Final Project-R Outputs
Group Number: 302
Name: Arpitha Jagadish
CWID: A20453142

Table of Figures

- Loading Dataset to R Environment

```
> #Load the Dataset to R Environment
> car_price <- read.csv ("cars-price.csv", header = T)
> #Display first 6 records
> head(car_price)
  manufacturer_name model_name transmission  color odometer_value year_produced engine_fuel engine_has_gas engine_type engine_capacity body_type
1            Subaru    Outback    automatic silver         190000          2010    gasoline          FALSE    gasoline             2.5 universal
2            Subaru    Outback    automatic   blue         290000          2002    gasoline          FALSE    gasoline             3.0 universal
3            Subaru   Forester    automatic    red         402000          2001    gasoline          FALSE    gasoline             2.5       suv
4            Subaru    Impreza   mechanical   blue          10000          1999    gasoline          FALSE    gasoline             3.0     sedan
5            Subaru     Legacy    automatic  black         280000          2001    gasoline          FALSE    gasoline             2.5 universal
6            Subaru    Outback    automatic silver         132449          2011    gasoline          FALSE    gasoline             2.5 universal
  has_warranty state drivetrain price_usd is_exchangeable location_region number_of_photos up_counter feature_0 feature_1 feature_2 feature_3
1        FALSE owned        all  10900.00           FALSE    Минская обл.               9         13     FALSE      TRUE      TRUE      TRUE
2        FALSE owned        all   5000.00            TRUE    Минская обл.              12         54     FALSE      TRUE     FALSE     FALSE
3        FALSE owned        all   2800.00            TRUE    Минская обл.               4         72     FALSE      TRUE     FALSE     FALSE
4        FALSE owned        all   9999.00            TRUE    Минская обл.               9         42      TRUE     FALSE     FALSE     FALSE
5        FALSE owned        all   2134.11            TRUE Гомельская обл.              14          7     FALSE      TRUE     FALSE      TRUE
6        FALSE owned        all  14700.00            TRUE    Минская обл.              20         56     FALSE      TRUE     FALSE     FALSE
  feature_4 feature_5 feature_6 feature_7 feature_8 feature_9 duration_listed
1     FALSE      TRUE     FALSE      TRUE      TRUE      TRUE             16
2      TRUE      TRUE     FALSE     FALSE     FALSE      TRUE             83
3     FALSE     FALSE     FALSE     FALSE      TRUE      TRUE            151
4     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE             86
5      TRUE     FALSE     FALSE     FALSE     FALSE      TRUE              7
6     FALSE      TRUE     FALSE      TRUE      TRUE      TRUE             67
>
```

Figure 1: Loading Dataset

Research Problem 1:

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

Box Plot:

