Fuel Economy Analysis

Jyoti Joshi

10/28/2018

Introduction

The dataset used in this analysis comes from https://www.fueleconomy.gov/feg/download.shtml. The particular file is https://www.fueleconomy.gov/feg/epadata/vehicles.csv.zip "Datasets for All Model Years (1984–2019)".The data dictionary is here: https://www.fueleconomy.gov/feg/ws/index.shtml#vehicle

In this analysis, we are trying to find out which manufacturer produces the most efficient fleet of cars. Also, looking for some interesting trends or insights like how fuel economy changed over time.

Load Data

```r
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```r
library(reshape2)
library(glm2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(ggplot2)
library(DataExplorer)
library(xtable)
library(car)
```

```
## Loading required package: carData
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# library(randomForest)

setwd("/Users/jyoti/Project1/")
```

```
filepath <- "/Users/jyoti/Project1/vehicles.csv"
vehicles_data = read.csv(filepath)
```

Exploratory Analysis

Explore the raw data by checking number of rows, columns, printing and plotting some content of variables and finding is there are any missing values.

```
print(paste0(paste0("Number of rows of the raw dataset: ", nrow(vehicles_data)), paste0(" Number of colu
```

```
## [1] "Number of rows of the raw dataset: 40081 Number of columns of raw dataset: 83"
```

```
names(vehicles_data)
```

```
##  [1] "barrels08"       "barrelsA08"      "charge120"      
##  [4] "charge240"       "city08"          "city08U"        
##  [7] "cityA08"         "cityA08U"        "cityCD"         
## [10] "cityE"           "cityUF"          "co2"            
## [13] "co2A"            "co2TailpipeAGpm" "co2TailpipeGpm" 
## [16] "comb08"          "comb08U"         "combA08"        
## [19] "combA08U"        "combE"           "combinedCD"     
## [22] "combinedUF"      "cylinders"       "displ"          
## [25] "drive"           "engId"           "eng_dscr"       
## [28] "feScore"         "fuelCost08"      "fuelCostA08"    
## [31] "fuelType"        "fuelType1"       "ghgScore"       
## [34] "ghgScoreA"       "highway08"       "highway08U"     
## [37] "highwayA08"      "highwayA08U"     "highwayCD"      
## [40] "highwayE"        "highwayUF"       "hlv"            
## [43] "hpv"             "id"              "lv2"            
## [46] "lv4"             "make"            "model"          
## [49] "mpgData"         "phevBlended"     "pv2"            
## [52] "pv4"             "range"           "rangeCity"      
## [55] "rangeCityA"      "rangeHwy"        "rangeHwyA"      
## [58] "trany"           "UCity"           "UCityA"         
## [61] "UHighway"        "UHighwayA"       "VClass"         
## [64] "year"            "youSaveSpend"    "guzzler"        
## [67] "trans_dscr"      "tCharger"        "sCharger"       
## [70] "atvType"         "fuelType2"       "rangeA"         
## [73] "evMotor"         "mfrCode"         "c240Dscr"       
## [76] "charge240b"      "c240bDscr"       "createdOn"      
## [79] "modifiedOn"      "startStop"       "phevCity"       
## [82] "phevHwy"         "phevComb"
```

