Name: Maninderpreet Singh Puri
Student ID: 20494381

BUS5PA Assignment 1 Submission (Part 2)

Q1.

a. Which variables are continuous/numerical? Which are ordinal? Which are nominal?

Solution: From the data set '*used_cars*', below listed variables are continuous variables as these are all in numerical form and don't have any order or fall under categories. Some of these are whole numbers while others are float numbers. The parameters with range are listed below:

Continuous/numerical:

- Price- (Range: 1500- 240000)
- back_legroom- (Range: 27.1- 44.4)
- city_fuel_economy- (Range: 11-70)
- daysonmarket- (Range: 0-1587)
- engine_displacement- (Range: 1000-1600)
- Front_legroom- (Range: 38-52.5)
- fuel_tank_volume- (Range: 9.5-33.5)
- height- (Range: 53.7- 88)
- highway_fuel_economy- (Range: 11-75)
- horsepower- (Range: 99-621)
- length- (Range: 150.6-221.9)
- mileage- (Range:79- 123203)
- wheelbase- (Range: 93.3-132.5)
- width- (Range: 65.4- 93.4)

Below listed variables are ordinal variables as these are in an increasing order (as ordered below in the list). For example, condition of car; '*Fair'* is the lowest level, '*Good'* is better and *'Excellent'* is the best. The parameters with levels are listed below:

Ordinal:

- Condition- (Levels: Fair -> Good -> Excellent)
- engine_cylinders- (Levels: 3 -> 4 -> 5 -> 6 -> 8-> 12 Cylinders)
- maximum_seating- (Levels: 4 -> 5 -> 6 -> 7 -> 8 seats)
- owner_count- (Levels: 1 -> 2-> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 owners)
- year- (Levels: 1990- 2021)

Below listed variables are nominal variables as their value are independent, that means values are not related to each other. For example, body_type: Hatchback car body type is different in terms of features to Sedan and SUV/Crossover. The parameters are justified below:

Nominal:

- body_type- (Categories: Hatchback, Sedan, SUV/ Crossover)
- fuel_type- (Categories: Diesel/Gasoline/Hybrid)
- make_name- (Categories: Audi, BMW, Ford, Mercedes-Benz, Toyota, Volkswagen)
- salvage: - (Categories: TRUE, FALSE)
- transmission- (Categories: Automatic, Continuously Variable Transmission, Dual clutch, Manual)
- wheel_system- (Categories: All- Wheel Drive, Four- Wheel Drive, Front- Wheel Drive, Rear- Wheel Drive)

b. What are the methods for transforming categorical variables?

Solution: Categorical variable cannot be used straight away in most cases as the computer can't understand text. So, according to the business scenario we can transform categorical variables to numbers (binary or multiple numbers) or split them into bins.

Techniques for transforming categorical variables:

1. Ordinal encoding: We can use ordinal encoding: for example, '*Fair*' is 0, '*Good*' is 1, and '*Excellent*' is 2. This is called ordinal encoding or integer encoding as describes in the article (Ray, 2020). This method is particularly useful for ordinal data as the variables still can maintain the order or levels.
2. One-Hot encoding: For categorical variables where no order exists or is nominal, we can use one-hot encoding. Each category is assigned a column where it is represented by 1 if it is true or 0 if it is false. Example for this is: *'Hatchback'* is encoded as [1,0,0], *'Sedan'* is encoded as [0,1,0], and *'SUV/Crossover'* is encoded as [0,0,1].
3. Dummy variable encoding: It is an improvement in one-hot encoding and eliminates redundancy. For example, if we know that [1, 0, 0] represents '*Hatchback*' and [0, 1, 0] represents '*Sedan*' we don't need another binary variable to represent '*SUV/Crossover*", instead we could use 0 values for both '*Hatchback*' and '*Sedan*' alone, e.g. [0, 0]. So, it can re- encode as: '*Hatchback'* is encoded as [1,0], *'Sedan'* is encoded as [0,1], and '*SUV/Crossover'* is encoded as [0,0]
4. Splitting an ordinal variable into bins- Ordinal variables can have large range of values. So, we can transform the large rang into small bins. For example, *'year'* variable has range 1992-2021. We can split the into six bins: 1992-1996, 1997-2001, 2002-2006, 2007-2011, 2012-2016, 2017-2021. Further, we can transform them into order. "1992-1996" is 0, "1997-2001" is 1, "2002-2006" is 2, "2007-2011" is 3, "2012-2016" is 4, and "2017-2021" is 5. This method can also be applied to continuous/numerical vales to transform them into ordinal values.

I choose following transformations for nominal and ordinal variables:

<u>Ordinal:</u>

- Condition- (Levels: Fair -> 1, Good -> 2, Excellent-> 3)
- engine_cylinders- (Levels: 3 Cylinders -> 3, 4 Cylinders -> 4, 5 Cylinders -> 5, 6 Cylinders -> 6, 8 Cylinders -> 8, 12 Cylinders -> 12). The integers 3, 4, 5, 6, 8, 12 are used as it retains the level as well as the number of cylinders.
- maximum_seating- (Levels: 4 seats -> 4, 5 seats -> 5, 6 seats -> 6, 7 seats -> 7, 8 seats -> 8). The integers 4, 5, 6, 7, 8 are used as it retains the level as well as the number of seats.
- owner_count- (Levels: 1 owner -> 1, 2 owners -> 2, 3 owners -> 3, 4 owners -> 4, 5 owners -> 5, 6 owners -> 6, 7 owners -> 7, 8 owners -> 8, 9 owners -> 9). The integers 1, 2, 3, 4, 5, 6, 7, 8, 9 are used as it retains the level as well as the number of owners.
- year- (Levels: "1990" is 31, "1997" is 24, "1998" is 23, "1999" is 22, "2000" is 21, "2001" is 20, "2002" is 19, "2003" is 18, "2004" is 17, "2005" is 16, "2006" is 15, "2007" is 14, "2008" is 13, "2009" is 12, "2010" is 11, "2011" is 10, "2012" is 9, "2013" is 8, "2014" is 7, "2015" is 6, "2016" is 5, "2017" is 4, "2018" is 3, "2019" is 2, "2020" is 1, "2021" is 0). I have transformed it into age of car as this can be analysed easily rather than the make year of car.

<u>Nominal:</u>

- I have used one-hot encoding as the transformation technique. This is best as we can use the count of each category to identify check the distribution of the parameter.
- body_type- (Categories: Hatchback, Sedan, SUV/ Crossover) encoded as [1,0,0], [0,1,0], [0,0,1].
- fuel_type- (Categories: Diesel/Gasoline/Hybrid) encoded as [1,0,0], [0,1,0], [0,0,1].
- make_name- (Categories: Audi, BMW, Ford, Mercedes-Benz, Toyota, Volkswagen) encoded as [1,0,0,0,0,0], [0,1,0,0,0,0], [0,0,1,0,0,0], [0,0,0,1,0,0], [0,0,0,0,1,0], [0,0,0,0,0,1].
- salvage: Categories- (Categories: "TRUE" is 1, "FALSE" is 0)
- transmission- (Categories: Automatic, Continuously Variable Transmission, Dual clutch, Manual) encoded as [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1].
- wheel_system- (Categories: All- Wheel Drive, Four- Wheel Drive, Front- Wheel Drive, Rear- Wheel Drive) encoded as [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1].

c. Carry out and demonstrate data transformation where necessary.

Solution: In attached R script.

Q2.

a. Calculate following summary statistics: mean, median, max, and standard deviation for each of the continuous variables, and count for each categorical variable.
Solution: In attached R script.

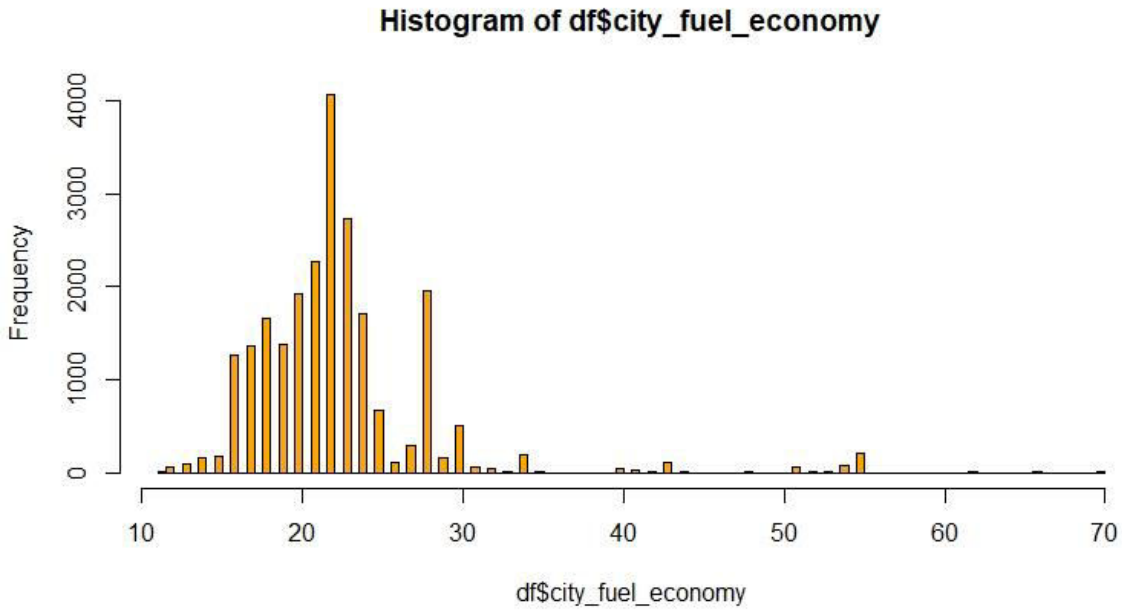
b. Is there any evidence of extreme values? Briefly discuss.
Solution:

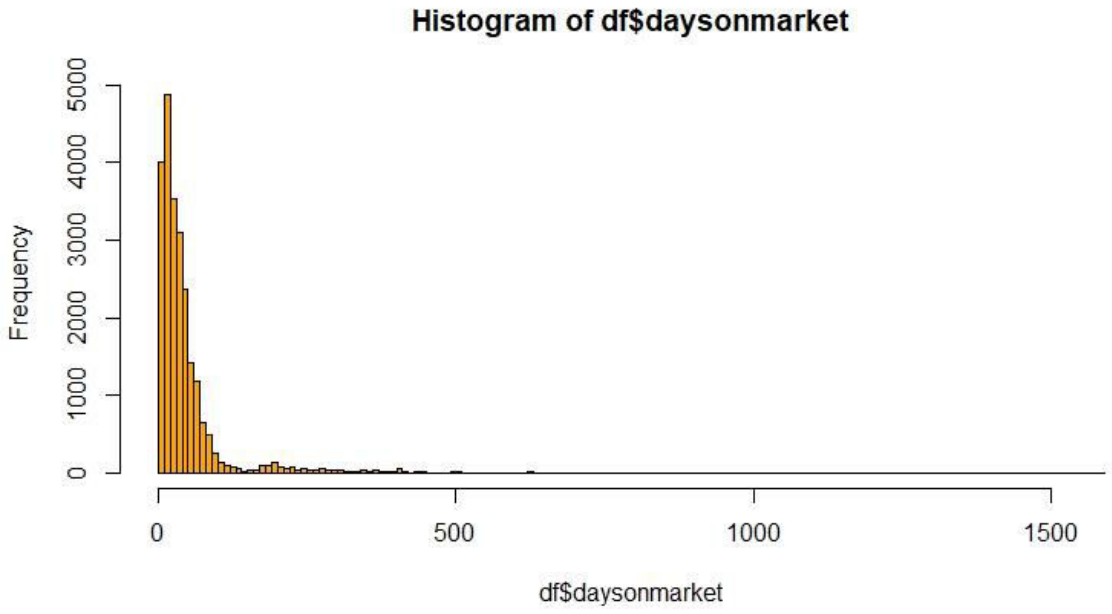
There are extreme values in a few parameters. Following are the parameters with extremities:

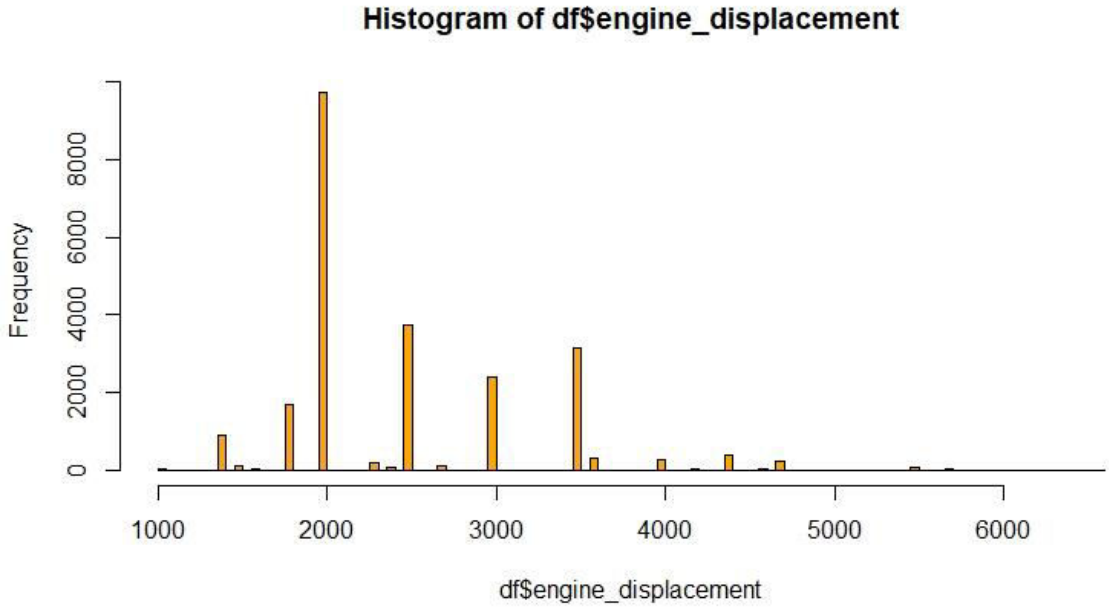
- condition: *'Fair'* value is occurring 39 times, while *'Excellent'* value is occurring 20444 times.
- engine_cylinders: *'12 Cydiners'* value is occurring 9 times, while *'4 Cydiners'* value is occurring 16419 times.
- maximum_seating: *'6 seats'* value is occurring 4 times, while *'5 seats'* value is occurring 18820 times.
- owner_count: *'9'* value is occurring 1 time, while *'1'* value is occurring 20175 times.
- age_car: *'31', '23', '22' & '0'* values are occurring 1 time, while *'4'* value is occurring 11790 times.
- body_type: '*Hatchback'* value is occurring 655 times, while *'SUV / Crossover'* value is occurring 12601 times.
- fuel_type: '*Diesel'* value is occurring 308 times, while *'Gasoline'* value is occurring 22642 times.
- wheel_system: '*Four-Wheel Drive'* value is occurring 252 times, while *'All-Wheel_Drive'* value is occurring 6861 times.

