Group 302: A Statistical Analysis of Used Car Sales Data

First Name	Last Name	Email (hawk.iit.edu)	Student ID
Arpitha	Jagadish	ajagadish@hawk.iit.edu	A20453142

Table of Contents

Table of Figures

1. Introduction

Used car sales is one of the largest turnover business in the automotive industry. There are numerous car companies evolved over the years all over the globe and Belarus auto market is one of the fastest growing vehicle markets in Europe, which is very well known for famous/prestigious car models and brands. In this project I am going to analyze the dataset collected from Belarus online catalog and verify the association between different factors and how these variables affect the selling price of the car. Also, to build a price-prediction model for vehicles sold by the online sellers.

As an aspiring car buyer, I always visit numerous websites to look for available car options. I wonder how the seller decides the car price and what are the variables that influence the selling cost. So, this analysis will help me to better understand the pricing strategy used by the car sellers. Also, many times when used car is sold, buyer or seller takes an advantage. So, the prediction model should help in optimizing the price, so that neither buyer nor seller will lose money.

2. Data

This dataset is collected from various popular online used car catalogs in Belarus (West Europe).

- Reference of this data is from Kaggle; Provided by Kirill Lepchenkov, who gathered the information from various web sources.
  Kaggle Link: https://www.kaggle.com/lepchenkov/usedcarscatalog

- The data set was collected in December 2019, which includes the 38,531 rows of car selling information with 29 influencing factors that affects the price of the car.

Following are the different variables listed in this data set.

- manufacturer_name: The name of car manufacturer.
- model_name: The name of the model.
- transmission: Type of the transmission.
- Color: Body color.
- odometer_value: Odometer state in kilometers.
- year_produced: The year the car has been produced.
- engine_fuel: Fuel type of the engine.
- engine_has_gas: Is the car equipped with propane tank and tubing?
- engine_type: Engine type.
- engine_capacity: The capacity of the engine in liters, numerical column.
- body_type: Type of the body (hatchback, sedan, etc.
- has_warranty: Does the car have warranty?
- State: New/owned/emergency. Emergency means the car has been damaged, sometimes severely.
- Drivetrain: Front/rear/all drivetrain, categorical column.
- price_usd: The price of a car as listed in the catalog in USD.

- is_exchangeable: If is exchangeable is True the owner of the car is ready to exchange this car to other cars with little or no additional payment.
- location_region: Categorical column, location_region is a region in Belarus where the car is listed for sale.
- number_of_photos: Number of photos the car has. numerical
- up_counter: Number of times the car has been upped, numerical.
- feature_0: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_1: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_2: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_3: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_4: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_5: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_6: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_7: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_8: Is the option like alloy wheels, conditioner, etc. is present in the car.
- feature_9: Is the option like alloy wheels, conditioner, etc. is present in the car.
- duration_listed: Number of days the car is listed in the catalog.

3. Problems to be Solved

- Whether there is a price difference between the automatic transmission cars and mechanical transmission cars.

- The average price of the cars produced between 2000-2010 are greater than cars produced between 1990-2000.

- Analyze how the average price of the car differs among different types of drive trains.

- Build the multiple linear regression model for used car price prediction

4. Solutions

- Whether there is a price difference between the automatic transmission cars and mechanical transmission cars.

In this statistical study, I am comparing the price difference between automatic transmission cars and mechanical transmission cars. Initially I draw the box plot to compare the mean price of automatic and manual transmission cars.

From the box plot,

- If the variance of sample is small, then I compare the median value from the box plot to see if there is a price difference between automatic transmission cars and mechanical transmission cars
- If the variance of sample is large, then It won't be reliable to compare the median values to do the comparison. So, in that case I prefer to use the hypothesis testing to do the analysis.

Therefore, I will use the box plot or two independent sample-two tail hypothesis testing to solve this research problem, as the samples are different and don't have any dependency on each other.

- The average price of the cars produced between 2000-2010 are greater than cars produced between 1990-2000.

In this research problem, I am comparing the price difference between the cars produced between 2000-2010 are greater than cars produced between 1990-2000. Initially I draw the box plot to compare the mean price of automatic and manual transmission cars.

From the box plot,

- If the variance of sample is small, then I compare the median value from the box plot to see if there average price of the cars produced between 2000-2010 are greater than cars produced between 1990-2000.

- If the variance of sample is large, then It won't be reliable to copmare the median values to do the comparison. So, in that case I prefer to use the hypothesis testing to do the analysis

  Therefore, I will use the box plot or two independent sample-one tail hypothesis testing to solve this research problem, as the samples are not same, and we are comparing to see if one thing is greater than the other.

- Analyze how the average price of the car differs among different types of drive trains

From the dataset we see there are 3 different types of drive trains- Front, Rear and All. Initially I draw the box plot to compare the mean price of front, rear and all-wheel drive train cars.

From the box plot,

- If the variance of sample is small, then I compare the median value from the box plot to see if how the average price of the car differs among different types of drive trains.

- If the variance of sample is large, then It won't be reliable to compare the median values to do the comparison. So, in that case I prefer to use the hypothesis testing to do the analysis

  Therefore, I will use the box plot or ANOVA regression model to solve this research problem, as there I am comparing more than two samples in this analysis. In this model, I consider the Price_usd as dependent variable and different types of drive train as independent variable.

  Steps for ANOVA regression model:

- Pre-processing the data

    - Fill out any missing values
    - Assign price values for different types of drive train
- Build ANOVA linear regression model
- Goodness of Fit Test
- Residual Analysis
- Transformation of variable if any of the above condition is not satisfied.

Also, I will perform the Individual t-test to figure out the significant variables.

- Build the multiple linear regression model for used car price prediction

I am going to build Multi linear regression model for used car price prediction, to figure out what are the different variables that affect the used car price.

For building the regression model, I consider

- Price_usd as the dependent variable
- Following factors are considered as independent variables

```
> str(car_price)
'data.frame':   38531 obs. of  30 variables:
 $ manufacturer_name: Factor w/ 57 levels "Acura","Alfa Romeo",..: 48 48 48 48 48 48 48 48 48 48 ...
 $ model_name       : Factor w/ 1118 levels "100","1007","100NX",..: 773 773 518 613 669 773 518 669 773 518 ...
 $ transmission     : Factor w/ 2 levels "automatic","mechanical": 1 1 1 2 1 1 1 1 1 1 ...
 $ color            : Factor w/ 12 levels "black","blue",..: 9 2 8 2 1 9 1 9 5 9 ...
 $ odometer_value   : int  190000 290000 402000 10000 280000 132449 318280 350000 179000 571317 ...
 $ year_produced    : int  2010 2002 2001 1999 2001 2011 1998 2004 2010 1999 ...
 $ engine_fuel      : Factor w/ 6 levels "diesel","electric",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ engine_has_gas   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ engine_type      : Factor w/ 3 levels "diesel","electric",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ engine_capacity  : num  2.5 3 2.5 3 2.5 2.5 2.5 2.5 2.5 2.5 ...
 $ body_type        : Factor w/ 12 levels "cabriolet","coupe",..: 11 11 10 9 11 11 11 9 11 11 ...
 $ has_warranty     : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ state            : Factor w/ 3 levels "emergency","new",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ drivetrain       : Factor w/ 3 levels "all","front",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ price_usd        : num  10900 5000 2800 9999 2134 ...

 $ location_region  : Factor w/ 6 levels "Брестская обл.",..: 5 5 5 5 3 5 5 1 5 5 ...
 $ number_of_photos : int  9 12 4 9 14 20 8 7 17 8 ...
 $ up_counter       : int  13 54 72 42 7 56 147 29 33 11 ...
 $ feature_0        : logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
 $ feature_1        : logi  TRUE TRUE TRUE FALSE TRUE TRUE ...
 $ feature_2        : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
 $ feature_3        : logi  TRUE FALSE FALSE FALSE TRUE FALSE ...
 $ feature_4        : logi  FALSE TRUE FALSE FALSE TRUE FALSE ...
 $ feature_5        : logi  TRUE TRUE FALSE FALSE FALSE TRUE ...
 $ feature_6        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ feature_7        : logi  TRUE FALSE FALSE FALSE FALSE TRUE ...
 $ feature_8        : logi  TRUE FALSE TRUE FALSE FALSE TRUE ...
 $ feature_9        : logi  TRUE TRUE TRUE FALSE TRUE TRUE ...
 $ duration_listed  : int  16 83 151 86 7 67 307 73 87 43 ...
```

Figure 1: Structure of Dataset

- Among these independent variables, I am excluding the feature_0, feature_1, feature_2, feature_3, feature_4, feature_5, feature_6, feature_7, feature_8, feature_9 variables as the description of these variables is incomplete.
- Also, removing the manufacturer_name and model_name variables by creating a new variable called Manufacturing continent and assigning the manufacturer and models to this column.
- I will categorize 12 different colors based on its contrast – Dark, Light and Other

- To convert all the nominal variables to categorical variables, I will create dummy variables to differentiate each level separately into n columns and consider n-1 columns for the regression model.
- I will group the columns for Year_produced, Odometer_value, number_of_photos, up_counter and duration_listed variables for better analysis.

Following are the steps I will follow to build multi linear regression model for used car price prediction

- Data preprocessing
  - Optimizing variable columns
  - Translate the names for better understanding
  - Filling out missing values
  - Creating Dummy variables
  - Group the columns
- Build the multi linear regression model using Feature selection (N-fold cross validation, as the data size is small and to avoid multicollinearity problems)
- Finalizing the optimal model using RMSE and MAE values.
- Residual Analysis
- Transformation of variable if any of the above condition is not satisfied.
- Use the final regression model to predict the price of the used car.

5. Experiments and Results

5.1. Methods and Process

5.1.1 Whether there is a price difference between the automatic transmission cars and mechanical transmission cars.

In order to determine if there is a price difference between automatic transmission cars and mechanical transmission cars

Using Boxplots:

Load the Car price data info to the R environment and plot the transmission type v/s Price to compare the average price of automatic and mechanical transmission cars.

```
library(ggplot2)
ggplot(car_price,aes(x=car_price$transmission, y=car_price$price_usd)) +
geom_boxplot(aes(fill = car_price$transmission)) +
stat_summary(fun.y = mean, geom="point", size=2) +
xlab('transmission') +
ylab('Price') +
ggtitle('Price vs.Transmission Type')
```

Hypothesis testing:

In this problem, I used two tail hypothesis testing to test whether the type of transmission affect the selling price of the car. This belongs to two-sided two independent sample hypothesis testing.

Two-sided Two tail Hypothesis testing:

Null Hypothesis: Ho**:** For the null hypothesis we consider the average car price for both types of transmission are equal.
i.e μ1 = μ2

Alternate Hypothesis: Ha: So, the alternate hypothesis we consider the average car price for both types of transmission are not equal.
i.e μ1 ≠ μ2

I have conducted Z-statistical test to determine the hypothesis.
For Z-test, I have considered 95% confidence interval.
As this is two tail hypothesis, the level of significance (α/2) = 0.025

Z-test in R:

```
install.packages("PASWR2")
library(PASWR2)

# Assigning Mechanical Transmission Rows
mechanical_cars <- car_price[ which(car_price$transmission=="mechanical"),]

# Assigning Automatic Transmission Rows
automatic_cars <- car_price[ which(car_price$transmission=="automatic"),]

# Accessing Price of Mechanical Cars
mechanical_car_price <- mechanical_cars$price_usd

# Accessing Price of Automatic Cars
automatic_car_price <- automatic_cars$price_usd

# Z test for price prediction of transmission and automatic tranmission types
z.test(mechanical_car_price,automatic_car_price,alternative="two.sided",mu=0,sigma.x=sd(mechanical_car_price
),sigma.y=sd(automatic_car_price),conf.level=0.95,paired=F)
```

If the p-value of z-test is less than α/2, then we reject null hypothesis. As per the alternate hypothesis, the average price of the automatic transmission cars and mechanical transmission cars are not equal.

If the p-value of z-test is less than α/2, then we accept null hypothesis. As per the null hypothesis, the average price of the automatic transmission cars and mechanical transmission cars are equal.

5.1.2 The average price of the cars produced between 2000-2010 are greater than cars produced between 1990-2000.

In order to determine if the average price of 2000-2010 cars are greater than 1990-2000 cars.

Using Boxplots:

Load the Car price data info to the R environment and plot the price v/s year produced to compare the average price of 1990-2000 cars(Ninties cars) and 2000-2010 cars(Y2K cars)

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

boxplot(Y2k_cars_price,main="Year Produced vs Car Price",
      xlab="Year Produced(Y2k_cars_price)", ylab="Price")
boxplot(Ninties_cars_price,main="Year Produced vs Car Price",
      xlab="Year Produced(Ninties_cars)", ylab="Price"
```

Hypothesis testing:

In this problem, I used one tail hypothesis testing to test whether the year in which car produced affect the selling price of the car. This belongs to one-sided two independent sample hypothesis testing.

Two-sided One tail Hypothesis testing:

Null Hypothesis: $H0$: For the null hypothesis we consider, the average price of the cars produced in the year 2000-2010 is same as the cars produced in the year 1990-2000
i.e μ1 =μ2

Alternate Hypothesis: Ha: For the alternate hypothesis we consider, the average price of the cars produced in the year 2000-2010 are greater than cars produced in the year 1990-2000
i.e μ1 > μ2

I have conducted Z-statistical test to determine the hypothesis.
For Z-test, I have considered 95% confidence interval.
As this is one tail hypothesis, the level of significance (α) = 0.05

Z-test in R:

```
install.packages("PASWR2")
library(PASWR2)

# Assigning 2000-2010 manufactured Rows to Y2K_cars
Y2K_cars <- car_price[ which(car_price$year_produced %in%
c("2001","2002","2003","2004","2005","2006","2007","2008","2009","2010")),]

# Assigning 1990-2000 manufactured Rows to Ninties_cars
Ninties_cars <- car_price[ which(car_price$year_produced %in%
c("1991","1992","1993","1994","1995","1996","1997","1998","1999","2000")),]

#Accessing price of Y2K cars
Y2k_cars_price <- Y2K_cars$price_usd

#Accessing price of Ninties cars
Ninties_cars_price <- Ninties_cars$price_usd

#Z test for price prediction of Y2K cars and Ninties cars
z.test(Y2k_cars_price,Ninties_cars_price,alternative="greater",mu=0,sigma.x=sd(Y2k_cars_price),sigma.y=sd(Ninti
es_cars_price),conf.level=0.95,paired=F)
```

If the p-value of z-test is less than α, then we reject null hypothesis. As per the alternate hypothesis, the average price of the cars produced in the year 2000-2010 are greater than cars produced in the year 1990-2000.

If the p-value of z-test is less than α, then we accept null hypothesis. As per the null hypothesis, the average price of the cars produced in the year 2000-2010 is same as the cars produced in the year 1990-2000.

5.1.3 Analyze how the average price of the car differs among different types of drive trains

In order to determine if the average price differs among different types of drive train

Using Boxplots:

Load the Car price data info to the R environment and plot the price v/s drive train type to compare the mean average price of front train cars, rear train cars and all train cars.