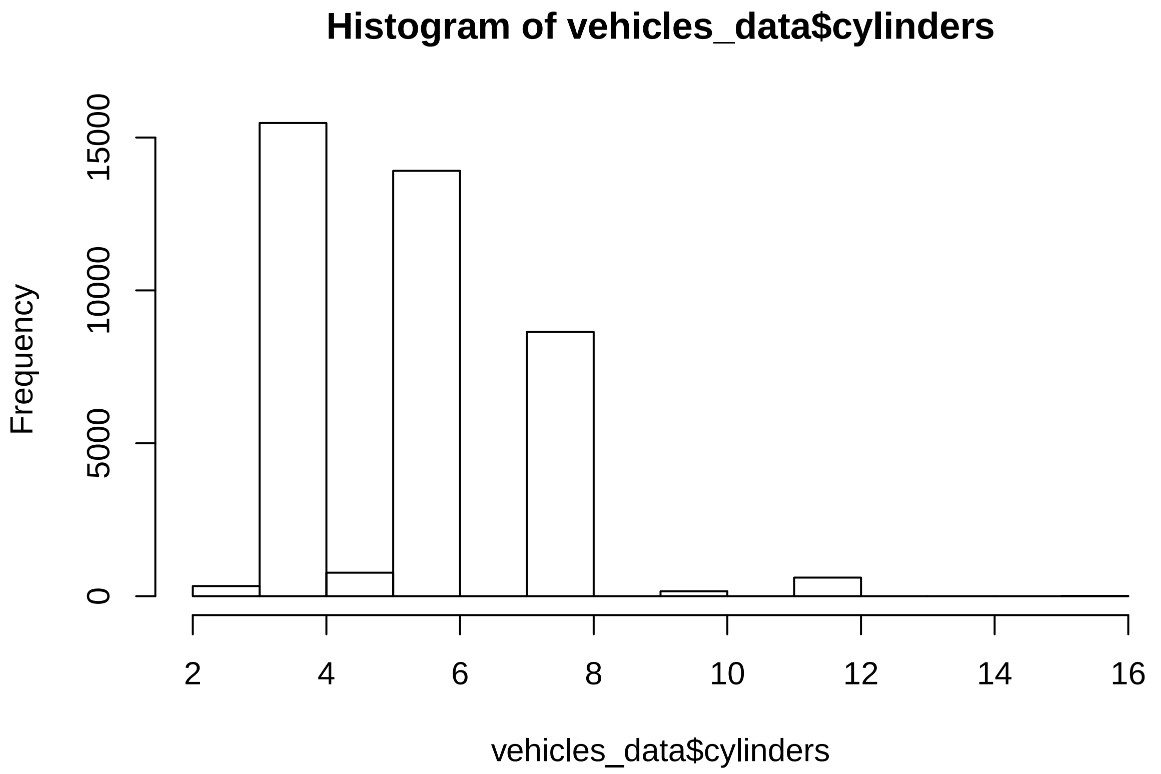
```
head(vehicles_data)
```

```
##   barrels08 barrelsA08 charge120 charge240 city08 city08U cityA08 cityA08U
## 1  15.69571          0         0         0     19       0       0        0
## 2  29.96455          0         0         0      9       0       0        0
## 3  12.20778          0         0         0     23       0       0        0
## 4  29.96455          0         0         0     10       0       0        0
## 5  17.34789          0         0         0     17       0       0        0
## 6  14.98227          0         0         0     21       0       0        0
##   cityCD cityE cityUF co2 co2A co2TailpipeAGpm co2TailpipeGpm comb08
## 1      0     0      0  -1   -1               0       423.1905     21
## 2      0     0      0  -1   -1               0       807.9091     11
## 3      0     0      0  -1   -1               0       329.1481     27
## 4      0     0      0  -1   -1               0       807.9091     11
## 5      0     0      0  -1   -1               0       467.7368     19
## 6      0     0      0  -1   -1               0       403.9545     22
##   comb08U combA08 combA08U combE combinedCD combinedUF cylinders displ
## 1       0       0        0     0          0          0         4   2.0
## 2       0       0        0     0          0          0        12   4.9
## 3       0       0        0     0          0          0         4   2.2
## 4       0       0        0     0          0          0         8   5.2
## 5       0       0        0     0          0          0         4   2.2
## 6       0       0        0     0          0          0         4   1.8
##                         drive engId   eng_dscr feScore fuelCost08
## 1           Rear-Wheel Drive  9011      (FFS)      -1       2000
## 2           Rear-Wheel Drive 22020  (GUZZLER)      -1       3850
## 3          Front-Wheel Drive  2100      (FFS)      -1       1550
## 4           Rear-Wheel Drive  2850                 -1       3850
## 5 4-Wheel or All-Wheel Drive 66031 (FFS,TRBO)      -1       2700
## 6          Front-Wheel Drive 66020      (FFS)      -1       1950
##   fuelCostA08 fuelType        fuelType1 ghgScore ghgScoreA highway08
## 1           0  Regular Regular Gasoline       -1        -1        25
## 2           0  Regular Regular Gasoline       -1        -1        14
## 3           0  Regular Regular Gasoline       -1        -1        33
## 4           0  Regular Regular Gasoline       -1        -1        12
## 5           0  Premium Premium Gasoline       -1        -1        23
## 6           0  Regular Regular Gasoline       -1        -1        24
##   highway08U highwayA08 highwayA08U highwayCD highwayE highwayUF hlv hpv
## 1          0          0           0         0        0         0   0   0
## 2          0          0           0         0        0         0   0   0
## 3          0          0           0         0        0         0  19  77
## 