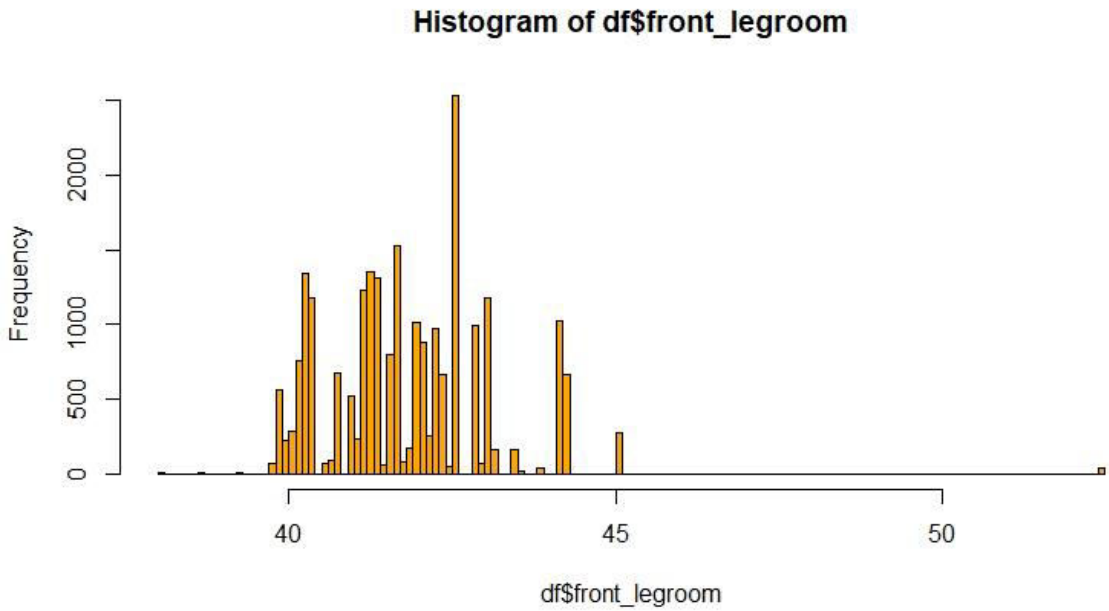
Q3.

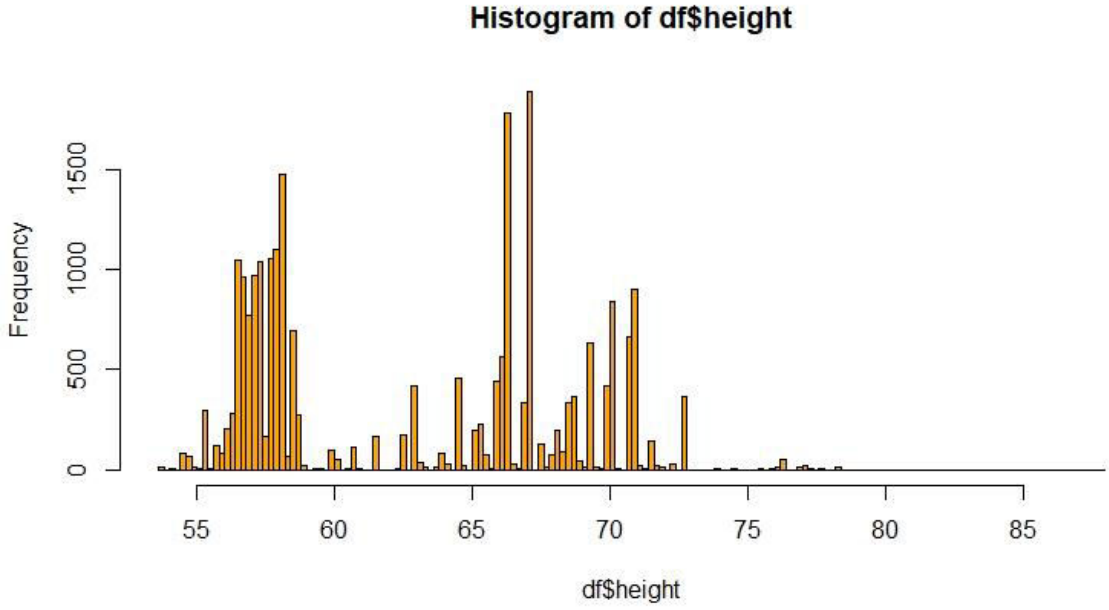
Plot histograms for each of the continuous variables and create summary statistics. Based on the histogram and summary statistics answer the following and provide brief explanations:

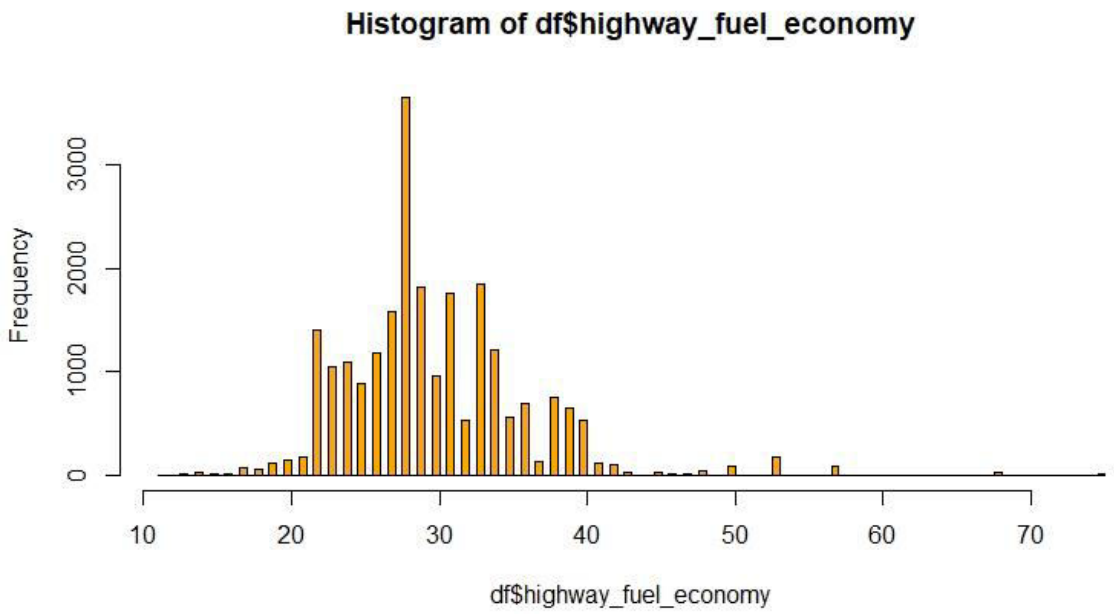


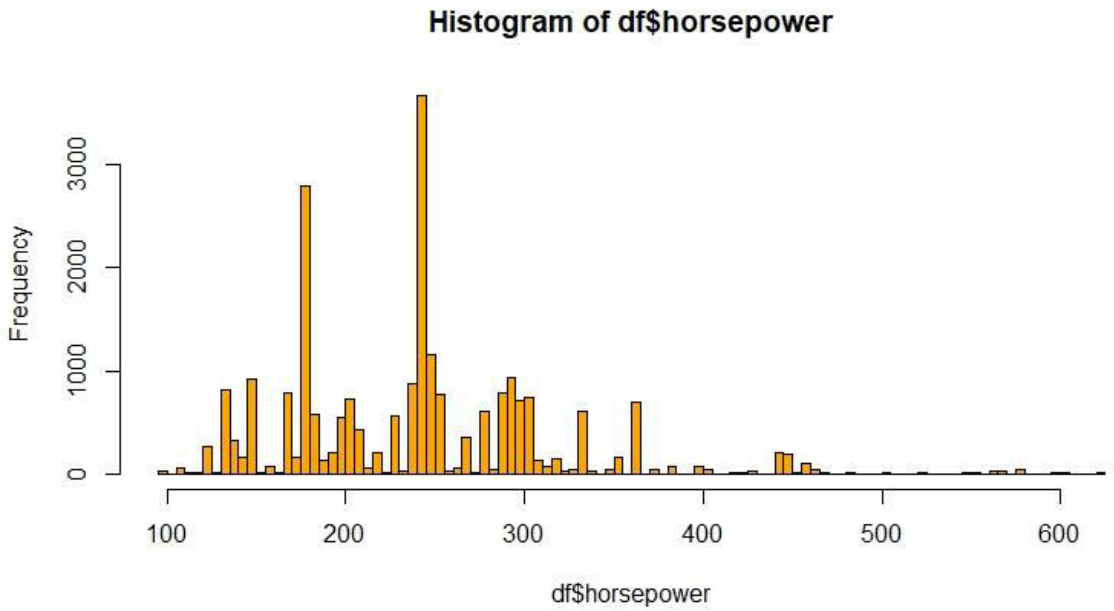


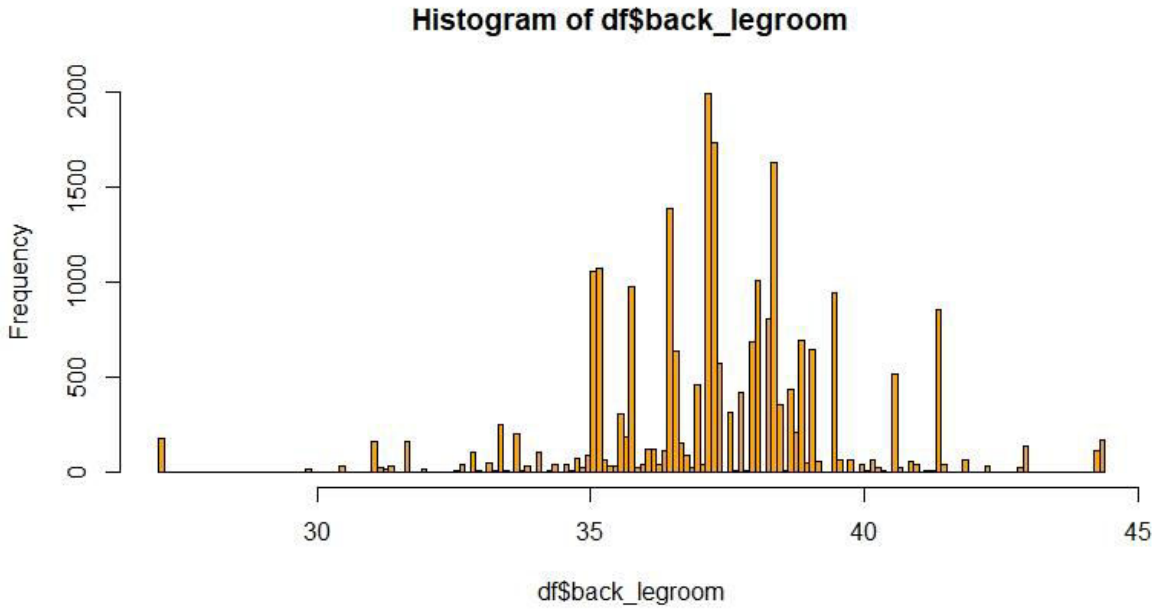

Histogram of df$engine_displacement
Frequency
0
2000
4000
6000
8000
1000
2000
3000
4000
5000
6000
df$engine_displacement

