UNIVERSITY OF PADUA

Department of Mathematics
Master's Degree in Data Science

Statistical learning project (Mod B)

Car Price Prediction

Students

Roya Ghamari - Alireza Saberi – Nahid Jahanianarange

July 2023

Contents

project

2023-04-18

```
## Loading required package: Matrix

## Loaded glmnet 4.1-7

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.1, built: 2022-11-18)
## ## Copyright (C) 2005-2023 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## The following objects are masked from 'package:Metrics':
##
##     precision, recall

## Loading required package: MASS

## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':
##
##     melanoma

## Loading required package: survey

## Loading required package: grid

## Loading required package: survival
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

## The following object is masked from 'package:caret':
##
##     cluster

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric
pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.

## Loaded gbm 2.1.8.1

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.1     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ tidyr::expand() masks Matrix::expand()
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ caret::lift()   masks purrr::lift()
## ✖ tidyr::pack()   masks Matrix::pack()
## ✖ dplyr::select() masks MASS::select()
## ✖ tidyr::unpack() masks Matrix::unpack()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

1. Objectives of the study

In this study, we will explore the dataset of Car Dekho. The dataset is about secondhand cars and various features of the Cars. Our goal is to identify the most important variables in determining the price of used cars, and to model data based on these variables in order to predict the selling price of used cars with different manufacturing date. We first start with analysis of the factors that influence price of used cars by exploring the dataset and all the components in order to discover correlations between data with a simple linear regression model and then we will use multiple and polynomial regressions as well as a lasso model to find a proper model to predict the price.

2. Preparation of the Dataset

The website of Car Dekho is https://www.cardekho.com/ and the dataset is also available on https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho in a public domain. This dataset consists of sale prices of 8128 cares sold between 1983 and 2020. It also has some features like km the car has been diven, name of the car, type of the fuel for cars. So, the dataset has 8128 rows and 13 columns.

In the code, as first steps we started importing some libraries and the dataset.

```
library(stringr)
library(purrr)
library(Amelia)
library(GGally)
library(caret)
library(relaimpo)
library(gbm)
library(broom)
library(knitr)
library(ggplot2)
library(tidyverse)
library(glmnet)
library(Metrics)
```

Importing dataset

```
car <- read.csv('Car details v3.csv')
attach(car)
```

We made a first inspection on the dataset.

```
str(car)
```

```
## 'data.frame':    8128 obs. of  13 variables:
##  $ name          : chr  "Maruti Swift Dzire VDI" "Skoda Rapid 1.5 TDI
Ambition" "Honda City 2017-2020 EXi" "Hyundai i20 Sportz Diesel" ...
##  $ year          : int  2014 2014 2006 2010 2007 2017 2007 2001 2011 2013
...
```

```
##  $ selling_price: int  450000 370000 158000 225000 130000 440000 96000
45000 350000 200000 ...
##  $ km_driven    : int  145500 120000 140000 127000 120000 45000 175000
5000 90000 169000 ...
##  $ fuel         : chr  "Diesel" "Diesel" "Petrol" "Diesel" ...
##  $ seller_type  : chr  "Individual" "Individual" "Individual" "Individual"
...
##  $ transmission : chr  "Manual" "Manual" "Manual" "Manual" ...
##  $ owner        : chr  "First Owner" "Second Owner" "Third Owner" "First
Owner" ...
##  $ mileage      : chr  "23.4 kmpl" "21.14 kmpl" "17.7 kmpl" "23.0 kmpl"
...
##  $ engine       : chr  "1248 CC" "1498 CC" "1497 CC" "1396 CC" ...
##  $ max_power    : chr  "74 bhp" "103.52 bhp" "78 bhp" "90 bhp" ...
##  $ torque       : chr  "190Nm@ 2000rpm" "250Nm@ 1500-2500rpm" "12.7@
2,700(kgm@ rpm)" "22.4 kgm at 1750-2750rpm" ...
##  $ seats        : int  5 5 5 5 5 5 5 4 5 5 ...

car$name <- word(car$name,1)
```

2.1. Variables description

There is a description of variables of the dataset:

Name: Name of the cars

Year: Year of the car when it was bought

Selling_price: Price at which the car is being sold

Km_driven: Number of Kilometers the car is driven

Fuel: Fuel type of car (petrol / diesel / CNG / LPG / electric)

Seller_type: Tells if a Seller is Individual or a Dealer or a TrustMark dealer. TrustMark is a certification and warranty programme launched by CarDekho.com and Gaadi.com to make your used car purchase safer

Transmission: Gear transmission of the car (Automatic/Manual)

Owner: Number of previous owners of the car.

Mileage: mileage of the car (kmpl) which is the number of miles that it can travel using one gallon or litre of fuel.

Engine: engine capacity of the car (cc)

Max_power: the amount of power that a car's engine generates to move it (bhp)

Torque: a physical quantity that indicates the traction in an engine design (Nm)

Seats: The number of seats in a car

2.2. Preprocessing

For the preprocessing part, we first analyzed the dimension of the dataset and checked for the presence of null values. There are no null values.

