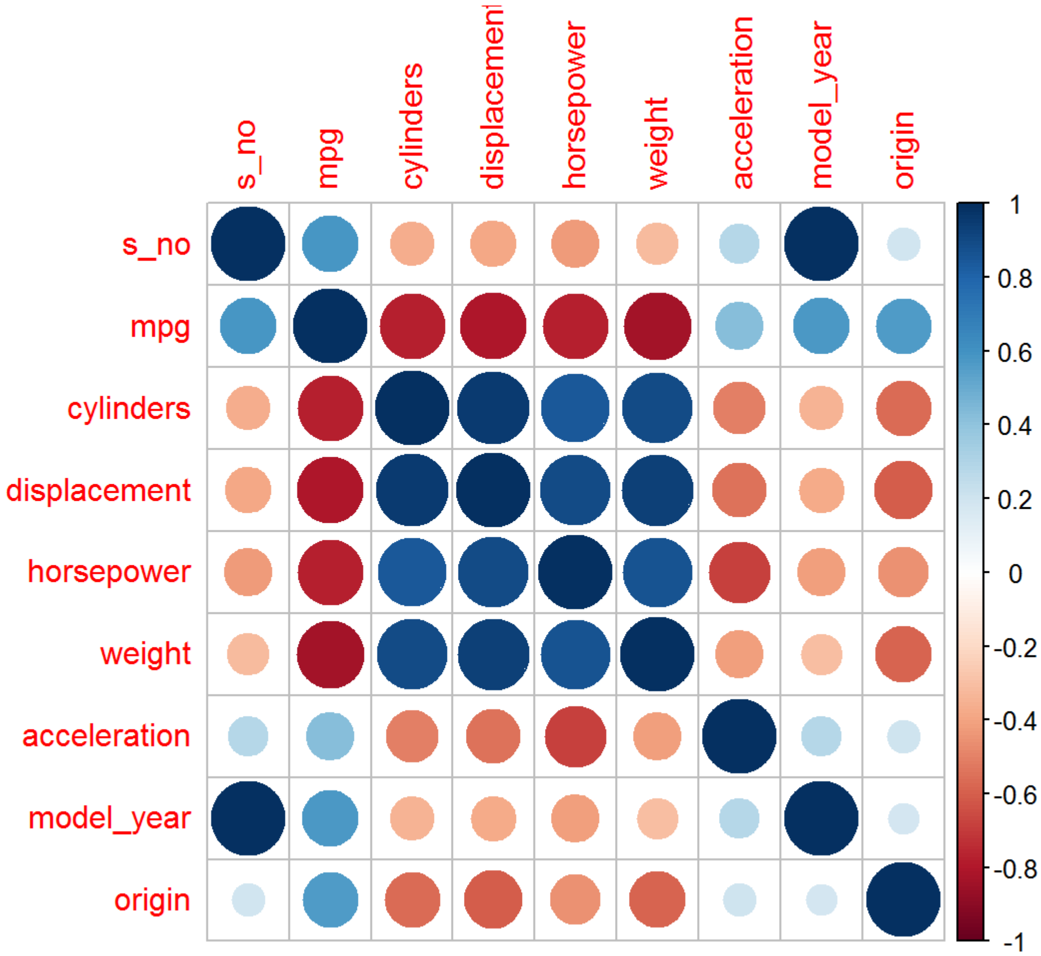

```
boxplot(mpg ~ model_year, data = train_data, xlab = "Model Year",ylab = "Miles Per Gallon", main
 = "Mileage Data")
```



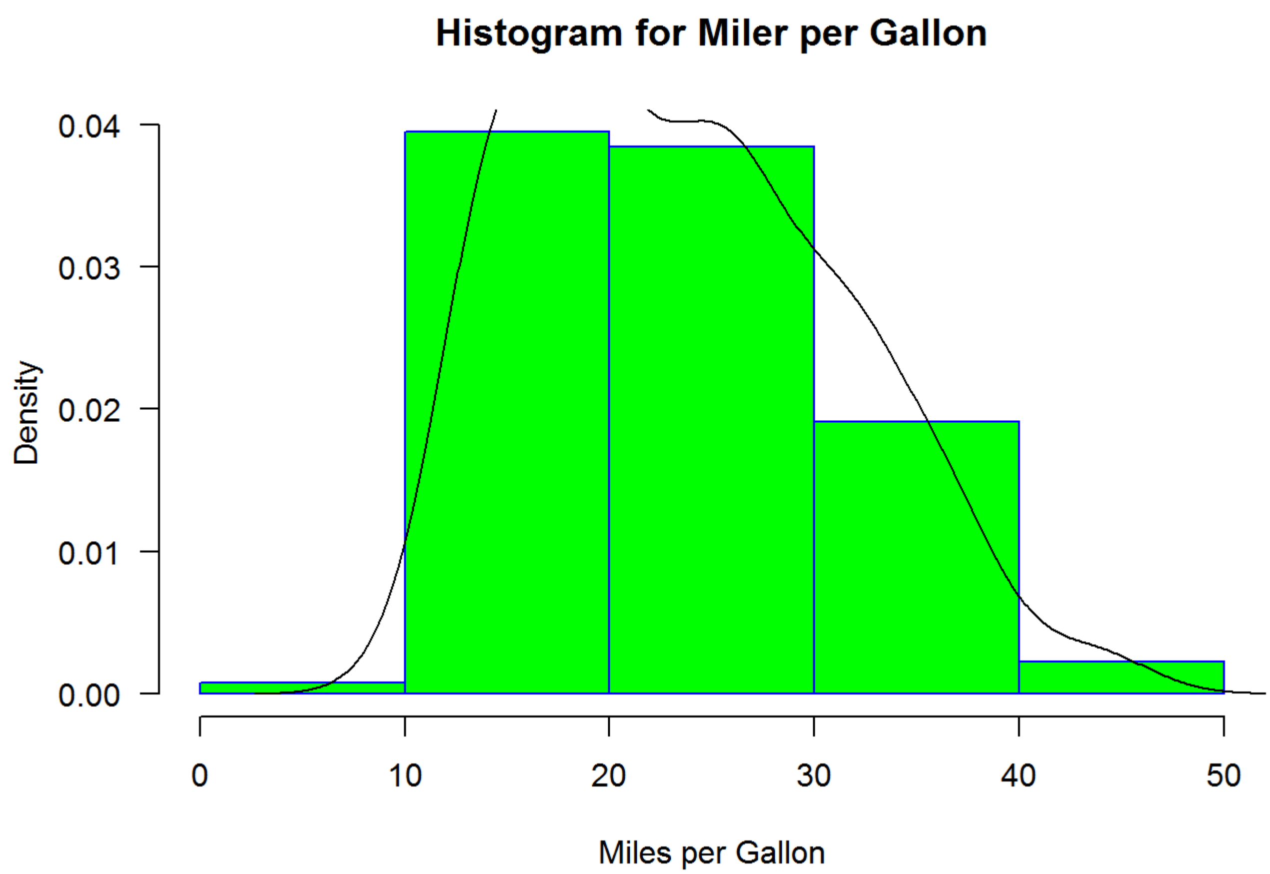

```
boxplot(mpg ~ origin, data = train_data, xlab = "Origin",ylab = "Miles Per Gallon", main = "Mile
age Data")
```




```
#correlation plot
corrplot(cor(train_data, use = 'pairwise.complete.obs', method = 'pearson'))
```




```
#histogram
hist(train_data$mpg, main="Histogram for Miler per Gallon", xlab="Miles per Gallon", border="blu
e", col="green", las=1, breaks=5, freq = FALSE)
lines(density(train_data$mpg))
```




```
# chi-sq test
chisq.test(train_data$displacement, train_data$cylinders, correct=FALSE)
```

```
## Warning in chisq.test(train_data$displacement, train_data$cylinders,
## correct = FALSE): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  train_data$displacement and train_data$cylinders
## X-squared = 1451.9, df = 324, p-value < 2.2e-16
```

```
chisq.test(train_data)
```

```
## Warning in chisq.test(train_data): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  train_data
## X-squared = 53850, df = 3176, p-value < 2.2e-16
```

```
#t test
t.test(train_data$displacement, train_data$cylinders, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  train_data$displacement and train_data$cylinders
## t = 36.531, df = 397, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  177.8551 198.0871
## sample estimates:
## mean of the differences
##                187.9711
```

```
t.test(train_data$displacement, train_data$cylinders)
```

```
##
##  Welch Two Sample t-test
##
## data:  train_data$displacement and train_data$cylinders
## t = 35.96, df = 397.21, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  177.6945 198.2477
## sample estimates:
##  mean of x  mean of y
## 193.425879   5.454774
```

```
#t.test(train_data$mpg~train_data$origin)
```