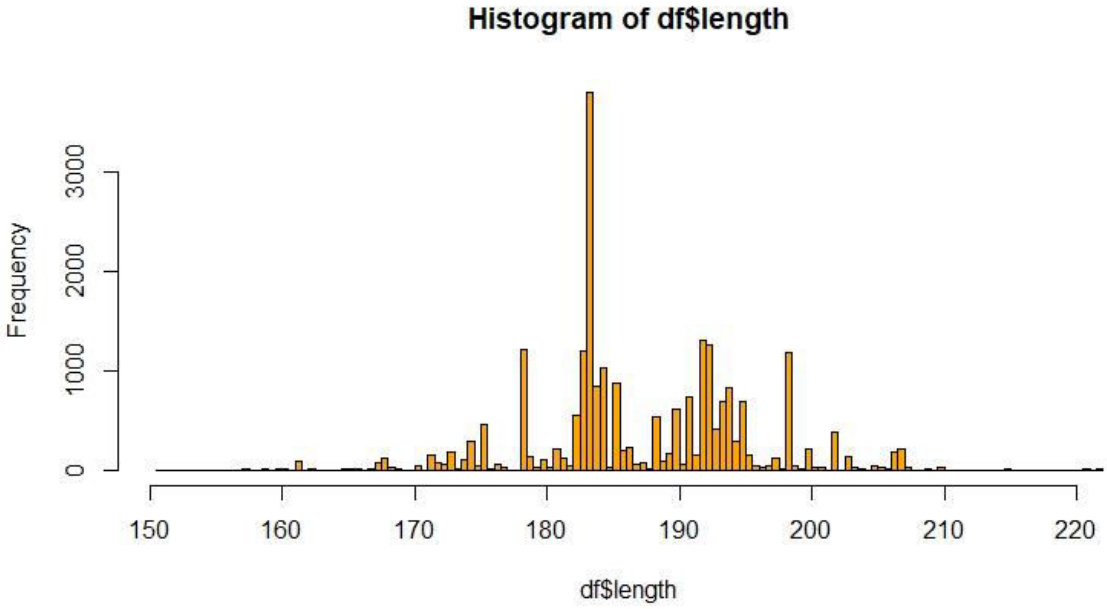

Histogram of df$front_legroom
Frequency
0
500
1000
2000
40
45
50
df$front_legroom

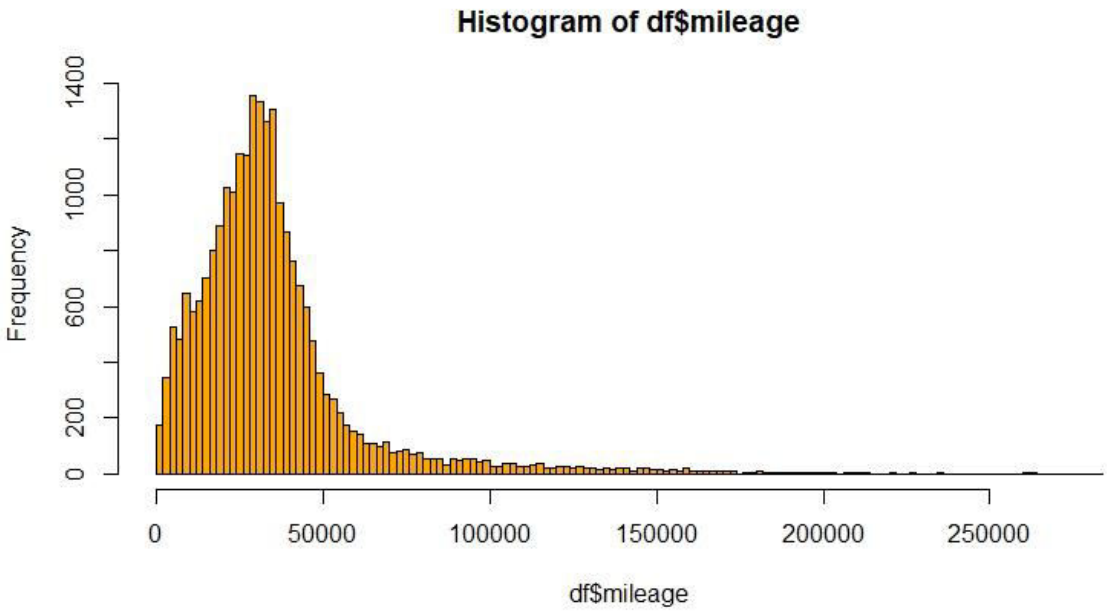

Histogram of df$height
Frequency
0
500
1000
1500
55
60
65
70
75
80
85
df$height

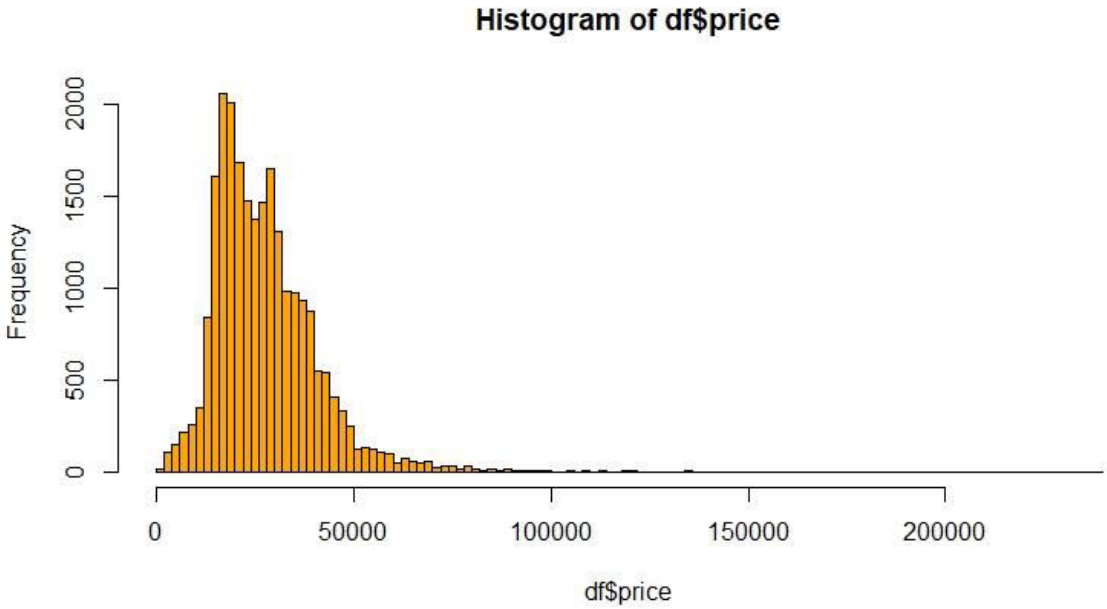

Histogram of df$highway_fuel_economy
Frequency
0
1000
2000
3000
10
20
30
40
50
60
70
df$highway_fuel_economy

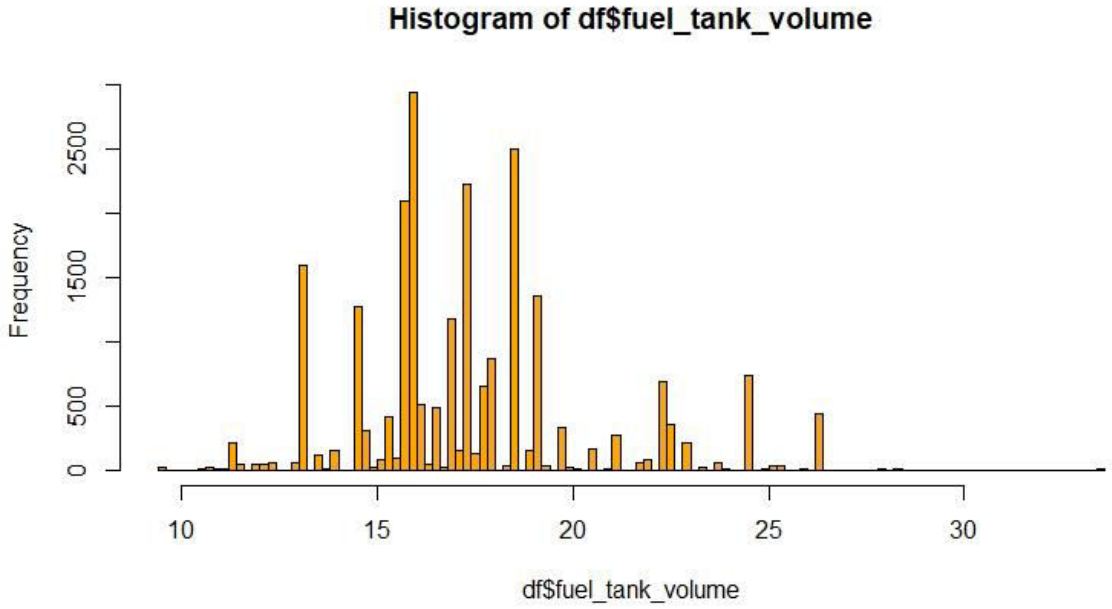

Histogram of df$horsepower
Frequency
0
1000
2000
3000
100
200
300
400
500
600
df$horsepower

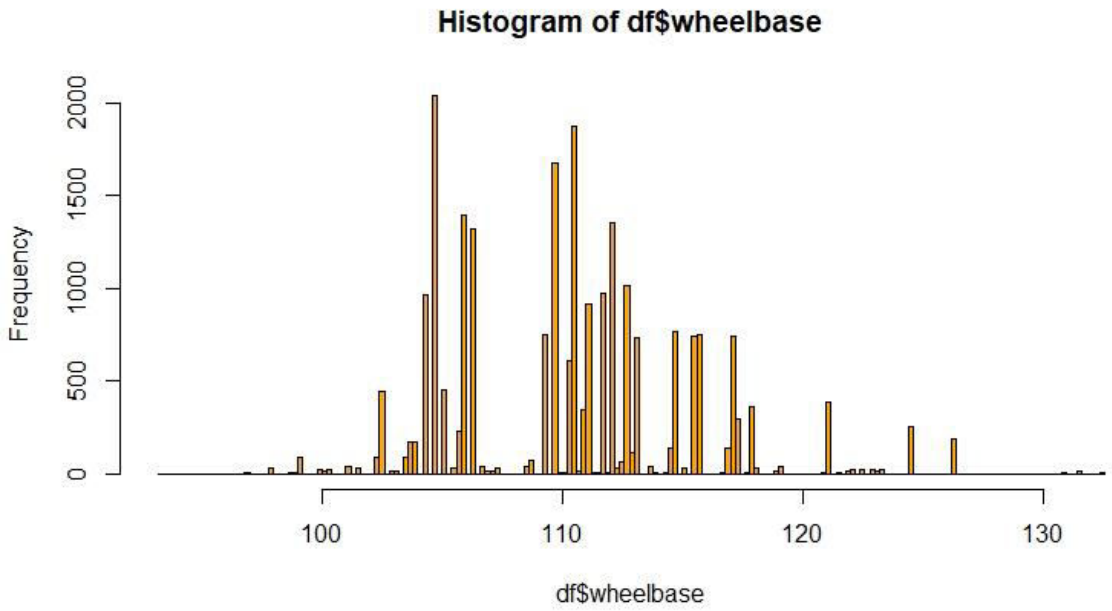

Histogram of df$back_legroom
Frequency
0
500
1000
1500
2000
30
35
40
45
df$back_legroom

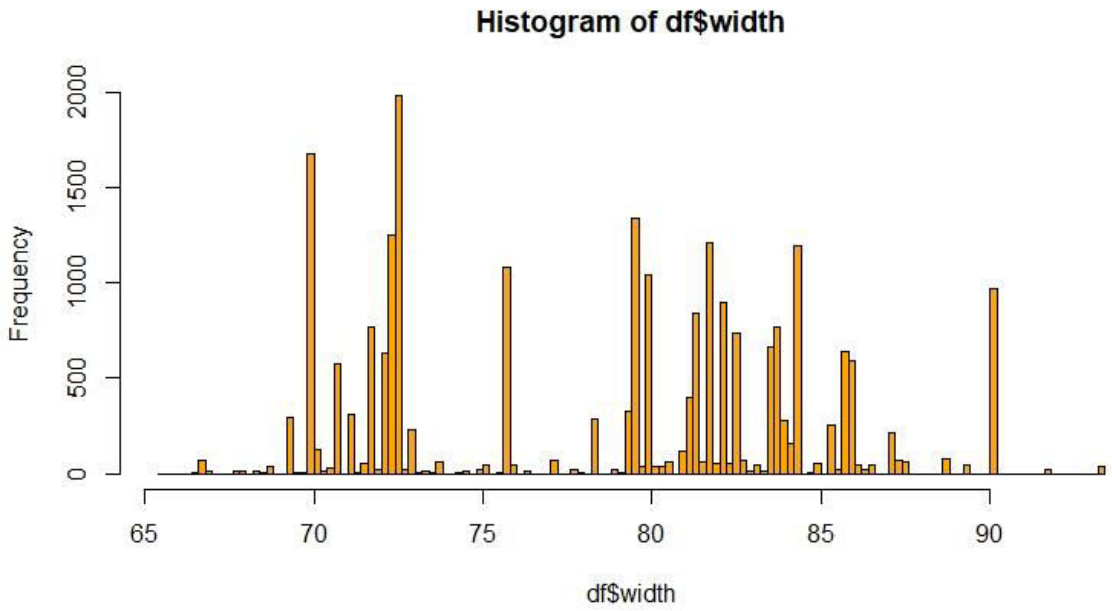

Histogram of df$length
Frequency
0
1000
2000
3000
150
160
170
180
190
200
210
220
df$length


Histogram of df$mileage
Frequency
0
200
600
1000
1400
0
50000
100000
150000
200000
250000
df$mileage


Histogram of df$price
Frequency
0
500
1000
1500
2000
0
50000
100000
150000
200000
df$price


Histogram of df$fuel_tank_volume
Frequency
0
500
1500
2500
10
15
20
25
30
df$fuel_tank_volume


Histogram of df$wheelbase
Frequency
0
500
1000
1500
2000
100
110
120
130
df$wheelbase