```
# dimension of the dataset
dim(car)

## [1] 8128   13

sum(is.na(car))

## [1] 221
```

It is clear that our dataset has 221 not available value, which means we should take care of them in the preprocessing step. After that, we inspect the response variable price and make a log transformation.
check on the response variable

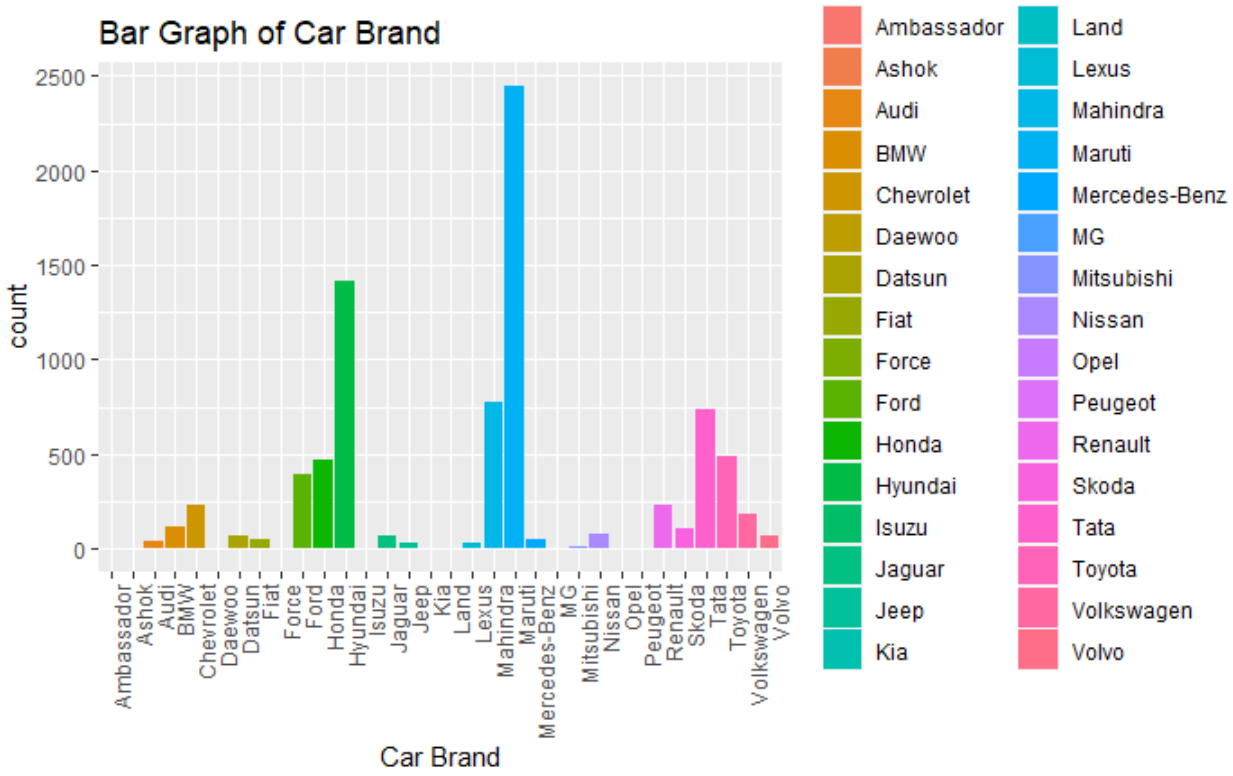
```
summary(car$selling_price)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    29999   254999   450000   638272   675000 10000000
```

and now log transformation of :

```
# logarithmic transformation of the response variable price
car$log10_price = log10(selling_price)
```

EDA
In this step we plot the car name column to check the distribution of the cars and their brands

```
ggplot(data = car, aes(x=name, fill = name)) +
  geom_bar() + labs(x='Car Brand') + labs(title = "Bar Graph of Car Brand") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



According to the graph, highest numbers of cars fall into Maruti brand followed by Hyundai, Mahindra and Tata brands

