Regression

Kelly Trinh

*The data set used in this assignment can be accessed **here (https://www.kaggle.com/datasets/ahmettyilmazz/fuel-consumption)***

What is linear regression?

Linear regression is essentially an analysis technique used on data to predict the value of a certain attribute based off of the value of another attribute. Linear regression works by attempting to find a relationship between two variables, x and y, and the ultimate goal is to find a model of that data, or the line of best fit. The relationship between these two variables are also defined by parameters w, which represents the amount of change between y and x, and b, which represents the intercept. One advantage of using linear regression is that it is very simple. If two variables do indeed have a linear relationship, then it is very easy to train an algorithm to predict upcoming values. However, there are also disadvantages to using linear regression. One disadvantage to using linear regression is that not all data can be applied to a linear regression algorithm, such as qualitative data. Furthermore, outliers heavily affect the outcome of linear regression models.

This analysis uses a fuel consumption data set and goes through a regression analysis of the data set using different column within the data.

Importing the data

First, import the data from the csv file downloaded from Kaggle. I named the data set **df** for easier reference.

```
df <- read.csv('/Users/kellytrinh/Desktop/school/Regression/Fuel_Consumption_2000-2022.c
sv', na.strings="NA", header=TRUE)
```

Data cleaning

Although there are many different columns within this data set, I'm going to be focusing on EMISSIONS, ENGINE.SIZE, and FUEL.CONSUMPTION for the first to models that I'm building. I'll be using other factors when trying to improve my results later on. I'm also checking for any NAs within the columns, and through the code below, we can see that there are none, so removing NAs or replacing them is unnecessary.

Note: EMISSIONS are CO2 emissions per car, and is measured in g/km. ENGINE.SIZE is the size of a car's engine, and FUEL.CONSUMPTION is the amount of fuel consumed by a car in units of L/100km.

```
# Count the NAs
sapply(df, function(x) sum(is.na(x)==TRUE))
```

```
##              YEAR              MAKE             MODEL    VEHICLE.CLASS
##                 0                 0                 0                0
##       ENGINE.SIZE         CYLINDERS      TRANSMISSION             FUEL
##                 0                 0                 0                0
## FUEL.CONSUMPTION   HWY..L.100.km.  COMB..L.100.km.       COMB..mpg.
##                 0                 0                 0                0
##         EMISSIONS
##                 0
```

A. Dividing the data set into 80/20 train and test sets

The below code block portrays how I divided the data set into an 80/20 train/test set. The rows are randomly sampled to vector *i* with row indices. These are used to divide the data set.

```
set.seed(1234) # Seed of 1234
i <- sample(1:nrow(df), nrow(df)*0.80, replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

B. Data exploration using the train set

After completing the train and test section, we can explore the data using different R commands. The data exploration of this data set is shown below.

```
dim(df) # Show the dimensions of the data frame
```

```
## [1] 22556    13
```

```
names(df) # Names of the columns in the data
```

```
##  [1] "YEAR"              "MAKE"              "MODEL"             "VEHICLE.CLASS"
##  [5] "ENGINE.SIZE"       "CYLINDERS"         "TRANSMISSION"      "FUEL"
##  [9] "FUEL.CONSUMPTION" "HWY..L.100.km."    "COMB..L.100.km."   "COMB..mpg."
## [13] "EMISSIONS"
```

```
str(df) # Show information about the structure of the data frame
```

```
## '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" ...
##  $ FUEL.CONSUMPTION: num  9.2 8.5 12.2 13.4 10 9.3 9.4 13.6 13.8 11.4 ...
##  $ HWY..L.100.km.  : num  6.7 6.5 7.4 9.2 7 6.8 7 9.2 9.1 7.2 ...
##  $ COMB..L.100.km. : 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 ...
```