Histogram of df$width
Frequency
0
500
1000
1500
2000
65
70
75
80
85
90
df$width

Summary Statistics:

	df (N = 23,531)
price	
minimum	1,500
median (IQR)	25,750 (18,500.00, 34,493.50)
mean (sd)	27,802.91 ± **13,283.03**
maximum	240,000
back_legroom	
minimum	27.10
median (IQR)	37.30 (36.50, 38.40)
mean (sd)	37.39 ± 2.29
maximum	44.40
city_fuel_economy	
minimum	11
median (IQR)	22.00 (19.00, 24.00)
mean (sd)	22.58 ± 6.13
maximum	70
Unknown/Missing	5 (0.02%)
daysonmarket	
minimum	0
median (IQR)	28 (14.00, 49.00)
mean (sd)	45.56 ± 68.47
maximum	1,587
engine_displacement	
minimum	1,000
median (IQR)	2,000 (2,000.00, 3,000.00)
mean (sd)	2,480.21 ± 744.95
maximum	6,600
front_legroom	
minimum	38.00
median (IQR)	41.70 (41.10, 42.60)
mean (sd)	41.87 ± 1.27
maximum	52.50
Unknown/Missing	7 (0.03%)
fuel_tank_volume	
minimum	9.50

median (IQR)	17.00 (15.80, 18.60)
mean (sd)	17.43 ± 3.04
maximum	33.50
height	
minimum	53.70
median (IQR)	62.90 (57.40, 67.10)
mean (sd)	62.88 ± 5.57
maximum	88.00
highway_fuel_economy	
minimum	11
median (IQR)	29 (26.00, 33.00)
mean (sd)	29.87 ± 6.02
maximum	75
horsepower	
minimum	99
median (IQR)	241 (178.00, 290.00)
mean (sd)	240.67 ± 74.50
maximum	621
length	
minimum	150.60
median (IQR)	185.10 (183.10, 192.70)
mean (sd)	187.35 ± 8.03
maximum	221.90
mileage	
minimum	79
median (IQR)	30,350 (20,167.50, 40,959.50)
mean (sd)	34,973.21 ± **26,512.74**
maximum	283,030
Unknown/Missing	108 (0.46%)
wheelbase	
minimum	93.30
median (IQR)	110.60 (105.90, 112.90)
mean (sd)	110.45 ± 5.02
maximum	132.50
width	
minimum	65.40

median (IQR)	80.00 (72.40, 83.50)
mean (sd)	78.61 ± 6.02
maximum	93.40

Q3.

a. Which variables have the largest variability?

Solution:

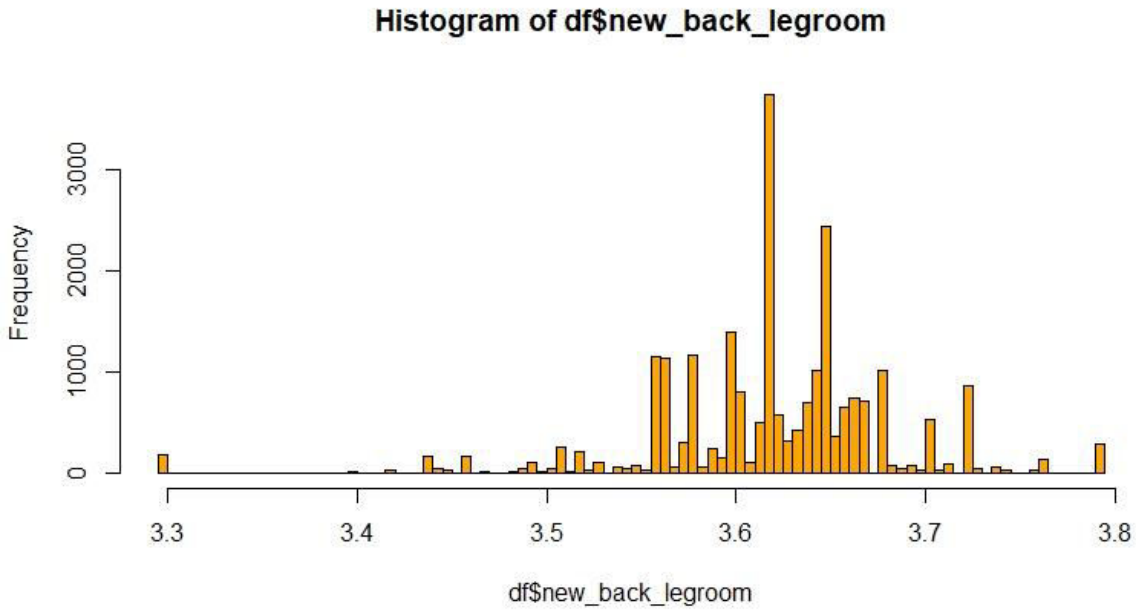
According to the measures of variability calculated in R studio, we can tell the order of parameters from highest to lowest variability are (this order is for standard deviation):

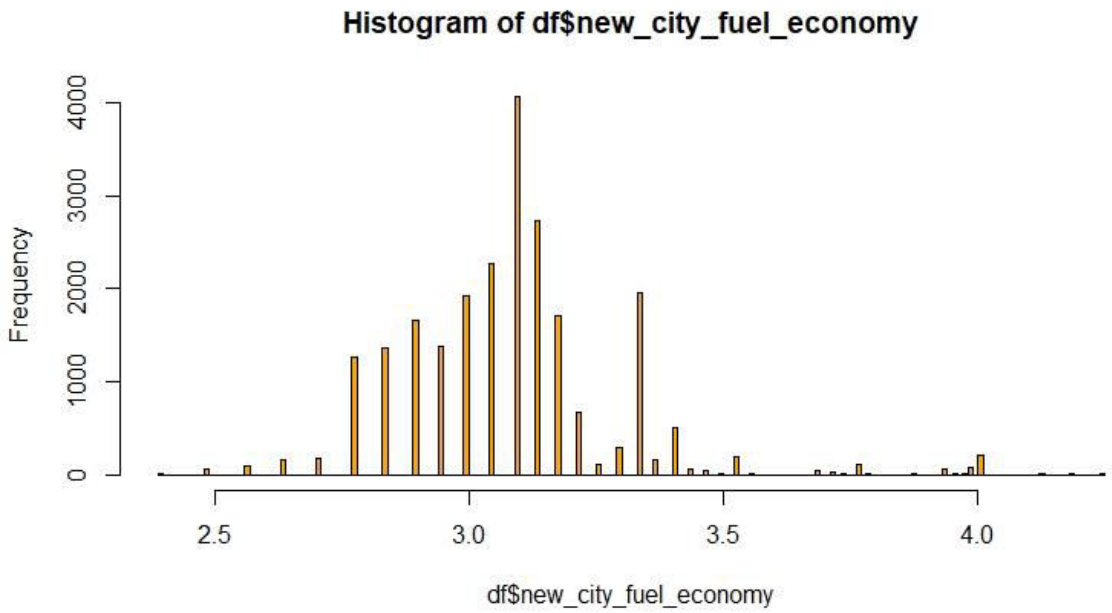
1. mileage
2. price
3. engine_displacement
4. horsepower
5. daysonmarket
6. length
7. city_fuel_economy
8. highway_fuel_economy
9. width
10. height
11. wheelbase
12. fuel_tank_volume
13. back_legroom
14. front_legroom

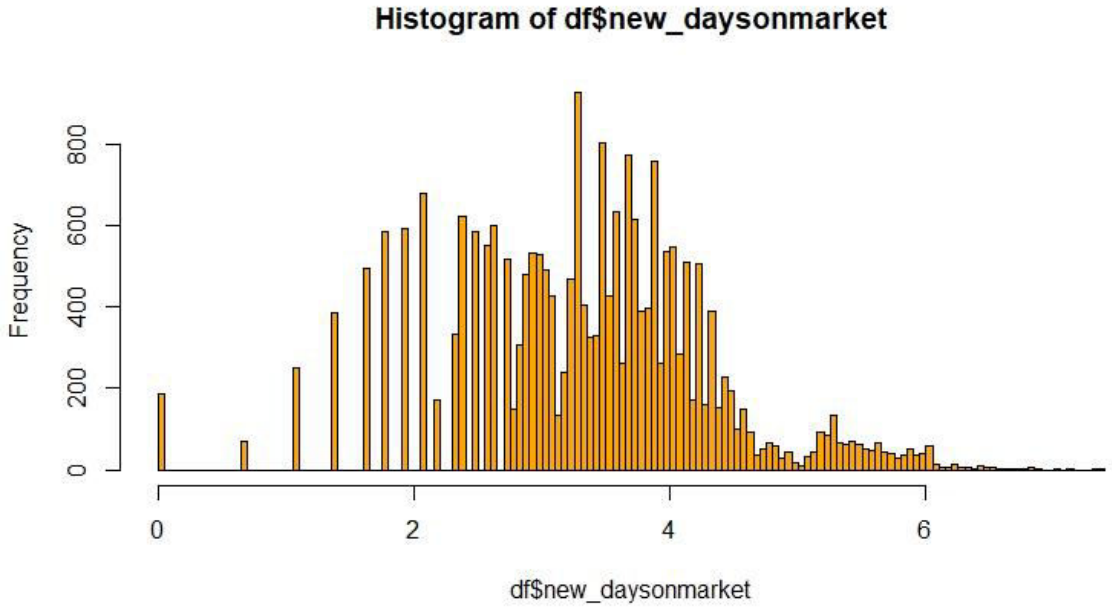
b. Which variables seems skewed?

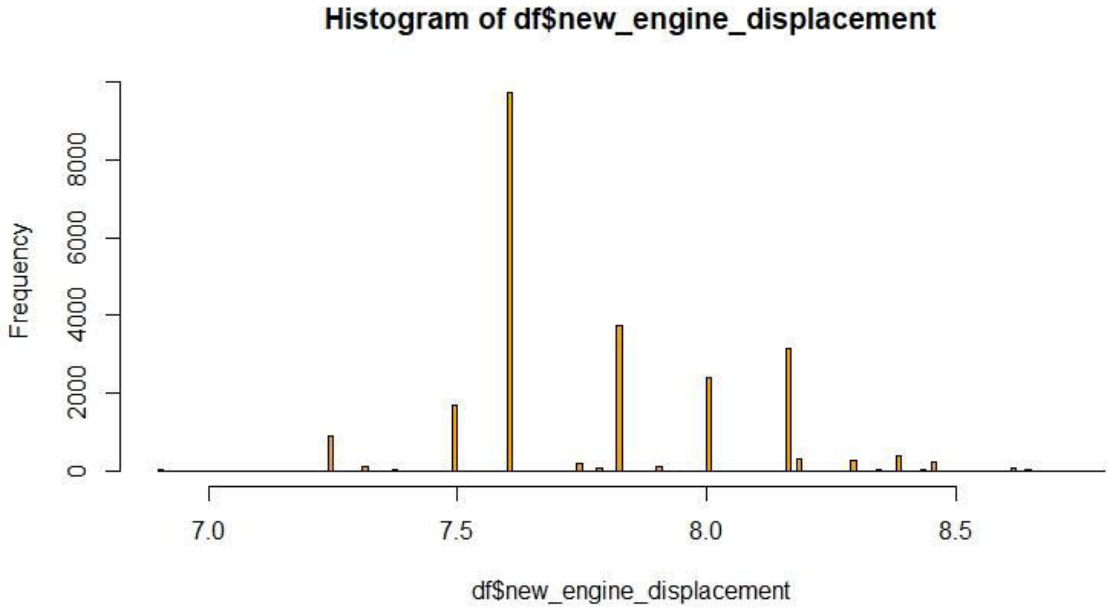
Solution: In a right skewed data, the median is smaller than the mean and in a left skewed data, the mean is smaller than the median. This difference should be significant to categorise a data as skewed. In *'used_car'* dataset, '*price'*, *'daysonmarket'*, *'engine_dsiplacement'*, *'mileage'* and *'length'* are right skewed as their median is significantly smaller than mean. The parameter *'width'* is left skewed as its mean is significantly smaller than median. We can apply log transformation to deal with skewness and compare the accuracy: whether it improved using transformed variables. If in case the model didn't perform well, we can always use the original parameters. I choose to apply log transformation to all the variables as it will transform the variables with smallest skewness into a normal shaped parameter. We should usually log transform all positive values, as suggest in this article (Andrew, 2019).