2.3. Cleaning the Data

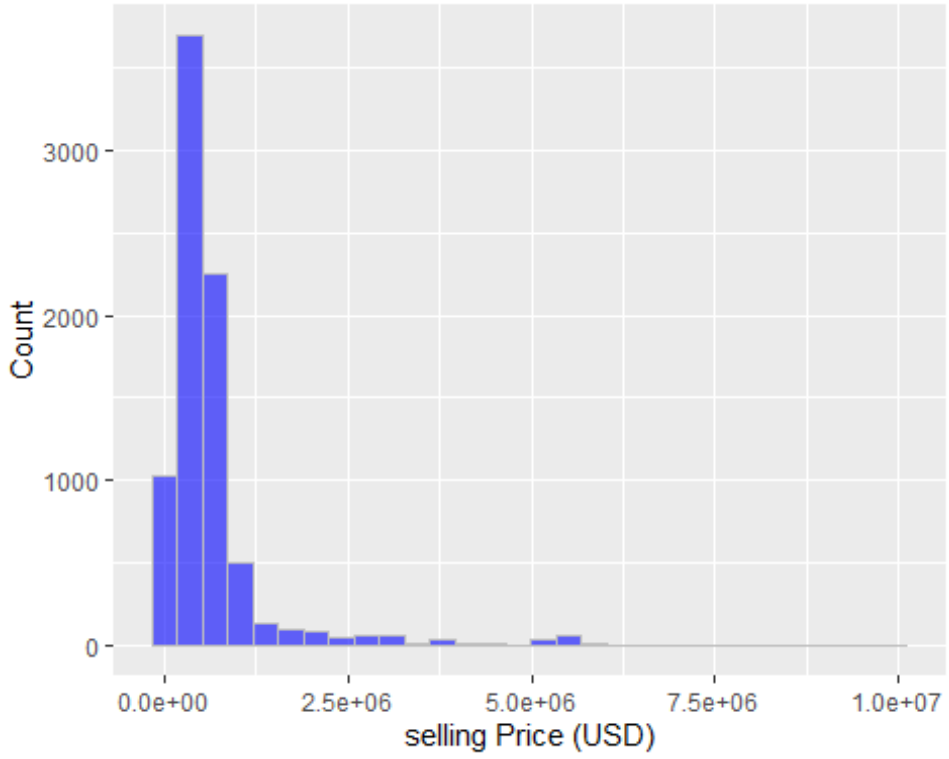
As we know the computer works with 0,1 and in specefic does not understand name of the cars. Then, we should encode our Categorical variables. In other words, we substitute the name of the cars with numbers.

```
car$name <- str_replace(car$name, 'Maruti', '0')
car$name <- str_replace(car$name, 'Skoda', '1')
car$name <- str_replace(car$name, 'Honda', '2')
car$name <- str_replace(car$name, 'Hyundai', '3')
car$name <- str_replace(car$name, 'Toyota', '4')
car$name <- str_replace(car$name, 'Ford', '5')
car$name <- str_replace(car$name, 'Renault', '6')
car$name <- str_replace(car$name, 'Mahindra', '7')
car$name <- str_replace(car$name, 'Tata', '8')
car$name <- str_replace(car$name, 'Chevrolet', '9')
car$name <- str_replace(car$name, 'Fiat', '10')
car$name <- str_replace(car$name, 'Datsun', '11')
car$name <- str_replace(car$name, 'Jeep', '12')
car$name <- str_replace(car$name, 'Mercedes-Benz', '13')
car$name <- str_replace(car$name, 'Mitsubishi', '14')
car$name <- str_replace(car$name, 'Audi', '15')
car$name <- str_replace(car$name, 'Volkswagen', '16')
```

```
car$name <- str_replace(car$name, 'BMW', '17')
car$name <- str_replace(car$name, 'Nissan', '18')
car$name <- str_replace(car$name, 'Lexus', '19')
car$name <- str_replace(car$name, 'Jaguar', '20')
car$name <- str_replace(car$name, 'Land', '21')
car$name <- str_replace(car$name, 'MG', '22')
car$name <- str_replace(car$name, 'Volvo', '23')
car$name <- str_replace(car$name, 'Daewoo', '24')
car$name <- str_replace(car$name, 'Kia', '25')
car$name <- str_replace(car$name, 'Force', '26')
car$name <- str_replace(car$name, 'Ambassador', '27')
car$name <- str_replace(car$name, 'Ashok', '28')
car$name <- str_replace(car$name, 'Isuzu', '29')
car$name <- str_replace(car$name, 'Opel', '30')
car$name <- str_replace(car$name, 'Peugeot', '31')

#Converting car name from categorical to numerical value

car$name <- as.numeric(car$name)
table(car$name)

##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
15
## 2448  105  467 1415  488  397  228  772  734  230   47   65   31   54   14
40
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
31
##  186  120   81   34   71    6    3   67    3    4    6    4    1    5    1
1
```

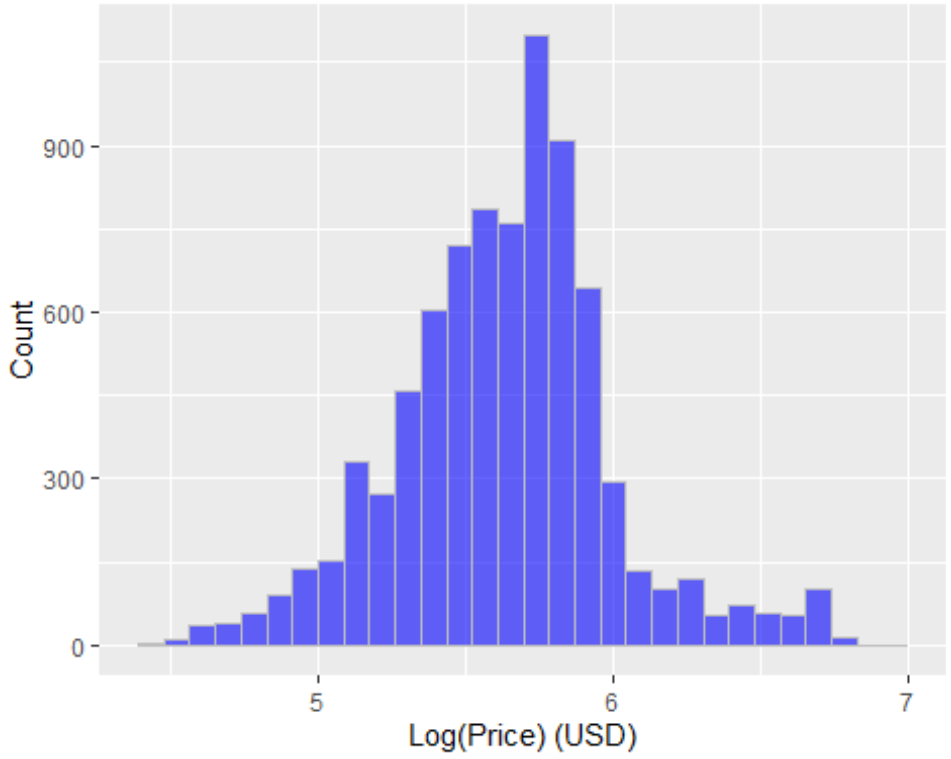
We start with plotting the distribution of some main variables:

```
ggplot(car, aes(x= selling_price)) +
geom_histogram(fill="blue", color="grey", alpha=0.6) +
labs(x ="selling Price (USD)", y="Count")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



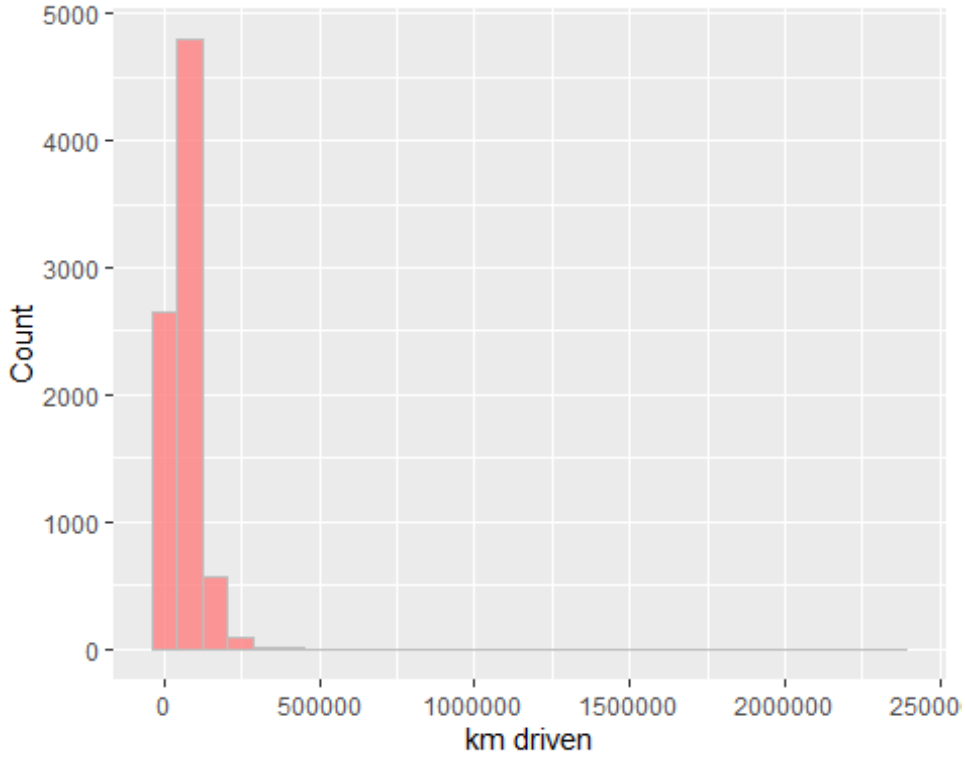
From the plot above we can observe that the selling price of majority of cars is less than two and half million dollars. Also it is clear that the distribution of the target variable price is right-skewd.

