# Regression

Code

Sunni Magan

03/07/2023

# Introduction

This notebook explores the relationship between fuel consumption and carbon dioxide emissions of retail cars in Canada

Source: https://www.kaggle.com/datasets/ahmettyilmazz/fuel-consumption (https://www.kaggle.com/datasets/ahmettyilmazz/fuel-consumption)

## About The Data

### YEAR

- 2000 - 2022

### MAKE

- 52 well known car brands

### MODEL

- 4WD/4X4 = Four-wheel drive
- AWD = All-wheel drive
- CNG = Compressed natural gas
- FFV = Flexible-fuel vehicle
- NGV = Natural gas vehicle
- # = High output engine that provides more power than the standard engine of the same size

### VEHICLE.CLASS

- compact
- full-size
- pickup truck - standard
- pickup truck - small
- mid-size
- minicompact
- minivan
- special purpose vehicle
- station wagon - mid-size
- station wagon - small
- subcompact
- suv
- suv - small
- suv - standard
- two-seater
- van - cargo
- van - passenger

## ENGINE.SIZE

- Cylinder volume of engine in liters

## CYLINDERS

- # of cylinders the engine has

## TRANSMISSION

- A = Automatic
- AM = Automated manual
- AS = Automatic with select shift
- AV = Continuously variable
- M = Manual
- 3 - 10 = Number of gears

## FUEL

- X = Regular gasoline
- Z = Premium gasoline
- D = Diesel
- E = Ethanol (E85)
- N = Natural Gas

## FUEL.CONSUMPTION

### HWY.LP100KM

- Highway fueld consumption in L/100km

### COMB.LP100KM

- Combined city/highway fuel consumption in L/100km

### COMB.MPG

- Combined city/highway fuel consumption in mpg

## EMISSIONS

- Estimated tailpipe carbon dioxide emissions in g/km

Hide

```
df <- read.csv("Fuel_Consumption.csv")
str(df)
```

```
'data.frame':   22556 obs. of  13 variables:
 $ YEAR         : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
 $ MAKE         : chr  "ACURA" "ACURA" "ACURA" "ACURA" ...
 $ MODEL        : chr  "1.6EL" "1.6EL" "3.2TL" "3.5RL" ...
 $ VEHICLE.CLASS: chr  "COMPACT" "COMPACT" "MID-SIZE" "MID-SIZE" ...
 $ ENGINE.SIZE  : num  1.6 1.6 3.2 3.5 1.8 1.8 1.8 3 3.2 1.8 ...
 $ CYLINDERS    : int  4 4 6 6 4 4 4 6 6 4 ...
 $ TRANSMISSION : chr  "A4" "M5" "AS5" "A4" ...
 $ FUEL         : chr  "X" "X" "Z" "Z" ...
 $ CITY.LP100KM : num  9.2 8.5 12.2 13.4 10 9.3 9.4 13.6 13.8 11.4 ...
 $ HWY.LP100KM  : num  6.7 6.5 7.4 9.2 7 6.8 7 9.2 9.1 7.2 ...
 $ COMB.LP100KM : num  8.1 7.6 10 11.5 8.6 8.2 8.3 11.6 11.7 9.5 ...
 $ COMB.MPG     : int  35 37 28 25 33 34 34 24 24 30 ...
 $ EMISSIONS    : int  186 175 230 264 198 189 191 267 269 218 ...
```

Hide

```
#getwd()
```

# Data Cleaning

Hide

```
df$MAKE <- tolower(df$MAKE)
df$MAKE <- as.factor(df$MAKE)

df$MODEL <- tolower(df$MODEL)
df$MODEL <- as.factor(df$MODEL)

df$VEHICLE.CLASS <- tolower(df$VEHICLE.CLASS)
df$VEHICLE.CLASS <- gsub(":", " -", df$VEHICLE.CLASS)
df$VEHICLE.CLASS <- as.factor(df$VEHICLE.CLASS)

df$TRANSMISSION <- as.factor(df$TRANSMISSION)
df$FUEL <- as.factor(df$FUEL)
```

Before the qualitative data can be converted to factors, it needs to be cleaned. Some of the features have inconsistent capitalization or punctuation which needs to be addressed.

Hide

```
str(df)
```

```
'data.frame':	22556 obs. of  13 variables:
 $ YEAR         : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
 $ MAKE         : Factor w/ 52 levels "acura","alfa romeo",..: 1 1 1 1 1 1 1 1 1 4 ...
 $ MODEL        : Factor w/ 3730 levels "1 series m coupe",..: 2 2 55 56 1914 1914 1916 2390 239
0 372 ...
 $ VEHICLE.CLASS: Factor w/ 17 levels "compact","full-size",..: 1 1 3 3 11 11 11 11 11 1 ...
 $ ENGINE.SIZE  : num  1.6 1.6 3.2 3.5 1.8 1.8 1.8 3 3.2 1.8 ...
 $ CYLINDERS    : int  4 4 6 6 4 4 4 6 6 4 ...
 $ TRANSMISSION : Factor w/ 30 levels "A10","A3","A4",..: 3 28 16 3 3 28 28 15 29 4 ...
 $ FUEL         : Factor w/ 5 levels "D","E","N","X",..: 4 4 5 5 4 4 5 5 5 5 ...
 $ CITY.LP100KM : num  9.2 8.5 12.2 13.4 10 9.3 9.4 13.6 13.8 11.4 ...
 $ HWY.LP100KM  : num  6.7 6.5 7.4 9.2 7 6.8 7 9.2 9.1 7.2 ...
 $ COMB.LP100KM : num  8.1 7.6 10 11.5 8.6 8.2 8.3 11.6 11.7 9.5 ...
 $ COMB.MPG     : int  35 37 28 25 33 34 34 24 24 30 ...
 $ EMISSIONS    : int  186 175 230 264 198 189 191 267 269 218 ...
```