```
library(ggplot2)

ggplot(car_price,aes(x=car_price$drivetrain, y=car_price$price_usd)) +
geom_boxplot(aes(fill = car_price$drivetrain)) +
xlab('Drive Train') +
ylab('Price') +
ggtitle('Price vs.Drive Train')
```

Anova F-Test hypothesis testing:

Null Hypothesis: H_0: For the null hypothesis we consider the average car price for all types of drive train are equal.
i.e μk= μ1 = μ2 = μ3

Alternate Hypothesis: H_a: So, the alternate hypothesis would be the average car price for all the drive train types are not equal
i.e μk≠μ1 ≠ μ2 ≠ μ3

I have considered 95% confidence interval. So, in this hypothesis, the level of significance (α) = 0.05
ANOVA in R:

```
anov=lm(car_price$price_usd~car_price$drivetrain)
summary(anov)

#Residual Analysis
#validating Constant- variance
plot(fitted(anov),rstandard(anov),main="p vs r")
abline(a=0,b=0,col='red')
```

```
#Normal Probability plot of residuals:
qqnorm(rstandard(anov))
qqline(rstandard(anov),col=2
```

If the p-value is less than α, then we reject null hypothesis. As per the alternate hypothesis, the average price of all the drive train types are not equal.

If the p-value is less than α, then we accept null hypothesis. As per the null hypothesis, the average price of all the drive train types are equal.

As the linear model didn't satisfy the normality condition, conducted Y transformation and rebuild the model as all the x variables are categorical.

Log transformation:

```
logtrans<-log10(car_price$price_usd)

anov_log=lm(logtrans~car_price$drivetrain)
summary(anov_log)

#Residual Analysis

plot(fitted(anov_log),rstandard(anov_log),main="p vs r")
abline(a=0,b=0,col='red')

qqnorm(rstandard(anov_log))
qqline(rstandard(anov_log),col=2)
```

Sqrt transformation:

```
sqtrans<-sqrt(car_price$price_usd)

anov_sqr=lm(sqtrans~car_price$drivetrain)
summary(anov_sqr)

#Residual Analysis

plot(fitted(anov_sqr),rstandard(anov_sqr),main="p vs r")
abline(a=0,b=0,col='red')

qqnorm(rstandard(anov_sqr))
qqline(rstandard(anov_sqr),col=2
```

Inverse transformation:

```
inver<- 1/(car_price$price_usd)

anov_inv=lm(inver~car_price$drivetrain)
summary(anov_inv)

#Residual Analysis
```

```
plot(fitted(anov_inv),rstandard(anov_inv),main="p vs r")
abline(a=0,b=0,col='red')

qqnorm(rstandard(anov_inv))
qqline(rstandard(anov_inv),col=2)
```

ANOVA- Individual t-Test:

Conducting individual t-test to figure out the variables which are significant in the analysis

I have considered,

DTALL ← car_price$drivetrainall
DTF ← car_price$drivetrainfront
DTR ← car_price$drivetrainrear

Considering DTALL as the base variable,

Fit Regression model for price_usd, by introducing 2 dummy variables

DTF = 1 for drivetrainfront and DTF = 0 otherwise
DTR = 1 for drivetrainrear and DTR = 0 otherwise

$$\text{Price\_usd} = \beta 0 + \beta 1\text{DTF} + \beta 2\text{DTR}$$

Group 1 (drivetrainall): All dummy variables are zero : μ1 = $\beta 0$
Group 2 (drivetrainfront): All dummy variables are zero except DTF : μ2= $\beta 0 + \beta 1$
Group 3 (drivetrainrear): All dummy variables are zero except DTR: μ3= $\beta 0 + \beta 2$

$\beta 1$ = μ2 − μ1 = estimated difference between the average price of drivetrainfront and drivetrainall
$\beta 2$ = μ3 − μ1 = estimated difference between the average hours of drivetrainrear and drivetrainall

Hypothesis testing: Individual t-test

Null Hypothesis: $H0$: The average price of dummy variable and base variable are not equal.

Alternate Hypothesis: Ha : The average price of dummy variable and base variable is equal.
As I considered 95% confidence for the hypothesis testing, level of significance (α) =1-0.95= 0.05

For all the variables which has P-value < α , we accept null hypothesis. So, as per null hypothesis these variables are significant, and their corresponding coefficients are not equal to zero.

For all the variables which has P-value > α , we reject null hypothesis. So, as per alternate hypothesis these variables are not significant, and their corresponding coefficients are equal to zero.

5.1.4 Build the multiple linear regression model for used car price prediction

Building the linear regression model to predict the used car price considering 29 other independent variables.

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + e_i \quad \text{for i=1,….,n}$$

I have used N-fold cross validation method to build the regression model as the sample size is small And I considered N as 10 to build this model.

Preprocessing:

- Assigning manufacturer names and model names to respective continents:

Using R:

```
# Pre-Processing (Assigning the manufacturers to corresponding continents)

#Listing unique manfacurer name
unique(car_price$manufacturer_name)

#Replacing manufacturer name yA3 with YA3 as they both are same, but displayed as duplicate
levels(car_price$manufacturer_name)[levels(car_price$manufacturer_name)=="УА3"]<-"YA3"

#Lisitng unique manufacturer name, yA3 has been changed to YA3
unique(car_price$manufacturer_name)

#Sorting and assigning different manufacturers to their corresponding continents-Asia,Europe and America based
on country of origin
car_price$Manufacturing_Continent<-ifelse(car_price$manufacturer_name %in%
c("Subaru","Kia","Acura","Lexus","Mitsubishi","SsangYong","Daewoo","Geely","Lifan","Toyota",
  "Great Wall","Hyundai","Nissan","Suzuki","Mazda","Infiniti","Chery","Honda"),"Asia",
ifelse(car_price$manufacturer_name %in% c("LADA", "YA3","Opel","Alfa Romeo","Dacia","Lancia",
" Rover","Muscovite","GAZ","Citroen","Mini","Jaguar","Porscche","BA3","Fiat","Renault","Seat","Volkswagen",
"Audi","ЗА3","ГА3",  "Volvo", "BMW" , "Land Rover", "Iveco", "Skoda","Saab", "Mercedes-
Benz","Peugeot"),"Europe","America"))

#Deleting manufacturer_name and model_name columns as we are seperating the cars with respect to continent
car_price$manufacturer_name <- NULL
car_price$model_name <- NULL


#Check the data set to see if manufacturer_name and model_name columns are deleted
#Viewing the dataset to check if the manufacturing continent column has been added

head(car_price)
```

- Optimizing the model by eliminating the variable feature 0 to feature 9:

Using R code:

```
#Pre-Processing(Optimizing the model by removing the unuseful data for prediction)

#Dimension of the car_price dataset(number of rows and columns)
dim(car_price)

#Deleting the columns feature_0 to feature_9, as the type of feature is not assigned to these columns in the dataset.So this wouldn't help to make the prediction.
car_price<-car_price[,-c(18:27)]

#Verifying the changes and dimension of the car_price(number of rows and columns) after the changes
head(car_price)
dim(car_price)
```

- Translating the location and engine fuel names for better understanding:

Using R code:

```
#Pre-Processing (Translating the region names to English)

#Translating the location information to English to better understand the names
levels(car_price$location_region)[levels(car_price$location_region)=="Минская обл."]<-"Minsk_Region"
levels(car_price$location_region)[levels(car_price$location_region)=="Гомельская обл."]<-"Gomel_Region"
levels(car_price$location_region)[levels(car_price$location_region)=="Брестская обл."]<-"Brest_Region"
levels(car_price$location_region)[levels(car_price$location_region)=="Могилевская обл."]<-"Mogilev_Region"
levels(car_price$location_region)[levels(car_price$location_region)=="Гродненская обл."]<-"Grodno_Region"
levels(car_price$location_region)[levels(car_price$location_region)=="Витебская обл."]<-"Vitebsk_Region"

#Renaming the engine_fuel names
levels(car_price$engine_fuel)[levels(car_price$engine_fuel)=="hybrid-diesel"]<-"hybrid_diesel"
levels(car_price$engine_fuel)[levels(car_price$engine_fuel)=="hybrid-petrol"]<-"hybrid_petrol"

#Viewing the dataset to check if the location information is updated with English names
head(car_price)
```

The original location names were in German and it was difficult to understand. So, translated all the names to English.

- Assigning different colors based on contrast

Using R-Code

```
#Pre-Processing(Assigning different colors based on contrast)

#listing unique colors of the car
unique(car_price$color)

#Based on the contrast, assigning the colors as Dark, Light and Other
car_price$color<-ifelse(car_price$color %in% c("silver","yellow","white"),"Light",
 ifelse(car_price$color %in% c("red","black","grey","brown","voilet","orange","green","blue"),"Dark", "Other"))

#Viewing the dataset to check if the color information is updated
head(car_price)
```

The 12 different colors are separated as dark, light and other based on their contrast.

- Finding missing variables

Using R Code:

```
#Pre-Processing(Filling the missing values)

# Checking for missing values
na_count <- sapply(car_price, function(y) sum(length(which(is.na(y)))))

#column with the missing values
na_count

#Though there is no fuel capacity for the electric cars, trying to fill it with the appropriate value by taking average value
car_price$engine_capacity = ifelse(is.na(car_price$engine_capacity), ave(car_price$engine_capacity, FUN= function(x) mean(x, na.rm=T)), car_price$engine_capacity)

sum(is.na(car_price$engine_capacity))

#Verifying the columns after the changes
na_count2 <- sapply(car_price, function(y) sum(length(which(is.na(y)))))
na_count2
```

The electric cars were not having fuel capacity information. Filled the missing values using the mean value.

- Creating dummy variables for categorical variables:

Using R Code:

```
#Pre-Processing(Creating dummy variables)

library(dummies)

#Checking number of rows and columns before creating dummy variables
dim(car_price)

# Creating Dummy variables for transmission
car_price <- dummy.data.frame(car_price, names=c("transmission"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (2)]

# Creating Dummy variables for color
car_price <- dummy.data.frame(car_price, names<-c("color"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (2)]

# Creating Dummy variables for engine fuel
car_price <- dummy.data.frame(car_price, names<-c("engine_fuel"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (7)]

# Creating Dummy variables for engine has gas
car_price <- dummy.data.frame(car_price, names<-c("engine_has_gas"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (12)]

# Creating Dummy variables for engine type
car_price <- dummy.data.frame(car_price, names<-c("engine_type"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (13)]

# Creating Dummy variables for body type
car_price <- dummy.data.frame(car_price, names=c("body_type"))
# Removing one dummy variable(N-1)
car_price <- car_price [, -c (15)]

# Creating Dummy variables for has warranty
car_price <- dummy.data.frame(car_price, names<-c("has_warranty"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (26)]
# Creating Dummy variables for state
car_price <- dummy.data.frame(car_price, names<-c("state"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (27)]
```

```
# Creating Dummy variables for drive train
car_price <- dummy.data.frame(car_price, names<-c("drivetrain"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (31)]


# Creating Dummy variables for is exchangeable
car_price <- dummy.data.frame(car_price, names<-c("is_exchangeable"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (32)]

# Creating Dummy variables for location region
car_price <- dummy.data.frame(car_price, names=c("location_region"))
#Removing one dummy variable(N-1)
car_price <- car_price [, -c (33)]

## Creating Dummy variables for manufacturing continent
car_price <- dummy.data.frame(car_price, names=c("Manufacturing_Continent"))
# Removing one dummy variable(N-1)
car_price <- car_price [, -c (41)]

#Verifying the data set after creating dummy variables and checking number of rows and columns after creating
dummy variables
head(car_price)
dim(car_price)
```

Created dummy variables for all the categorical variables and used (N-1) variables for model creation.

- Dividing the column values into different intervals and creating dummy variables

Using R Code:

```
#Though the variables year, odometer, number of photos and up counter are numerical variables, it is treated as
categorical and converted it to dummy variables in the regression model

#Dividing the years into 4 groups

#1940-1960 is 40_60

#1960-1980 is 60_80

#1980-2000 is 80_00

#2000-2020 is 00_20

car_price$year_produced<-cut(car_price$year_produced,breaks=c(1940,1960,1980,2000,2020),labels =
c('40_60','60_80','80_00','00_20'))

car_price$year_produced[1:10]

#Creating dummy variables and removing one dummy variable(N-1)
```

```
car_price <- dummy.data.frame(car_price, names=c("year_produced"))

car_price <- car_price [, -c (5)]
```

Divided the years from 1940 to 2020 into 4 groups and created dummy variables for each group variable and used (N-1) variables to create the model.