```
ggplot(car, aes(x= log10_price)) +
geom_histogram(fill="blue", color="grey", alpha=0.6) +
labs(x ="Log(Price) (USD)", y="Count")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Since our response variable (price) is right-skewed, we decided to apply a logarithm transformation on it. As we can observe, the distribution of the logarithmic transformation of price becomes bell-shaped.

```
ggplot(car, aes(x= km_driven)) +
geom_histogram(fill="#FF8080", color="grey",alpha=0.8) +
labs(x ="km driven",y="Count")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



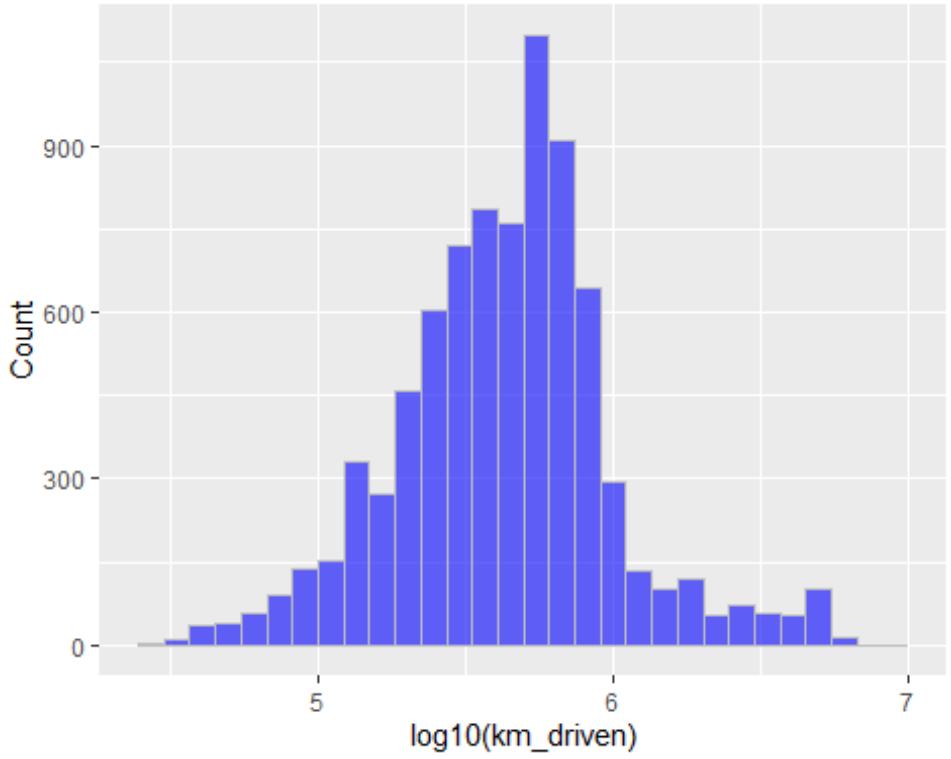