# Train/Test Split

Hide

```
set.seed(1234)
i <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Performing an 80/20 split on the data to create training and testing sets

# Data Exploration

Now that the data is formatted in more appropriate datatypes, we can now explore the data to find relationships

## Data Summary

Hide

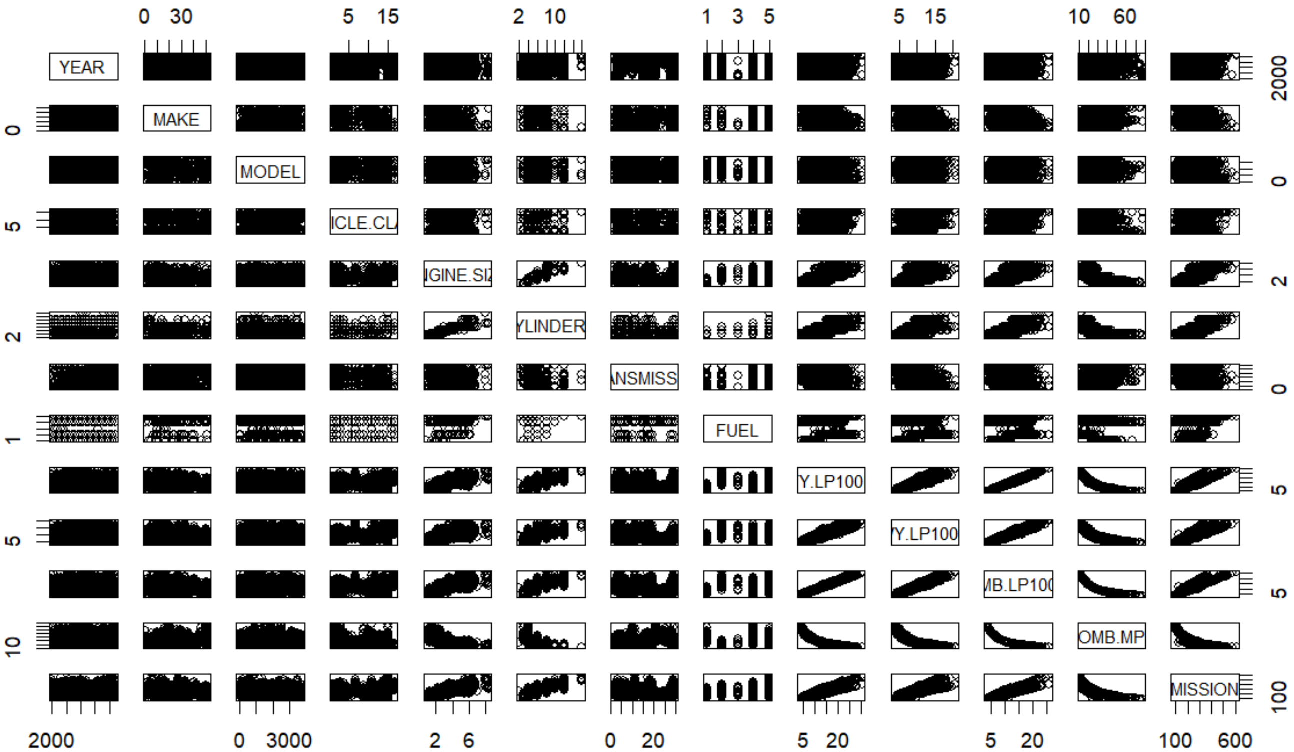
```
summary(train)
```

```
      YEAR                 MAKE                  MODEL                        VEHICLE.CLASS   ENGI
NE.SIZE      CYLINDERS       TRANSMISSION  FUEL      CITY.LP100KM
 Min.   :2000   chevrolet    : 1724   mustang      :   85   compact                 :2507   Min.
:0.800   Min.   : 2.000   A4     :2821   D: 250   Min.   : 3.50
 1st Qu.:2006   ford         : 1381   jetta        :   76   mid-size                :2347   1st Q
u.:2.300   1st Qu.: 4.000   AS6    :2264   E: 868   1st Qu.:10.40
 Median :2012   bmw          : 1192   sierra       :   62   suv                     :2113   Media
n :3.000   Median : 6.000   M6     :2069   N:  29   Median :12.30
 Mean   :2012   gmc          : 1072   silverado    :   61   pickup truck - standard:1763   Mean
:3.353   Mean   : 5.847   M5     :1685   X:9525   Mean   :12.75
 3rd Qu.:2017   mercedes-benz: 1006   silverado 4wd:   61   subcompact              :1617   3rd Q
u.:4.200   3rd Qu.: 8.000   A6     :1582   Z:7372   3rd Qu.:14.70
 Max.   :2022   toyota       :  776   sentra       :   59   suv - small             :1445   Max.
:8.400   Max.   :16.000   AS8    :1380            Max.   :30.60
                (Other)      :10893   (Other)      :17640   (Other)                 :6252
(Other):6243
  HWY.LP100KM      COMB.LP100KM      COMB.MPG      EMISSIONS
 Min.   : 3.200   Min.   : 3.60   Min.   :11.0   Min.   : 83.0
 1st Qu.: 7.300   1st Qu.: 9.10   1st Qu.:22.0   1st Qu.:209.0
 Median : 8.400   Median :10.50   Median :27.0   Median :242.0
 Mean   : 8.914   Mean   :11.03   Mean   :27.4   Mean   :249.8
 3rd Qu.:10.200   3rd Qu.:12.70   3rd Qu.:31.0   3rd Qu.:288.0
 Max.   :20.900   Max.   :26.10   Max.   :78.0   Max.   :608.0
```

# Finding Correlations

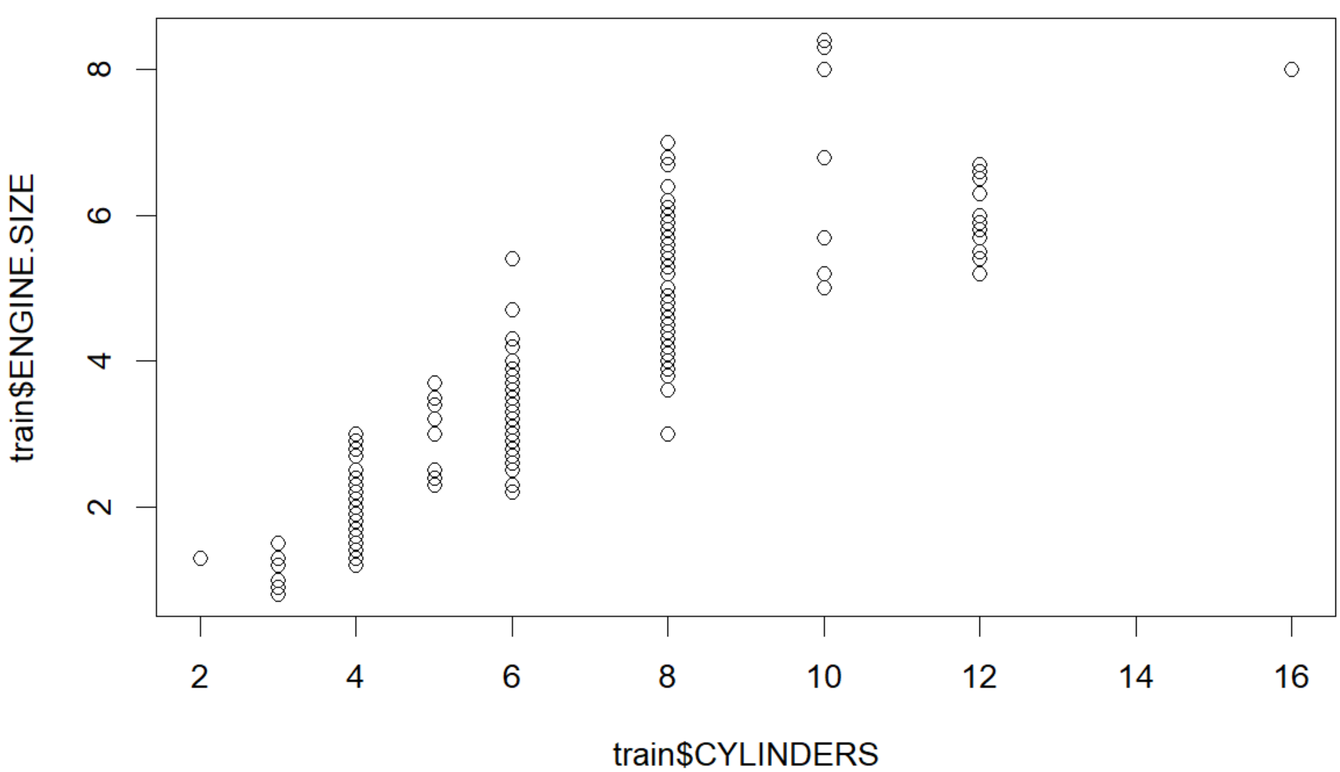
Hide