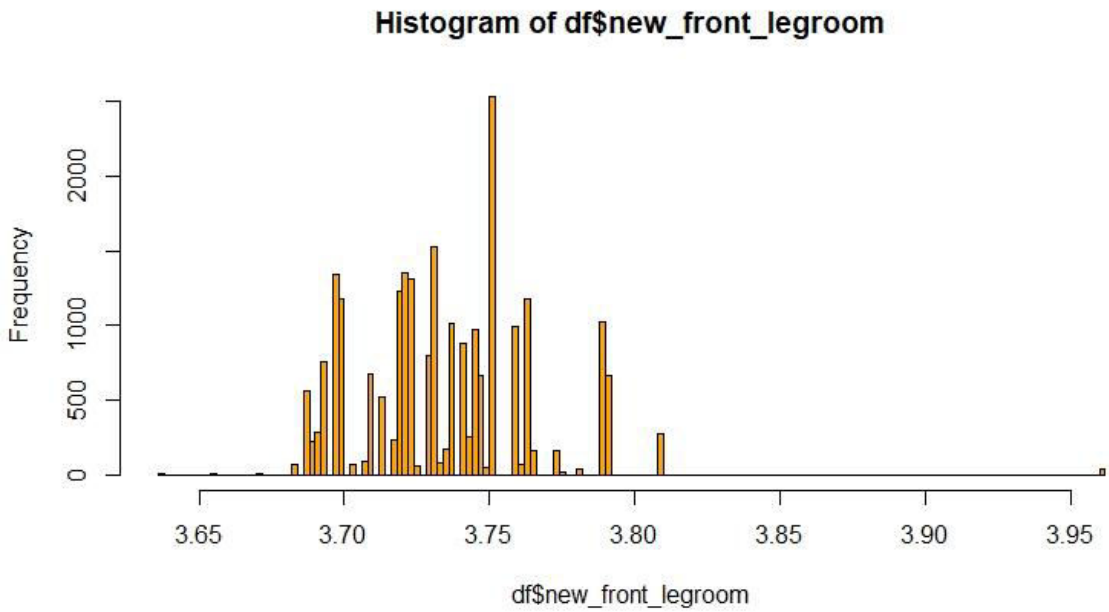
The histograms after applying log transformation are:

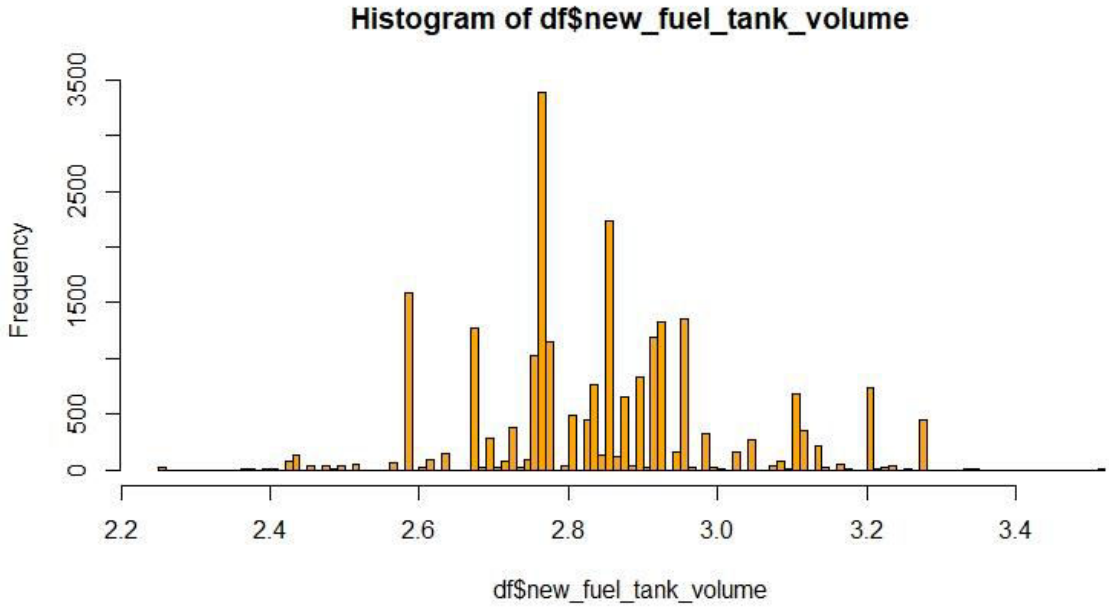

Histogram of df$new_back_legroom
Frequency
0
1000
2000
3000
3.3
3.4
3.5
3.6
3.7
3.8
df$new_back_legroom

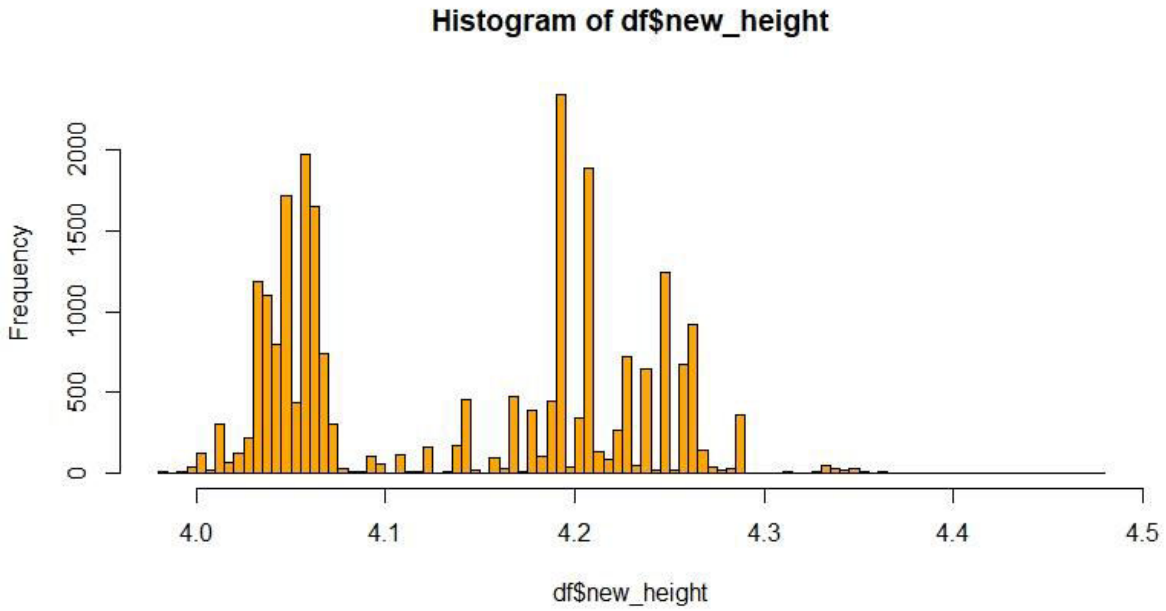

Histogram of df$new_city_fuel_economy
Frequency
0
1000
2000
3000
4000
2.5
3.0
3.5
4.0
df$new_city_fuel_economy

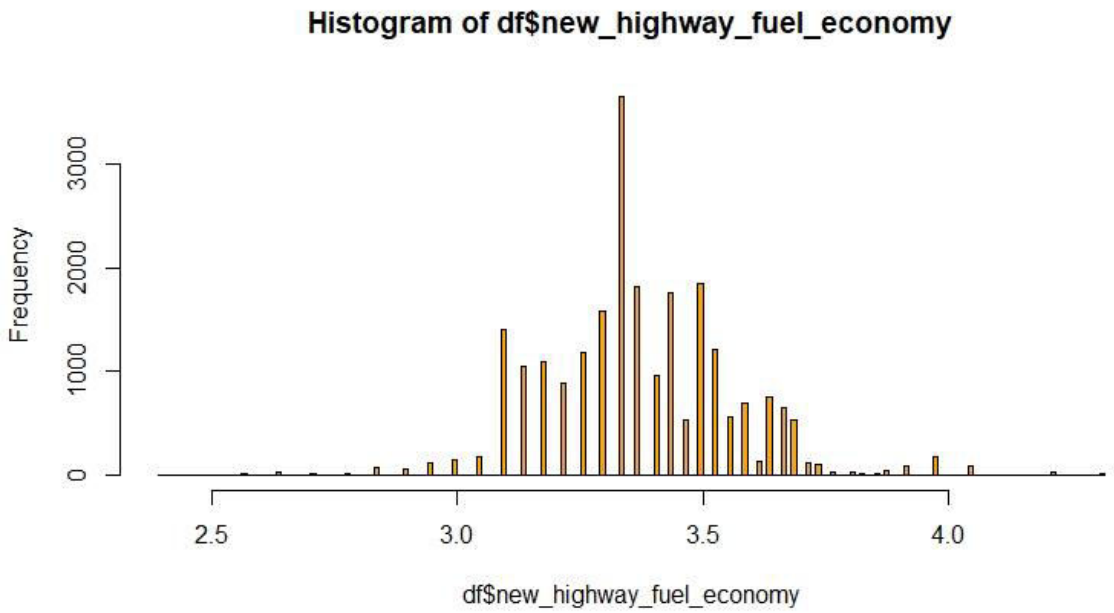

Histogram of df$new_daysonmarket
Frequency
0
200
400
600
800
0
2
4
6
df$new_daysonmarket

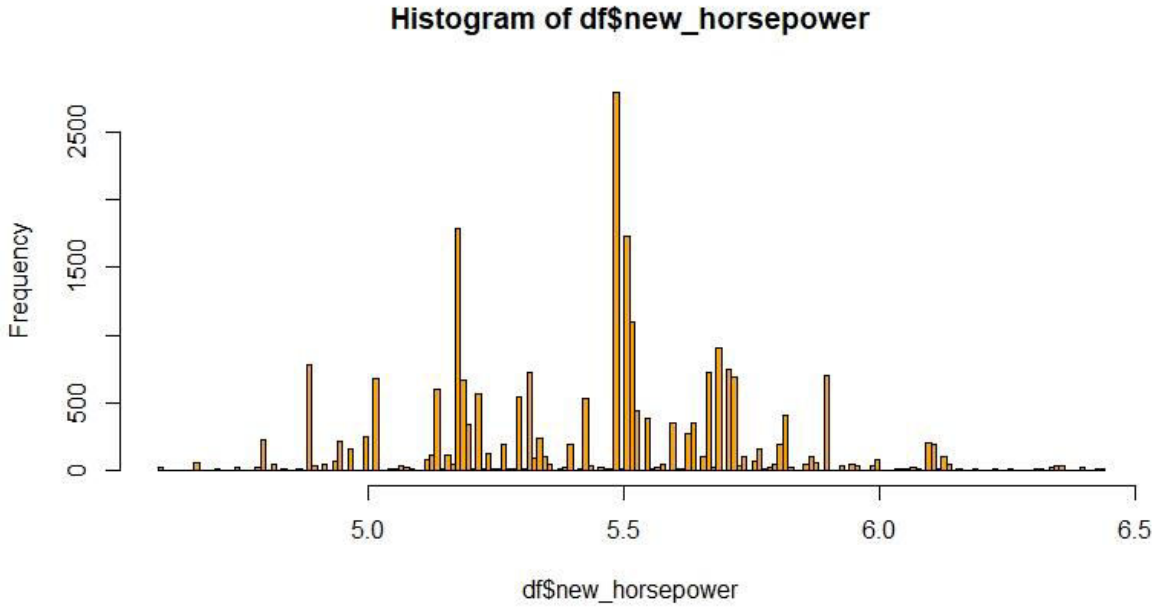

Histogram of df$new_engine_displacement
Frequency
0
2000
4000
6000
8000
7.0
7.5
8.0
8.5
df$new_engine_displacement

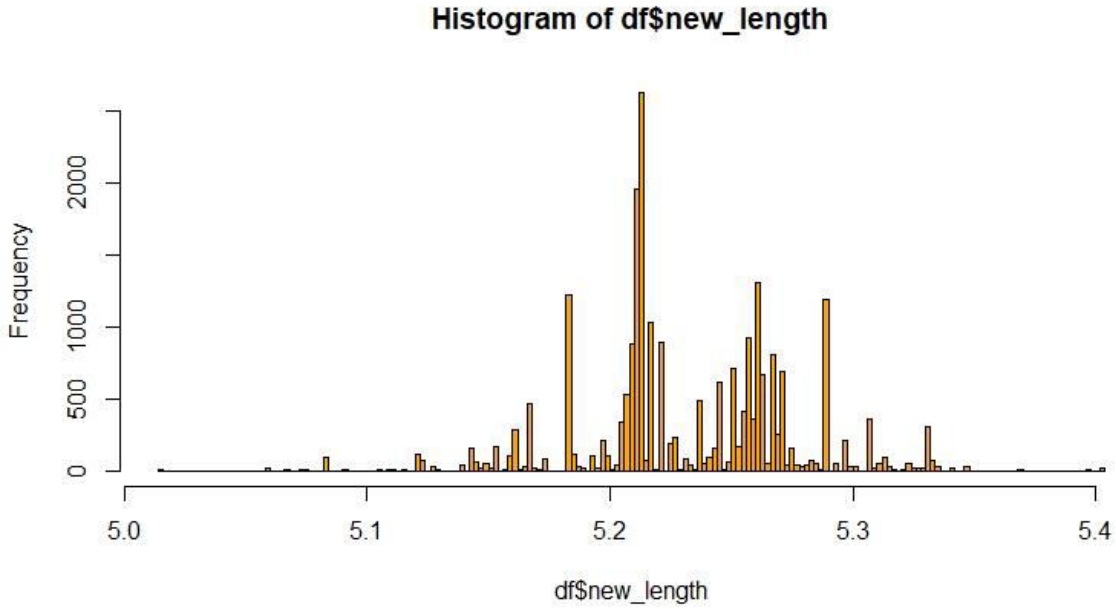

Histogram of df$new_front_legroom
Frequency
0
500
1000
2000
3.65
3.70
3.75
3.80
3.85
3.90
3.95
df$new_front_legroom

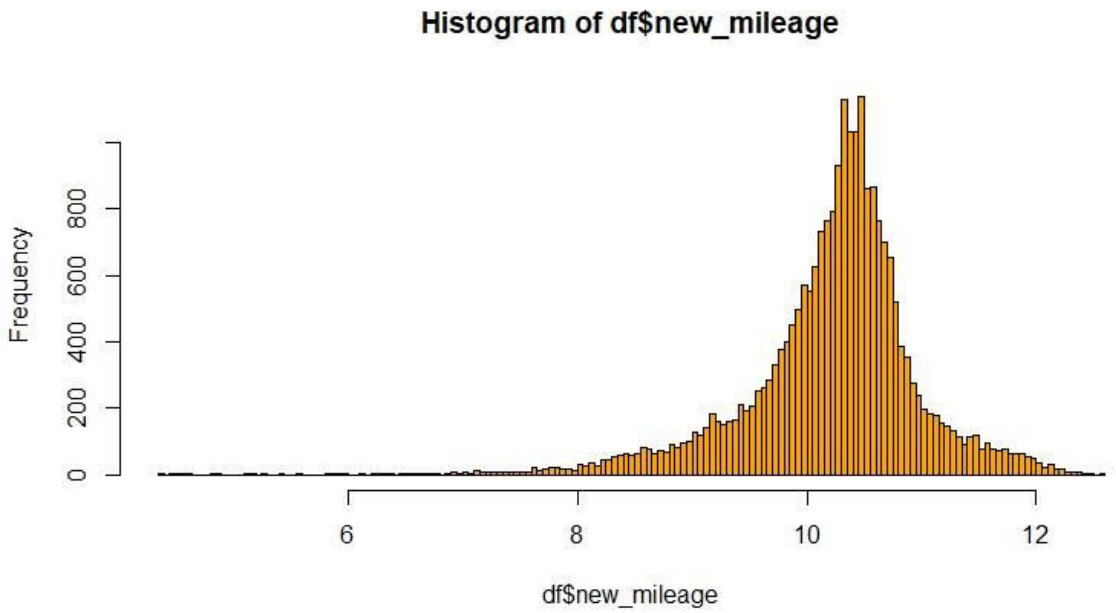

Histogram of df$new_fuel_tank_volume
Frequency
0
500
1500
2500
3500
2.2
2.4
2.6
2.8
3.0
3.2
3.4
df$new_fuel_tank_volume