```
6
```

```
## [1] 6
```

As it is shown in the graph, most of the cars have been driven less than 500000 kms, and it is obvious that the graph is right-skewd. Then we do the same procedure we did toward the price of the car, which is using logarithm. As we can observe, the distribution of the logarithmic transformation of km driven becomes bell-shaped.

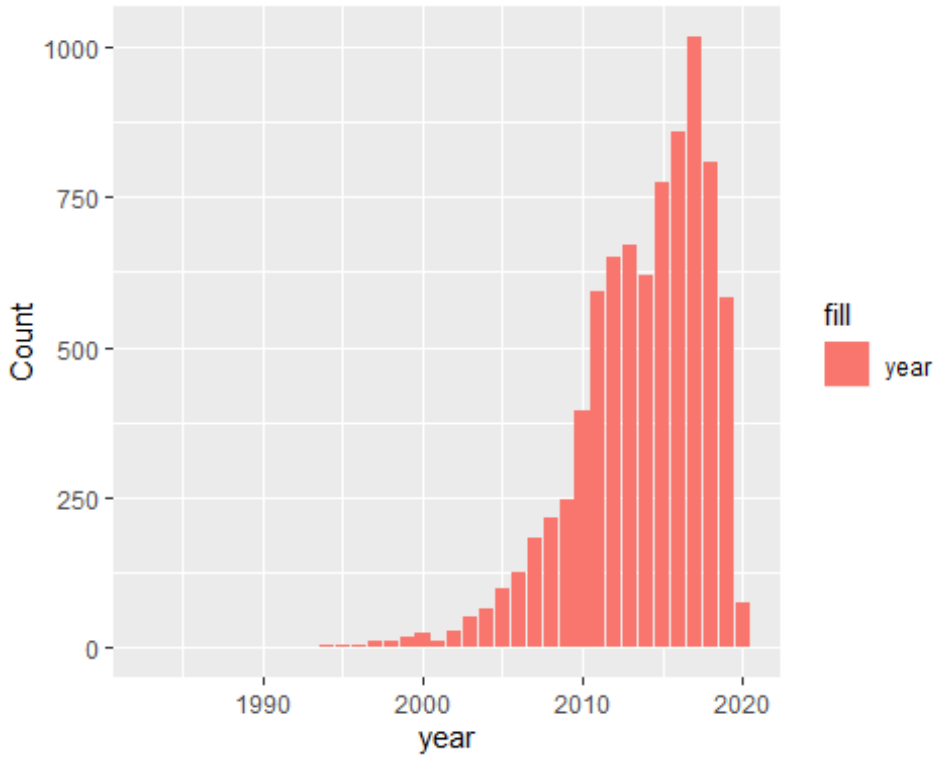
```
log10_km_driven <- log10(km_driven)
ggplot(car, aes(x= log10_price)) +
geom_histogram(fill="blue", color="grey", alpha=0.6) +
labs(x ="log10(km_driven)", y="Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The next variable of our dataset is the year that car has been sold. First of all, we draw the graph of the year column.

```
ggplot(car, aes(year, fill='year'))+
geom_bar() +
labs(x = 'year', y = 'Count')
```



Our graph is left-skewed and also the year of most of the sold cars in our dataset is after 2010.

```
car <- subset (car, select = -torque)
```

In our dataset, we have some values that their types are strings. As we know, we should give the computer numeric values. Then, we remove “kmpl” which stands for km per liter, and also “km/kg”. Another thing that should be considered is we should take care of missing values. we replaced the missing values with the mean value of their regarding column.

```
car$mileage <- str_replace(car$mileage, 'kmpl', '')
car$mileage <- str_replace(car$mileage, 'km/kg', '')
car$mileage <- as.numeric(car$mileage)
car$mileage[is.na(car$mileage)]<-mean(car$mileage,na.rm=TRUE)
```

In the Column of engine, we had a value with type of string which was “cc”. In order to have only numeric value, the string value should be removed. We also replaced not avilable value in engine column with the mean value of engine column.

```
car$engine <- str_replace(car$engine, 'CC', '')
car$engine <- as.numeric(car$engine)
car$engine[is.na(car$engine)]<-mean(car$engine,na.rm=TRUE)
```

Removing unit from max_power, converting it to numeric value and replacing the missing values with mean of max_power column

```
car$max_power <- str_replace(car$max_power, 'bhp', '')
car$max_power <- as.numeric(car$max_power)
car$max_power[is.na(car$max_power)]<-mean(car$max_power,na.rm=TRUE)
```

Converting seats to numeric value and replacing the missing values with mean value. The other thing that should be considered is that maybe in our dataset we have some null value but with not “NA” type, like null string. We should replace them with “NA” in order to use specific R function like is.na().

```
car$seats <- as.numeric(car$seats)
car$seats[is.na(car$seats)]<-median(car$seats,na.rm=TRUE)
car$mileage[car$mileage == ""] <- NA
car$engine[car$engine == ""] <- NA
car$max_power[car$max_power == ""] <- NA
```