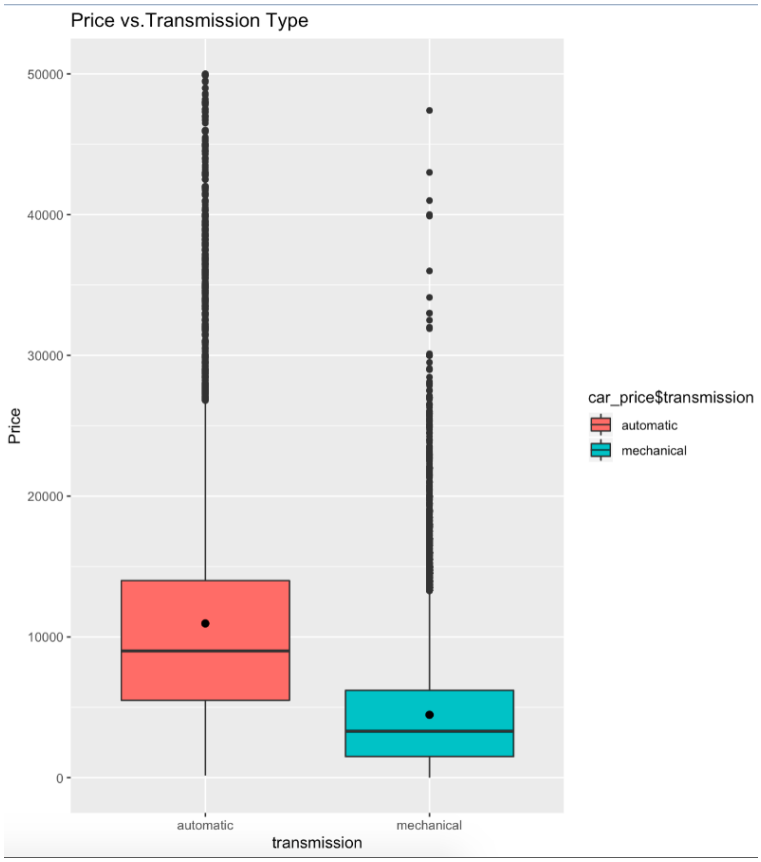


Figure 2 : Box plot for Transmission vs Car price

Z-test:

```
> # 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

Research Problem 2:

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

Box Plot

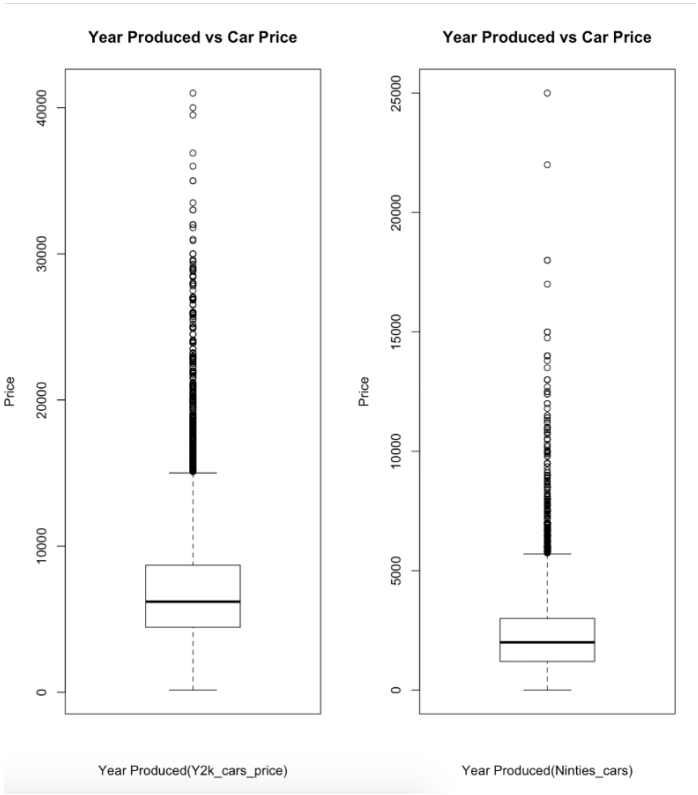


Figure 4: Box plot for Year produced vs Car price

Z-test

```
> #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

Research Problem 3:
Analyze how the average price of the car differs among different types of drive trains

Box Plot:

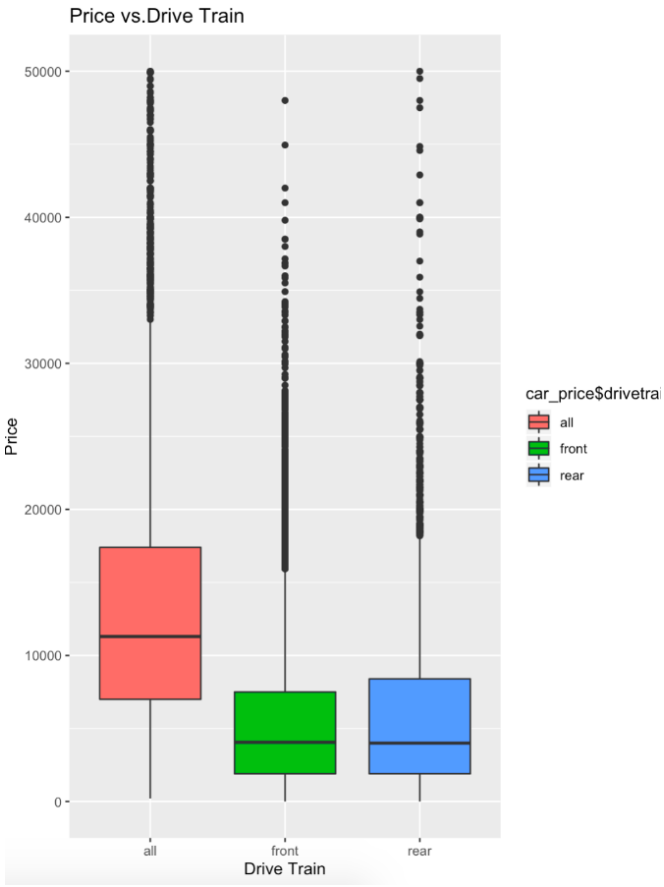


Figure 6: Box plot for different drive train vs Car price

ANOVA F-Test Hypothesis

```
> 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:

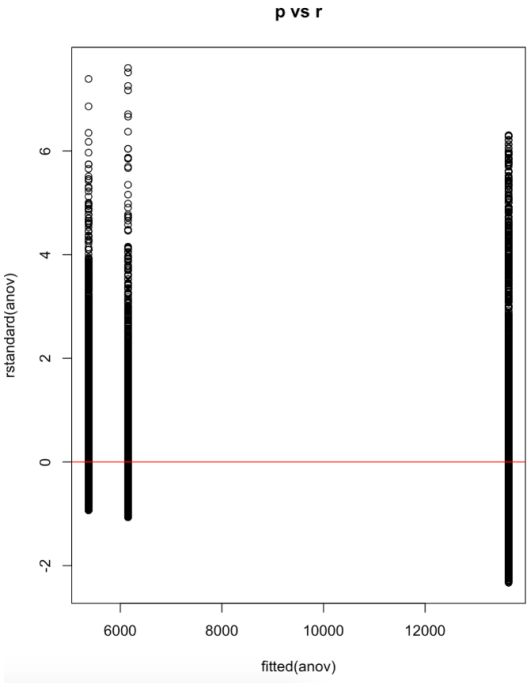


Figure 8: anov -Residuals vs predicted values

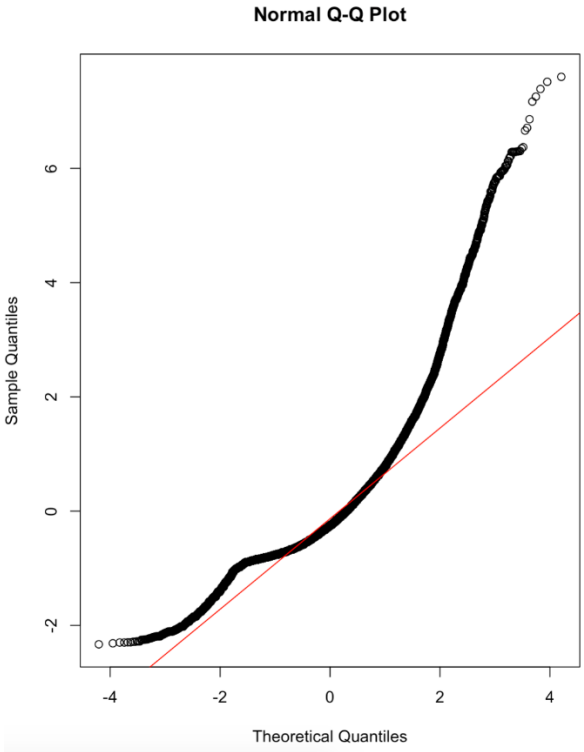


Figure 9: anov-Normal Q-Q plot

Y- Transformation:

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:

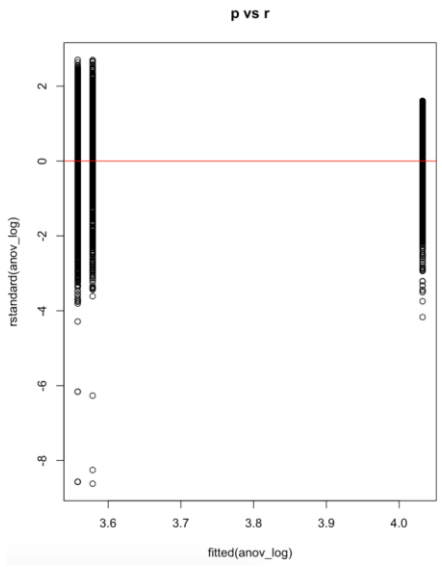


Figure 11: anov_log Residuals vs predicted values

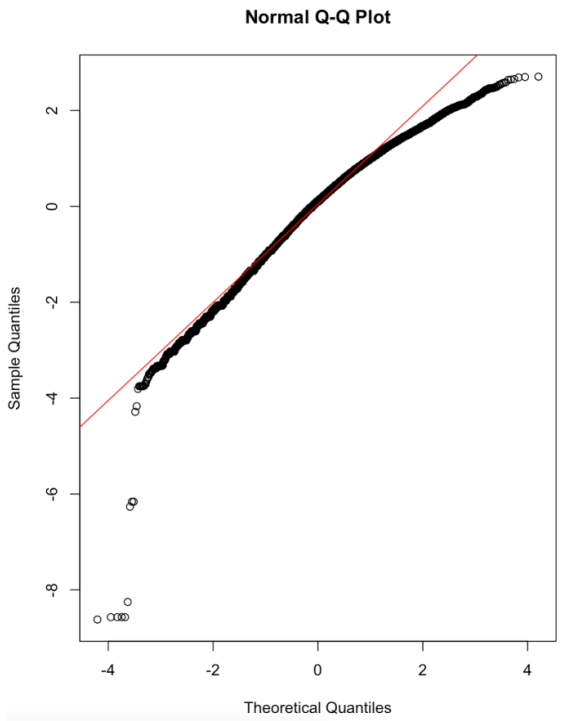


Figure 12: anov_log Normal Q-Q plot