```
pairs(train)
```




Hide

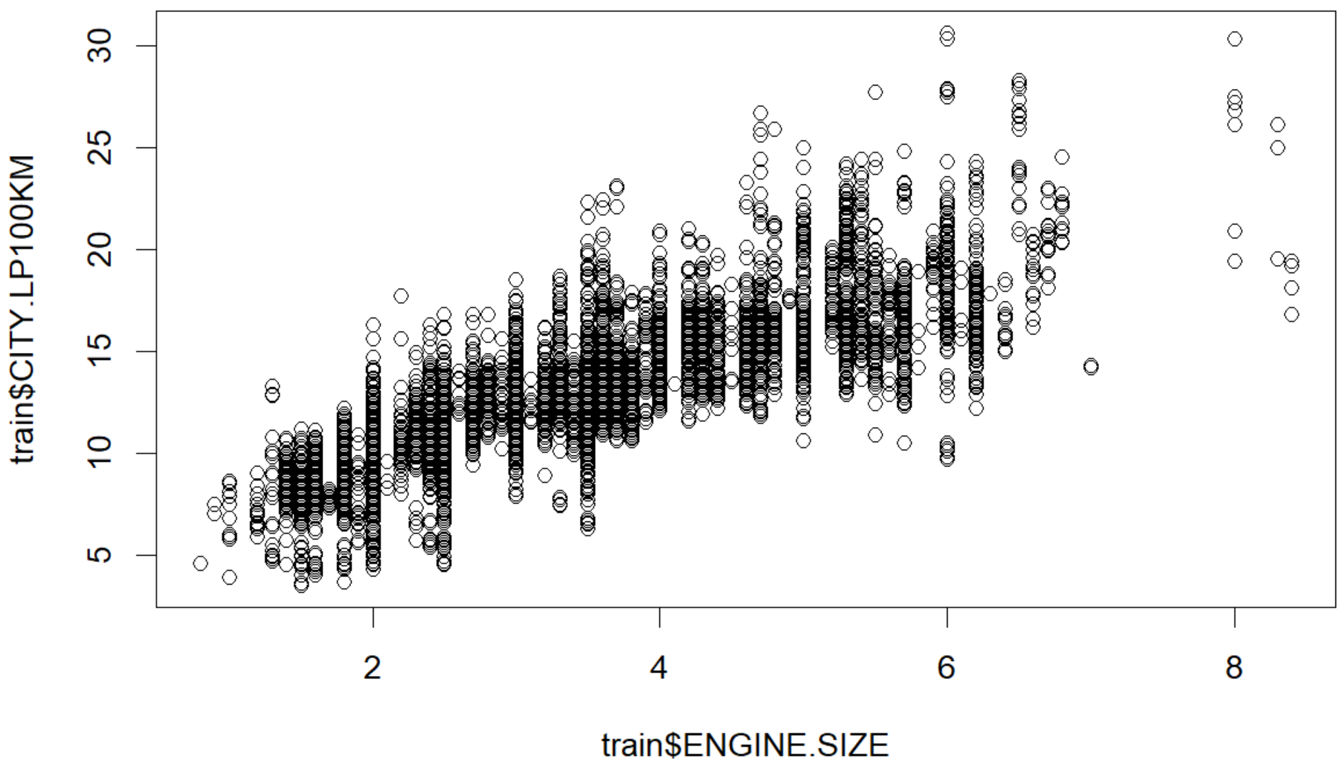
```
plot(train$CYLINDERS, train$ENGINE.SIZE)
```



There are trivial correlation such as engine size and number of cylinders which we will ignore.

Hide