4          0          0           0         0        0         0   0   0
## 5          0          0           0         0        0         0   0   0
## 6          0          0           0         0        0         0   0   0
##      id lv2 lv4       make               model mpgData phevBlended pv2 pv4
## 1     1   0   0 Alfa Romeo  Spider Veloce 2000       Y       false   0   0
## 2    10   0   0    Ferrari          Testarossa       N       false   0   0
## 3   100   0   0      Dodge             Charger       Y       false   0   0
## 4  1000   0   0      Dodge B150/B250 Wagon 2WD       N       false   0   0
## 5 10000   0  14     Subaru    Legacy AWD Turbo       N       false   0  90
## 6 10001   0  15     Subaru              Loyale       N       false   0  88
##   range rangeCity rangeCityA rangeHwy rangeHwyA           trany   UCity
## 1     0         0          0        0         0    Manual 5-spd 23.3333
## 2     0         0          0        0         0    Manual 5-spd 11.0000
## 3     0         0          0        0         0    Manual 5-spd 29.0000
## 4     0         0          0        0         0 Automatic 3-spd 12.2222
## 5     0         0          0        0         0    Manual 5-spd 21.0000
## 6     0         0          0        0         0 Automatic 3-spd 27.0000
##   UCityA UHighway UHighwayA          VClass year youSaveSpend guzzler
## 1      0  35.0000         0     Two Seaters 1985        -2250
## 2      0  19.0000         0     Two Seaters 1985       -11500       T
## 3      0  47.0000         0 Subcompact Cars 1985            0
## 4      0  16.6667         0            Vans 1985       -11500
## 5      0  32.0000         0    Compact Cars 1993        -5750
## 6      0  33.0000         0    Compact Cars 1993        -2000
##   trans_dscr tCharger sCharger atvType fuelType2 rangeA evMotor mfrCode
## 1                  NA
## 2                  NA
## 3        SIL       NA
## 4                  NA
## 5                TRUE
## 6                  NA
##   c240Dscr charge240b c240bDscr                    createdOn
## 1                   0           Tue Jan 01 00:00:00 EST 2013
## 2                   0           Tue Jan 01 00:00:00 EST 2013
## 3                   0           Tue Jan 01 00:00:00 EST 2013
## 4                   0           Tue Jan 01 00:00:00 EST 2013
## 5                   0           Tue Jan 01 00:00:00 EST 2013
## 6                   0           Tue Jan 01 00:00:00 EST 2013
##                     modifiedOn startStop phevCity phevHwy phevComb
## 1 Tue Jan 01 00:00:00 EST 2013                  0       0        0
## 2 Tue Jan 01 00:00:00 EST 2013                  0       0        0
## 3 Tue Jan 01 00:00:00 EST 2013                  0       0        0
## 4 Tue Jan 01 00:00:00 EST 2013                  0       0        0
## 5 Tue Jan 01 00:00:00 EST 2013                  0       0        0
## 6 Tue Jan 01 00:00:00 EST 2013                  0       0        0
```