```
summary(df) # Show the statistics of each numeric column in the data set
```

```
##       YEAR          MAKE              MODEL           VEHICLE.CLASS     
##  Min.   :2000   Length:22556       Length:22556       Length:22556      
##  1st Qu.:2006   Class :character   Class :character   Class :character  
##  Median :2012   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :2012                                                           
##  3rd Qu.:2017                                                           
##  Max.   :2022                                                           
##   ENGINE.SIZE      CYLINDERS      TRANSMISSION           FUEL          
##  Min.   :0.800   Min.   : 2.000   Length:22556       Length:22556      
##  1st Qu.:2.300   1st Qu.: 4.000   Class :character   Class :character  
##  Median :3.000   Median : 6.000   Mode  :character   Mode  :character  
##  Mean   :3.357   Mean   : 5.854                                        
##  3rd Qu.:4.200   3rd Qu.: 8.000                                        
##  Max.   :8.400   Max.   :16.000                                        
##  FUEL.CONSUMPTION HWY..L.100.km.   COMB..L.100.km.   COMB..mpg.   
##  Min.   : 3.50    Min.   : 3.200   Min.   : 3.60   Min.   :11.00  
##  1st Qu.:10.40    1st Qu.: 7.300   1st Qu.: 9.10   1st Qu.:22.00  
##  Median :12.30    Median : 8.400   Median :10.60   Median :27.00  
##  Mean   :12.76    Mean   : 8.919   Mean   :11.03   Mean   :27.37  
##  3rd Qu.:14.72    3rd Qu.:10.200   3rd Qu.:12.70   3rd Qu.:31.00  
##  Max.   :30.60    Max.   :20.900   Max.   :26.10   Max.   :78.00  
##    EMISSIONS    
##  Min.   : 83.0  
##  1st Qu.:209.0  
##  Median :243.0  
##  Mean   :250.1  
##  3rd Qu.:288.0  
##  Max.   :608.0  
```

```
head(df) # Show the first six instances of each column in the data set
```

	Y...	M...	MODEL	VEHICLE.CLA...	ENGINE.SIZE	CYLIND...	TRANSMISS...	F...	FUEL.CONSUM
	<int>	<chr>	<chr>	<chr>	<dbl>	<int>	<chr>	<chr>	<
1	2000	ACURA	1.6EL	COMPACT	1.6	4	A4	X	
2	2000	ACURA	1.6EL	COMPACT	1.6	4	M5	X	
3	2000	ACURA	3.2TL	MID-SIZE	3.2	6	AS5	Z	
4	2000	ACURA	3.5RL	MID-SIZE	3.5	6	A4	Z	
5	2000	ACURA	INTEGRA	SUBCOMPACT	1.8	4	A4	X	
6	2000	ACURA	INTEGRA	SUBCOMPACT	1.8	4	M5	X	

6 rows | 1-10 of 14 columns

C. Plotting the data based on the data exploration, using the training data

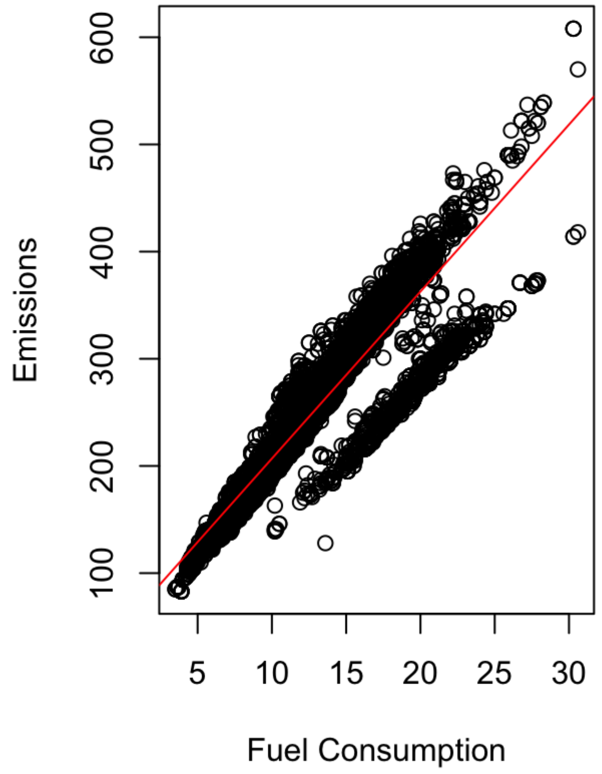
Before I start building the linear regression models for this data set, I am going to create some graphs so that I can visualize and explore the data further. The below code block creates these plots. Here, I created plots using ENGINE.SIZE against EMISSIONS, and FUEL.CONSUMPTION against EMISSIONS.

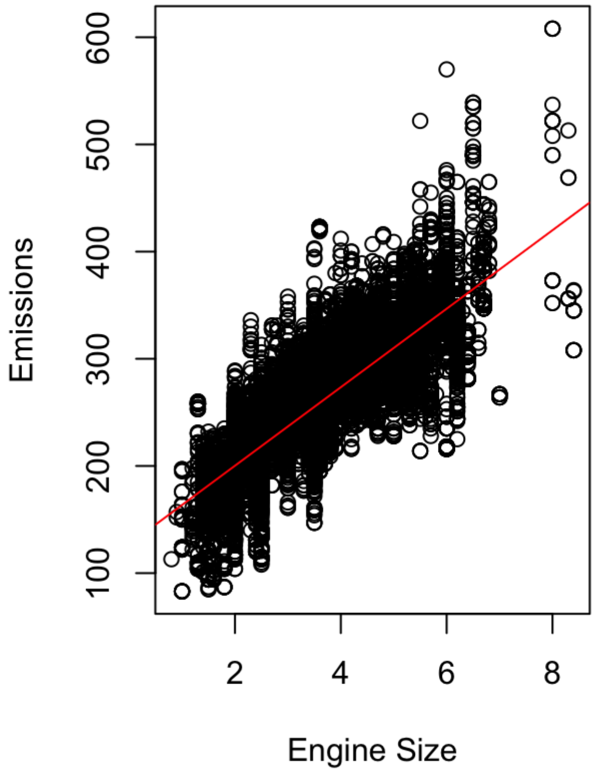
We can attempt to see a linear relationship by plotting these graphs.