```
plot(train$ENGINE.SIZE, train$CITY.LP100KM)
```




There are also slight linear relationships between engine size and fuel consumption, however there is a lot of noise.

Hide

```
cor(train[,9:13])
```

```
             CITY.LP100KM HWY.LP100KM COMB.LP100KM   COMB.MPG  EMISSIONS
CITY.LP100KM    1.0000000   0.9427948    0.9929838 -0.9210320  0.9187051
HWY.LP100KM     0.9427948   1.0000000    0.9752563 -0.8842794  0.8949005
COMB.LP100KM    0.9929838   0.9752563    1.0000000 -0.9204309  0.9226617
COMB.MPG       -0.9210320  -0.8842794   -0.9204309  1.0000000 -0.9006844
EMISSIONS       0.9187051   0.8949005    0.9226617 -0.9006844  1.0000000
```

There does seem to be some strong linear relationships between fuel consumption and emission

Hide

```
par(mfrow = c(2,3))
plot(train$COMB.LP100KM, train$EMISSIONS)
plot(train$COMB.LP100KM[train$FUEL == 'D'], train$EMISSIONS[train$FUEL == 'D'], main = "Fuel Typ
e D")
```

Hide

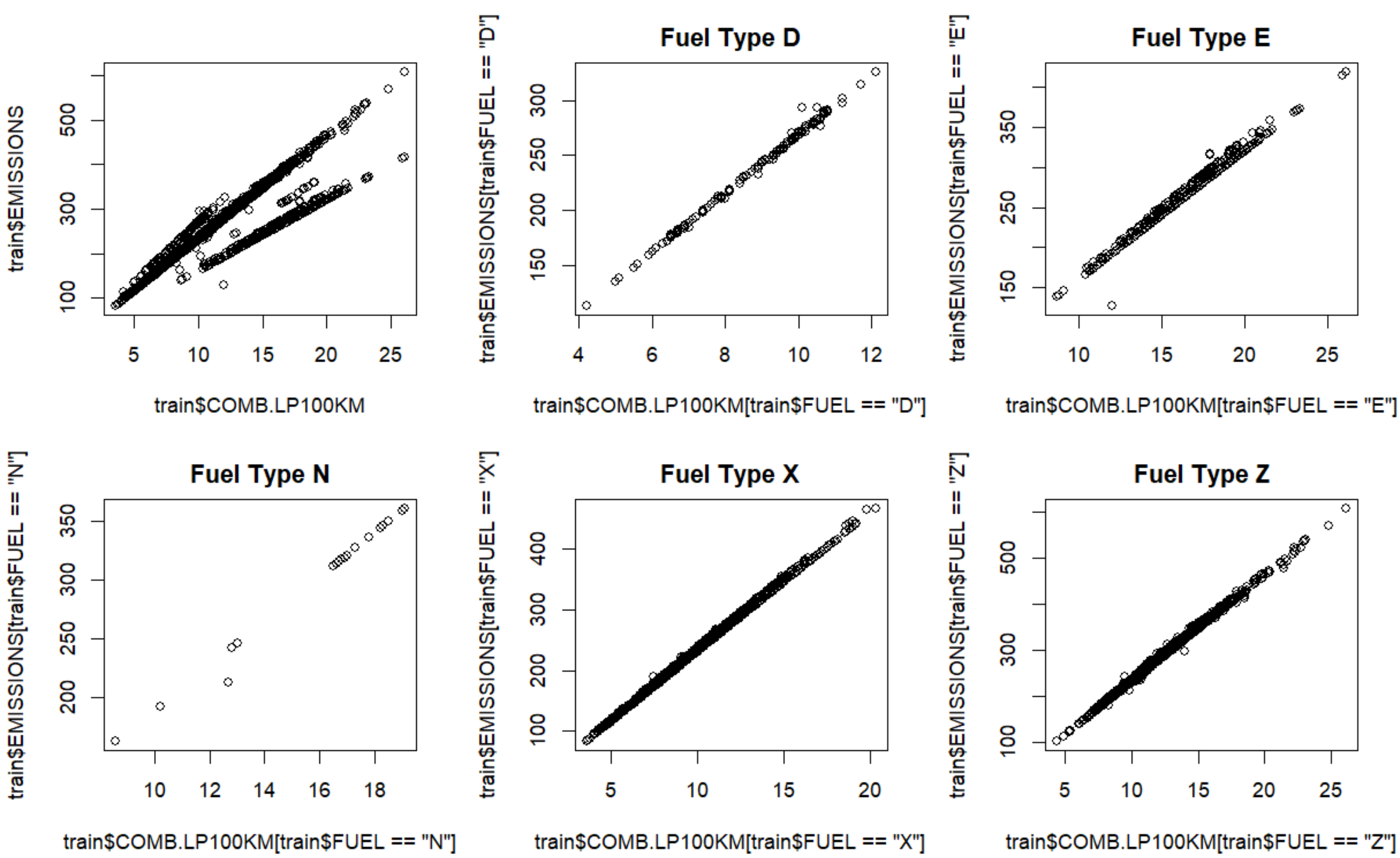
```
plot(train$COMB.LP100KM[train$FUEL == 'E'], train$EMISSIONS[train$FUEL == 'E'], main = "Fuel Typ
e E")
plot(train$COMB.LP100KM[train$FUEL == 'N'], train$EMISSIONS[train$FUEL == 'N'], main = "Fuel Typ
e N")
```

Hide