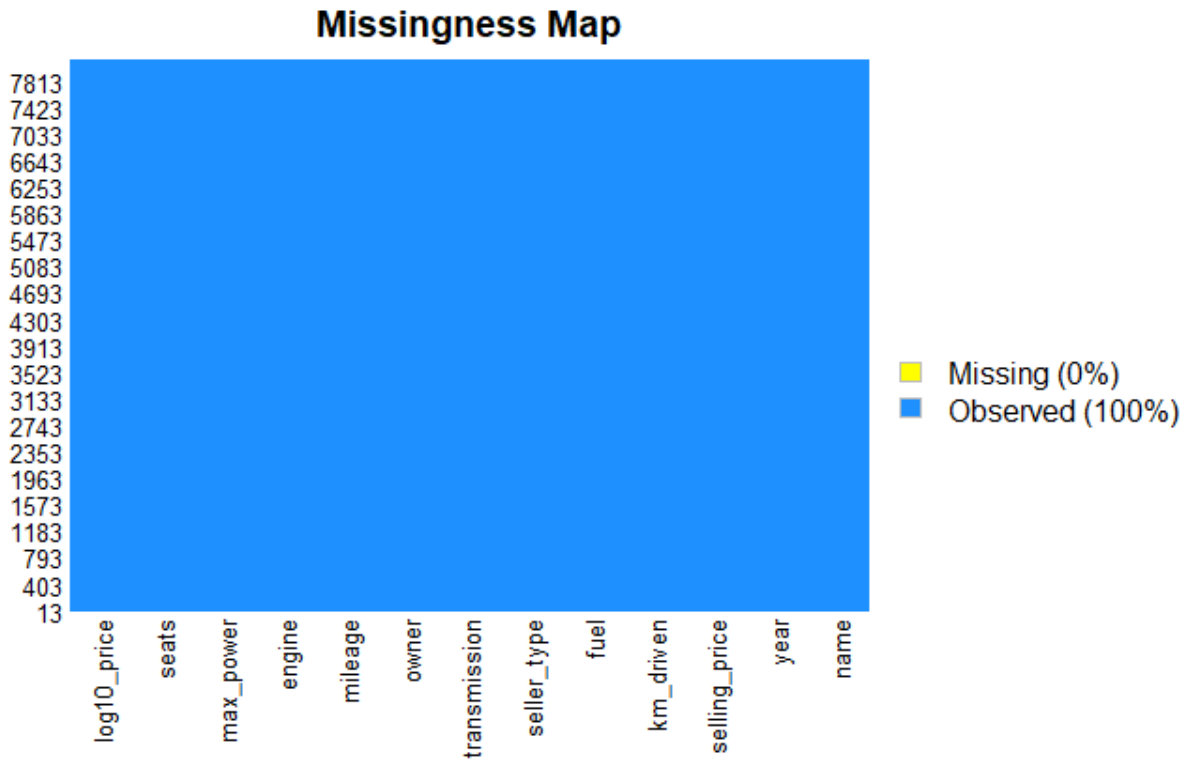
Checking for missing values

```
sapply(car, function(x) sum(is.na(x)))
```

```
##          name          year selling_price     km_driven          fuel
##             0             0             0             0             0
##   seller_type  transmission         owner       mileage        engine
##             0             0             0             0             0
##     max_power         seats   log10_price
##             0             0             0
```

Missing values map

```
missmap(car, legend = TRUE, col = c("yellow", "dodgerblue"))
```

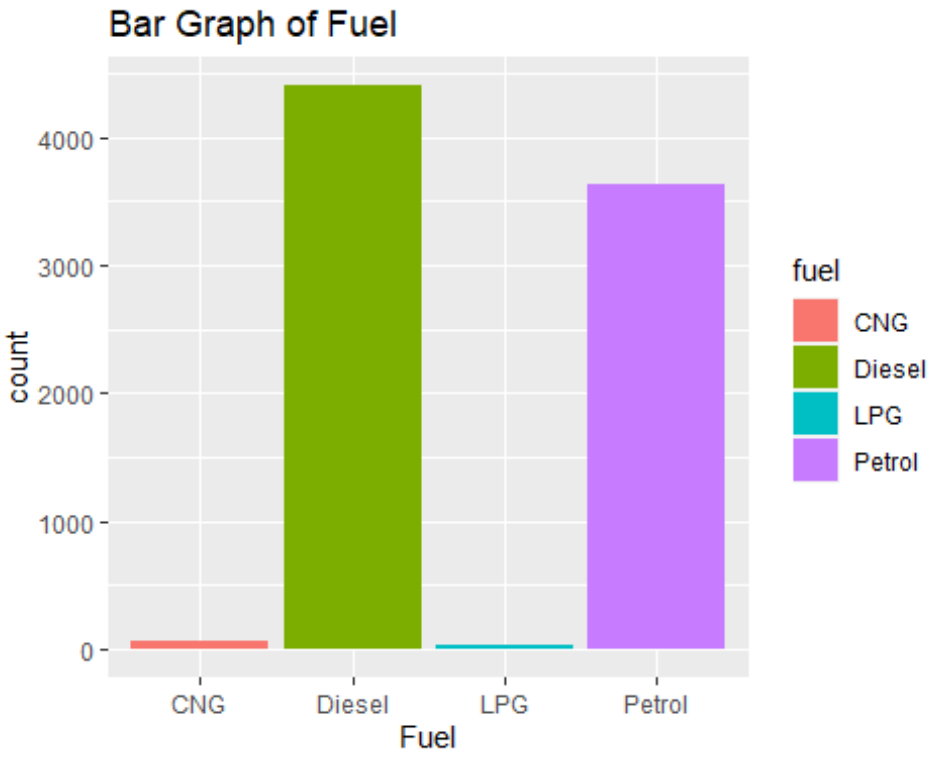


Now we ensure that there are no missing values in the columns.