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:

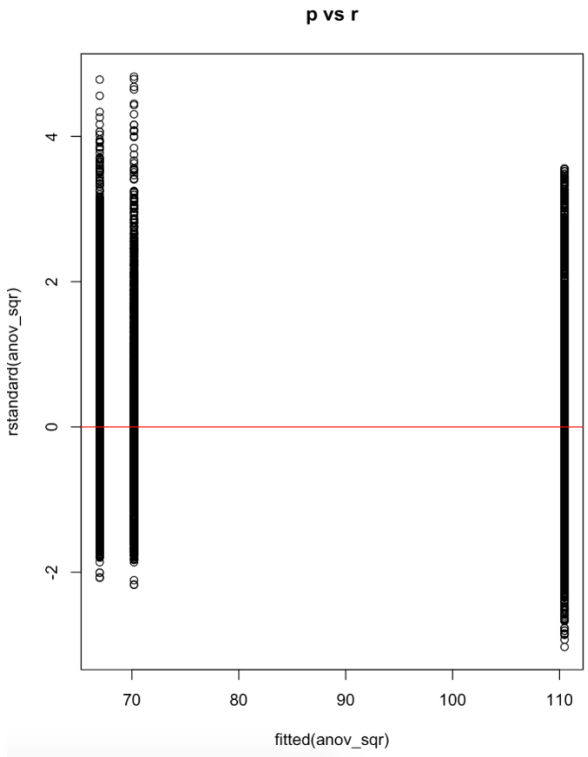


Figure 14: anov_sqr residulas vs predicted valve

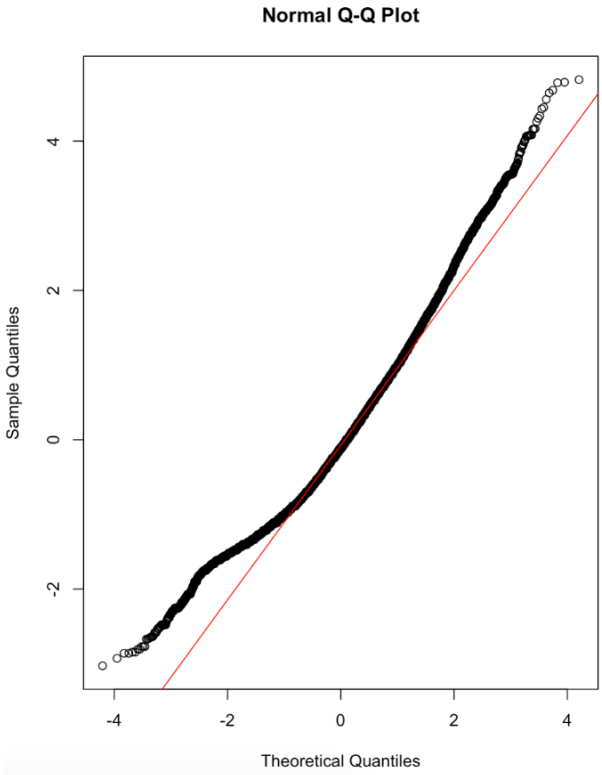


Figure 15: anov_sqr Noraml Q-Q plot

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:

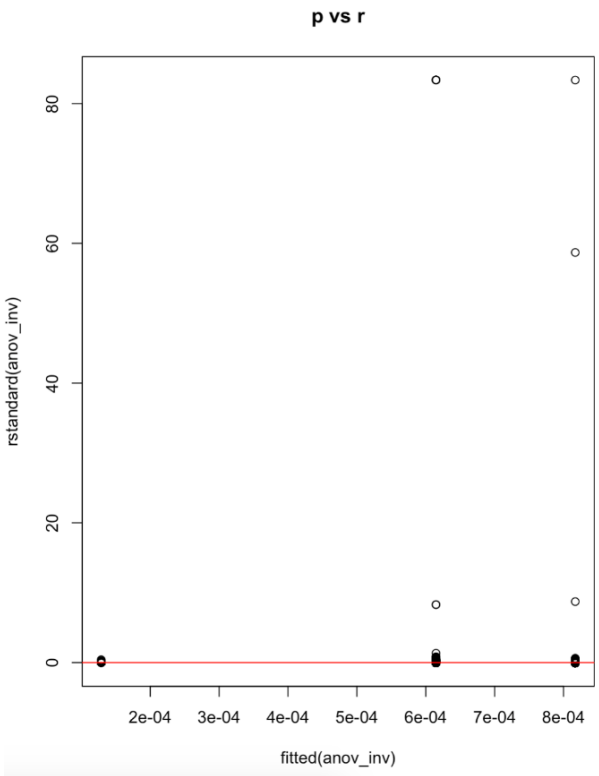


Figure 17: anov_inv residual vs predicted value

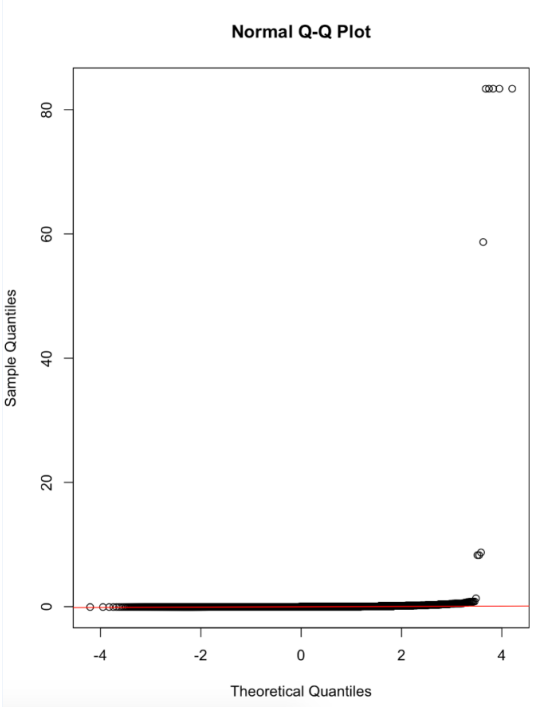


Figure18: anov_inv Normal Q-Q plot

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

Research Problem 4:
Build the multiple linear regression model for used car price prediction

Preprocessing:

```
> #-----------------------------------------------------------Pre-Processing(Assigning the manufacturers to corresponding continents)-----------------------------------------------
>
> #Listing unique manfacurer name
> unique(car_price$manufacturer_name)
 [1] Subaru        LADA          Dodge         YA3           УА3           Kia           Opel          Muscovite     Alfa Romeo    Acura         Dacia         Lexus         Mitsubishi    Lancia
[15] Citroen       Mini          Jaguar        Porsche       SsangYong     Daewoo        Geely         ВА3           Fiat          Ford          Renault       Seat          Rover         Volkswagen
[29] Lifan         Jeep          Cadillac      Audi          3A3           Toyota        ГА3           GAZ           Volvo         Chevrolet     Great Wall    Buick         Pontiac       Lincoln
[43] Hyundai       Nissan        Suzuki        BMW           Mazda         Land Rover    Iveco         Skoda         Saab          Infiniti      Chery         Honda         Mercedes-Benz Peugeot
[57] Chrysler
57 Levels: Acura Alfa Romeo Audi BMW Buick Cadillac Chery Chevrolet Chrysler Citroen Dacia Daewoo Dodge Fiat Ford GAZ Geely Great Wall Honda Hyundai Infiniti Iveco Jaguar Jeep Kia LADA Lancia ... УА3
>
> #Replacing manufacturer name yA3 with YA3 as they both are same, but displayed as duplicate
> levels(car_price$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        ГА3           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",
+                                                                                     "ГА3",  "Volvo", "BMW" , "Land Rover", "Iveco", "Skoda","Saab",
+                                                                                    "Mercedes-Benz","Peugeot"),"Europe","America"))
>
> #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

```
> #-------------------------------------------------------------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

```
> #-------------------------------------------------------------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

```
> #-----------------------------------------------------------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

```
> #-----------------------------------------------------------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

```
> #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

```
> #-----------------------------------------------------------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*

```
> ##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*

Building Regression Model:

```
> #----------------------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
> # 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:
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
> # 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

Residual Analysis:

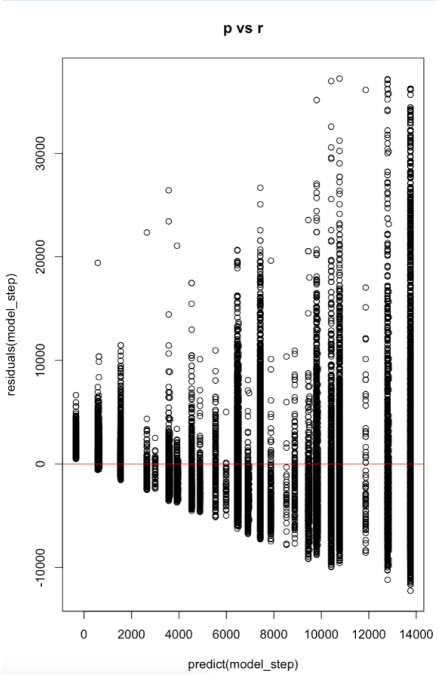
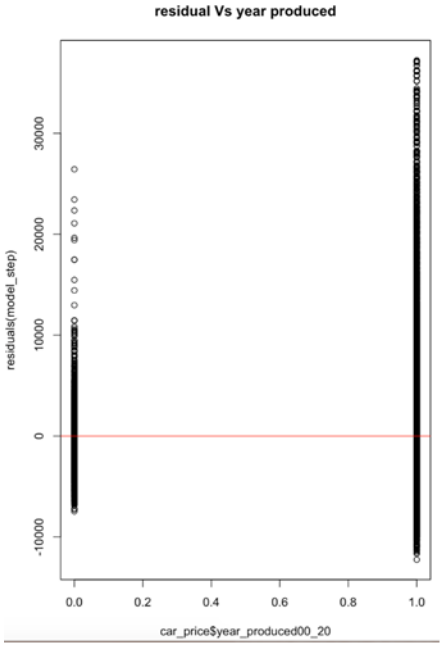
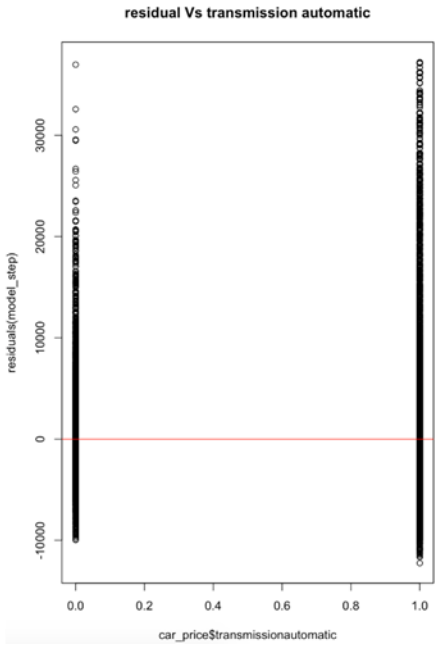