```
plot(train$COMB.LP100KM[train$FUEL == 'X'], train$EMISSIONS[train$FUEL == 'X'], main = "Fuel Typ
e X")
plot(train$COMB.LP100KM[train$FUEL == 'Z'], train$EMISSIONS[train$FUEL == 'Z'], main = "Fuel Typ
e Z")
```

Hide

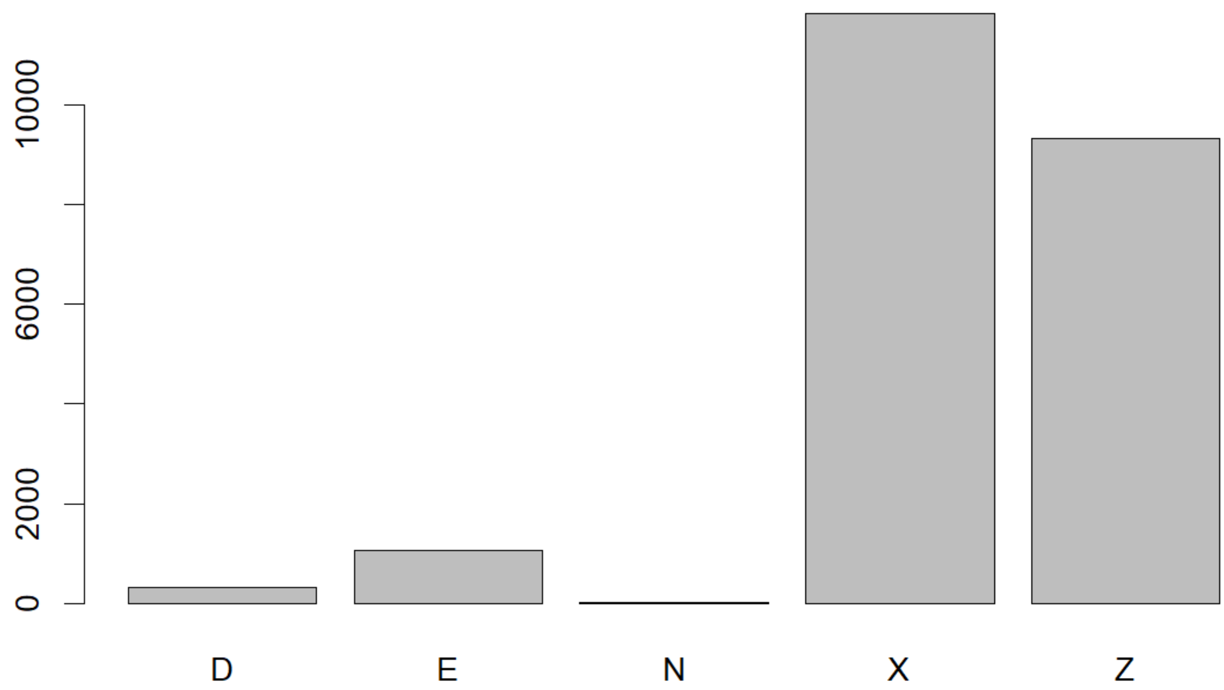
```
par(mfrow = c(1,1))
```




As we can see, there are multiple separate linear relationships between fuel consumption and CO2 emissions. This may mean one of the other features may be affecting this relationship. The separate relationships appear to be discrete. There are about 5 distinct linear relationship, which also correspond with the 5 fuel types.

Hide

```
barplot(table(df$FUEL))
```




Hide

```
summary(df$train)
```

```
Length  Class   Mode
     0   NULL   NULL
```

Based on the distribution of the fuel type on the data set, it seems like a better idea to to create a linear model for each fuel type. For this notebook, let's focus only on the most common fuel types, regular (X). This means we will have to re-sample our testing and training data to get an 80/20 split on the selected fuel type.

# Resampling Data

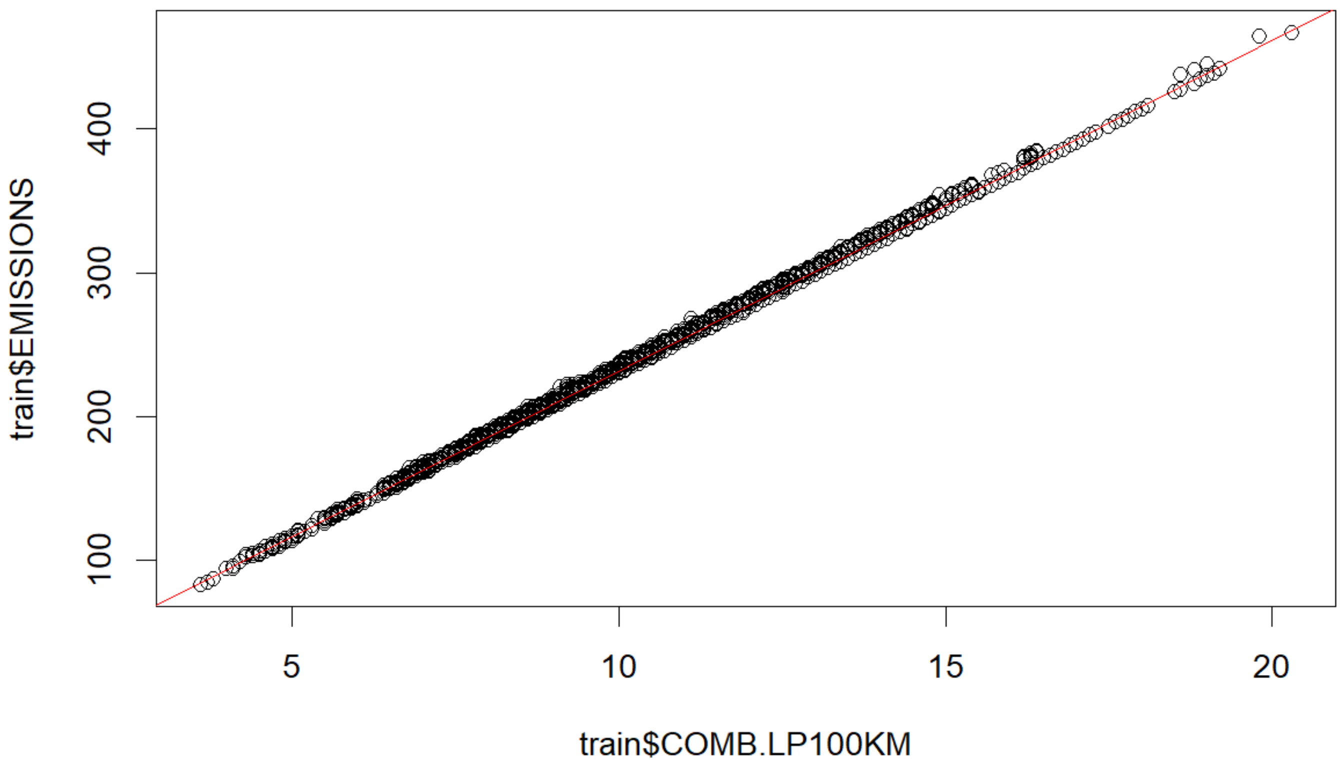
Hide