```
#Dividing the odometer value into 4 groups

# Creating Dummy variables
car_price$odometer_value<-cut(car_price$odometer_value,
                    quantile(car_price$odometer_value, probs = c(0, .25, .50,.75, 1)),
                    labels = c('grp1','grp2','grp3', 'grp4'),
                    include.lowest = TRUE)

car_price$odometer_value[1:10]

#Creating dummy variables and removing one dummy variable(N-1)
car_price <- dummy.data.frame(car_price, names=c("odometer_value"))
car_price <- car_price [, -c (4)]
```

Divided the odometer values into 4 groups by considering 25% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

```
#Dividing the number of photos into 2 groups

car_price$number_of_photos<-cut(car_price$number_of_photos,breaks=c(0,43,86))

car_price$number_of_photos[1:10]

#Creating dummy variables and removing one dummy variable(N-1)
car_price <- dummy.data.frame(car_price, names=c("number_of_photos"))
car_price <- car_price [, -c (43)]
```

Divided the number of photos into 2 groups by considering 50% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

```
#Dividing the up counter into 4 groups

car_price$up_counter<-cut(car_price$up_counter,breaks=c(0,465,931,1396,1861))

car_price$up_counter[1:10]

#Creating dummy variables and removing one dummy variable(N-1)
car_price <- dummy.data.frame(car_price, names=c("up_counter"))
car_price <- car_price [, -c (46)]
```

Divided the up counter into 4 groups by considering 25% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

```
#Dividing the duration listed into 4 groups

car_price$duration_listed<-cut(car_price$duration_listed,    quantile(car_price$duration_listed, probs = c(0, .25,
.50,.75, 1)), labels = c('grp1','grp2','grp3', 'grp4'), include.lowest = TRUE)


car_price$duration_listed[1:10]

#Creating dummy variables and removing one dummy variable(N-1)
car_price <- dummy.data.frame(car_price, names=c("duration_listed"))
car_price <- car_price [, -c (48)]
```

Divided the up counter into 4 groups by considering 25% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

- Using Feature selection to build the regression model

We use N-fold cross validation method to build regression model for life expectancy with N value as 10. We are considering Backward Selection and Forward Selection and stepwise to build the model.

Using R Code:

```
set.seed(10001)
train.control <- trainControl(method = "cv", number = 10)

#Backward Selection
# Train the model backward selection
model_backward <- train(price_usd~., data = car_price, method = "leapBackward",
                        trControl = train.control)
# Summarize the results
print(model_backward)

#Forward selection
# Train the model forward selection
model_forward <- train(price_usd~., data = car_price, method = "leapForward",
                       trControl = train.control)
# Summarize the results
print(model_forward)

#Stepwise
# Train the model stepwise selection
model_step <- train(price_usd~., data = car_price, method = "leapSeq",
                    trControl = train.control)
#Summarize the results
print(model_step)
```

Regression analysis:

```
#checking for constant variance
plot(predict(model_step),residuals(model_step),main="Predicted vs Residuals")
abline(a=0,b=0,col='red')

#linear plot for each x variable
plot(car_price$year_produced00_20,y=residuals(model_step),main="residual Vs year produced")
abline(a=0,b=0,col='red')

plot(car_price$transmissionautomatic,y=residuals(model_step),main="residual Vs transmission automatic")
abline(a=0,b=0,col='red')

plot(car_price$drivetrainfront,y=residuals(model_step),main="residual Vs drive train front")
abline(a=0,b=0,col='red')

plot(car_price$body_typesedan,y=residuals(model_step),main="residual Vs body type sedan")
abline(a=0,b=0,col='red')

plot(car_price$location_regionVitebsk_Region,y=residuals(model_step),main="residual Vs location region Vitebsk
Region")
abline(a=0,b=0,col='red')

#normality test
qqnorm(residuals(model_step))
qqline(residuals(model_step),col=2)
```

After transformation:

```
#Stepwise
# Train the model stepwise selection
model_step2 <- train(price_usd~., data = car_price, method = "leapSeq",
                     trControl = train.control)
#Summarize the results
print(model_step2)
```

Regression analysis:

```
#Verifying constant variance
plot(predict(model_step2),residuals(model_step2),main="Predicted Vs Residuals")
abline(a=0,b=0,col='red')

#Linearity check for each X variable
plot(car_price$year_produced00_20,y=residuals(model_step2),main="residual Vs year produced")
abline(a=0,b=0,col='red')

plot(car_price$body_typesedan,y=residuals(model_step2),main="residual Vs body type")
abline(a=0,b=0,col='red')

plot(car_price$drivetrainfront,y=residuals(model_step2),main="residual Vs drive train front")
```

```
abline(a=0,b=0,col='red')

plot(car_price$has_warrantyTRUE,y=residuals(model_step2),main="residual Vs has warranty")
abline(a=0,b=0,col='red')

plot(car_price$transmissionautomatic,y=residuals(model_step2),main="residual Vs transmission")
abline(a=0,b=0,col='red')

#Normality Test
qqnorm(residuals(model_step2))
qqline(residuals(model_step2),col=2)
```

After removing outliers:

```
#Stepwise
# Train the model stepwise selection
model_step3 <- train(price_usd~., data = no_outliers, method = "leapSeq",
                     trControl = train.control)
#Summarize the results
print(model_step3)
```

Regression analysis:

```
#Verifying constant variance
plot(predict(model_step3),residuals(model_step3),main="p vs r")
abline(a=0,b=0,col='red')

#Linearity check for each X variable
plot(no_outliers$year_produced00_20,y=residuals(model_step3),main="residual Vs year produced")
abline(a=0,b=0,col='red')

plot(no_outliers$body_typesuv,y=residuals(model_step3),main="residual Vs body type")
abline(a=0,b=0,col='red')

plot(no_outliers$statenew,y=residuals(model_step3),main="residual Vs drive train front")
abline(a=0,b=0,col='red')

plot(no_outliers$is_exchangeableTRUE,y=residuals(model_step3),main="residual Vs is exchangable")
abline(a=0,b=0,col='red')

plot(no_outliers$transmissionautomatic,y=residuals(model_step3),main="residual Vs transmission")
abline(a=0,b=0,col='red')

#Normality Test
qqnorm(residuals(model_step3))
qqline(residuals(model_step3),col=2)
```

5.2. Evaluations and Results

5.2.1 Whether there is a price difference between the automatic transmission cars and mechanical transmission cars.

Using Box-plot:

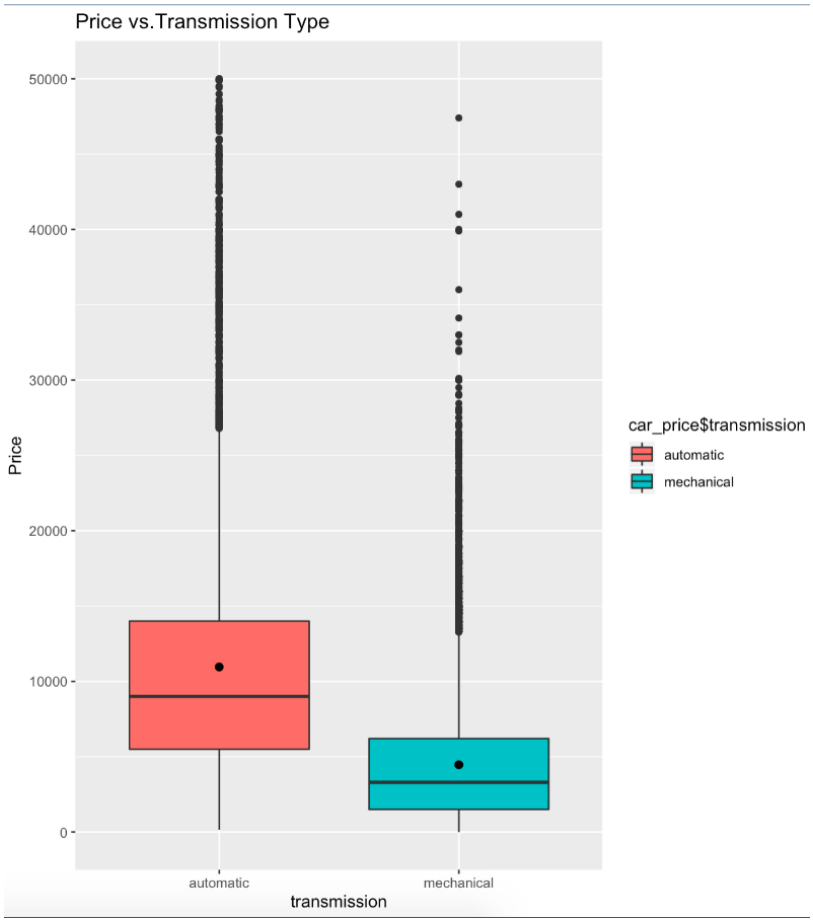


Figure 2 : Box plot for Transmission vs Car price

From the box-plot, we see that the variance of the automatic transmission cars is larger than the variance of mechanical transmission cars. Also, both these samples have numerous outliers which affects the comparison. So, comparing the mean average price values using medians in box-plot will not be reliable. Therefore I have used hypothesis testing to compare the mean values of these samples.

From Z-statistic Hypothesis Testing:

```
> # Z test for price prediction of Mechanical Transmission and Automatic Transmission types
> z.test(mechanical_car_price,automatic_car_price,alternative="two.sided",mu=0,sigma.x=sd(mechanical_car_price
+        sigma.y=sd(automatic_car_price),conf.level=0.95,paired=F)

        Two Sample z-test

data:  mechanical_car_price and automatic_car_price
z = -87.389, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6635.088 -6343.994
sample estimates:
mean of x mean of y
  4467.64  10957.18
```

Figure 3: Z-test for price prediction of mechanical and automatic transmission cars

5.2.2 The average price of the cars produced between 2000-2010 are greater than cars produced between 1990-2000.

Using Box-plot:

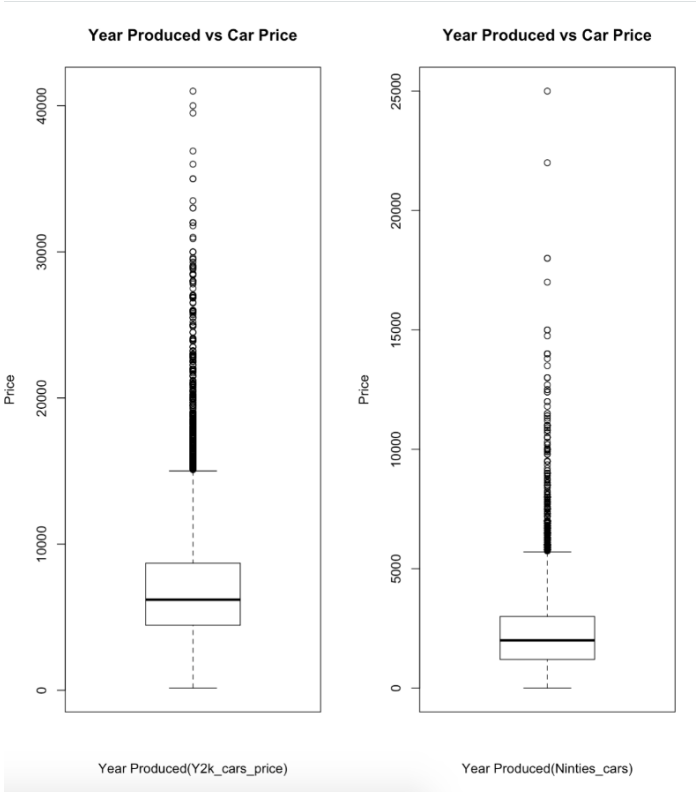


Figure 4: Box plot for Year produced vs Car price

From the box-plot, we see that the variance of the 2000-2010's cars is larger than the variance of 1990-2000's cars. Also, both these samples have numerous outliers which affects the comparison. So, comparing the mean average price values using medians in box-plot will not be reliable. Therefore, I have used hypothesis testing to compare the mean values of these samples.

From Z-test hypothesis testing:

```
> #Z test for price prediction of Y2K cars and Ninties cars
> z.test(Y2k_cars_price,Ninties_cars_price,alternative="greater",mu=0,sigma.x=sd(Y2k_cars_price),
+        sigma.y=sd(Ninties_cars_price),conf.level=0.95,paired=F)

        Two Sample z-test

data:  Y2k_cars_price and Ninties_cars_price
z = 137.74, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 4704.465      Inf
sample estimates:
mean of x mean of y
 7090.293  2328.970
```

Figure 5: Z-test for price prediction of cars produced in Nineties and Y2K

5.2.3 Analyze how the average price of the car differs among different types of drive trains

Using Box-plot:

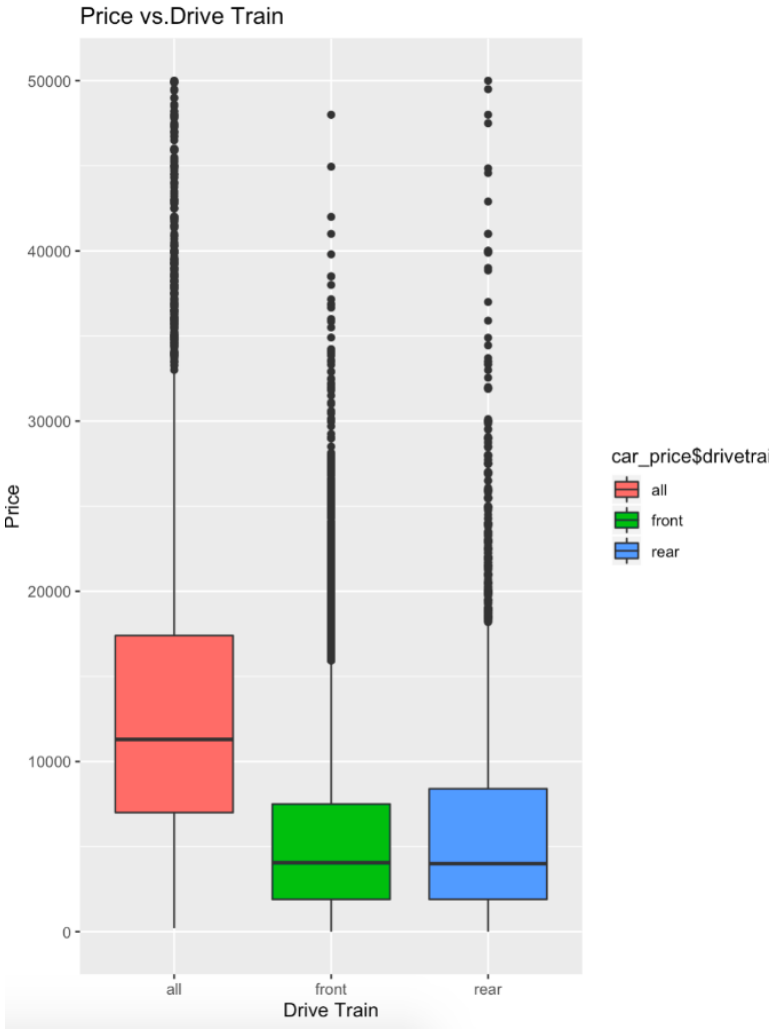


Figure 6: Box plot for different drive train vs Car price

From the box-plot, we see that the variance of the all wheel train is larger than the variance of front and rear train cars. Also, both these samples have numerous outliers which affects the comparison. So, comparing the mean average price values using medians in box-plot will not be reliable. Therefore I have used ANOVA hypothesis testing to compare the mean values of these samples.

ANOVA Regression model:

```
> anov=lm(car_price$price_usd~car_price$drivetrain)
> summary(anov)

Call:
lm(formula = car_price$price_usd ~ car_price$drivetrain)