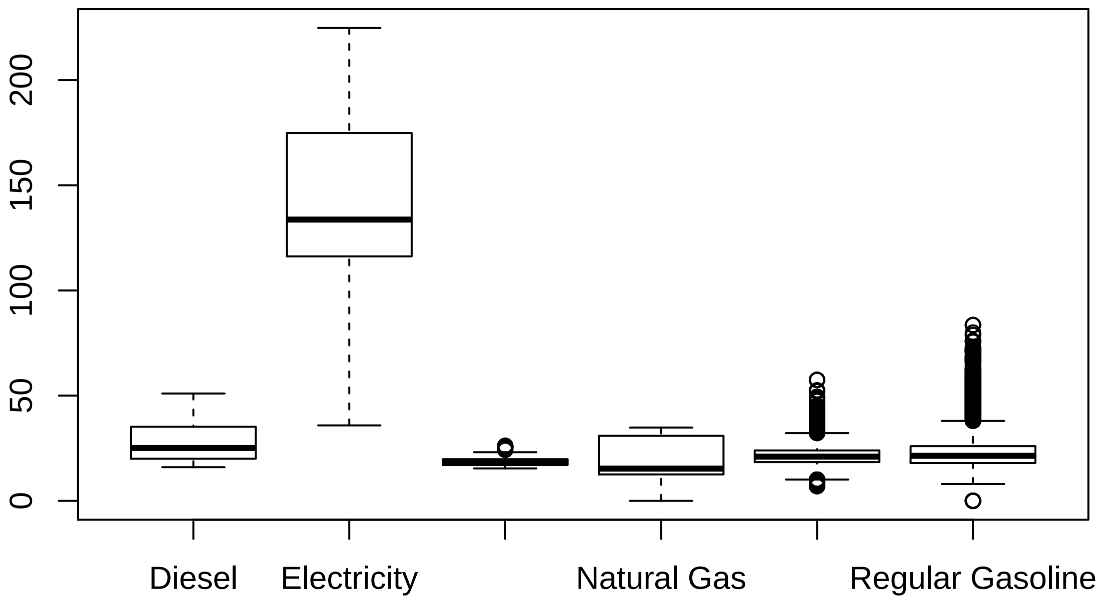
```
# str(vehicles_data)
# summary(vehicles_data)
glimpse(vehicles_data)
```

```
## Observations: 40,081
## Variables: 83
## $ barrels08       <dbl> 15.69571, 29.96455, 12.20778, 29.96455, 17.347...
## $ barrelsA08      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ charge120       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ charge240       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ city08          <int> 19, 9, 23, 10, 17, 21, 22, 23, 23, 23, 23, 18,...
## $ city08U         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ cityA08         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ cityA08U        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ cityCD          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ cityE           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ cityUF          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ co2             <int> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1...
## $ co2A            <int> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1...
## $ co2TailpipeAGpm <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ co2TailpipeGpm  <dbl> 423.1905, 807.9091, 329.1481, 807.9091, 467.73...
## $ comb08          <int> 21, 11, 27, 11, 19, 22, 25, 24, 26, 25, 26, 21...
## $ comb08U         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ combA08         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ combA08U        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ combE           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ combinedCD      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ combinedUF      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ cylinders       <int> 4, 12, 4, 8, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 8, ...
## $ displ           <dbl> 2.0, 4.9, 2.2, 5.2, 2.2, 1.8, 1.8, 1.6, 1.6, 1...
## $ drive           <fct> Rear-Wheel Drive, Rear-Wheel Drive, Front-Whee...
## $ engId           <int> 9011, 22020, 2100, 2850, 66031, 66020, 66020, ...
## $ eng_dscr        <fct> (FFS), (GUZZLER), (FFS), , (FFS,TRBO), (FFS), ...
## $ feScore         <int> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1...
## $ fuelCost08      <int> 2000, 3850, 1550, 3850, 2700, 1950, 1700, 1750...
## $ fuelCostA08     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ fuelType        <fct> Regular, Regular, Regular, Regular, Premium, R...
## $ fuelType1       <fct> Regular Gasoline, Regular Gasoline, Regular Ga...
## $ ghgScore        <int> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1...
## $ ghgScoreA       <int> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1...
## $ highway08       <int> 25, 14, 33, 12, 23, 24, 29, 26, 31, 30, 30, 26...
## $ highway08U      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ highwayA08      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ highwayA08U     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ highwayCD       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ highwayE        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ highwayUF       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ hlv             <int> 0, 0, 19, 0, 0, 0, 0, 0, 0, 0, 17, 17, 0, 0...
## $ hpv             <int> 0, 0, 77, 0, 0, 0, 0, 0, 0, 0, 88, 88, 0, 0...
## $ id              <int> 1, 10, 100, 1000, 10000, 10001, 10002, 10003, ...
## $ lv2             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ lv4             <int> 0, 0, 0, 0, 14, 15, 15, 13, 13, 13, 13, 0, 0, ...
## $ make            <fct> Alfa Romeo, Ferrari, Dodge, Dodge, Subaru, Sub...
## $ model           <fct> Spider Veloce 2000, Testarossa, Charger, B150/...
## $ mpgData         <fct> Y, N, Y, N, N, N, Y, Y, Y, Y, Y, N, Y, N, N, N...
## $ phevBlended     <fct> false, false, false, false, false, false, fals...
## $ pv2             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ pv4             <int> 0, 0, 0, 0, 90, 88, 88, 89, 89, 89, 89, 0, 0, ...
## $ range           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ rangeCity       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ rangeCityA      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ rangeHwy        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ rangeHwyA       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ trany           <fct> Manual 5-spd, Manual 5-spd, Manual 5-spd, Auto...
## $ UCity           <dbl> 23.3333, 11.0000, 29.0000, 12.2222, 21.0000, 2...
## $ UCityA          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ UHighway        <dbl> 35.0000, 19.0000, 47.0000, 16.6667, 32.0000, 3...
## $ UHighwayA       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ VClass          <fct> Two Seaters, Two Seaters, Subcompact Cars, Van...
## $ year            <int> 1985, 1985, 1985, 1985, 1993, 1993, 1993, 1993...
## $ youSaveSpend    <int> -2250, -11500, 0, -11500, -5750, -2000, -750, ...
## $ guzzler         <fct> , T, , , , , , , , , , , , , , , , , , , T, T,...
## $ trans_dscr      <fct> , , SIL, , , , , , , , , 2MODE CLKUP, , 2MODE ...
## $ tCharger        <lgl> NA, NA, NA, NA, TRUE, NA, NA, NA, NA, NA, NA, ...
## $ sCharger        <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ atvType         <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ fuelType2       <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ rangeA          <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ evMotor         <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ mfrCode         <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ c240Dscr        <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ charge240b      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ c240bDscr       <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ createdOn       <fct> Tue Jan 01 00:00:00 EST 2013, Tue Jan 01 00:00...
## $ modifiedOn      <fct> Tue Jan 01 00:00:00 EST 2013, Tue Jan 01 00:00...
## $ startStop       <fct> , , , , , , , , , , , , , , , , , , , , , , , ,
## $ phevCity        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ phevHwy         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ phevComb        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
```

```
hist(vehicles_data$cylinders)
```



```
plot(vehicles_data$fuelType1, vehicles_data$UCity)
```



```
#Find missing values in data
missing_values <- sapply(vehicles_data, function(x) sum(is.na(x)))
missing_values[missing_values >0]
```

```
## cylinders     displ  tCharger
##       171       169     33779
```