```
df <- df[df$FUEL == 'X',]
set.seed(1234)
i <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

# Linear Regression Model

Hide

```
plot(train$COMB.LP100KM, train$EMISSIONS)
model1 = lm(EMISSIONS~COMB.LP100KM, data=train)
abline(model1, col='red')
```




Hide

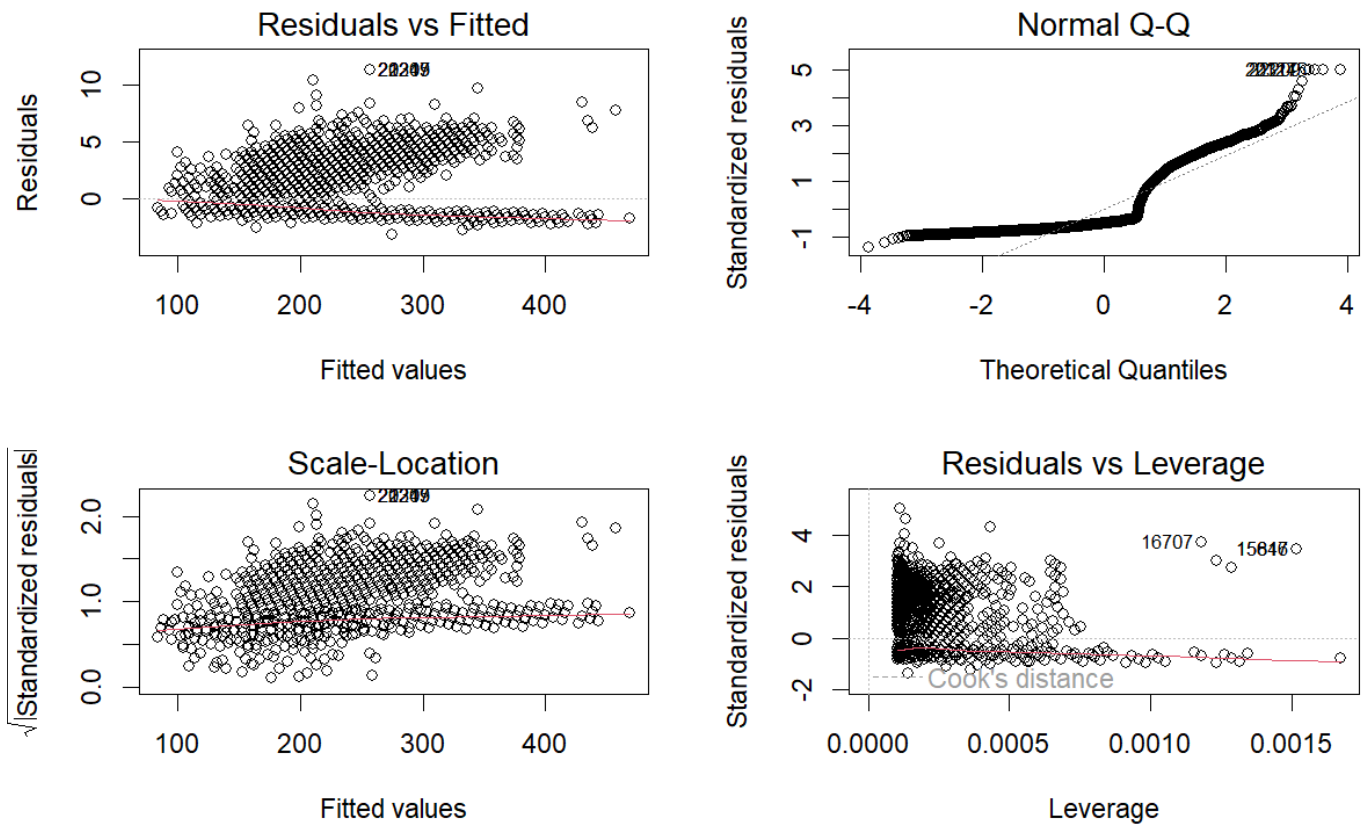
```
summary(model1)
```

```

Call:
lm(formula = EMISSIONS ~ COMB.LP100KM, data = train)

Residuals:
   Min     1Q Median     3Q    Max 
-3.096 -1.469 -1.113  1.480 11.345 

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept)   0.792301   0.096600    8.202 2.68e-16 ***
COMB.LP100KM 23.050740   0.009037 2550.774  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.267 on 9455 degrees of freedom
Multiple R-squared:  0.9985,	Adjusted R-squared:  0.9985 
F-statistic: 6.506e+06 on 1 and 9455 DF,  p-value: < 2.2e-16
```

The summary of this model shows promising results. First of all, the Std. Error values are low, which indicates a low variance in the estimate and its; actual value. Secondly, we have a '***' p-value. This means we have strong evidence to reject the null hypothesis and that our predictor and target variable are related. Thirdly, the t-value, which measures the amount of standard deviations away from zero our estimate is. In this case, the t-value is very high. Lastly, the Multiple R-squared statistic shows that more than 99% of the variance is explained by the predictor.

Hide

```
par(mfrow=c(2,2))
plot(model1)
```




These graphs are difficult to interpret. When looking at the Residual vs Fitted graph. The line seemed to be horizontal, however, the points do not seem to be even distributed on either size of the line. The Scaled-Location is similar in that the points are not even distributed on a horizontal line. This could mean there is more than just a linear relationship between the predictor and target variable. The Normal Q-Q Plot seems to have trouble following the diagonal line, especially in the extreme cases. By looking at the Residuals vs Leverage graph, it can be seen that all of the points are well withing Cook's distance since the boundary lines do not even appear on the graph.