```
par(mfrow=c(1,2))

plot(train$FUEL.CONSUMPTION, train$EMISSIONS, xlab="Fuel Consumption", ylab="Emissions")
abline(lm(EMISSIONS~FUEL.CONSUMPTION, data=train), col = "red")

plot(train$ENGINE.SIZE, train$EMISSIONS, xlab="Engine Size", ylab="Emissions")
abline(lm(EMISSIONS~ENGINE.SIZE, data=train), col = "red")
```





D. Building a linear regression model using one predictor

Now, a linear regression model is built. In simple linear regression, only one predictor is used, and here, I wanted to see the impact of ENGINE.SIZE on EMISSIONS.

```
# Linear Regression Model using FUEL.CONSUMPTION against EMISSIONS
lm1 <- lm(EMISSIONS~FUEL.CONSUMPTION, data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = EMISSIONS ~ FUEL.CONSUMPTION, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -134.997   -7.235    0.585   12.212   84.793
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      51.08890    0.65953   77.46   <2e-16 ***
## FUEL.CONSUMPTION 15.58147    0.04987  312.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.46 on 18042 degrees of freedom
## Multiple R-squared:  0.844,  Adjusted R-squared:  0.844
## F-statistic: 9.763e+04 on 1 and 18042 DF,  p-value: < 2.2e-16
```

Explaining the summary

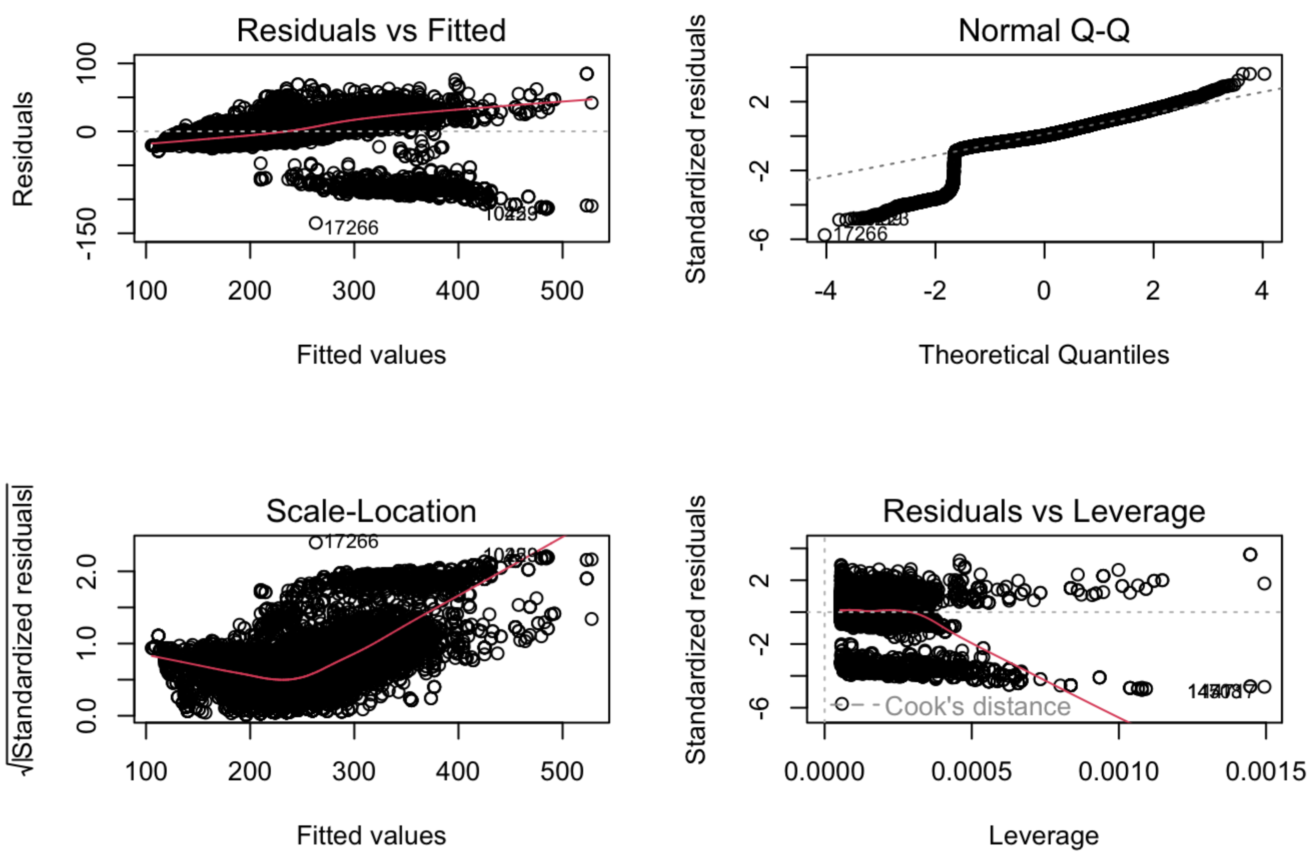
The summary of the model tells us many things. We are able to see the estimated value, the intercept, the standard error, and the p and t values. We can see that the p-value for FUEL.CONSUMPTION is small, which is good because it gives us evidence to reject the null hypothesis, indicating