Subset and Clean Data

Looking at the data, the data dictionary and the problem statement, we can now select the features that we intuitively find relevant to the analysis. Below are the columns selected from dataset:

- "cylinders" - engine cylinders
- "displ" - engine displacement in liters
- "drive" - drive axle type
- "feScore" - EPA Fuel Economy Score (-1 = Not available)
- "make" - manufacturer (division)
- "trany" - transmission
- "fuelType1" - fuel type 1. For single fuel vehicles, this will be the only fuel. For dual fuel vehicles, this will be the conventional fuel
- "phevBlended" - if true, this vehicle operates on a blend of gasoline and electricity in charge depleting mode
- "VClass" - EPA vehicle size class
- "UCity" - unadjusted city MPG for fuelType1
- "year" - model year

```
required <- c('cylinders', 'displ', 'drive', 'feScore', 'make', 'trany', 'fuelType1', 'phevBlended', 'V

#Subset the data to get desired features
vehicles_desired <- vehicles_data[, (names(vehicles_data) %in% required)]
# names(vehicles_desired)

#get count of NAs in feScore column (with value = -1)
sum(vehicles_desired$feScore==-1)
```

```
## [1] 32027
```

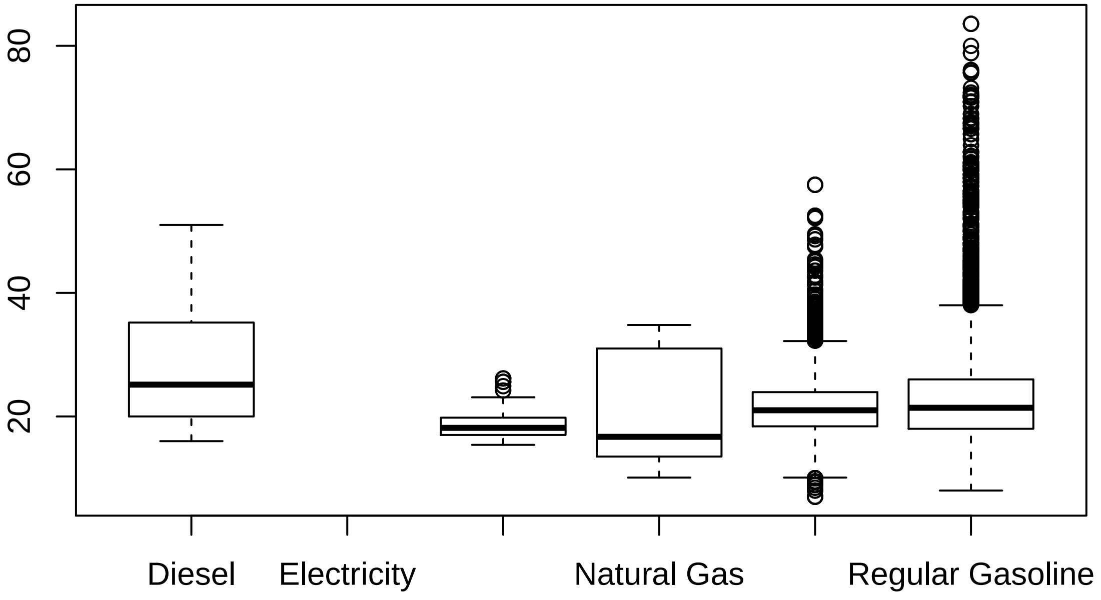
```
#drop feScore from features
vehicles_desired <- vehicles_desired[, !(names(vehicles_desired) %in% c("feScore"))]

#get rid of NAs from Cylinders and displ columns by dropping rows (since the NAs are few)
vehicles_desired <- vehicles_desired[complete.cases(vehicles_desired),] #This also gets rid of Electric

#Treating 0 UCity as bad data
sum(vehicles_desired$UCity==0)
```

```
## [1] 25
```

```
#Drop rows with 0 Ucity
vehicles_desired <- vehicles_desired[!vehicles_desired$UCity==0,]

summary(vehicles_desired)
```

```
##    cylinders          displ                               drive
##  Min.   : 2.000   Min.   :0.6   Front-Wheel Drive         :13876
##  1st Qu.: 4.000   1st Qu.:2.2   Rear-Wheel Drive          :13475
##  Median : 6.000   Median :3.0   4-Wheel or All-Wheel Drive: 6642
##  Mean   : 5.721   Mean   :3.3   All-Wheel Drive           : 2675
##  3rd Qu.: 6.000   3rd Qu.:4.3   4-Wheel Drive             : 1325
##  Max.   :16.000   Max.   :8.4                             : 1181
##                                 (Other)                   :  711
##              fuelType1            make        phevBlended
##  Diesel           : 1142   Chevrolet: 3933   false:39809
##  Electricity      :    0   Ford     : 3257   true :   76
##  Midgrade Gasoline:  100   Dodge    : 2555
##  Natural Gas      :   54   GMC      : 2465
##  Premium Gasoline :11267   Toyota   : 2003
##  Regular Gasoline :27322   BMW      : 1848
##                            (Other)  :23824
##              trany            UCity
##  Automatic 4-spd:11021   Min.   : 7.00
##  Manual 5-spd   : 8348   1st Qu.:18.10
##  Automatic 3-spd: 3150   Median :21.20
##  Automatic (S6) : 2984   Mean   :22.50
##  Manual 6-spd   : 2671   3rd Qu.:25.59
##  Automatic 5-spd: 2198   Max.   :83.56
##  (Other)        : 9513
##                          VClass           year
##  Compact Cars                : 5738   Min.   :1984
##  Subcompact Cars             : 5016   1st Qu.:1991
##  Midsize Cars                : 4675   Median :2002
##  Standard Pickup Trucks      : 2354   Mean   :2001
##  Sport Utility Vehicle - 4WD: 2090   3rd Qu.:2011
##  Large Cars                  : 2032   Max.   :2019
##  (Other)                     :17980
```

```
plot(vehicles_desired$fuelType1, vehicles_desired$UCity)
```