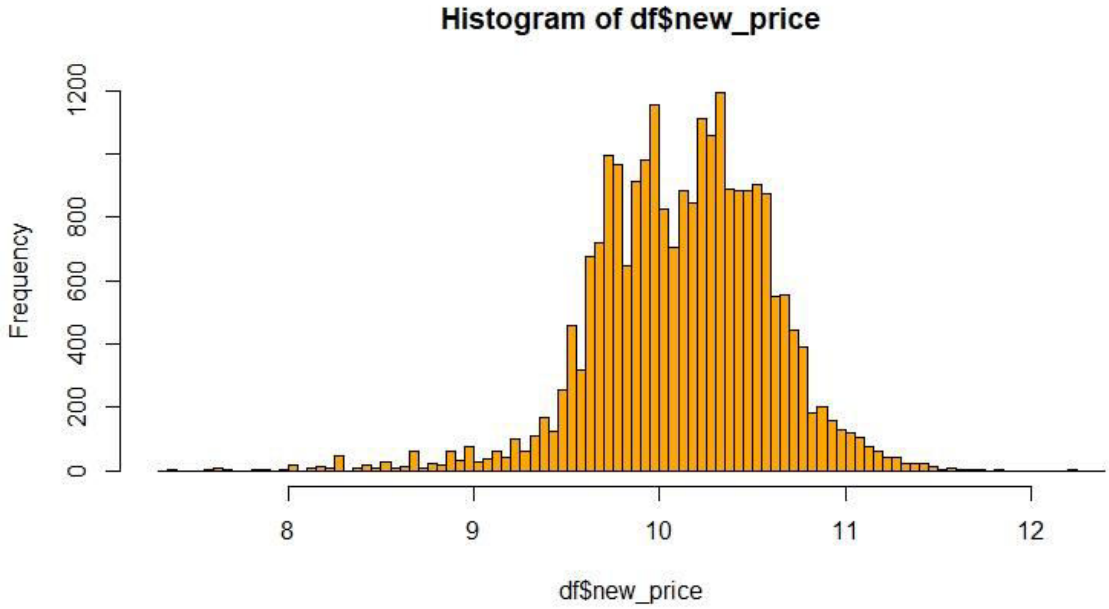

Histogram of df$new_height
Frequency
0
500
1000
1500
2000
4.0
4.1
4.2
4.3
4.4
4.5
df$new_height

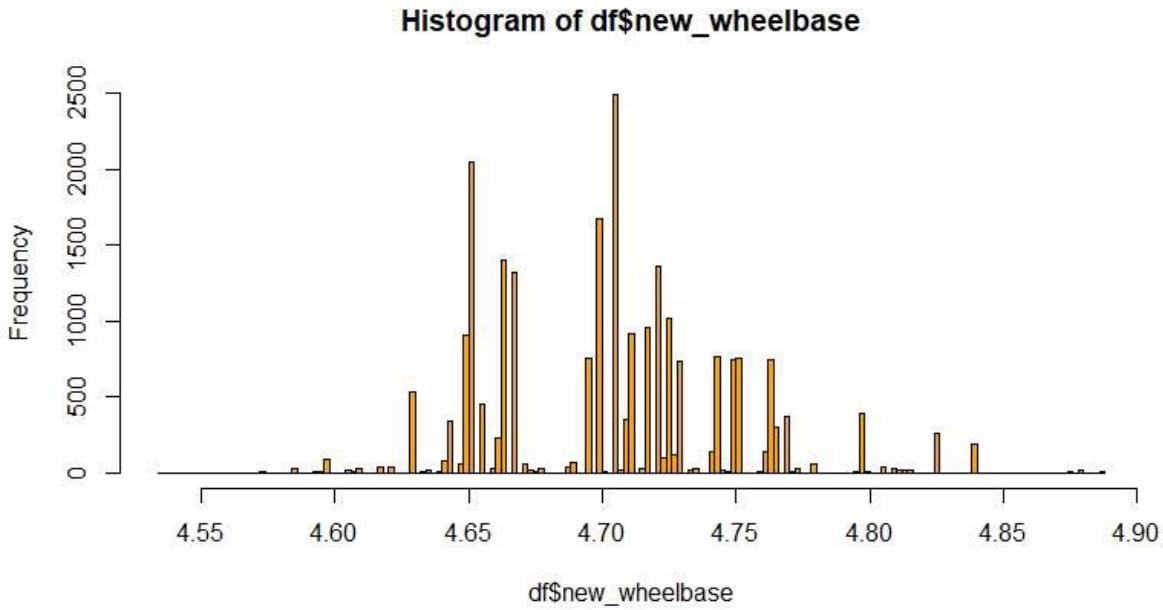

Histogram of df$new_highway_fuel_economy
Frequency
0
1000
2000
3000
2.5
3.0
3.5
4.0
df$new_highway_fuel_economy

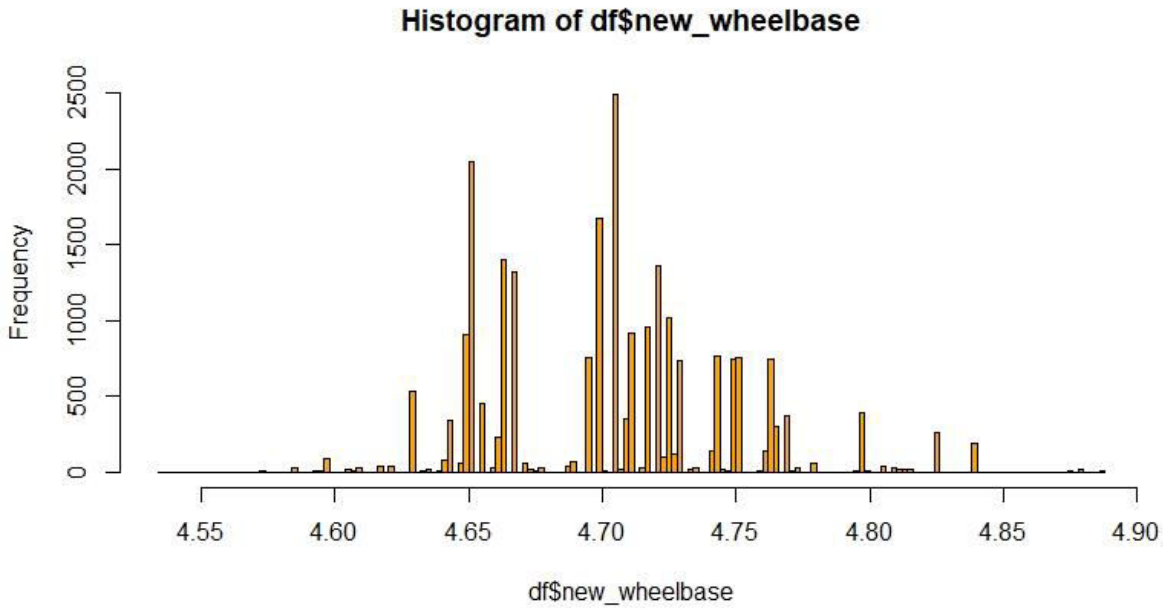

Histogram of df$new_horsepower
Frequency
0
500
1500
2500
5.0
5.5
6.0
6.5
df$new_horsepower


Histogram of df$new_length
Frequency
0
500
1000
2000
5.0
5.1
5.2
5.3
5.4
df$new_length


Histogram of df$new_mileage
Frequency
0
200
400
600
800
6
8
10
12
df$new_mileage


Histogram of df$new_price
Frequency
0
200
400
600
800
1200
8
9
10
11
12
df$new_price

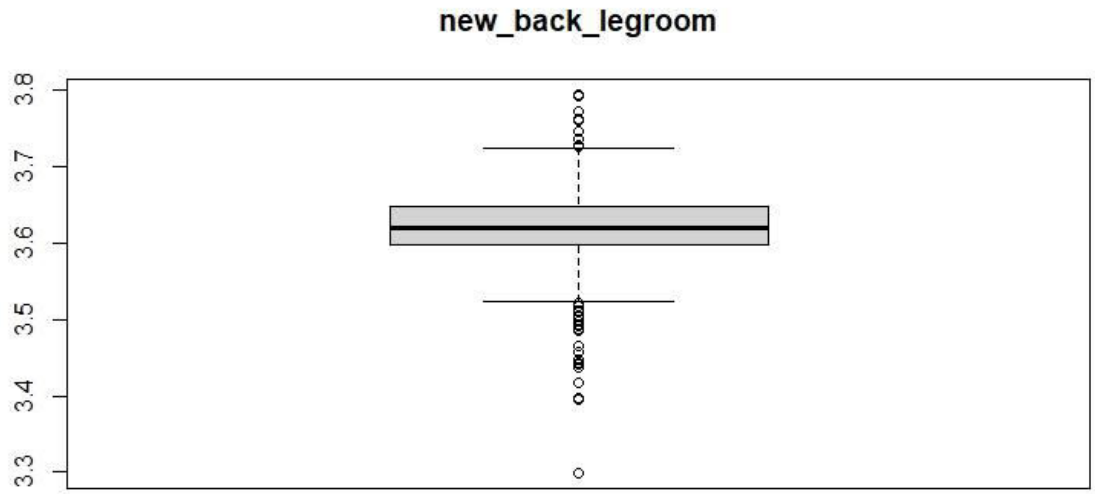

Histogram of df$new_wheelbase
Frequency
0
500
1000
1500
2000
2500
4.55
4.60
4.65
4.70
4.75
4.80
4.85
4.90
df$new_wheelbase

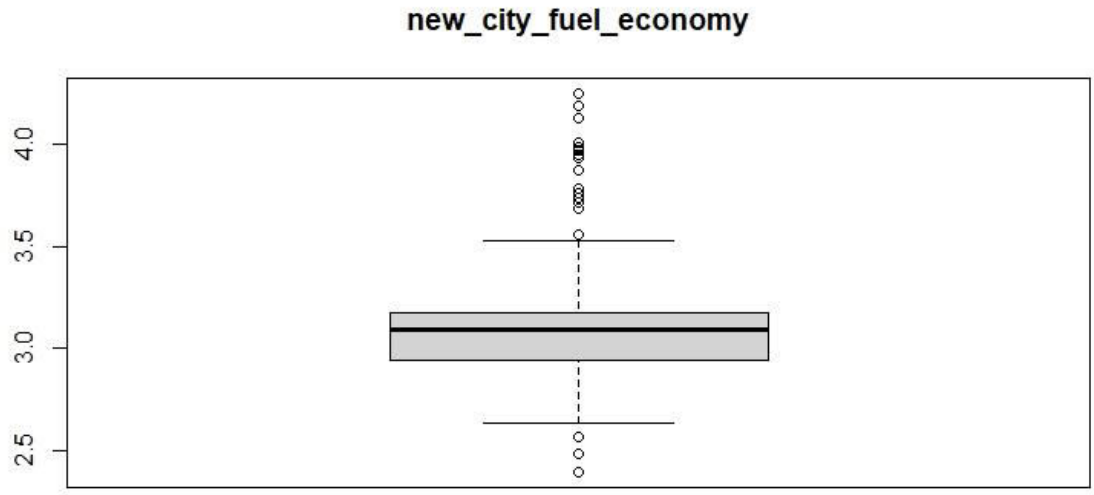

Histogram of df$new_wheelbase
Frequency
0
500
1000
1500
2000
2500
4.55
4.60
4.65
4.70
4.75
4.80
4.85
4.90
df$new_wheelbase

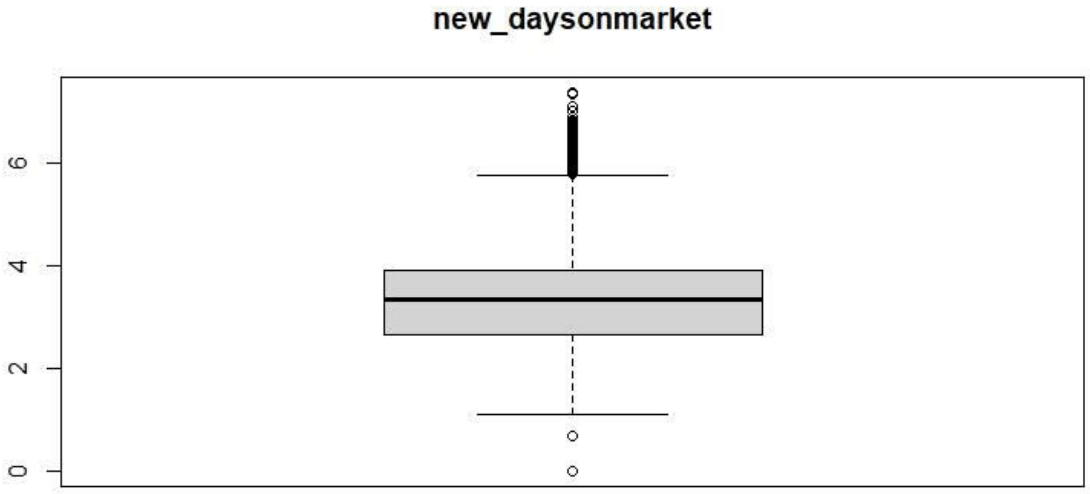
You can observe that the distribution of data is around centre after log transformation.