that the predictor can influence the target. This small p-value gives us confidence in seeing a relationship between FUEL.CONSUMPTION and EMISSIONS. Furthermore, we can see that our predictor value for FUEL.CONSUMPTION is 15.58147, meaning that for approximately 15.5 L/100km fuel consumed, 1g/km CO2 is emitted. The t-value for FUEL.CONSUMPTION is 312.45 away from zero, meaning that we can have more confidence in rejecting the null hypothesis.

E. Plotting the residuals

Now we plot the residuals for the singular linear regression model that we just created. I used a 2x2 grid to display all of the plots.

First looking at the Residuals vs Fitted plot, we can see that the red line is fairly horizontal, showing us that there is smaller amount of deviation in the data. Next, using the Normal Q-Q plot, the diagonal line is fairly straight until we reach the top right, in which it deviates from the straight line and goes higher. This means that the data is not completely normal distributed, as it does not follow a diagonal line. Then, by using the Scale-Location plot, we can see a fairly horizontal line with points distributed around it. The points are evenly distributed until the end of the graph, where we can see only some data points. This means that the data may be homoscedastic, excluding some data point. The last plot, Residuals vs Leverage, indicate that there may be many data points with unusual x vaues, as there is a mass amount of data towards the beginning of the graph that deviate from the horizontal line.