Residuals:
   Min     1Q Median     3Q    Max
-13444  -3844  -1475   2325  43848

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  13644.4       78.6   173.6   <2e-16 ***
car_price$drivetrainfront   -8269.5       85.9   -96.3   <2e-16 ***
car_price$drivetrainrear    -7492.2      111.1   -67.4   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5770 on 38528 degrees of freedom
Multiple R-squared:  0.195,     Adjusted R-squared:  0.195
F-statistic: 4.66e+03 on 2 and 38528 DF,  p-value: <2e-16
```

Figure 7: ANOVA regression model to predict car price for different drive trains

Residual Analysis:

- Validate the constant variance
- Validate Normal distribution of residuals.

Validate constant variance:

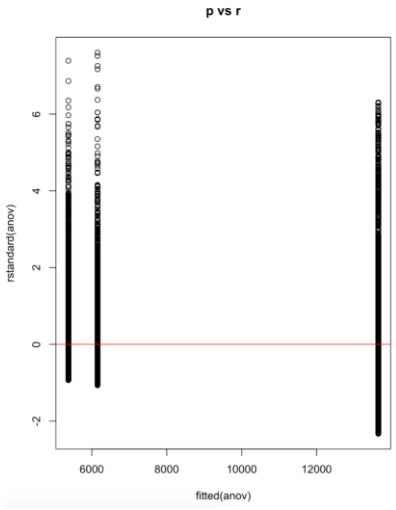


Figure 8: anov -Residuals vs predicted values

The observations are not scattered around the zero line. So, we can't say that the residuals don't have constant variance.

Normality test:

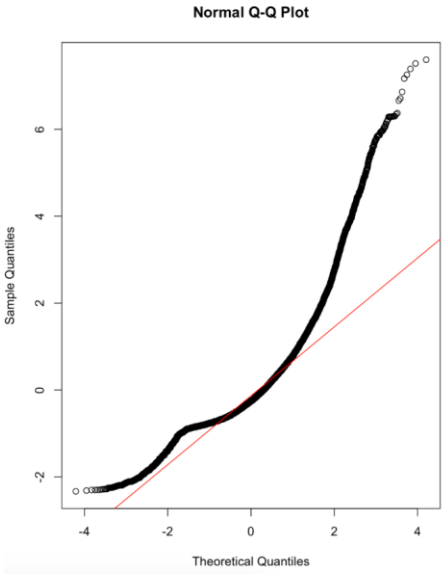


Figure 9: anov-Normal Q-Q plot

The model doesn't satisfy the normality condition, as most of the points are not aligned with normal line.

Transformation on Price_usd:

Log Transformation:

```
> #log transformation
>
> logtrans<-log10(car_price$price_usd)
> anov_log=lm(logtrans~car_price$drivetrain)
> summary(anov_log)

Call:
lm(formula = logtrans ~ car_price$drivetrain)

Residuals:
   Min     1Q Median     3Q    Max
-3.579 -0.268  0.044  0.305  1.123

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  4.03222    0.00566   712.8   <2e-16 ***
car_price$drivetrainfront -0.47408    0.00618   -76.7   <2e-16 ***
car_price$drivetrainrear  -0.45356    0.00800   -56.7   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.415 on 38528 degrees of freedom
Multiple R-squared:  0.134,     Adjusted R-squared:  0.134
F-statistic: 2.98e+03 on 2 and 38528 DF,  p-value: <2e-16
```

Figure 10: ANOVA log transformation model

Residual Analysis:

- **Validate constant variance:**

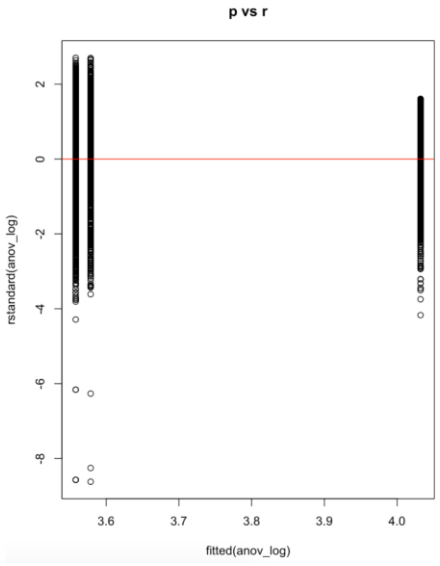


Figure 11: anov_log Residuals vs predicted values

- **Normality test:**

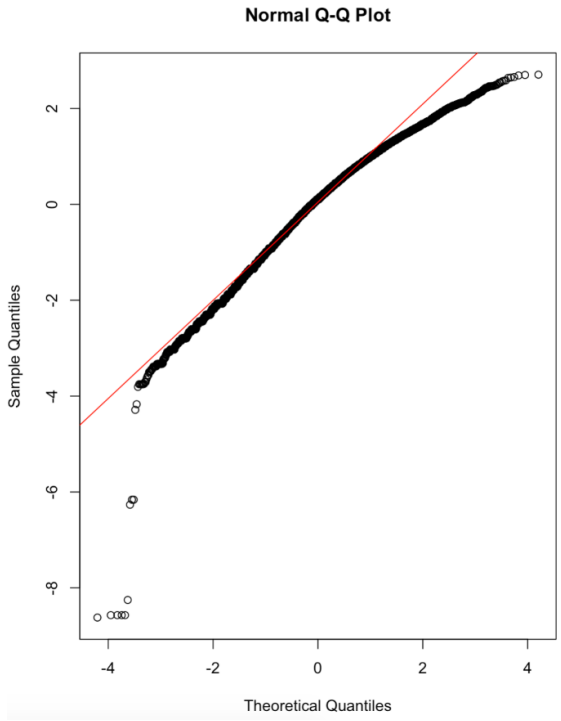


Figure 12: anov_log Normal Q-Q plot

Sqrt Transformation:

```
> sqtrans<-sqrt(car_price$price_usd)
> anov_sqr=lm(sqtrans~car_price$drivetrain)
> summary(anov_sqr)

Call:
lm(formula = sqtrans ~ car_price$drivetrain)

Residuals:
   Min     1Q Median     3Q    Max
-96.33 -24.56  -3.74  19.91 153.41

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  110.471      0.433   254.8   <2e-16 ***
car_price$drivetrainfront   -43.484      0.474   -91.8   <2e-16 ***
car_price$drivetrainrear    -40.275      0.613   -65.7   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 31.8 on 38528 degrees of freedom
Multiple R-squared:  0.181,     Adjusted R-squared:  0.181
F-statistic: 4.25e+03 on 2 and 38528 DF,  p-value: <2e-16
```

Figure 13: ANOVA sqrt transformation model

Residual analysis:

- **Validating constant variance**

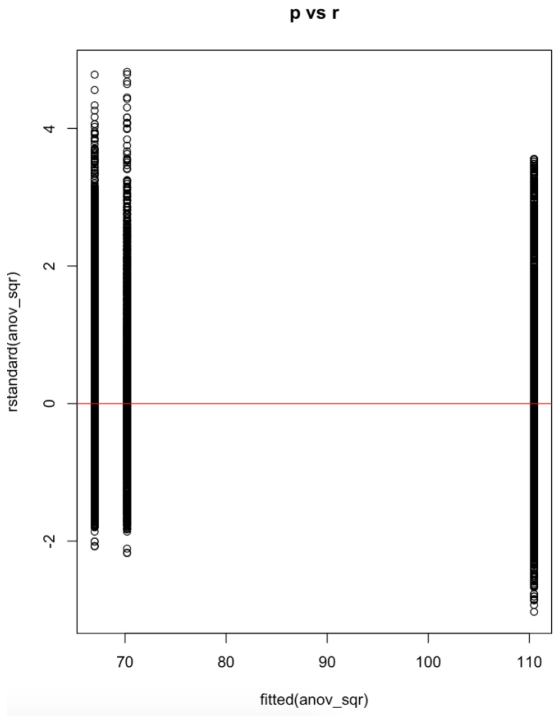


Figure 14: anov_sqr residulas vs predicted valve

- **Normality Test**

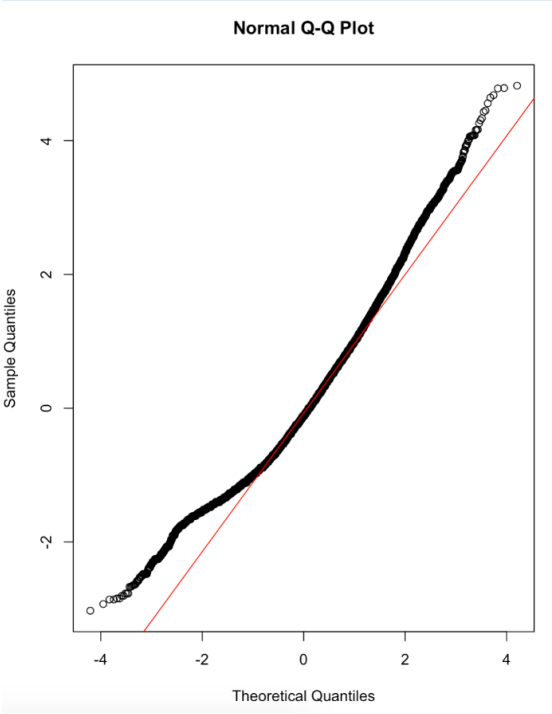


Figure 15: anov_sqr Noraml Q-Q plot

Inverse Transformation:

```
> #inverse
>
> inver<-1/(car_price$price_usd)
> anov_inv=lm(inver~car_price$drivetrain)
> summary(anov_inv)

Call:
lm(formula = inver ~ car_price$drivetrain)

Residuals:
    Min      1Q  Median      3Q     Max
-0.0008 -0.0005 -0.0003  0.0000  0.9994

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.000129   0.000163    0.79   0.4303
car_price$drivetrainfront 0.000486   0.000178    2.72   0.0065 **
car_price$drivetrainrear  0.000688   0.000231    2.98   0.0029 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.012 on 38528 degrees of freedom
Multiple R-squared:  0.000259,  Adjusted R-squared:  0.000207
F-statistic: 4.99 on 2 and 38528 DF,  p-value: 0.00683
```

Figure 16: ANOVA inverse transformation model

Residual Analysis:

- **Validate constant variance:**

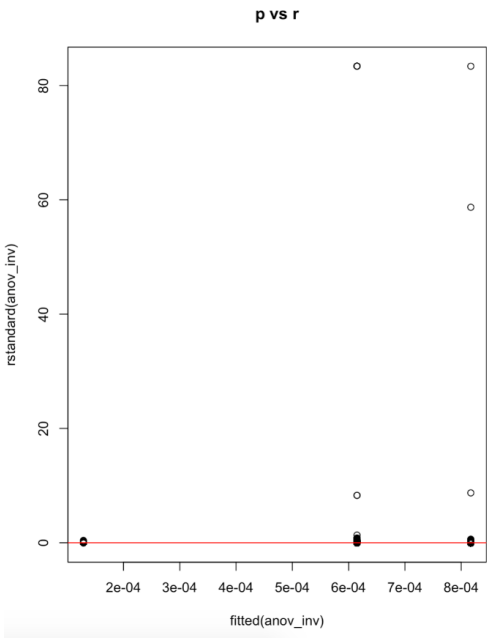


Figure 17: anov_inv residual vs predicted value

- **Normality Test:**

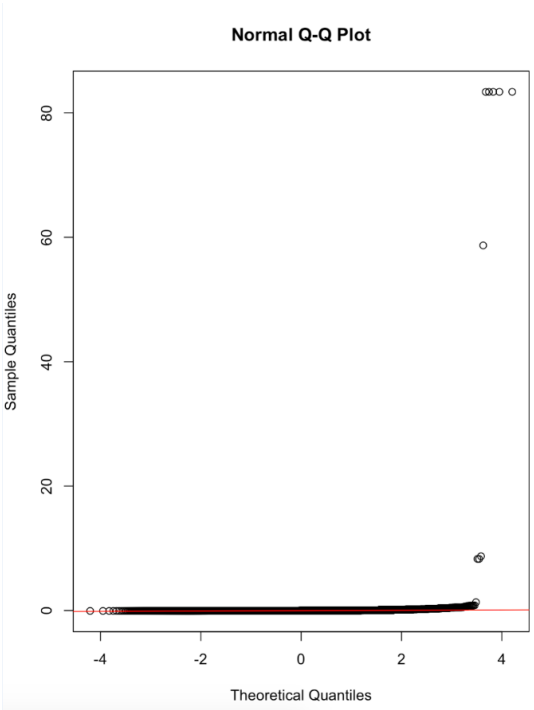


Figure18: anov_inv Normal Q-Q plot

- **ANOVA -Individual t-Test:**

```
> sqtrans<-sqrt(car_price$price_usd)
> anov_sqr=lm(sqtrans~car_price$drivetrain)
> summary(anov_sqr)

Call:
lm(formula = sqtrans ~ car_price$drivetrain)

Residuals:
   Min     1Q Median     3Q    Max
-96.33 -24.56  -3.74  19.91 153.41

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  110.471      0.433   254.8   <2e-16 ***
car_price$drivetrainfront    -43.484      0.474   -91.8   <2e-16 ***
car_price$drivetrainrear     -40.275      0.613   -65.7   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 31.8 on 38528 degrees of freedom
Multiple R-squared:  0.181,     Adjusted R-squared:  0.181
F-statistic: 4.25e+03 on 2 and 38528 DF,  p-value: <2e-16
```

Figure 19: ANOVA -Individual t-test

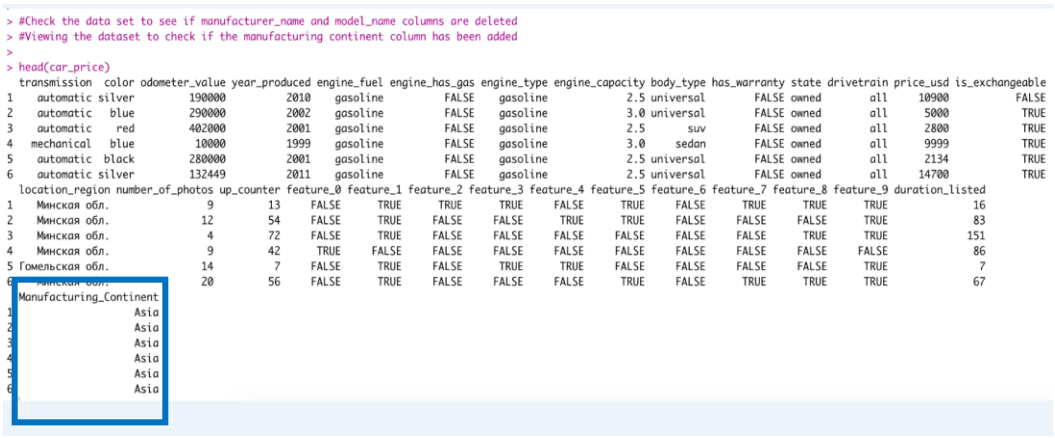
5.2.4 Build the multiple linear regression model for used car price prediction

- **Preprocessing**

- <u>Assigning manufacturer names and model names to respective continents:</u>