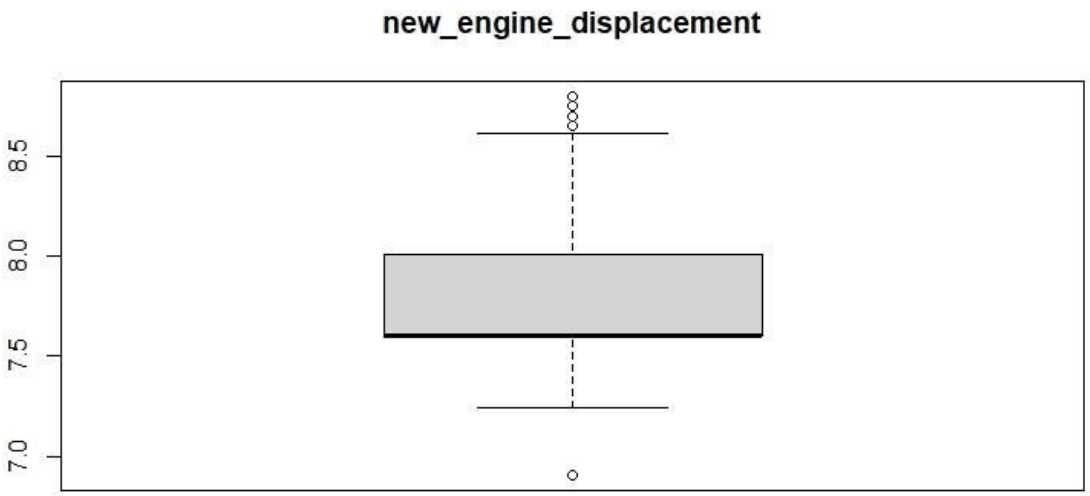
c. Are there any values that seem extreme?

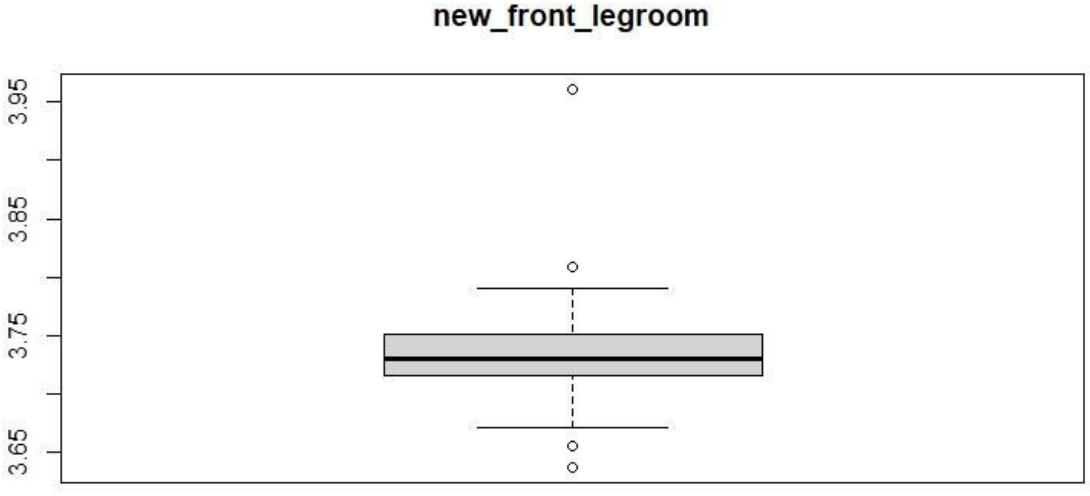
Solution: We can see outliers or extreme values in the dataset using a boxplot. Below are the boxplots for all the parameters in *'used_car'* dataset. I have used log transformed variables for plotting. We can see the circles outside the whiskers of the boxplot are the outliers. I explored whether eliminating outliers would help the analysis, but it eliminated most of the data points. So, I kept the parameters as it is.