3. Exploratory Data Analysis

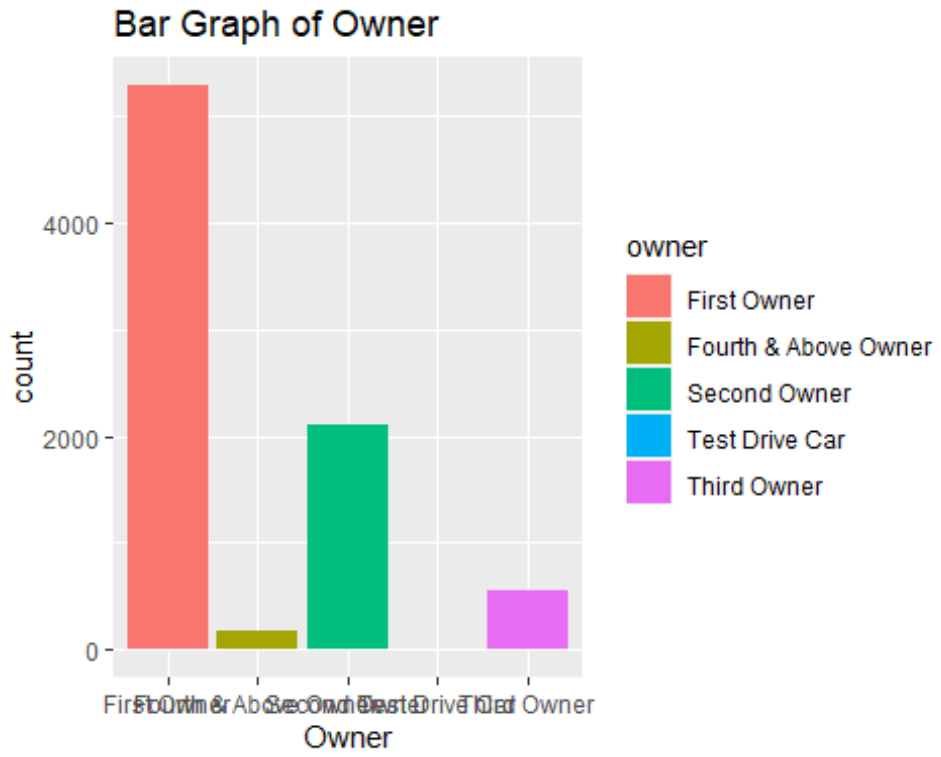
Bar graph of Fuel

```
ggplot(data = car, aes(x= fuel, fill = fuel)) +
  geom_bar() + labs(x='Fuel') + labs(title = "Bar Graph of Fuel")
```



Diesel and Petrol have the highest ownership for the fuel types.
Most of the cars fall into Diesel category followed by Petrol. Very few cars fall into CNG and LPG category.
Bar graph of Owner

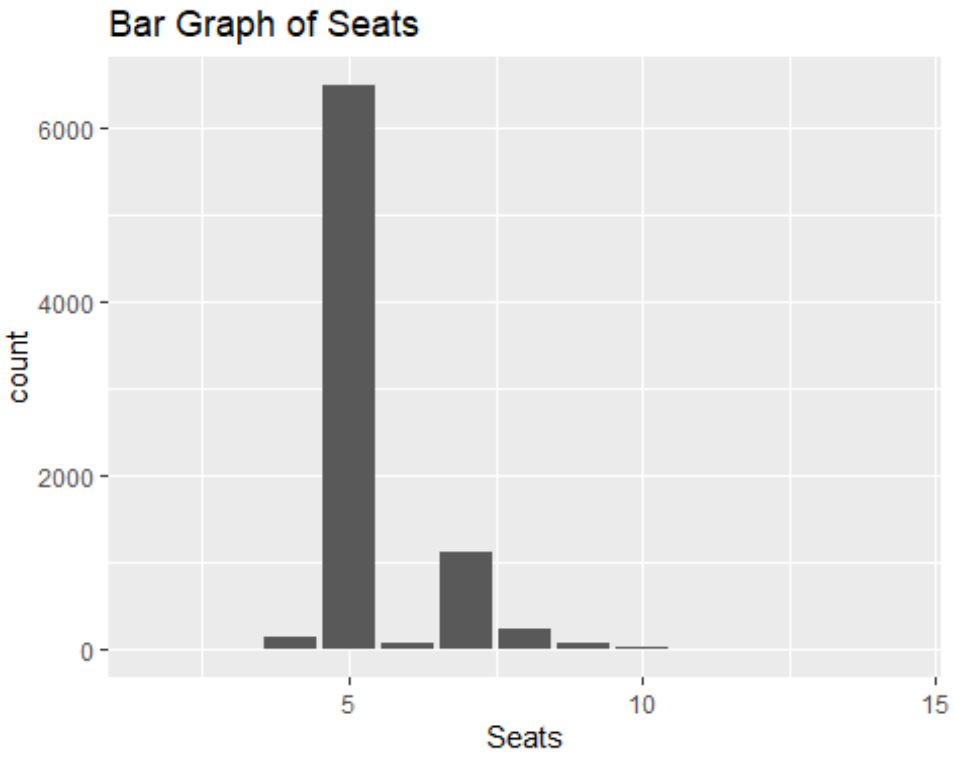
```
ggplot(data = car, aes(x=owner, fill = owner)) +
  geom_bar() + labs(x='Owner') + labs(title = "Bar Graph of Owner")
```



Most of the cars are owned by first owners.
Bar graph of seats

```
ggplot(data = car, aes(x=seats, fill =seats)) +
  geom_bar() + labs(x='Seats') + labs(title = "Bar Graph of Seats")
```

```
## Warning: The following aesthetics were dropped during statistical
transformation: fill
## ℹ This can happen when ggplot fails to infer the correct grouping
structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



Majority of the seats in the cars are 5. So compact cars are the most dominant one in the car market.
Converting transmission column into binary 0 if Manual and 1 if Automatic

```
car$transmission <- str_replace(car$transmission, 'Manual', "0")
car$transmission <- str_replace(car$transmission, 'Automatic', "1")
car$transmission <- as.numeric(car$transmission)
table(car$transmission)

##
##    0    1
## 7078 1050
```

Converting owner into Ordinal Encoder

```
car$owner <- str_replace(car$owner, 'First Owner', "0")
car$owner <- str_replace(car$owner, 'Second Owner', "1")
car$owner <- str_replace(car$owner, 'Third Owner', "2")
car$owner <- str_replace(car$owner, 'Fourth & Above Owner', "3")
car$owner <- str_replace(car$owner, 'Test Drive Car', "4")
car$owner <- as.numeric(car$owner)
table(car$owner)

##
##    0    1    2    3    4
## 5289 2105  555  174    5
```

Converting seller_type into Ordinal Encoder

```
car$seller_type <- str_replace(car$seller_type, "Trustmark Dealer", "0")
car$seller_type <- str_replace(car$seller_type, "Dealer", "1")
car$seller_type <- str_replace(car$seller_type, "Individual", "2")
car$seller_type <- as.numeric(car$seller_type)
table(car$seller_type)
```