```
par(mfrow=c(2,2)) # Create a 2x2 grid for the residuals
plot(lm1)
```



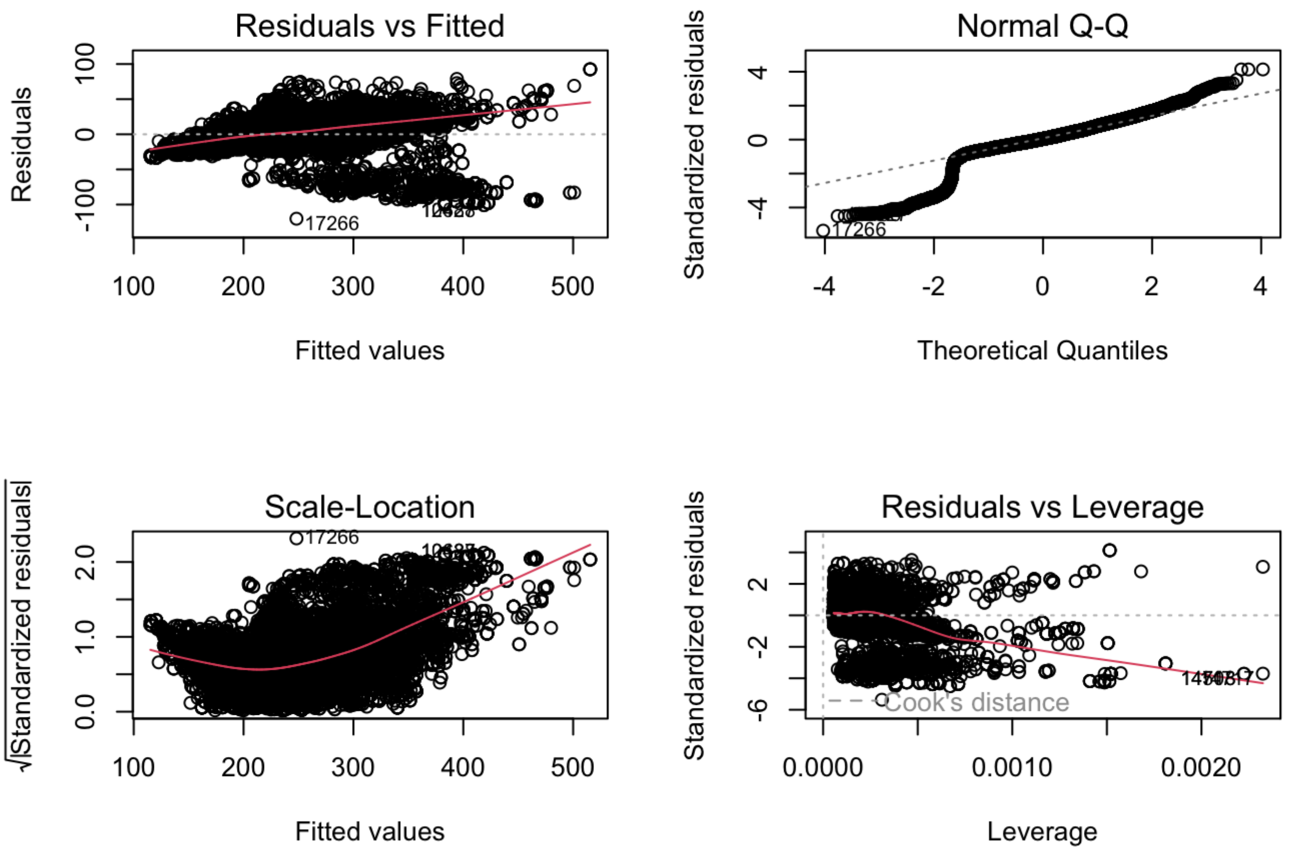
F. Building a multiple linear regression model using a combination of predictors

Now, we will use two predictors to create a multiple linear regression model. The two predictors that I am using in this regression model for median_house_value are median_income and housing_median_age, which is the median age of a house within a block.

```
# Second linear regression model using ENGINE.SIZE and FUEL.CONSUMPTION
lm2 <- lm(EMISSIONS~ENGINE.SIZE+FUEL.CONSUMPTION, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = EMISSIONS ~ ENGINE.SIZE + FUEL.CONSUMPTION, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -120.152   -8.120    0.969   11.785   92.532
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      56.93160    0.64521   88.24   <2e-16 ***
## ENGINE.SIZE       9.17267    0.21979   41.73   <2e-16 ***
## FUEL.CONSUMPTION 12.71138    0.08365  151.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.41 on 18041 degrees of freedom
## Multiple R-squared:  0.8578, Adjusted R-squared:  0.8577
## F-statistic: 5.439e+04 on 2 and 18041 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2)) # Create a 2x2 grid for the residuals
plot(lm2)
```



Now, I will use the anova() function to compare the two linear regression models that were generated.

```
anova(lm1, lm2) # Analysis of Variance between lm1 and lm2
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	18042	9931694	NA	NA	NA	NA
2	18041	9057313	1	874381	1741.654	0