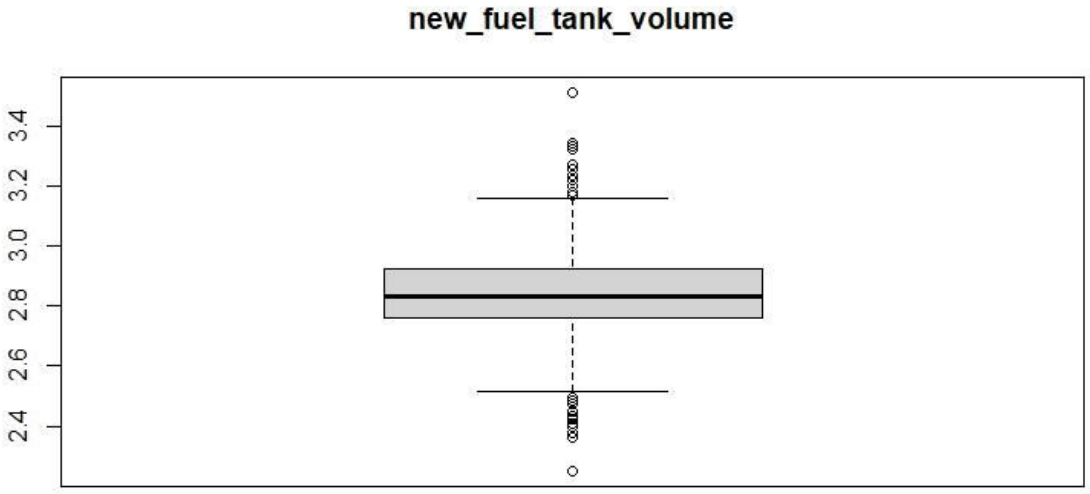


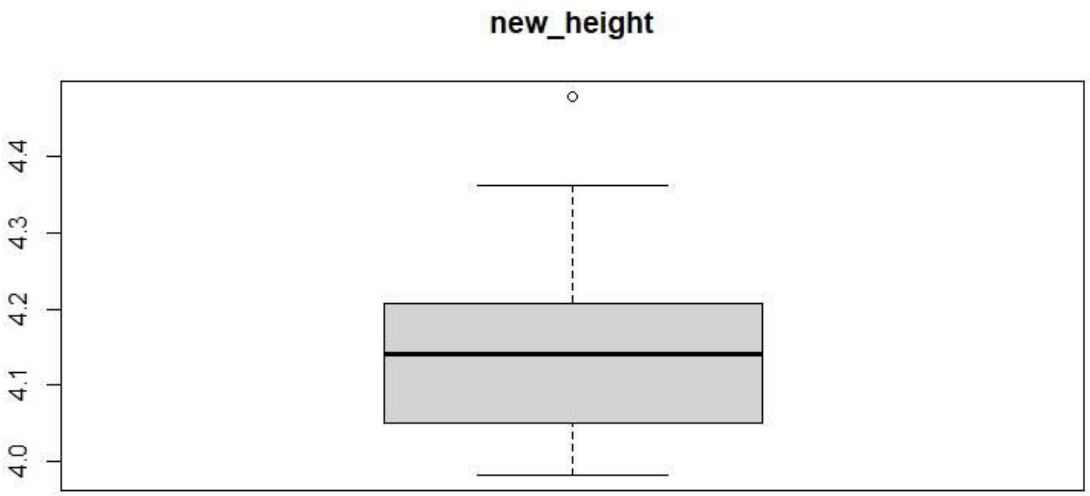


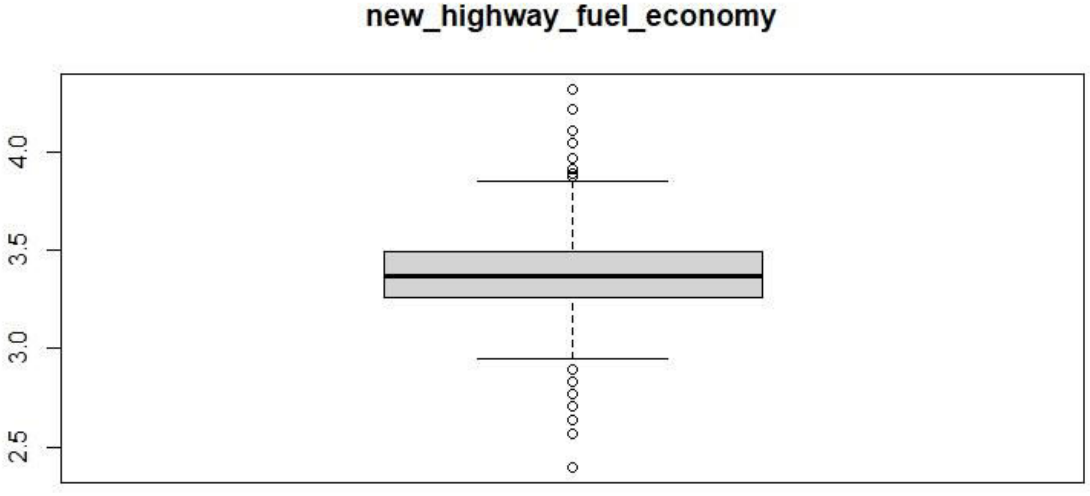

new_daysonmarket
0
2
4
6

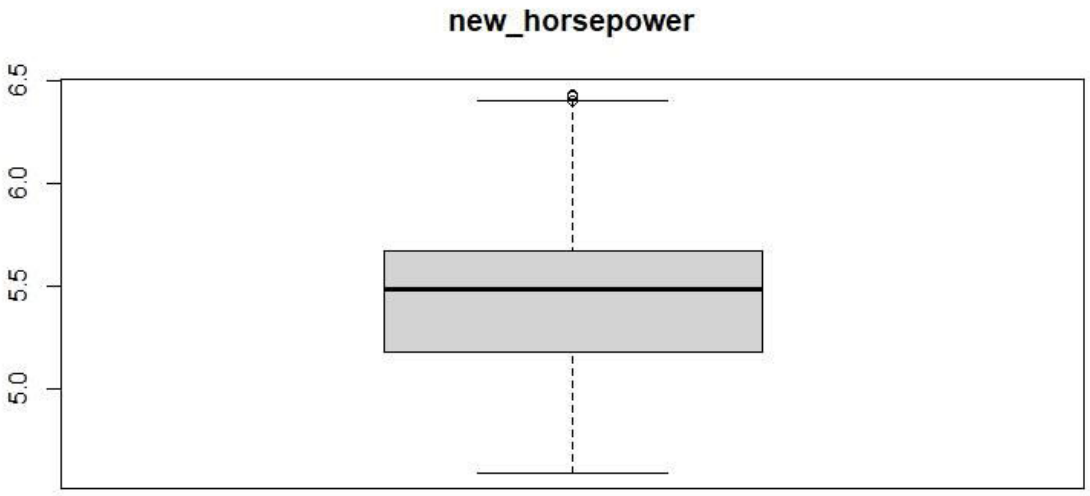

new_engine_displacement
7.0
7.5
8.0
8.5

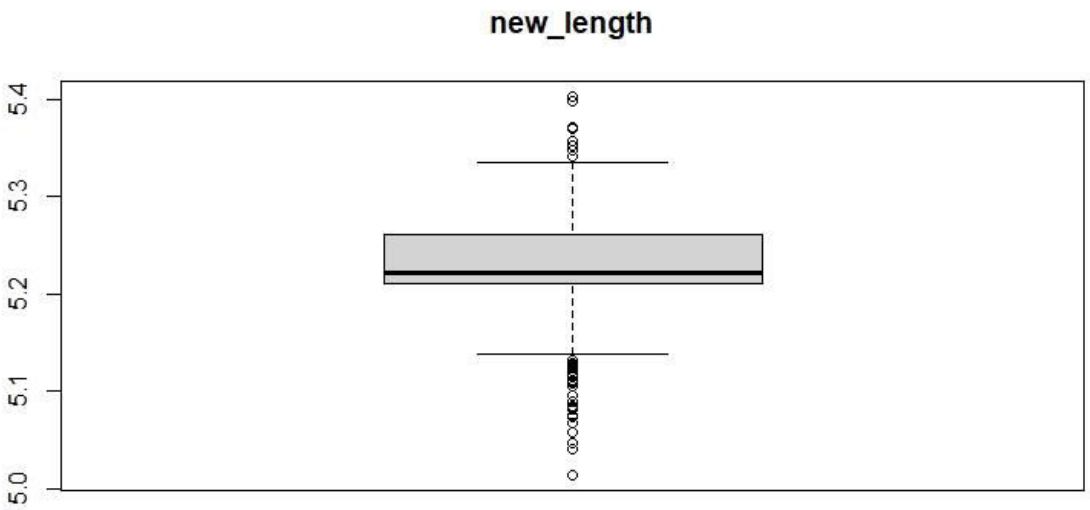

new_front_legroom
3.65
3.75
3.85
3.95

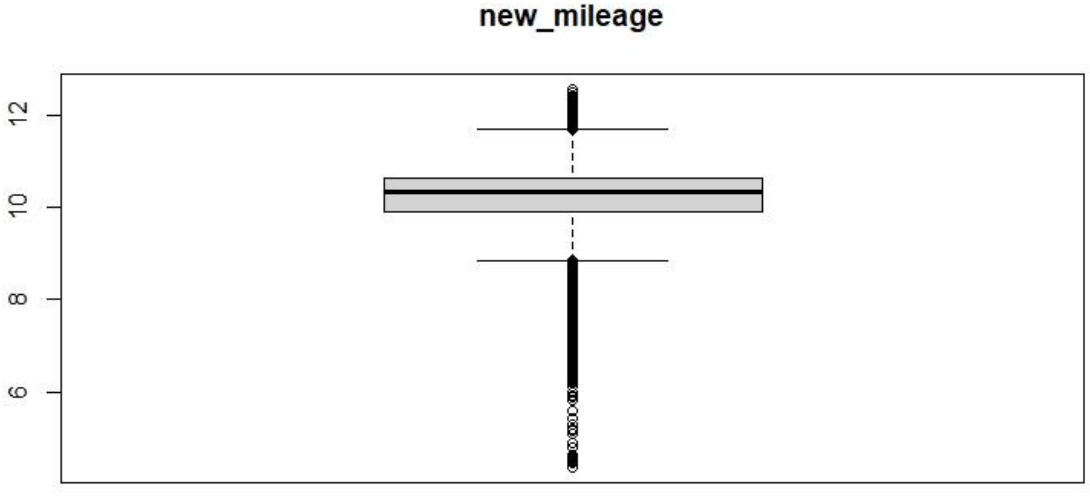

new_fuel_tank_volume
2.4
2.6
2.8
3.0
3.2
3.4

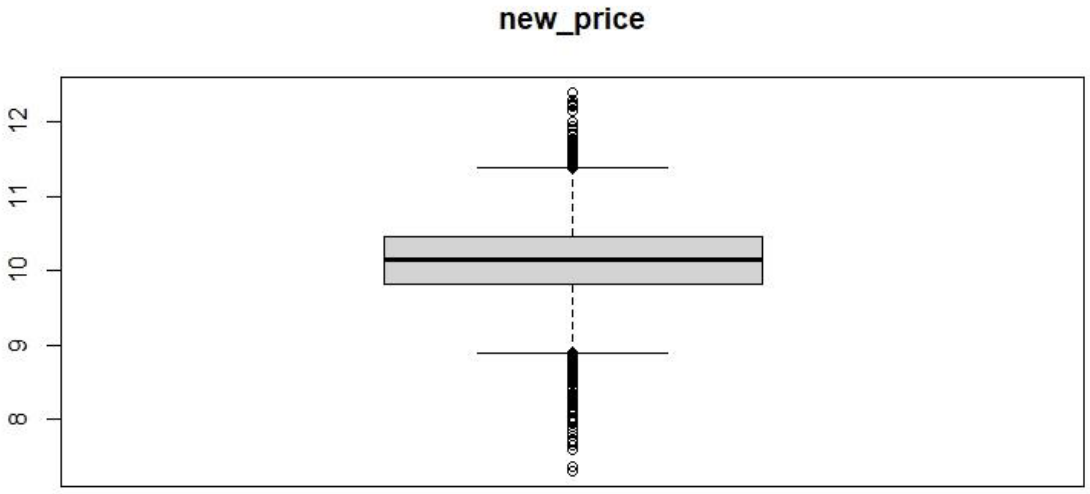

new_height
4.0
4.1
4.2
4.3
4.4

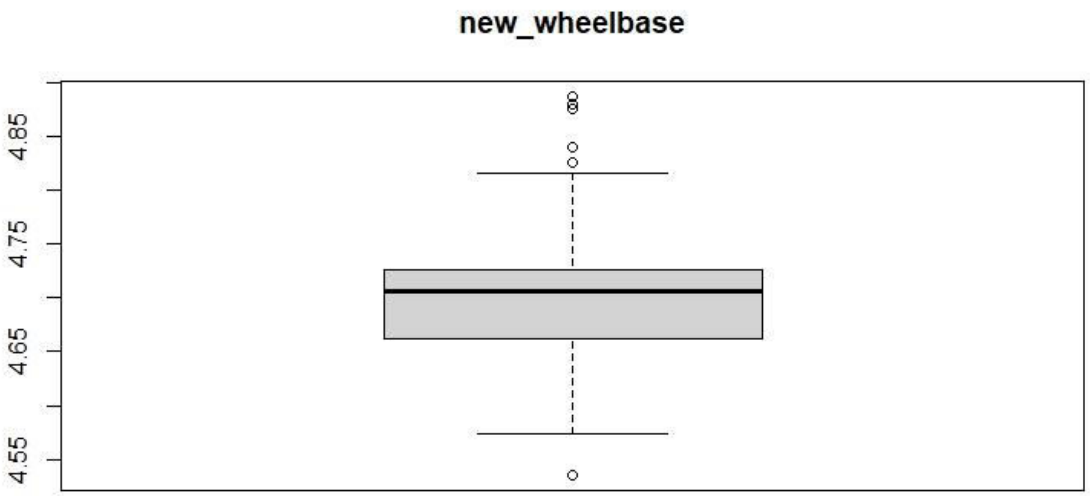

new_highway_fuel_economy
2.5
3.0
3.5
4.0

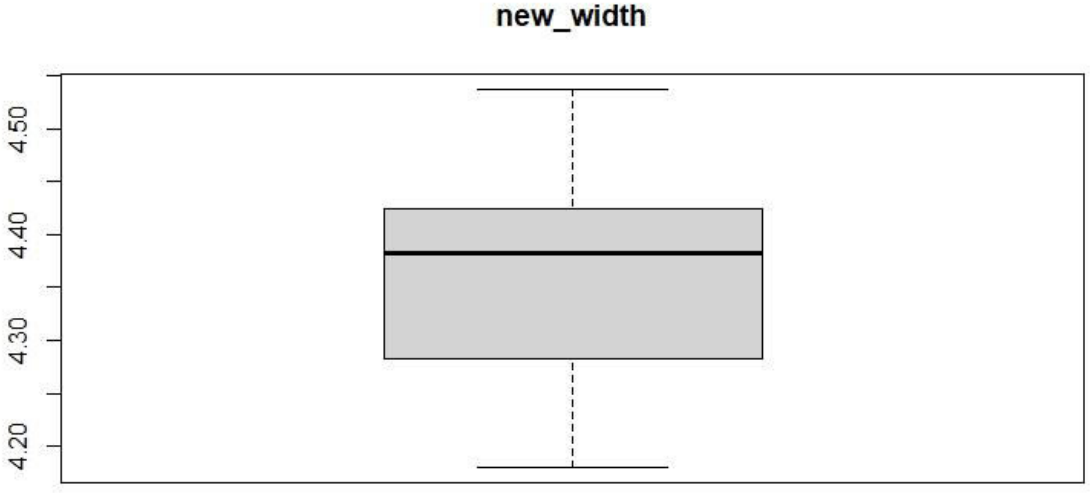

new_horsepower
6.5
6.0
5.5
5.0


new_length
5.4
5.3
5.2
5.1
5.0


new_mileage
12
10
8
6


new_price
12
11
10
9
8


new_wheelbase
4.85
4.75
4.65
4.55


new_width
4.50
4.40
4.30
4.20

Q4.

a. Which, if any, of the variables have missing values?

Solution: There are missing values in the *'used_cars'* data set. There are 120 missing values (*front_legroom: 7, city_fuel_economy: 5, mileage: 108*).

b. What are the methods of handling missing values?

Solution: We can consider three methods:

1. Replace NAs with a specific value, such as 0.

2. Delete records with NAs

3. Replace with mean of the column (can bias sample and misinterpretation)

4. If we have missing nominal variables, we need to use mode or we can use a special value such as unknown

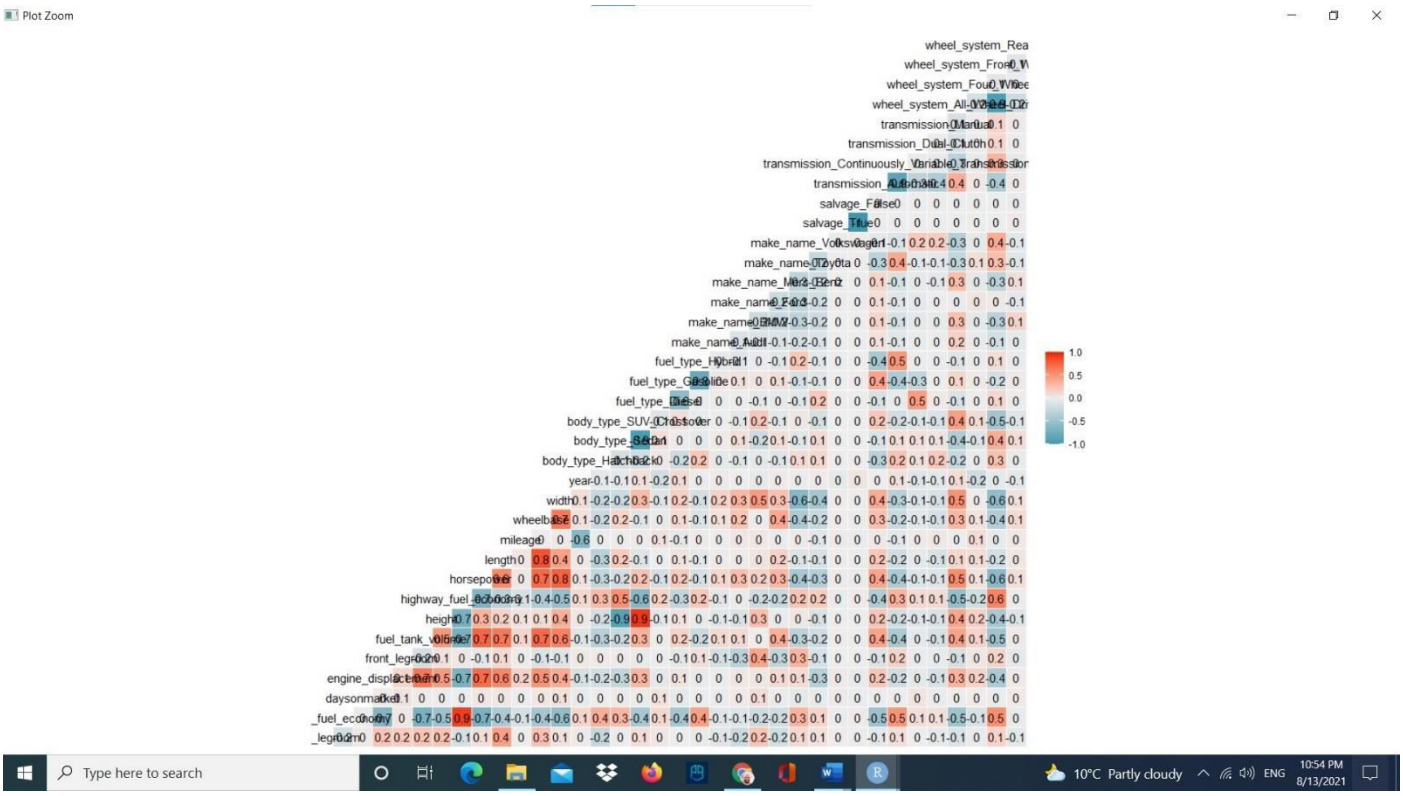
c. Apply the 3 methods of missing value and demonstrate the output (summary statistics and transformation plot) for each method in (4-b). (hint: the objective is to identify the impact of using each of the methods you mentioned in the 4-b on the summary statistics output above). Which method of handling missing values is most suitable for this data set? Discuss briefly referring to the data set.

Solution: I choose to remove the replace the NA values in front_legroom, city_fuel_economy and mileage with mean as the NA values in these parameters are very small (5, 108 and 7). This will not affect the overall dataset as the dataset is very large and it will also not introduce skewness as the values are replaced with mean value of the paramteres.

Q5.

a. Evaluate the correlations between the variables.

Solution: The evaluated correlation plot is:



We can see that there are some strong correlations in the data set (in red shade). Most of the parameters are neutral (in grey shade), while a few are negatively correlated as well (in blue shade).

b. Which variables should be used for dimension reduction and why? Carry out dimensionality reduction.

Solution: The following variables are highly correlated:

- city_fuel_economy
- highway_fuel_economy
- horsepower
- fuel_tank_volume
- width
- wheel_system_Front_Wheel_Drive
- wheelbase
- height
- transmission_Automatic
- body_type_SUV_Crossover
- fuel_type_Gasoline
- mileage
- salvage_False

We can remove one variable so the inter-correlation between variables will be minimum. Highly correlated variables have similar information, and this can bring down the performance of predictive models as described by (Sharma, 2020).

c. Explore the distribution of selected variables (from step 5-a) against the target variable. Explain.

Solution: We can see in the exploration that the mileage has the largest number of distinct values followed by target variable (price). The mean and median are almost same in each parameter. The target variable has second highest mean/median value after mileage (exception is the '*year'* which is an integer value and statistics are not true). Also, the variance is higher than target variable in case of *'daysonmarket'* parameter.

References

1. Ray, S. (2020, December 23). Simple Methods to deal with Categorical Variables in Predictive Modeling. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/
2. A. (2019, August 21). *You should (usually) log transform your positive data « Statistical Modeling, Causal Inference, and Social Science*. Https://Statmodeling.Stat.Columbia.Edu/. https://statmodeling.stat.columbia.edu/2019/08/21/you-should-usually-log-transform-your-positive-data/
3. Sharma, P. (2020, May 25). *The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/