```
> #-------------------------------------------------------------Pre-Processing(Assigning the manufacturers to corresponding continents)-----------------------------------------------
>
> #Listing unique manfacurer name
> unique(car_price$manufacturer_name)
 [1] Subaru        LADA          Dodge         YA3           YA3           Kia           Opel          Muscovite     Alfa Romeo    Acura         Dacia         Lexus         Mitsubishi    Lancia
[15] Citroen       Mini          Jaguar        Porsche       SsangYong     Daewoo        Geely         BA3           Fiat          Ford          Renault       Seat          Rover         Volkswagen
[29] Lifan         Jeep          Cadillac      Audi          3A3           Toyota        TA3           GAZ           Volvo         Chevrolet     Great Wall    Buick         Pontiac       Lincoln
[43] Hyundai       Nissan        Suzuki        BMW           Mazda         Land Rover    Iveco         Skoda         Saab          Infiniti      Chery         Honda         Mercedes-Benz Peugeot
[57] Chrysler
57 Levels: Acura Alfa Romeo Audi BMW Buick Cadillac Chery Chevrolet Chrysler Citroen Dacia Daewoo Dodge Fiat Ford GAZ Geely Great Wall Honda Hyundai Infiniti Iveco Jaguar Jeep Kia LADA Lancia ... YA3
>
> #Replacing manufacturer name yA3 with YA3 as they both are same, but displayed as duplicate
> levels(car_price$manufacturer_name)[levels(car_price$manufacturer_name)=="YA3"]<-"YA3"
>
> #Lisitng unique manufacturer name, yA3 has been changed to YA3
> unique(car_price$manufacturer_name)
 [1] Subaru        LADA          Dodge         YA3           Kia           Opel          Muscovite     Alfa Romeo    Acura         Dacia         Lexus         Mitsubishi    Lancia        Citroen
[15] Mini          Jaguar        Porsche       SsangYong     Daewoo        Geely         BA3           Fiat          Ford          Renault       Seat          Rover         Volkswagen    Lifan
[29] Jeep          Cadillac      Audi          3A3           Toyota        TA3           GAZ           Volvo         Chevrolet     Great Wall    Buick         Pontiac       Lincoln       Hyundai
[43] Nissan        Suzuki        BMW           Mazda         Land Rover    Iveco         Skoda         Saab          Infiniti      Chery         Honda         Mercedes-Benz Peugeot       Chrysler
56 Levels: Acura Alfa Romeo Audi BMW Buick Cadillac Chery Chevrolet Chrysler Citroen Dacia Daewoo Dodge Fiat Ford GAZ Geely Great Wall Honda Hyundai Infiniti Iveco Jaguar Jeep Kia LADA Lancia ... 3A3
>
> #Sorting and assigning different manufacturers to their corresponding continents-Asia,Europe and America based on country of origin
> car_price$Manufacturing_Continent<-ifelse(car_price$manufacturer_name %in% c("Subaru","Kia","Acura","Lexus","Mitsubishi","SsangYong","Daewoo","Geely","Lifan","Toyota",
+                                                                              "Great Wall","Hyundai","Nissan","Suzuki","Mazda","Infiniti","Chery","Honda"),"Asia",
+                                           ifelse(car_price$manufacturer_name %in% c("LADA", "YA3","Opel","Alfa Romeo","Dacia","Lancia"," Rover","Muscovite","GAZ","Citroen",
+                                                                                     "Mini","Jaguar","Porscche","BA3","Fiat", "Renault","Seat","Volkswagen", "Audi","3A3",
+                                                                                     "TA3", "Volvo", "BMW" , "Land Rover", "Iveco", "Skoda","Saab",
+                                                                                   "Mercedes-Benz","Peugeot"),"Europe","America"))
```

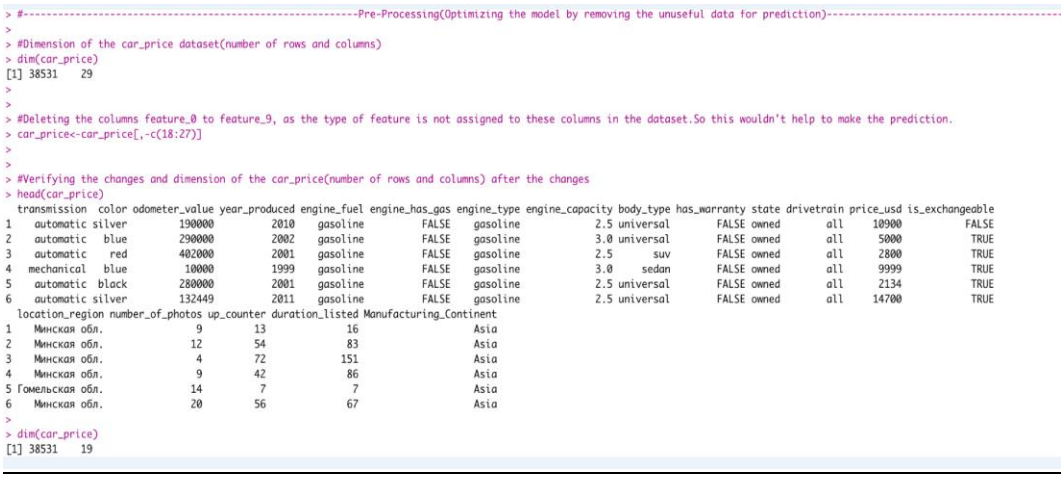
Sorted and assigned all the manufacturers to respective continents based on manufacturing location.



```
> #Check the data set to see if manufacturer_name and model_name columns are deleted
> #Viewing the dataset to check if the manufacturing continent column has been added
>
> head(car_price)
  transmission  color odometer_value year_produced engine_fuel engine_has_gas engine_type engine_capacity body_type has_warranty state drivetrain price_usd is_exchangeable
1    automatic silver         190000          2010    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     10900            FALSE
2    automatic   blue         290000          2002    gasoline          FALSE    gasoline             3.0 universal        FALSE owned        all      5000             TRUE
3    automatic    red         402000          2001    gasoline          FALSE    gasoline             2.5       suv        FALSE owned        all      2800             TRUE
4   mechanical   blue          10000          1999    gasoline          FALSE    gasoline             3.0     sedan        FALSE owned        all      9999             TRUE
5    automatic  black         280000          2001    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all      2134             TRUE
6    automatic silver         132449          2011    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     14700             TRUE
  location_region number_of_photos up_counter feature_0 feature_1 feature_2 feature_3 feature_4 feature_5 feature_6 feature_7 feature_8 feature_9 duration_listed
1    Минская обл.                9         13     FALSE      TRUE      TRUE      TRUE     FALSE      TRUE     FALSE      TRUE      TRUE      TRUE             16
2    Минская обл.               12         54     FALSE      TRUE     FALSE     FALSE      TRUE      TRUE     FALSE     FALSE     FALSE      TRUE             83
3    Минская обл.                4         72     FALSE      TRUE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE      TRUE      TRUE            151
4    Минская обл.                9         42      TRUE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE             86
5 Гомельская обл.               14          7     FALSE      TRUE     FALSE      TRUE      TRUE     FALSE     FALSE     FALSE     FALSE      TRUE              7
6    Минская обл.               20         56     FALSE      TRUE     FALSE     FALSE     FALSE      TRUE     FALSE      TRUE      TRUE      TRUE             67
  Manufacturing_Continent
1                    Asia
2                    Asia
3                    Asia
4                    Asia
5                    Asia
6                    Asia
```

Figure 20: Preprocessing -Assigning manufacturing continents

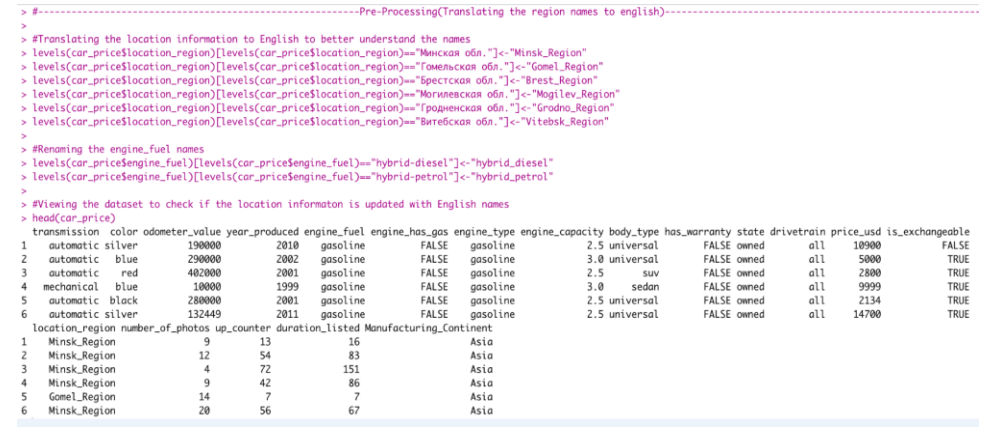
- <u>Removing all feature variables to optimize the model</u>



```
> #-----------------------------------------------------------Pre-Processing(Optimizing the model by removing the unuseful data for prediction)-----------------------------------
>
> #Dimension of the car_price dataset(number of rows and columns)
> dim(car_price)
[1] 38531    29
>
>
> #Deleting the columns feature_0 to feature_9, as the type of feature is not assigned to these columns in the dataset.So this wouldn't help to make the prediction.
> car_price<-car_price[,-c(18:27)]
>
>
> #Verifying the changes and dimension of the car_price(number of rows and columns) after the changes
> head(car_price)
  transmission  color odometer_value year_produced engine_fuel engine_has_gas engine_type engine_capacity body_type has_warranty state drivetrain price_usd is_exchangeable
1    automatic silver         190000          2010    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     10900            FALSE
2    automatic   blue         290000          2002    gasoline          FALSE    gasoline             3.0 universal        FALSE owned        all      5000             TRUE
3    automatic    red         402000          2001    gasoline          FALSE    gasoline             2.5       suv        FALSE owned        all      2800             TRUE
4   mechanical   blue          10000          1999    gasoline          FALSE    gasoline             3.0     sedan        FALSE owned        all      9999             TRUE
5    automatic  black         280000          2001    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all      2134             TRUE
6    automatic silver         132449          2011    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     14700             TRUE
  location_region number_of_photos up_counter duration_listed Manufacturing_Continent
1    Минская обл.                9         13             16                    Asia
2    Минская обл.               12         54             83                    Asia
3    Минская обл.                4         72            151                    Asia
4    Минская обл.                9         42             86                    Asia
5 Гомельская обл.               14          7              7                    Asia
6    Минская обл.               20         56             67                    Asia
>
> dim(car_price)
[1] 38531    19
```

Figure 21: Preprocessing -Eliminating feature variables

- <u>Translating the location and engine fuel names for better understanding:</u>



```
> #-----------------------------------------------------------Pre-Processing(Translating the region names to english)-----------------------------------------------------
>
> #Translating the location information to English to better understand the names
> levels(car_price$location_region)[levels(car_price$location_region)=="Минская обл."]<-"Minsk_Region"
> levels(car_price$location_region)[levels(car_price$location_region)=="Гомельская обл."]<-"Gomel_Region"
> levels(car_price$location_region)[levels(car_price$location_region)=="Брестская обл."]<-"Brest_Region"
> levels(car_price$location_region)[levels(car_price$location_region)=="Могилевская обл."]<-"Mogilev_Region"
> levels(car_price$location_region)[levels(car_price$location_region)=="Гродненская обл."]<-"Grodno_Region"
> levels(car_price$location_region)[levels(car_price$location_region)=="Витебская обл."]<-"Vitebsk_Region"
>
> #Renaming the engine_fuel names
> levels(car_price$engine_fuel)[levels(car_price$engine_fuel)=="hybrid-diesel"]<-"hybrid_diesel"
> levels(car_price$engine_fuel)[levels(car_price$engine_fuel)=="hybrid-petrol"]<-"hybrid_petrol"
>
> #Viewing the dataset to check if the location informaton is updated with English names
> head(car_price)
  transmission  color odometer_value year_produced engine_fuel engine_has_gas engine_type engine_capacity body_type has_warranty state drivetrain price_usd is_exchangeable
1    automatic silver         190000          2010    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     10900            FALSE
2    automatic   blue         290000          2002    gasoline          FALSE    gasoline             3.0 universal        FALSE owned        all      5000             TRUE
3    automatic    red         402000          2001    gasoline          FALSE    gasoline             2.5       suv        FALSE owned        all      2800             TRUE
4   mechanical   blue          10000          1999    gasoline          FALSE    gasoline             3.0     sedan        FALSE owned        all      9999             TRUE
5    automatic  black         280000          2001    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all      2134             TRUE
6    automatic silver         132449          2011    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     14700             TRUE
  location_region number_of_photos up_counter duration_listed Manufacturing_Continent
1    Minsk_Region                9         13             16                    Asia
2    Minsk_Region               12         54             83                    Asia
3    Minsk_Region                4         72            151                    Asia
4    Minsk_Region                9         42             86                    Asia
5    Gomel_Region               14          7              7                    Asia
6    Minsk_Region               20         56             67                    Asia
```

Figure 22: Preprocessing -Translating location names

All German names were translated to English for better understanding.

- Assigning different colors based on contrast