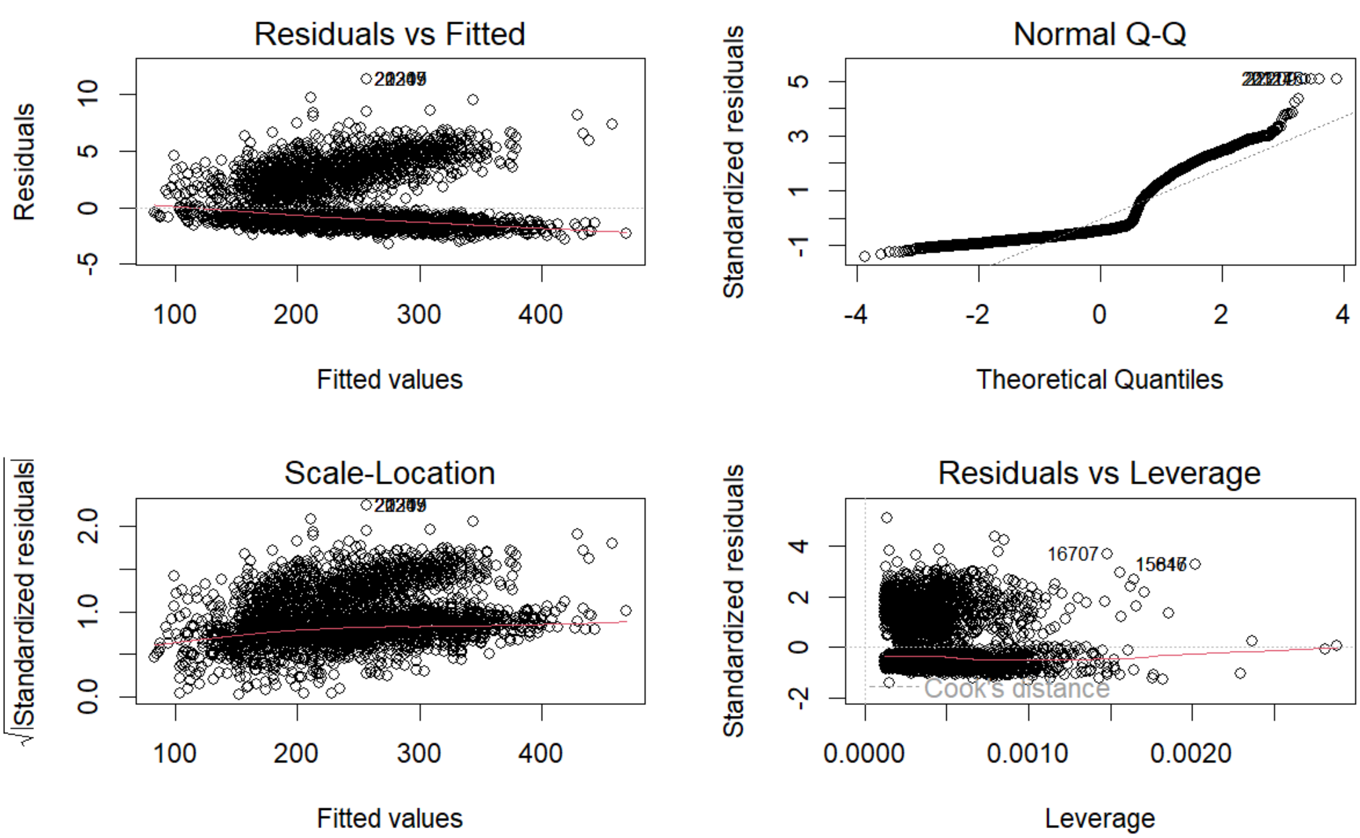
Hide

```
model2 <- lm(EMISSIONS~COMB.LP100KM+CYLINDERS+ENGINE.SIZE, data=train)
summary(model2)
```

```
Call:
lm(formula = EMISSIONS ~ COMB.LP100KM + CYLINDERS + ENGINE.SIZE,
    data = train)

Residuals:
   Min     1Q Median     3Q    Max
-3.190 -1.466 -1.044  1.366 11.359

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)   0.17182    0.13040    1.318    0.188
COMB.LP100KM 23.25441    0.01711 1359.417   <2e-16 ***
CYLINDERS     0.04153    0.04743    0.876    0.381
ENGINE.SIZE  -0.54405    0.06362   -8.552   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.243 on 9453 degrees of freedom
Multiple R-squared:  0.9986,	Adjusted R-squared:  0.9986
F-statistic: 2.214e+06 on 3 and 9453 DF,  p-value: < 2.2e-16
```