Figure 32: model_step: Validating constant variance





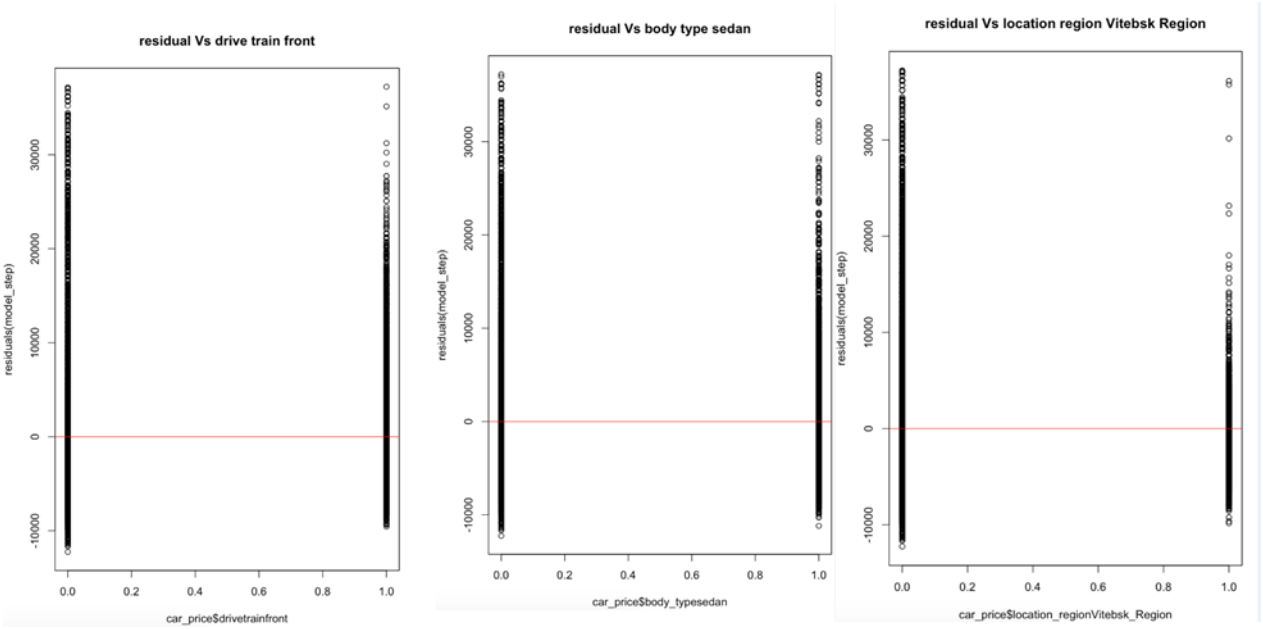


Figure 33: model_step: Validating linearity

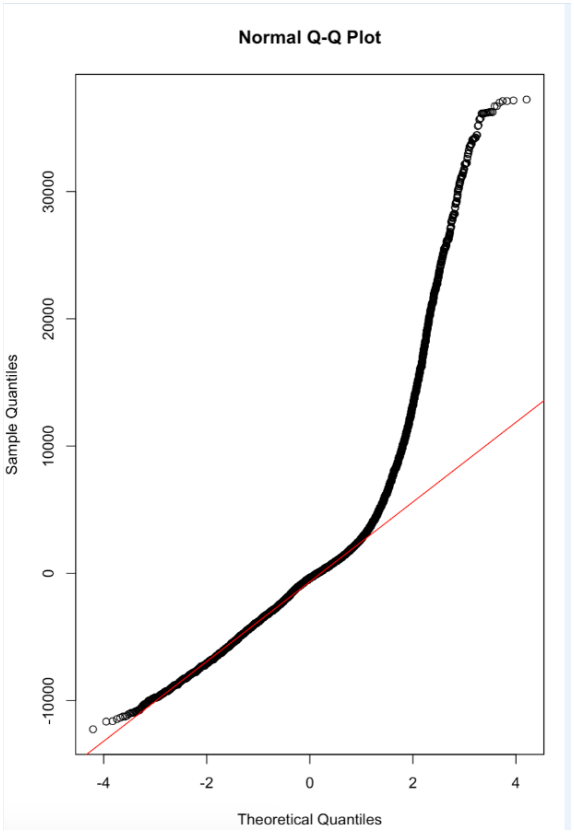


Figure 34: model_step: Normality test

Y-Transformation:

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

```
> #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

```
> #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

```
> #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 Type	RMSE
Log	4858.10
Sqrt	4505.35
Inverse	359015.7

Figure 41: Transformation comparison

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:

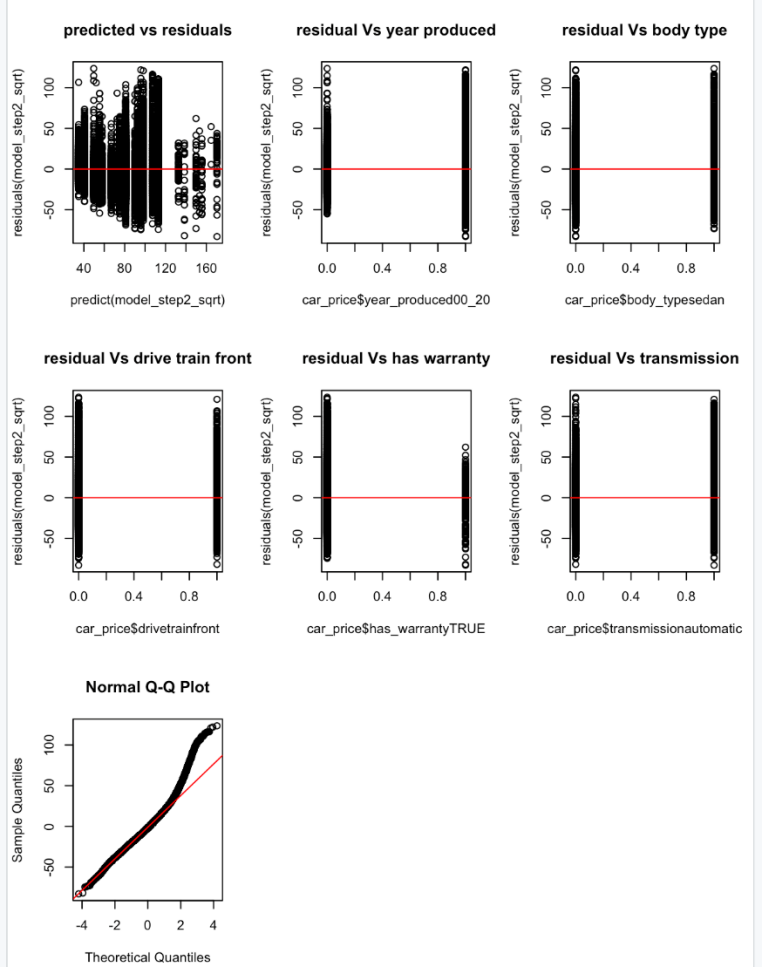


Figure 43: model_step2: Residual analysis

Removing Outliers:

```
> #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

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

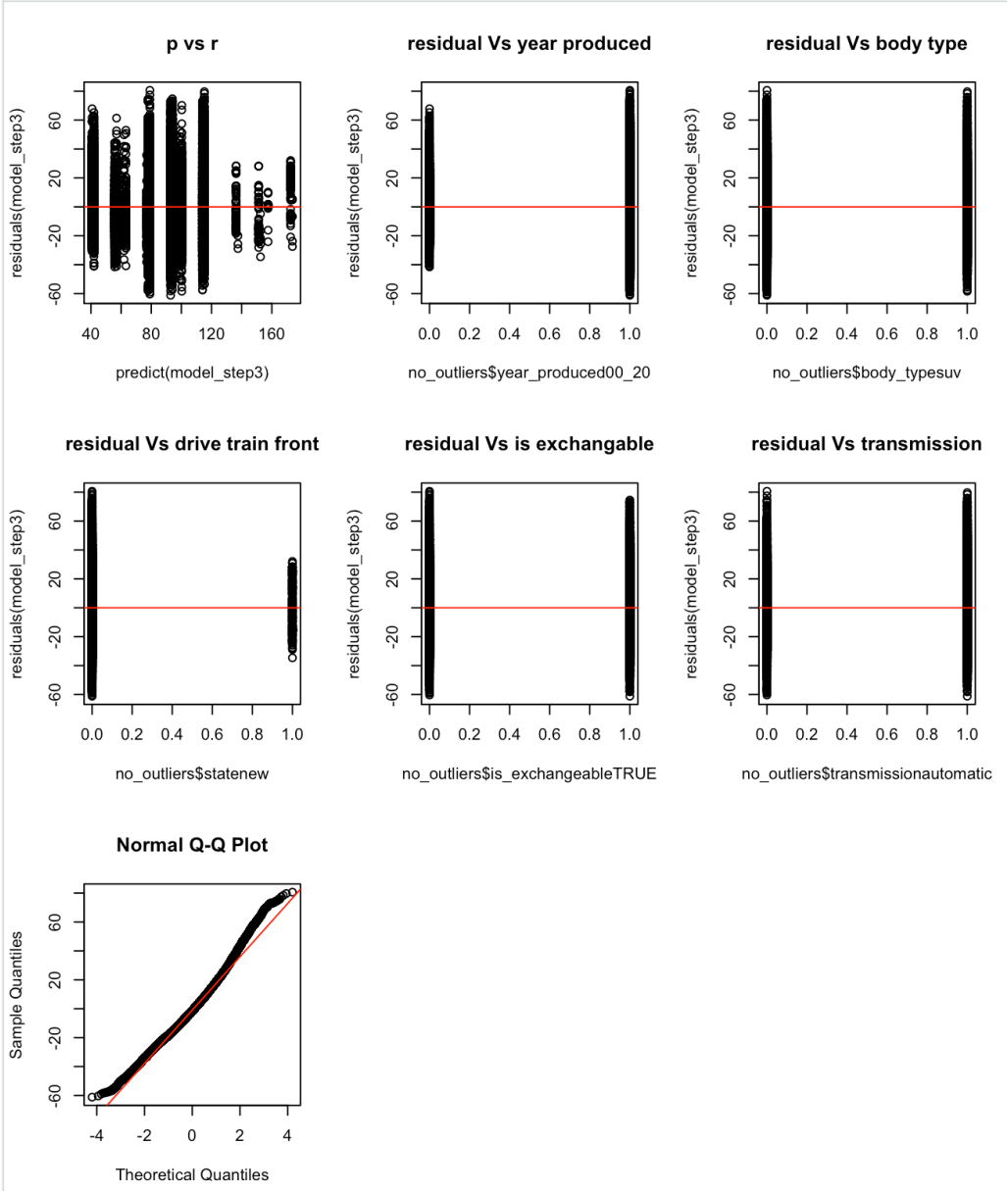


Figure 46: Residual Analysis:model_step3