```
> #-------------------------------------------------------------Pre-Processing(Assigning different colors based on contrast)-----------------------------------------------------
>
> #listing unique colors of the car
> unique(car_price$color)
 [1] silver blue   red    black  grey   other  brown  white  green  violet orange yellow
Levels: black blue brown green grey orange other red silver violet white yellow
>
> #Based on the contrast, assigning the colors as Dark, Light and Other
> car_price$color<-ifelse(car_price$color %in% c("silver","yellow","white"),"Light",
+                         ifelse(car_price$color %in% c("red","black","grey","brown","voilet","orange","green","blue"),"Dark", "Other"))
>
> #Viewing the dataset to check if the color information is updated
> head(car_price)
  transmission color odometer_value year_produced engine_fuel engine_has_gas engine_type engine_capacity body_type has_warranty state drivetrain price_usd is_exchangeable
1    automatic Light          190000          2010    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     10900            FALSE
2    automatic  Dark          290000          2002    gasoline          FALSE    gasoline             3.0 universal        FALSE owned        all      5000             TRUE
3    automatic  Dark          402000          2001    gasoline          FALSE    gasoline             2.5       suv        FALSE owned        all      2800             TRUE
4   mechanical  Dark           10000          1999    gasoline          FALSE    gasoline             3.0     sedan        FALSE owned        all      9999             TRUE
5    automatic  Dark          280000          2001    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all      2134             TRUE
6    automatic Light          132449          2011    gasoline          FALSE    gasoline             2.5 universal        FALSE owned        all     14700             TRUE
  location_region number_of_photos up_counter duration_listed Manufacturing_Continent
1    Minsk_Region                9         13              16                    Asia
2    Minsk_Region               12         54              83                    Asia
3    Minsk_Region                4         72             151                    Asia
4    Minsk_Region                9         42              86                    Asia
5    Gomel_Region               14          7               7                    Asia
6    Minsk_Region               20         56              67                    Asia
```

Figure 23: Preprocessing -Assigning colors based on contrast

Divided the colors into dark, light and others based on contrast.

- Filling the missing values

```
> #-------------------------------------------------------------Pre-Processing(Filling the missing values)-----------------------------------------------------------------
>
> # Checking for missing values
> na_count <- sapply(car_price, function(y) sum(length(which(is.na(y)))))
>
> #column with the missing values
> na_count
        transmission                color         odometer_value          year_produced            engine_fuel         engine_has_gas            engine_type
                   0                    0                      0                      0                      0                      0                      0
     engine_capacity            body_type          has_warranty                  state             drivetrain              price_usd         is_exchangeable
                  10                    0                      0                      0                      0                      0                      0
       location_region      number_of_photos           up_counter      duration_listed Manufacturing_Continent
                   0                    0                      0                      0                      0
>
>
> #Though there is no fuel capacity for the electric cars, trying to fill it with the appropiate value by taking average value
> car_price$engine_capacity = ifelse(is.na(car_price$engine_capacity), ave(car_price$engine_capacity, FUN= function(x) mean(x, na.rm=T)), car_price$engine_capacity)
>
> sum(is.na(car_price$engine_capacity))
[1] 0
>
> #Verifying the columns after the changes
> na_count2 <- sapply(car_price, function(y) sum(length(which(is.na(y)))))
> na_count2
        transmission                color         odometer_value          year_produced            engine_fuel         engine_has_gas            engine_type
                   0                    0                      0                      0                      0                      0                      0
     engine_capacity            body_type          has_warranty                  state             drivetrain              price_usd         is_exchangeable
                   0                    0                      0                      0                      0                      0                      0
       location_region      number_of_photos           up_counter      duration_listed Manufacturing_Continent
                   0                    0                      0                      0                      0
```

Figure 24: Preprocessing -Filling missing variables

Filled the missing values in the engine capacity using the mean values.

- Creating dummy variables

Created dummy variables for all the categorical variables and use N-1 columns of each variable to build the model. Following is how the data looks like after creating the dummy variables.

```
> #Verifying the data set after creating dummy variables
> head(car_price)
  transmissionautomatic colorLight colorOther odometer_value year_produced engine_fueldiesel engine_fuelgas engine_fuelgasoline engine_fuelhybrid_diesel engine_fuelhybrid_petrol
1                     1          1          0          190000          2010                 0              0                    1                        0                        0
2                     1          0          0          290000          2002                 0              0                    1                        0                        0
3                     1          0          0          402000          2001                 0              0                    1                        0                        0
4                     0          0          0           10000          1999                 0              0                    1                        0                        0
5                     1          0          0          280000          2001                 0              0                    1                        0                        0
6                     1          1          0          132449          2011                 0              0                    1                        0                        0
  engine_has_gasFALSE engine_typediesel engine_typegasoline engine_capacity body_typecoupe body_typehatchback body_typeliftback body_typelimousine body_typeminibus body_typeminivan
1                   1                 0                   1             2.5              0                   0                0                   0                0                0
2                   1                 0                   1             3.0              0                   0                0                   0                0                0
3                   1                 0                   1             2.5              0                   0                0                   0                0                0
4                   1                 0                   1             3.0              0                   0                0                   0                0                0
5                   1                 0                   1             2.5              0                   0                0                   0                0                0
6                   1                 0                   1             2.5              0                   0                0                   0                0                0
  body_typepickup body_typesedan body_typesuv body_typeuniversal body_typevan has_warrantyTRUE statenew stateowned drivetrainall drivetrainfront price_usd is_exchangeableTRUE
1               0               0            0                 1             0                0        0          1             1                0     10900                    0
2               0               0            0                 1             0                0        0          1             1                0      5000                    1
3               0               0            1                 0             0                0        0          1             1                0      2800                    1
4               0               1            0                 0             0                0        0          1             1                0      9999                    1
5               0               0            0                 1             0                0        0          1             1                0      2134                    1
6               0               0            0                 1             0                0        0          1             1                0     14700                    1
  location_regionVitebsk_Region location_regionGomel_Region location_regionGrodno_Region location_regionMinsk_Region location_regionMogilev_Region number_of_photos up_counter
1                             0                           0                           0                           1                           0               9         13
2                             0                           0                           0                           1                           0              12         54
3                             0                           0                           0                           1                           0               4         72
4                             0                           0                           0                           1                           0               9         42
5                             0                           1                           0                           0                           0              14          7
6                             0                           0                           0                           1                           0              20         56
  duration_listed Manufacturing_ContinentAsia Manufacturing_ContinentEurope
1              16                           1                             0
2              83                           1                             0
3             151                           1                             0
4              86                           1                             0
5               7                           1                             0
6              67                           1                             0
>
> #Checking number of rows and columns afer creating dummy variables
> dim(car_price)
[1] 38531    42
```

Figure 25: Preprocessing -Dataset after creating dummy variables

- Dividing the column values into different intervals and creating dummy variables

```
> #-------------------------------------------------------------Grouping the columns-------------------------------------------------------------------------------
>
> #Though the varibles year,odometer,number of photos and up counter are numerical varibles,it is treated as categorical and converted it to dummy varibles in the regeression model
>
> ##Divinding the years into 4 groups
> #1940-1960 is 40_60
> #1960-1980 is 60_80
> #1980-2000 is 80_00
> #2000-2020 is 00_20
>
> car_price$year_produced<-cut(car_price$year_produced,breaks=c(1940,1960,1980,2000,2020),labels = c('40_60','60_80','80_00','00_20'))
>
> car_price$year_produced[1:10]
 [1] 00_20 00_20 00_20 80_00 00_20 00_20 80_00 00_20 00_20 80_00
Levels: 40_60 60_80 80_00 00_20
>
> #Creating dummy variables and removing one dummy variable(N-1)
> car_price <- dummy.data.frame(car_price, names=c("year_produced"))
Warning message:
In model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE) :
  non-list contrasts argument ignored
> car_price <- car_price [, -c (5)]
>
> ##Dividing the odometer value into 4 groups
>
> # Creating Dummy variables
> car_price$odometer_value<-cut(car_price$odometer_value,
+                               quantile(car_price$odometer_value, probs = c(0, .25, .50,.75, 1)),
+                               labels = c('grp1','grp2','grp3', 'grp4'),
+                               include.lowest = TRUE)
>
> car_price$odometer_value[1:10]
 [1] grp2 grp3 grp4 grp1 grp3 grp1 grp3 grp4 grp2 grp4
Levels: grp1 grp2 grp3 grp4
>
> #Creating dummy variables and removing one dummy variable(N-1)
> car_price <- dummy.data.frame(car_price, names=c("odometer_value"))
Warning message:
In model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE) :
  non-list contrasts argument ignored
> car_price <- car_price [, -c (4)]
```

- *Figure 26: Preprocessing - Dividing the column values into different intervals and creating dummy variables*

Divided the years from 1940 to 2020 into 4 groups and created dummy variables for each group variable and used (N-1) variables to create the model.

- Divided the odometer values into 4 groups by considering 25% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

```
> ##Dividing the number of photos into 2 groups
>
> car_price$number_of_photos<-cut(car_price$number_of_photos,breaks=c(0,43,86))
>
> car_price$number_of_photos[1:10]
 [1] (0,43] (0,43] (0,43] (0,43] (0,43] (0,43] (0,43] (0,43] (0,43] (0,43]
Levels: (0,43] (43,86]
>
> #Creating dummy variables and removing one dummy variable(N-1)
> car_price <- dummy.data.frame(car_price, names=c("number_of_photos"))
Warning message:
In model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE) :
  non-list contrasts argument ignored
> car_price <- car_price [, -c (43)]
>
> ##Dividing the up counter into 4 groups
>
> car_price$up_counter<-cut(car_price$up_counter,breaks=c(0,465,931,1396,1861))
>
> car_price$up_counter[1:10]
 [1] (0,465] (0,465] (0,465] (0,465] (0,465] (0,465] (0,465] (0,465] (0,465] (0,465]
Levels: (0,465] (465,931] (931,1.4e+03] (1.4e+03,1.86e+03]
>
> #Creating dummy variables and removing one dummy variable(N-1)
> car_price <- dummy.data.frame(car_price, names=c("up_counter"))
Warning message:
In model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE) :
  non-list contrasts argument ignored
> car_price <- car_price [, -c (46)]
>
> ##Dividing the up counter into 4 groups
>
> car_price$duration_listed<-cut(car_price$duration_listed,    quantile(car_price$duration_listed, probs = c(0, .25, .50,.75, 1)),
+                                labels = c('grp1','grp2','grp3', 'grp4'),
+                                include.lowest = TRUE)
>
> car_price$duration_listed[1:10]
 [1] grp1 grp3 grp4 grp3 grp1 grp3 grp4 grp3 grp3 grp2
Levels: grp1 grp2 grp3 grp4
>
> #Creating dummy variables and removing one dummy variable(N-1)
> car_price <- dummy.data.frame(car_price, names=c("duration_listed"))
Warning message:
In model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE) :
  non-list contrasts argument ignored
> car_price <- car_price [, -c (48)]
```

- *Figure 27: Preprocessing – Dividing the column values into different intervals and creating dummy variables*

- Divided the number of photos into 2 groups by considering 50% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model

- Divided the up counter into 4 groups by considering 25% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

- Divided the up counter into 4 groups by considering 25% values on each group and created dummy variables for each group variable and used (N-1) variables to create the model.

- **Regression Model:**

We use N-fold cross validation method to build regression model for life expectancy with N value as 10.

We are considering Backward Selection and Forward Selection and stepwise to build the model.

Backward Selection:

```
> #----------------------Multi Linear regression models---------------------
>
> #.........................N-fold Validation-------------------
>
> library(caret)
>
>

> set.seed(10001)
> train.control <- trainControl(method = "cv", number = 10)
> #Backward Selection
> # Train the model backward selection
> model_backward <- train(price_usd~., data = car_price, method = "leapBackward",
+                         trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
> # Summarize the results
> print(model_backward)
Linear Regression with Backwards Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34677, 34679, 34679, 34677, 34678, 34679, ...
Resampling results across tuning parameters:

  nvmax  RMSE  Rsquared  MAE
  2      5303  0.320     3343
  3      4990  0.398     3234
  4      4764  0.451     3118

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
```

Figure 28: N-fold cross validation : model_backward

Forward Selection

```
> #Forward selection
> # Train the model backward selection
> model_forward <- train(price_usd~., data = car_price, method = "leapForward",
+                        trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
> # Summarize the results
> print(model_forward)
Linear Regression with Forward Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34677, 34677, 34678, 34678, 34679, 34677, ...
Resampling results across tuning parameters:

  nvmax  RMSE  Rsquared  MAE
  2      5303  0.320     3343
  3      4990  0.397     3234
  4      4791  0.445     3105

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
```

Figure 29: N-fold cross validation : model_forward

Stepwise Selection

```
> #Stepwise
> # Train the model stepwise selection
> model_step <- train(price_usd~., data = car_price, method = "leapSeq",
+                     trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
> #Summarize the results
> print(model_step)
Linear Regression with Stepwise Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34678, 34679, 34678, 34678, 34678, 34677, ...
Resampling results across tuning parameters:

  nvmax  RMSE  Rsquared  MAE
  2      5303  0.320     3343
  3      4990  0.398     3233
  4      4790  0.445     3105

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
```

Figure 30: N-fold cross validation : model_step

Model	Nvmax 4 values		
	RMSE	Adj-R squared	MAE
Backward	4764	45.1%	3118
Forward	4791	44.5%	3105
Stepwise	4790	44.5%	3105

Figure 31: Model comparison

Considering both **RMSE and MAE** values, I have considered Stepwise model to continue the analysis

Residual Analysis:

- Validating constant variance for model_step

Using R Code:

```
> par(mfrow=c(1,1))
>
> #checking for constant variance
> plot(predict(model_step),residuals(model_step),main="p vs r")
> abline(a=0,b=0,col='red')
```

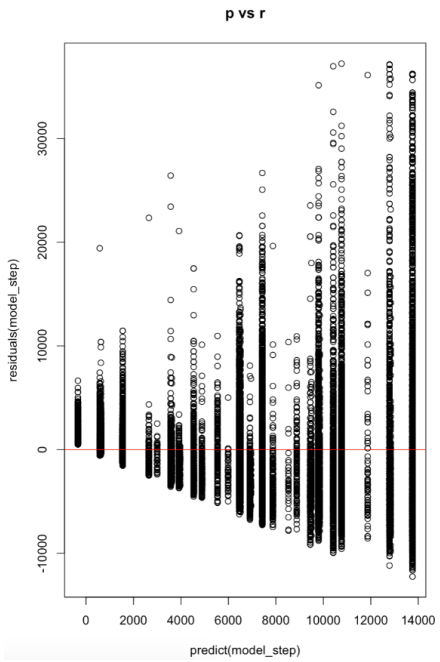
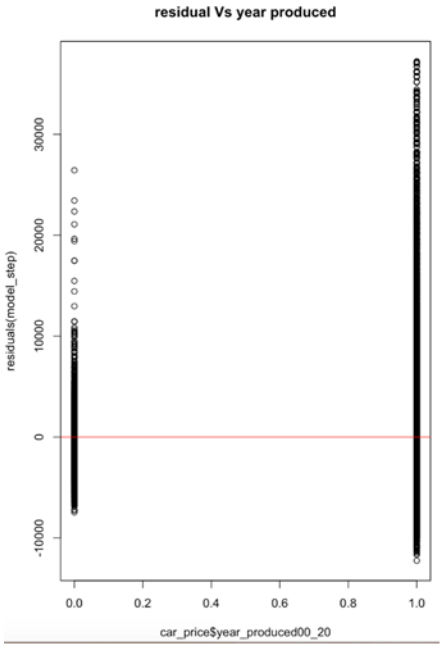


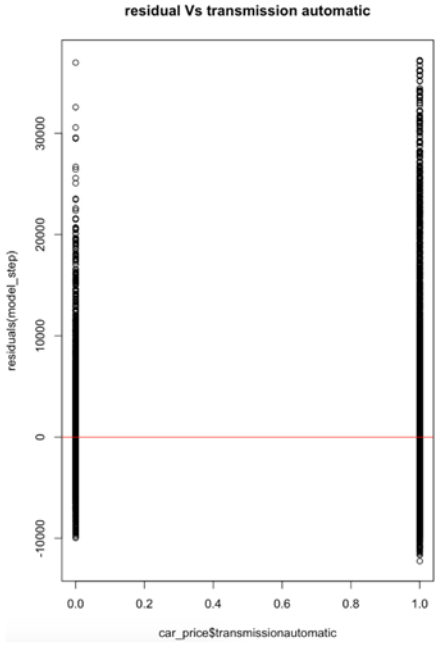
Figure 32: model_step: Validating constant variance

The observations are not scattered around the zero line. I see an incremental funnel type of pattern in the residual plot. So, the variance is not constant.

- Validating linear relationship of X variables

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

#checking for constant variance
plot(predict(model_step),residuals(model_step),main="p vs r")
abline(a=0,b=0,col='red')

#linear plot for each x variable
plot(car_price$year_produced00_20,y=residuals(model_step),main="residual Vs year produced")
abline(a=0,b=0,col='red')

plot(car_price$transmissionautomatic,y=residuals(model_step),main="residual Vs transmission automatic")
abline(a=0,b=0,col='red')

plot(car_price$drivetrainfront,y=residuals(model_step),main="residual Vs drive train front")
abline(a=0,b=0,col='red')

plot(car_price$body_typesedan,y=residuals(model_step),main="residual Vs body type sedan")
abline(a=0,b=0,col='red')

plot(car_price$location_regionVitebsk_Region,y=residuals(model_step),main="residual Vs location region Vitebsk Region")
abline(a=0,b=0,col='red')
```





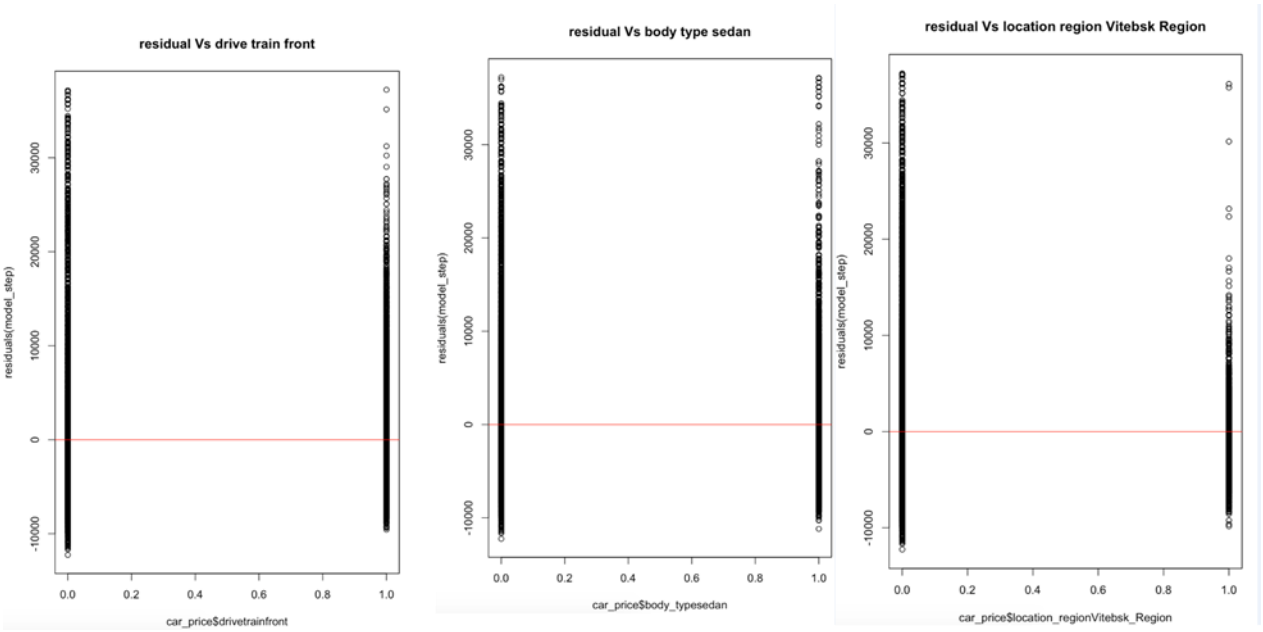


Figure 33: model_step: Validating linearity

After observing the plot of all X-variables, looks like the dummy values are randomly scattered around zero line. So, the varibles transmissionautomatic,year produced 00_20, body type sedan, drive train front and location Vitebsk region are all having linear relationship with Price_usd

- Normality Test-Q-Q plot:

```
> #normality test
> qqnorm(residuals(model_step))
> qqline(residuals(model_step),col=2)
```

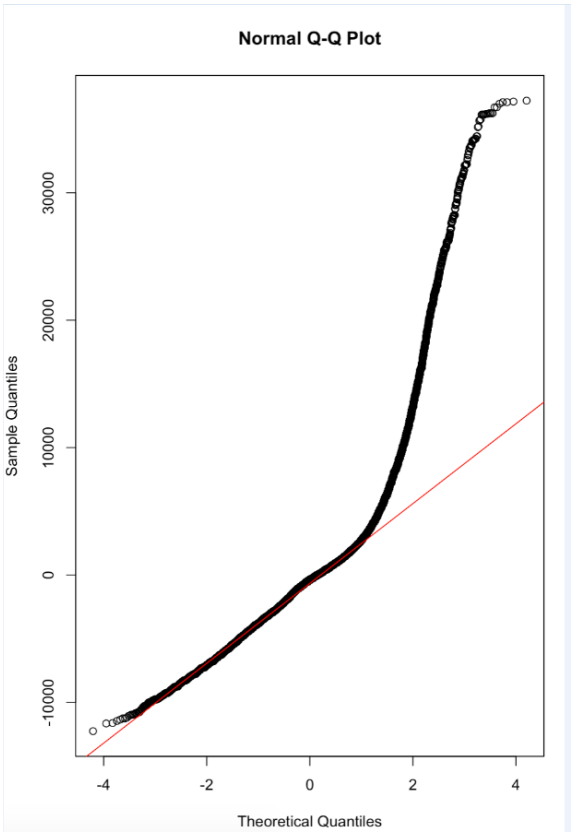


Figure 34: model_step: Normality test

From the Q-Q plot, I see that most of the values towards the end are not close to the normal line. So, the model is not normally distributed.

As model_step failed to satisfy constant variance and normality conditions, we cannot consider this as a fitted model.

Therefore, we need to make transformation on the variables and re-build the model

- **Transformation**

As most of the X variables are categorical, I am transforming Y-variable to improve the fitted model and satisfy the residual analysis condition.

Log Transformation:

```
> #Performing Y transrformation
>
> #log Transformation
> #After Y transformation(log Transformation)
>
> #Train the model stepwise selection
> model_step2_log<- train(log(price_usd)~., data =car_price, method = "leapSeq",
+                         trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
>
> # Summarize the results
> print(model_step2_log)
Linear Regression with Stepwise Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34678, 34678, 34678, 34678, 34677, 34679, ...
Resampling results across tuning parameters:

  nvmax  RMSE       Rsquared   MAE
  2      0.7191920  0.5071561  0.5620803
  3      0.8713125  0.2747997  0.6924456
  4      0.6428831  0.6085680  0.4971319

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
```

Figure 35: model_step2_log: log Y regression model

- Calculating RMSE for Y after log transformation:

```
> #price_usd prediction after the log transformation
> prelog<-predict(model_step2_log,car_price)
>
> head(cbind(actual=car_price$price_usd,prelog))
    actual   prelog
1 10900.00 9.413360
2  5000.00 9.412785
3  2800.00 9.412785
4  9999.00 7.625956
5  2134.11 9.412785
6 14700.00 9.412785
> #Back Transformed
> head(cbind(actual=car_price$price_usd,pred=exp(prelog)))
    actual     pred
1 10900.00 12250.97
2  5000.00 12243.93
3  2800.00 12243.93
4  9999.00  2050.74
5  2134.11 12243.93
6 14700.00 12243.93
>
> #Calculating the RMSE value
> RMSE(car_price$price_usd,exp(prelog))
[1] 4858.105
```

Figure 36: RMSE of model_step2_log after log transformation

- **Sqrt Transformation:**

```
> #Square root Transformation
> #After Y transformation(squareroot Transformation)
>
> #Train the model stepwise selection
> model_step2_sqrt<- train(sqrt(price_usd)~., data =car_price, method = "leapSeq",
+                          trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
>
> # Summarize the results
> print(model_step2_sqrt)
Linear Regression with Stepwise Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34677, 34679, 34678, 34680, 34676, 34679, ...
Resampling results across tuning parameters:

  nvmax  RMSE      Rsquared   MAE
  2      27.50514  0.3828238  21.02714
  3      23.04113  0.5703748  17.29246
  4      22.00927  0.6079153  16.71368

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
```

Figure 37: model_step2_sqrt: sqrt Y regression model

- Calculating RMSE for Y after Sqrt transformation:

```
>
> #price_usd prediction after the Square root transformation
> presqt<-predict(model_step2_sqrt,car_price)
>
> head(cbind(actual=car_price$price_usd,presqt))
    actual   presqt
1 10900.00 112.9516
2  5000.00 112.9516
3  2800.00 112.9516
4  9999.00  49.6297
5  2134.11 112.9516
6 14700.00 112.9516
> #Back Transformed
> head(cbind(actual=car_price$price_usd,pred=(presqt*presqt)))
    actual      pred
1 10900.00 12758.061
2  5000.00 12758.061
3  2800.00 12758.061
4  9999.00  2463.108
5  2134.11 12758.061
6 14700.00 12758.061
>
> #Calculating the RMSE value
> RMSE(car_price$price_usd,(presqt*presqt))
[1] 4505.352
```

Figure 38: RMSE of model_step2_sqrt after sqrt transformation

- **Inverse Transformation:**

```
> #Inverse Transformation
> #After Y transformation(Inverse Transformation)
>
> #Trani the model stepwise selection
> model_step2_inverse<- train(1/(price_usd)~., data =car_price, method = "leapSeq",
+                     trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
>
> # Summarize the results
> print(model_step2_inverse)
Linear Regression with Stepwise Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34678, 34678, 34678, 34679, 34677, 34679, ...
Resampling results across tuning parameters:

  nvmax  RMSE         Rsquared     MAE
  2      0.009073054  0.002067963  0.0005768380
  3      0.009051510  0.064257209  0.0005087624
  4      0.009051795  0.064242605  0.0005097960

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 3.
```

Figure 39: model_step2_inverse: inverse Y regression model

- Calculating RMSE for Y after inverse transformation:

```
> #price_usd prediction after the Inverse transformationn
> preinv<-predict(model_step2_inverse,car_price)
>
> head(cbind(actual=car_price$price_usd,preinv))
    actual       preinv
1 10900.00 1.886969e-04
2  5000.00 1.813202e-05
3  2800.00 1.813202e-05
4  9999.00 1.072630e-03
5  2134.11 1.813202e-05
6 14700.00 1.813202e-05
> #Back Transformed
> head(cbind(actual=car_price$price_usd,pred=(1/preinv)))
    actual       pred
1 10900.00  5299.5044
2  5000.00 55151.0601
3  2800.00 55151.0601
4  9999.00   932.2875
5  2134.11 55151.0601
6 14700.00 55151.0601
>
> #Calculating the RMSE value
> RMSE(car_price$price_usd,(1/preinv))
[1] 359015.7
```

Figure 40: RMSE of model_step2_inverse after inverse transformation

- **Transformation comparison:**

Transformation Type	RMSE
Log	4858.10
Sqrt	4505.35
Inverse	359015.7

Figure 41: Transformation comparison

Among all the transformation, Sqrt transformation provided low RMSE value. So, we consider Sqrt Y to build the regression model.

- **Building Stepwise model after transformation:**

```
> #Assigning sqtprice to price_usd in the dataset
>
> car_price[,'price_usd']<-sqtprice
> #After Y transformation
>
> #Train the model stepwise selection
>
> model_step2<- train(price_usd~., data =car_price, method = "leapSeq",
+                     trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
>
> # Summarize the results
>
> print(model_step2)
Linear Regression with Stepwise Selection

38531 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 34677, 34679, 34678, 34680, 34676, 34679, ...
Resampling results across tuning parameters:

  nvmax  RMSE  Rsquared  MAE
  2      27.5  0.383     21.0
  3      23.0  0.570     17.3
  4      22.0  0.608     16.7

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
```

Figure 42: N-fold Cross validation: model_step2

- Residual Analysis:

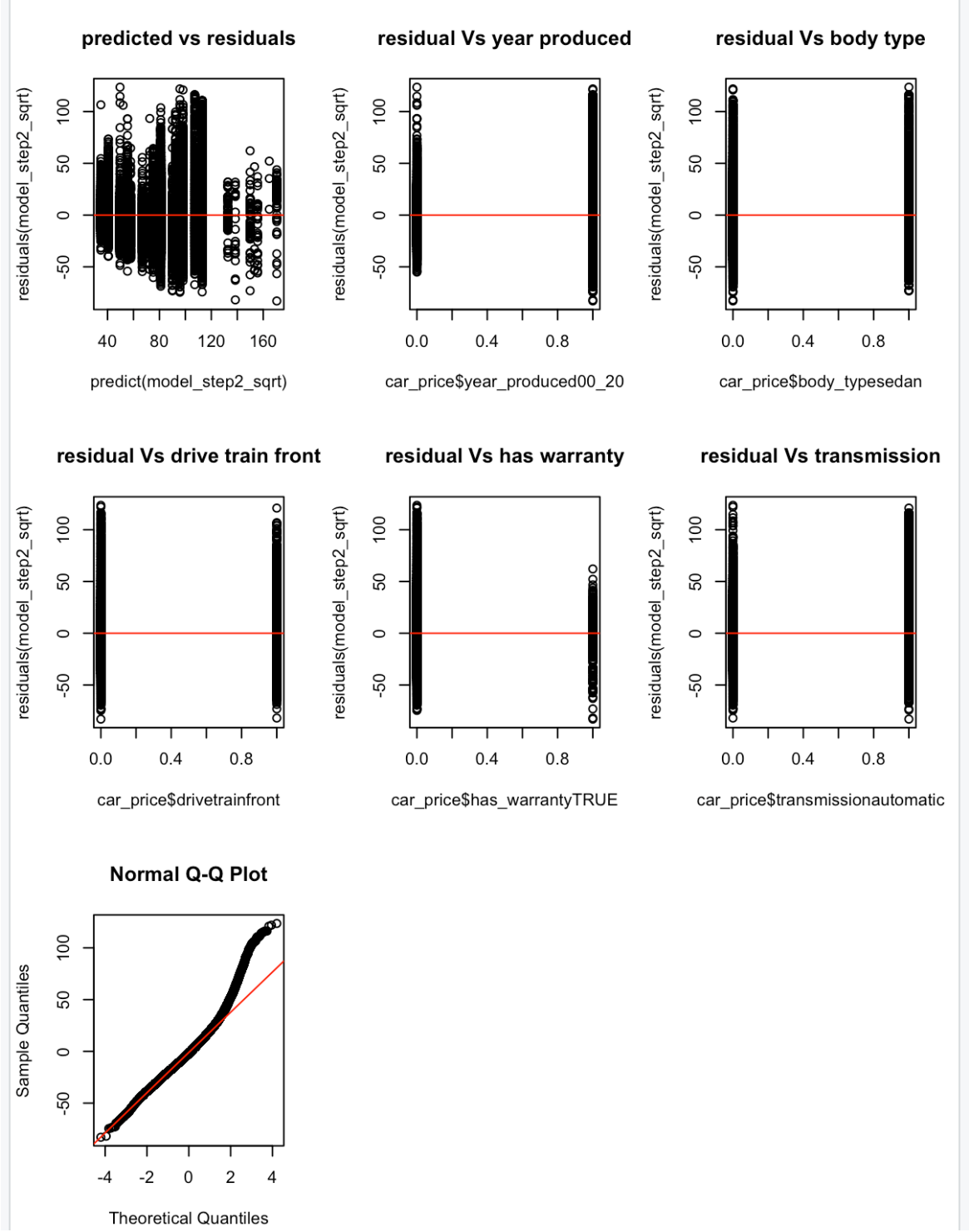


Figure 43: model_step2: Residual analysis

Validating Constant Variance:

All the observations are scattered around the zero line. So, the model is having constant variance

Linearity Test:

From the plots, we see that all the variables are having linear relationship with residuals

Normality Test:

From Q-Q plot, most of the observations at the end are not close to the normal line. So, normality condition is not satisfied.

<u>Checking Outliers:</u>

```
> #Checking Outliers
>
> quantile(car_price$price_usd, probs = seq(0, 1, by= 0.01))
      0%       1%       2%       3%       4%       5%       6%       7%       8%       9%      10%      11%      12%      13%      14%      15%      16%      17%
   1.000  332.545  450.000  500.000  600.000  650.000  750.000  800.000  900.000  999.000 1000.000 1100.000 1199.000 1233.050 1300.000 1400.000 1499.000 1500.000
     18%      19%      20%      21%      22%      23%      24%      25%      26%      27%      28%      29%      30%      31%      32%      33%      34%      35%
1590.000 1650.000 1700.000 1800.000 1900.000 2000.000 2000.000 2100.000 2200.000 2300.000 2450.000 2500.000 2600.000 2700.000 2800.000 2950.000 3000.000 3100.000
     36%      37%      38%      39%      40%      41%      42%      43%      44%      45%      46%      47%      48%      49%      50%      51%      52%      53%
3200.000 3300.000 3500.000 3500.000 3600.000 3700.000 3850.000 3999.000 4000.000 4200.000 4300.000 4450.000 4500.000 4650.000 4800.000 4950.000 5000.000 5200.000
     54%      55%      56%      57%      58%      59%      60%      61%      62%      63%      64%      65%      66%      67%      68%      69%      70%      71%
5300.000 5500.000 5539.224 5700.000 5900.000 6000.000 6200.000 6400.000 6500.000 6700.000 6900.000 7000.000 7199.000 7350.841 7500.000 7700.000 7900.000 8000.000
     72%      73%      74%      75%      76%      77%      78%      79%      80%      81%      82%      83%      84%      85%      86%      87%      88%      89%
8300.000 8500.000 8700.000 8990.000 9100.000 9450.000 9650.000 9950.000 10200.000 10500.000 10900.000 11200.000 11500.000 11999.000 12450.000 12850.000 13300.000 13900.000
     90%      91%      92%      93%      94%      95%      96%      97%      98%      99%     100%
14500.000 15000.000 15700.000 16500.000 17500.000 18700.000 20000.000 22700.000 25950.000 32900.000 50000.000
>
>
> #lesser value of RMSE for Square Root transformation
> car_price[,'price_usd']<-sqrt(car_price$price_usd)
>
>
> #----------removing the outliers------------------
> cooks<-lm(price_usd~.,data=car_price)
> cooksd <- cooks.distance(cooks)
>
> influential <- as.numeric(names(cooksd)[(cooksd > 4/nrow(car_price))])
>
> with_outliers<-car_price
>
> dim(with_outliers)
[1] 38531    50
>
> no_outliers <- with_outliers[-influential, ]
>
> dim(no_outliers)
[1] 36838    50
```

Figure 44: Removing Outliers

The previous normality plots shows majority of values at higher end are away from the normal line and from the above quantile report, looks like there is a significant increase in values from 98% to 100%. So, removing the outliers using “Cook distance” method to improve the model.

- **Building Stepwise model after removing outliers:**

```
> #Afterremoving outliers
>
> #Train the model stepwise selection
> model_step3<- train(price_usd~., data =no_outliers, method = "leapSeq",
+                     trControl = train.control)
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
Reordering variables and trying again:
There were 11 warnings (use warnings() to see them)
>
> # Summarize the results
> print(model_step3)
Linear Regression with Stepwise Selection

36838 samples
   49 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 33154, 33155, 33155, 33153, 33154, 33153, ...
Resampling results across tuning parameters:

  nvmax  RMSE      Rsquared   MAE
  2      21.74228  0.5501488  16.93582
  3      27.87436  0.2607332  22.46610
  4      18.95750  0.6579604  14.92100

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was nvmax = 4.
>
> coef(model_step3$finalModel,5)
         (Intercept) transmissionautomatic    year_produced00_20          body_typesuv               statenew   is_exchangeableTRUE
           41.967819             14.959450             37.098419             21.212809             58.469553             -1.212689
```

Figure 45: N-fold cross validation:model_step3

- **Residual analysis**

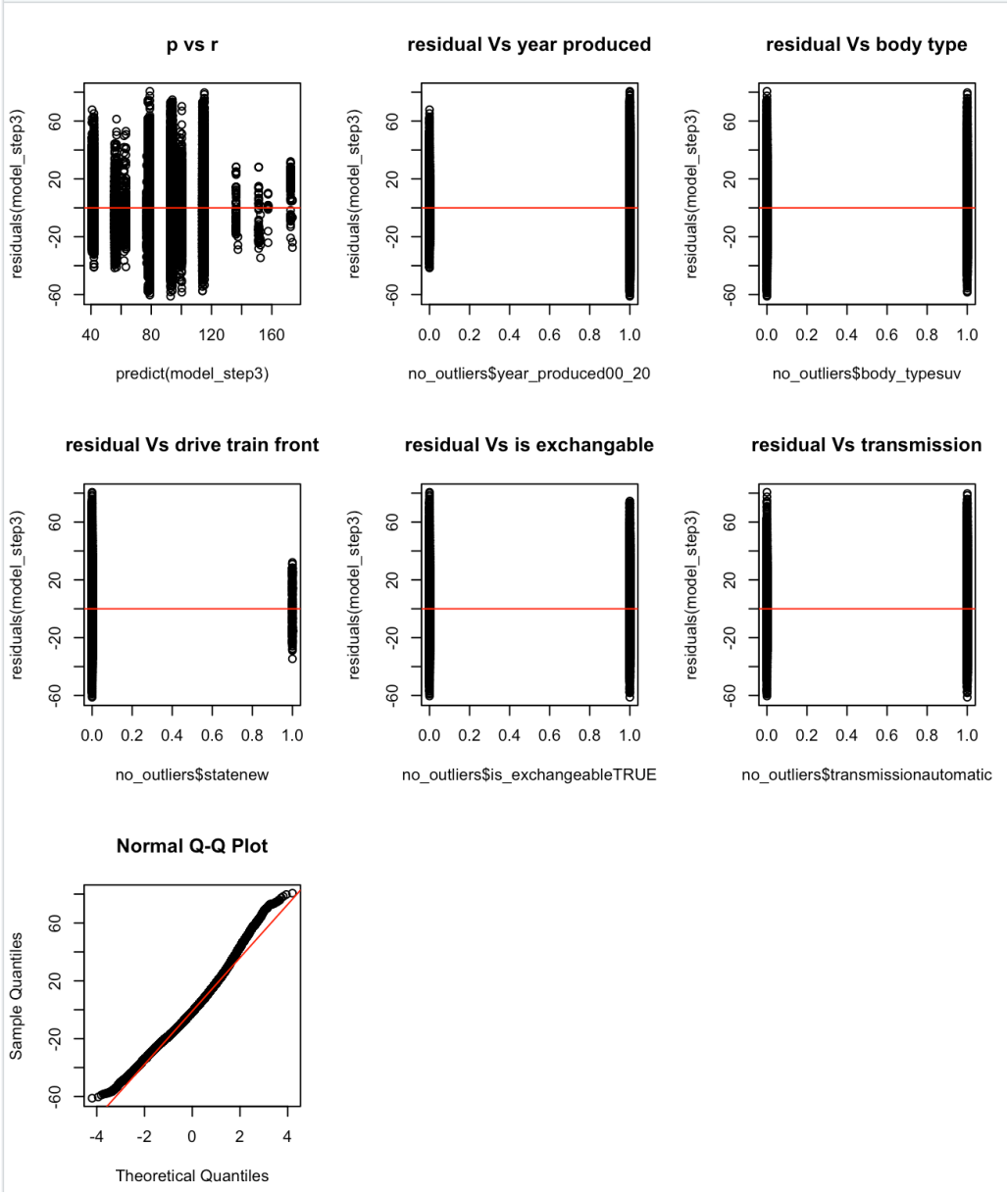


Figure 46: Residual Analysis:model_step3

Validating Constant Variance:

All the observations are scattered around the zero line. So, the model is having constant variance

Linearity Test:

From the plots, we see that all the variables are having linear relationship with residuals

Normality Test:

From Q-Q plot, most of the observations are close to the normal line.

5.3. Findings

5.3.1 Whether there is a price difference between the automatic transmission cars and mechanical transmission cars.

From Z-test, the p-value is calculated as 2.2e-16 (refer Figure 3).

In this case, the P-value is less than α/2 (0.025). Therefore, we reject null hypothesis. As per the alternate hypothesis, the average price of the automatic transmission cars and mechanical transmission cars are not equal.

5.3.2 The average price of the cars produced between 2000-2010 are greater than cars produced between 1990-2000.

From Z-test, the p-value is calculated as 2.2e-16 (refer Figure 5).

In this case, the P-value is less than α (0.05). Therefore, we reject null hypothesis. As per the alternate hypothesis, the average price of the cars produced in the year 2000-2010 are greater than cars produced in the year 1990-2000.

From Figure 5, Positive Z value indicates the increase in used car price from 1990-2000 to 2000-2010.

5.3.3 Analyze how the average price of the car differs among different types of drive trains

Among all the transformation, Sqrt transformation satisfied both F-test and residual analysis conditions.

ANOVA F-test: Goodness of Fit Test

From the generated, the p-value is calculated as 2.2e-16 (refer Figure 7). As per the alternate hypothesis, the average price of all the drive train types are not equal.

ANOVA-Individual t-test

From Figure 19, as P-value < α, we accept null hypothesis. So, as per null hypothesis all these variables are significant, and their corresponding coefficients are not equal to zero.

5.3.4 Build the multiple linear regression model for used car price prediction

After performing all the transformation, the Sqrt transformation that generated the least RMSE value was considered to build the regression model.

Outliers had been identified and removed using Cook's distance method.

Finally, model_step3 was considered as the final fitted model to predict the used car price.

So, the final fitted regression model is:

Sqrt(Price_usd) = 41.97 + 14.96 (transmissionautomatic) + 37.10 (yearproduced00_20)

+ 21.10 (body_typeSUV) + 58.47 (Statenew)- 1.21 (is_exchangeableTRUE)+Error

After back transformation:

Price_usd = 1761.48 + 223.80 (transmissionautomatic) + 1376.41 (yearproduced00_20)

+ 445.21 (body_typeSUV) + 3418.74 (Statenew) - 1.46 (is_exchangeableTRUE)+Error

According to the final regression model,

- Whenever the car has automatic transmission, the price of the car increases by 223.80 units assuming all the other variables constant.
- If the car is produced between the year 2000-2020, then the price of the car increase by 1376.41 units assuming all the other variables constant.
- If the body type of the car is SUV, then the price of the car increases by 445.21 units assuming all the other variables constant.
- If the car is in a new state, then the price of the car increases by 3418.74 units assuming all the other variables constant.
- Whenever the car exchange is TRUE, the price of the car decreases by 1.46 units assuming all the other variables constant.

6. Conclusions and Future Work

6.1. Conclusions

Following are the conclusions I derived from this project

- The average price between automatic transmission cars and manual transmission cars are not equal
- The average price of the cars produced between 2000-2010 are greater than the cars produced between 1990-2000
- The average price of the front, rear and all wheel drive train are not equal.
- The price of the used car can be predicted using the following regression model

Price_usd = 1761.48 + 223.80 (transmissionautomatic) + 1376.41 (yearproduced00_20)
+ 445.21 (body_typeSUV) + 3418.74 (Statenew) - 1.46 (is_exchangeableTRUE)+Error

6.2. Limitations

The limitations of this project are,

- The features are excluded from the dataset as it contains incomplete descriptions.
- The data was limited to car sales in Belarus region.

6.3. Potential Improvements or Future Work

- The used car sales data from different countries/regions would add make the regression model more robust with higher dataset.
- The car models were divided based on the manufacturing continent. Country wise separation will provide more specific details.
- The data set contains cars manufactured from 1940-2020. So, separating the antique and modern cars would give more insight.