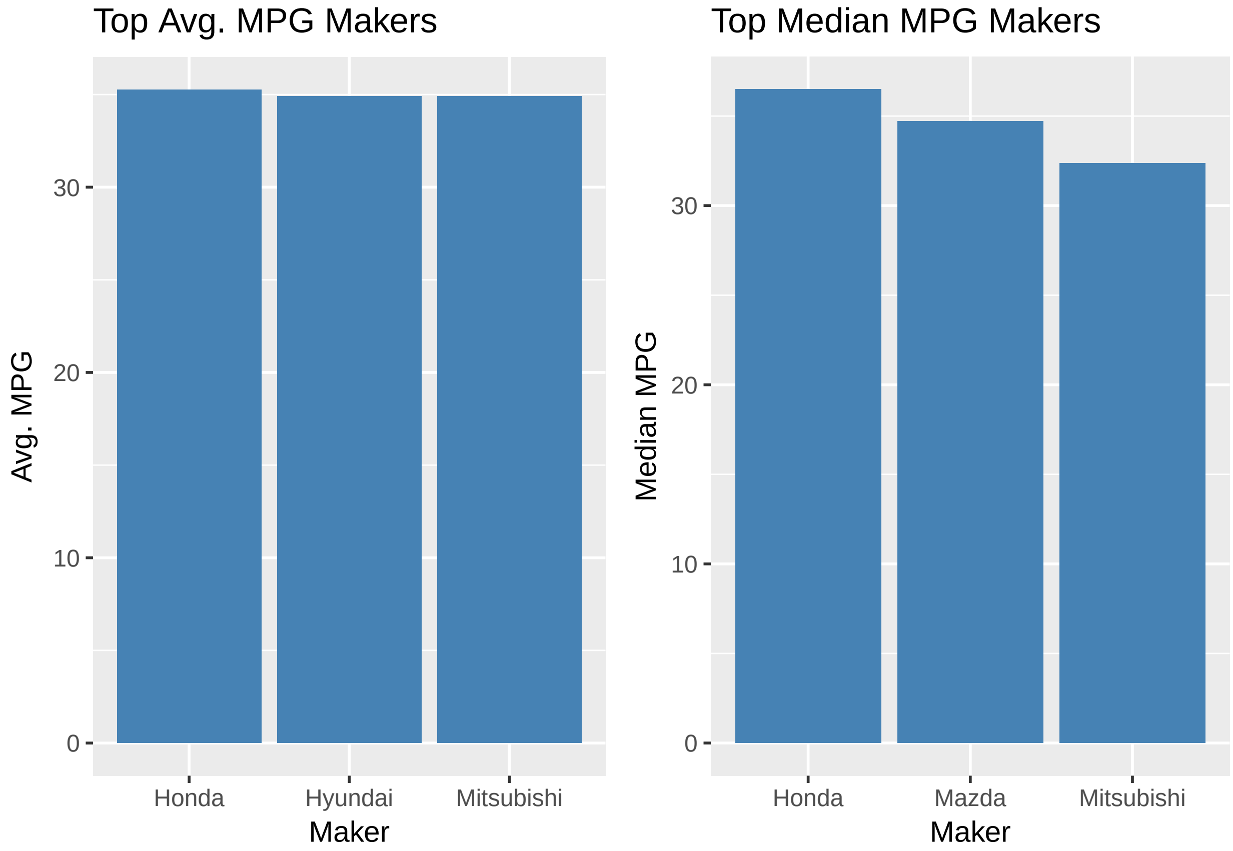
Finding manufacturer with most fuel efficient fleet

```
#Helper function for plotting multiple plots together
multiplot <- function(..., plotlist=NULL, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                    ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

To compare fuel efficiency among different car manufacturers, it only makes sense to do so over a particular model year. The following comparison thus is done for 2018 model year. Also, our data cleaning step excludes the 'Electricity' fueltype as it does not make sense to compare electric vehicle with other fuel type vehicles for MPG.