Hide

```
par(mfrow=c(2,2))
plot(model2)
```



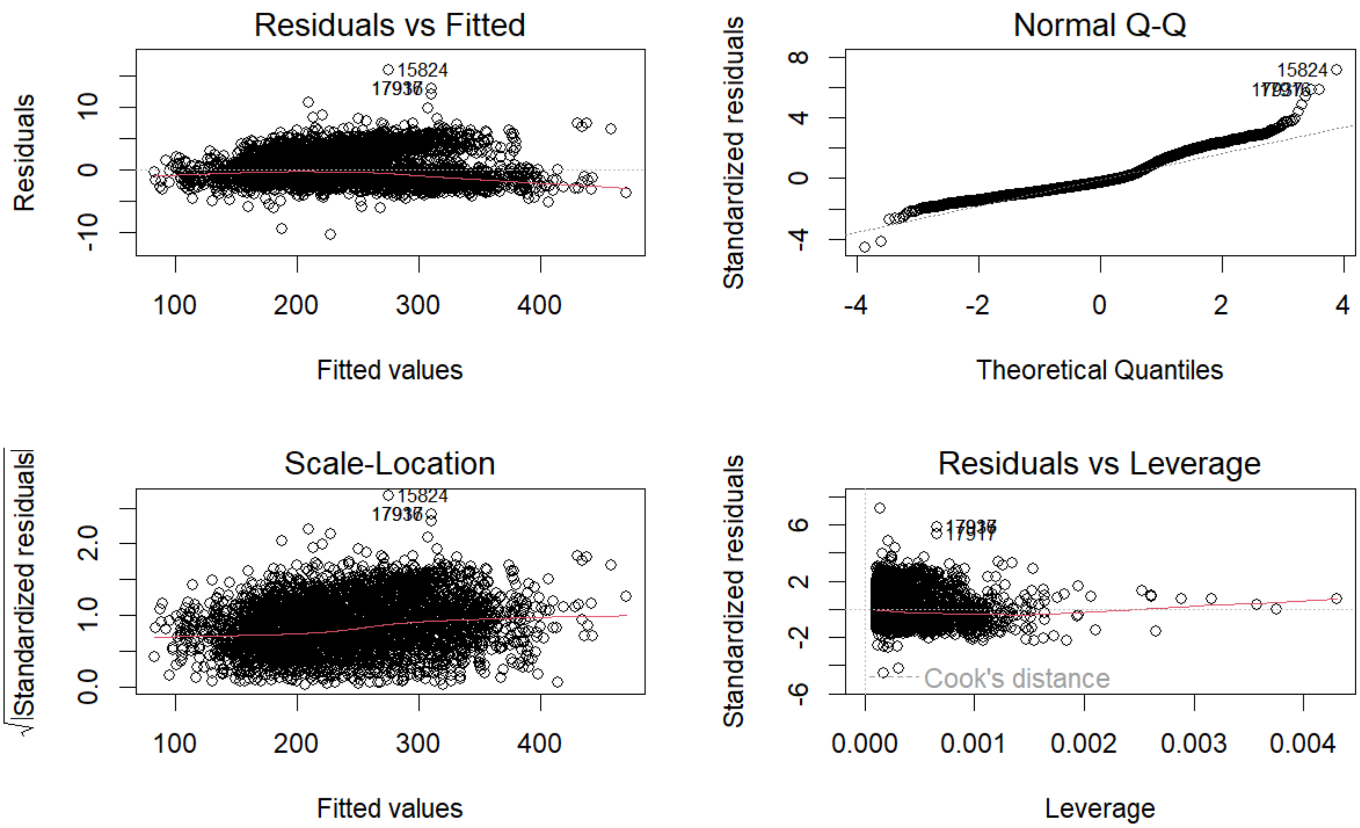
Hide

```
model3 <- lm(EMISSIONS~CITY.LP100KM+HWY.LP100KM, data=train)
summary(model3)
```

```
Call:
lm(formula = EMISSIONS ~ CITY.LP100KM + HWY.LP100KM, data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.2556  -1.5361  -0.5856   1.0985  15.9864 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.13629    0.09790   1.392    0.164    
CITY.LP100KM 12.04286    0.02209 545.206   <2e-16 ***
HWY.LP100KM  11.33617    0.03264 347.354   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.246 on 9454 degrees of freedom
Multiple R-squared:  0.9986,    Adjusted R-squared:  0.9986 
F-statistic: 3.312e+06 on 2 and 9454 DF,  p-value: < 2.2e-16
```

Hide

```
par(mfrow=c(2,2))
plot(model3)
```




Just by looking at the summaries of each model, it is not obvious which model it the best. Among all three models, the p-values, Multiple R-squared, and Std. Error are all around the same area. The t-value in the first model is very high. It lowers in the second model. Additionally, adding the engine size and number of cylinders to the model seems to be irrelevant because their respective t-values are quite low compared to the independent fuel consumption of model 1. It can also be seen when comparing the plots of models 1 and 2, that there are not many changes between the two models. There are more data points in model 2 but they have the same distribution problems that model 1 has. The Residual vs Fitted values are not as evenly distributed as one would like. In the Residuals vs Leverage graph however the points in model 2 are not as stretched as in model 1. This brings all of the highly influential points closer to the rest of the data.

The t-value in model 3, while not as significant as in model 2, is still less than the the singular fuel consumption t-value in model 1When comparing the plots of model 1 and 3, some of the problems in model 1 are solved in model 3. When looking at the Residuals vs Fitted graphs, in both the original and the scaled versions, the data points are more evenly distributed along the horizontal line. This is evident by looking at the Normal Q-Q graph. In model 3, the data points more closely follow the linear line. In the Residuals vs Leverage graph, the outliers are shrunk even further than both model 1 and 2 to the point where they almost blend in with the rest of the data.While there are still some far right points, they still fall easily within Cook’s distance. For these reason, it seems as though model 3 is the best choice out of the 3.

Hide

```
actual <- test$EMISSIONS
```

# Model 1 Metrics

Hide

```
predicted <- predict(model1, test)
residuals <- predicted - actual

correlation <- cor(predicted, actual)
paste("COR: ", correlation)
```

```
[1] "COR:  0.99918589760688"
```

Hide

```
mse <- mean(residuals^2)
paste("MSE: ", mse)
```

```
[1] "MSE:  5.60841284034346"
```

Hide

```
rmse <- sqrt(mse)
paste("RMSE: ", rmse)
```

```
[1] "RMSE:  2.3682087830982"
```

# Model 2 Metrics

Hide

```
predicted <- predict(model2, test)
residuals <- predicted - actual

correlation <- cor(predicted, actual)
paste("COR: ", correlation)
```

```
[1] "COR:  0.999209601350685"
```

Hide

```
mse <- mean(residuals^2)
paste("MSE: ", mse)
```

```
[1] "MSE:  5.44208779543504"
```

Hide

```
rmse <- sqrt(mse)
paste("RMSE: ", rmse)
```

```
[1] "RMSE:  2.33282828245781"
```

# Model 3 Metrics

Hide

```
predicted <- predict(model3, test)
residuals <- predicted - actual

correlation <- cor(predicted, actual)
paste("COR: ", correlation)
```

```
[1] "COR:  0.999180496747625"
```

Hide

```
mse <- mean(residuals^2)
paste("MSE: ", mse)
```

```
[1] "MSE:  5.64354047972001"
```

Hide

```
rmse <- sqrt(mse)
paste("RMSE: ", rmse)
```

```
[1] "RMSE:  2.37561370591264"
```

According to these metrics, model 2 preformed the best out of all three models. This contradicts the analysis deon in the previous step. It is not clear what the reason is for this.