2 rows

G. Building a third linear regression model to try and improve the results

In order to try and improve the results, I'm going to create a different combination of predictors for a multiple linear regression model, adding on from the second linear regression model that I created. In this model, I added CYLINDERS, which represents the amount of cylinders that the car has, and HWY..L.100.km., which is the highway fuel consumption of the car. This linear regression model has 4 predictors total. The below code builds the regression model:

```
 # Multiple linear regression model using ENGINE.SIZE, FUEL.CONSUMPTION, CYLINDERS, and HIGHWAY FUEL CONSUMPTION
lm3 <- lm(EMISSIONS~ENGINE.SIZE+FUEL.CONSUMPTION+CYLINDERS+HWY..L.100.km., data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = EMISSIONS ~ ENGINE.SIZE + FUEL.CONSUMPTION + CYLINDERS +
##     HWY..L.100.km., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -120.934   -5.933    2.291   10.892   81.922
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       39.7711     0.7271   54.70   <2e-16 ***
## ENGINE.SIZE        3.8225     0.3174   12.04   <2e-16 ***
## FUEL.CONSUMPTION   6.1339     0.1651   37.16   <2e-16 ***
## CYLINDERS          6.0130     0.2239   26.86   <2e-16 ***
## HWY..L.100.km.     9.4045     0.2146   43.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.15 on 18039 degrees of freedom
## Multiple R-squared:  0.8733, Adjusted R-squared:  0.8733
## F-statistic: 3.109e+04 on 4 and 18039 DF,  p-value: < 2.2e-16
```

Note: CYLINDERS is the amount of cylinders in a car's engine, and HWY..L.100.km. is highway fuel consumption of a car, measured in L/100km.

H. Comparing the results

We can see that the R-squared value is the best for the third linear regression model, which is a multiple linear regression using ENGINE.SIZE, FUEL.CONSUMPTION, CYLINDERS, and HWY..L.100.km. (highway fuel consumption) against EMISSIONS. The R-squared value for this model is 0.8733, which is better in comparison to the 0.8578 and 0.844 R-squared values from the second and first linear regression models. Furthermore, we see that, after using the anova() function with the first and second linear regression models, the second linear model (lm2) lowered the errors and the RSS value. This indicates that lm2 was a better model than lm1, however it is not as good as lm3.

I. Predicting and evaluating on the test data

Now, I'm going to predict and evaluate on the test data using the three linear regression models that I created so far. Below is the code that predicts and evaluates using the first model created in this project:

```
pred1 <- predict(lm1, newdata=test) # Predicting through lm1 and the test data set
cor1 <- cor(pred1, test$EMISSIONS) # Correlation between pred1 and EMISSIONS
mse1 <- mean((pred1-test$EMISSIONS)^2) # Mean Square Error using lm1 and the test data s
et
rmse1 <- sqrt(mse1) # Root Mean Square

print(paste('correlation:', cor1))
```

```
## [1] "correlation: 0.920693906117968"
```

```
print(paste('mse:', mse1))
```

```
## [1] "mse: 533.115826323111"
```

```
print(paste('rmse:', rmse1))
```

```
## [1] "rmse: 23.0893011224487"
```

Below is the code that predicts and evaluates using the second model created:

```
pred2 <- predict(lm2, newdata=test) # Predicting through lm2 and the test data set
cor2 <- cor(pred2, test$EMISSIONS) # Correlation between pred2 and EMISSIONS
mse2 <- mean((pred2-test$EMISSIONS)^2) # Mean Square Error using lm2 and the test data s
et
rmse2 <- sqrt(mse2) # Root Mean Square

print(paste('correlation:', cor2))
```

```
## [1] "correlation: 0.927392993119372"
```

```
print(paste('mse:', mse2))
```

```
## [1] "mse: 489.826718204576"
```

```
print(paste('rmse:', rmse2))
```

```
## [1] "rmse: 22.1320292382912"
```

Lastly, below is the code that predicts and evaluates using the third model created:

```
pred3 <- predict(lm3, newdata=test) # Predicting through lm3 and the test data set
cor3 <- cor(pred3, test$EMISSIONS) # Correlation between pred3 and EMISSIONS
mse3 <- mean((pred3-test$EMISSIONS)^2) # Mean Square Error using lm3 and the test data s
et
rmse3 <- sqrt(mse3) # Root Mean Square

print(paste('correlation:', cor3))
```

```
## [1] "correlation: 0.935649882877636"
```

```
print(paste('mse:', mse3))
```

```
## [1] "mse: 435.953293276862"
```

```
print(paste('rmse:', rmse3))
```

```
## [1] "rmse: 20.8794945646886"
```

We can see that the correlation values between the three models increase with each model. This indicates that lm3 had the best correlation out of all of the three models, and therefore it can be the best representation of the dataset and how correlated the data is. The RMSE and MSE values also decreased over each model, indicating that there was a smaller difference between the predicted and actual values for lm3.