```
#Subsetting for year 2018
vehicles_2018 <- vehicles_desired[vehicles_desired$year==2018,]

# mean(vehicles_2018$UCity)
mean_mpg_per_make <- aggregate(vehicles_2018$UCity, list(vehicles_2018$make), mean)
median_mpg_per_make <- aggregate(vehicles_2018$UCity, list(vehicles_2018$make), median)

top_mean <- mean_mpg_per_make[order(mean_mpg_per_make$x,decreasing=T)[1:3],]
top_median <- median_mpg_per_make[order(median_mpg_per_make$x,decreasing=T)[1:3],]

barplot1 <- ggplot(data=top_mean, aes(x=Group.1, y=x)) +
  geom_bar(stat="identity", fill="steelblue") +
  xlab("Maker") +
  ylab("Avg. MPG") +
  ggtitle("Top Avg. MPG Makers")
# print(barplot1)
```

```
barplot2 <- ggplot(data=top_median, aes(x=Group.1, y=x)) +
  geom_bar(stat="identity", fill="steelblue") +
  xlab("Maker") +
  ylab("Median MPG") +
  ggtitle("Top Median MPG Makers")
# print(barplot2)

multiplot(barplot1, barplot2, cols =2)
```



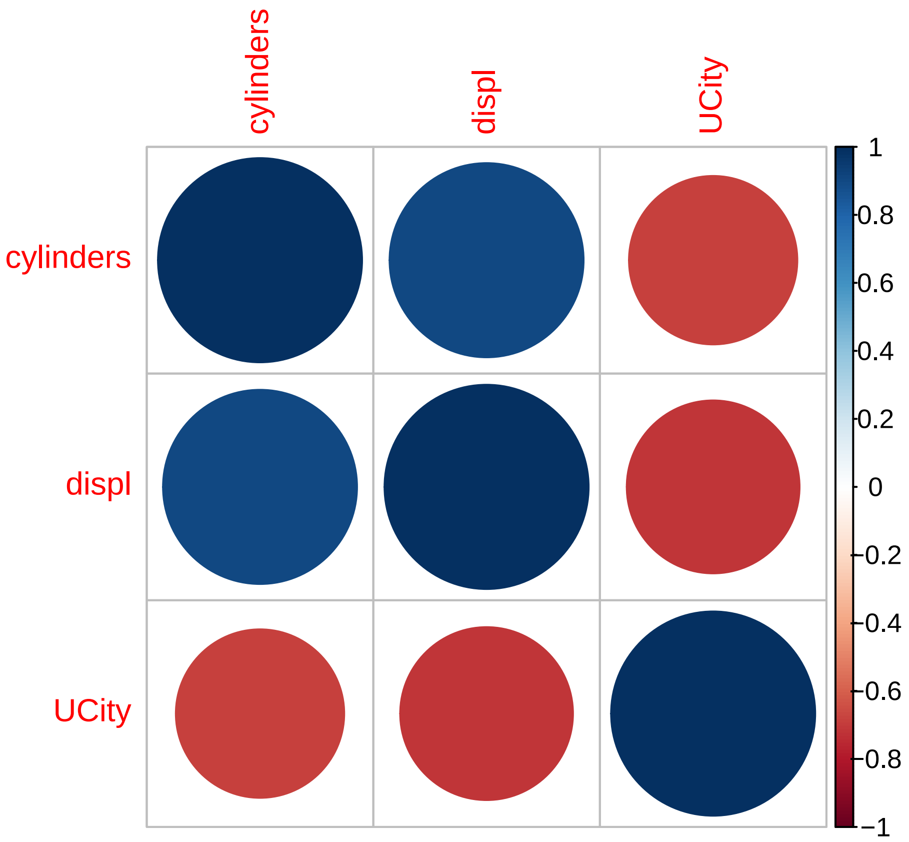
```
print(paste0(top_mean[1,1], paste0(" has the most fuel efficient fleet with average MPG as ", round(top_
```

```
## [1] "Honda has the most fuel efficient fleet with average MPG as 35.27"
```

Correlation Analysis

Checking for corelations between different variables.

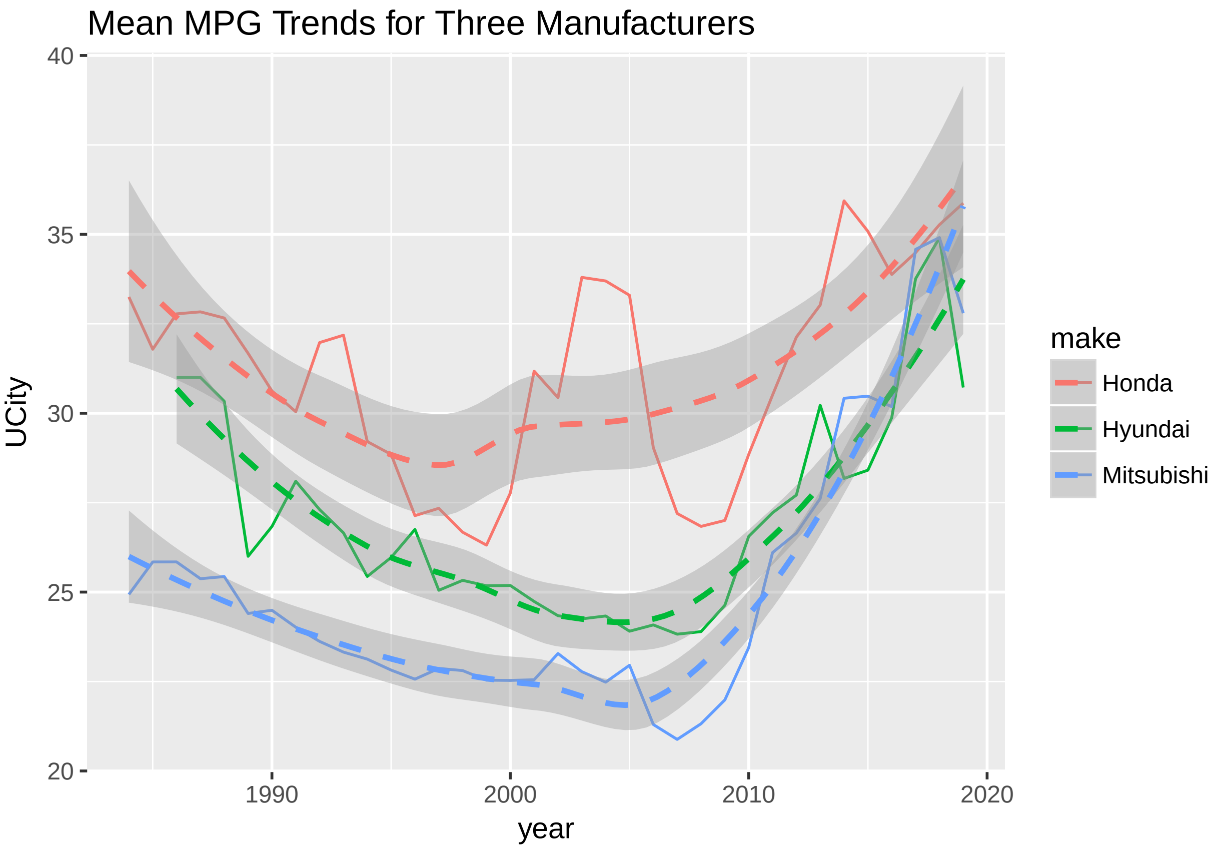
```
numeric_features <- c('cylinders', 'displ', 'UCity')
vehicles_numeric_data <- vehicles_desired[, names(vehicles_desired) %in% numeric_features]

correlations <- cor(vehicles_numeric_data)
corrplot(correlations)
```



The two numeric features, cylinders and displ, both are well correlated with target UCity (MPG) but also highly correlated with each other as seen from the above correlation graph. So, one of the two varibale might be required to be removed depending on Variance Inflation Factor of final model.

Other Trends

```
#Subsetting for only top 3 manufacturers as our previous analysis
vehicles_trends <- vehicles_desired[vehicles_desired$make %in% c("Honda", "Hyundai", "Mitsubishi"),]

mean_mpg <- aggregate( UCity ~ make + year, data = vehicles_trends, mean)
plot1 <- ggplot(data = mean_mpg, aes(x=year, y=UCity, colour = make)) +
  geom_line(aes(group = make)) +
  geom_smooth(method = 'loess', linetype = 2) +
  ggtitle("Mean MPG Trends for Three Manufacturers")
plot1
```



As seen from the graph, the average MPG is generally going up for the shown manufacturers since 2005, probably because of external factors like recent push from authorities to reduce fuel consumption or rising fuel costs. The future MPG thus should be higher than current values. One can further explore this with a time series model.