```
##
##    0    1    2
##  236 1126 6766
```

Converting fuel into Ordinal Encoder

```
car$fuel <- str_replace(car$fuel, 'Diesel', "0")
car$fuel <- str_replace(car$fuel, 'Petrol', "1")
car$fuel <- str_replace(car$fuel, 'CNG', "2")
car$fuel <- str_replace(car$fuel, 'LPG', "3")
car$fuel <- as.numeric(car$fuel)
table(car$fuel)
```

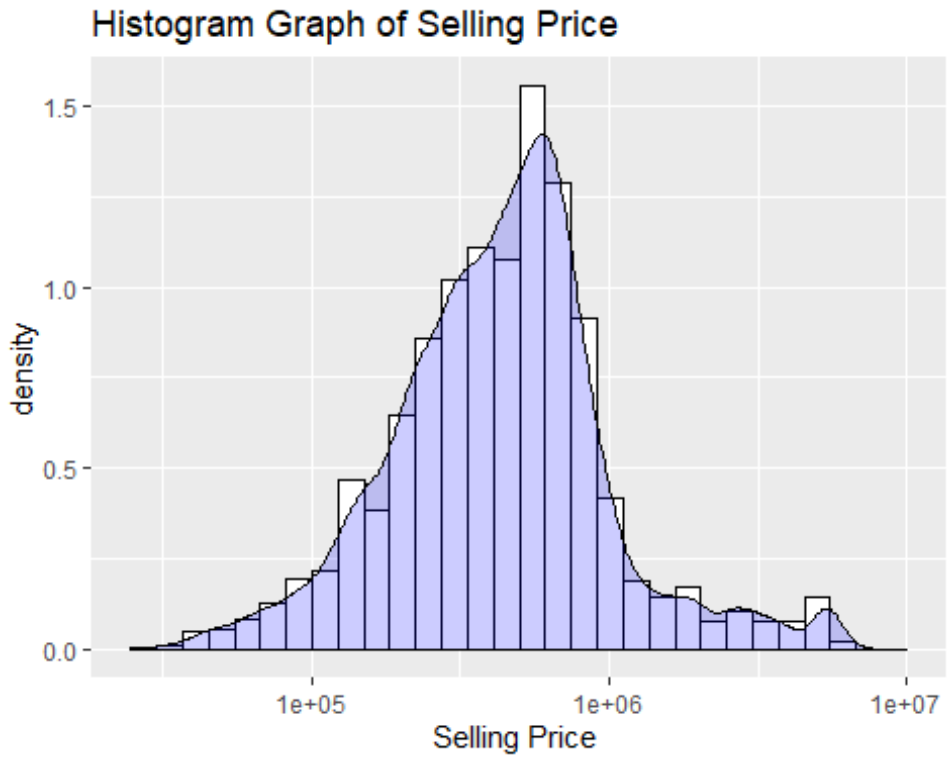
```
##
##    0    1    2    3
## 4402 3631   57   38
```

Histogram of Selling Price

```
ggplot(car, aes(x=selling_price)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="blue")+
  labs(x='Selling Price ') + labs(title = "Histogram Graph of Selling Price")
+
  scale_x_continuous(trans='log10')
```

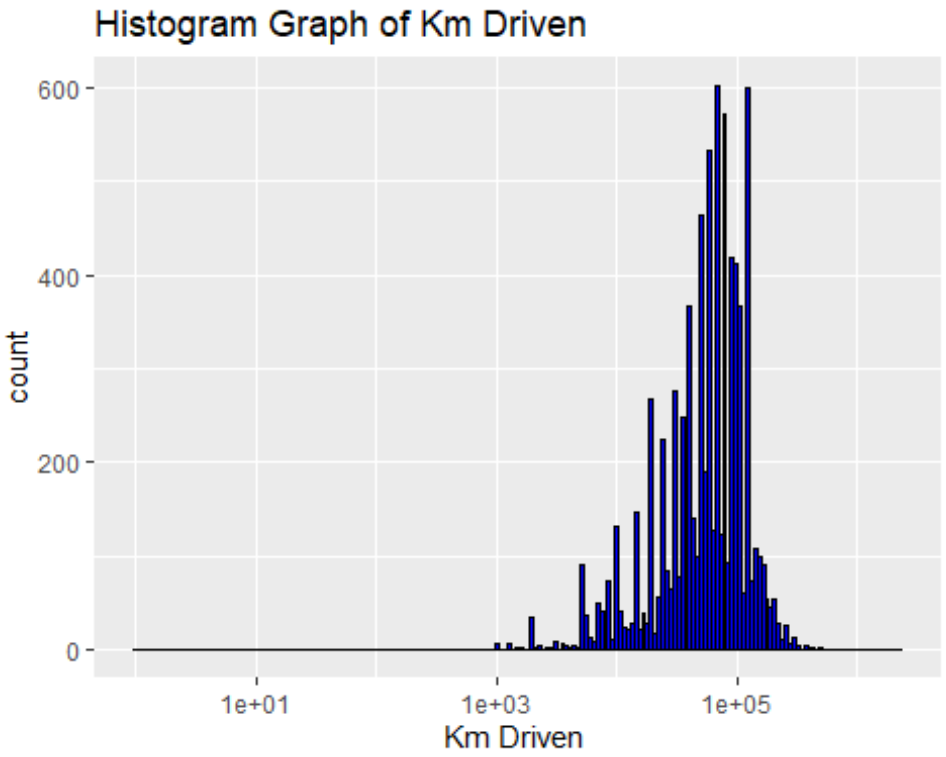
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2
3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram of Km Driven

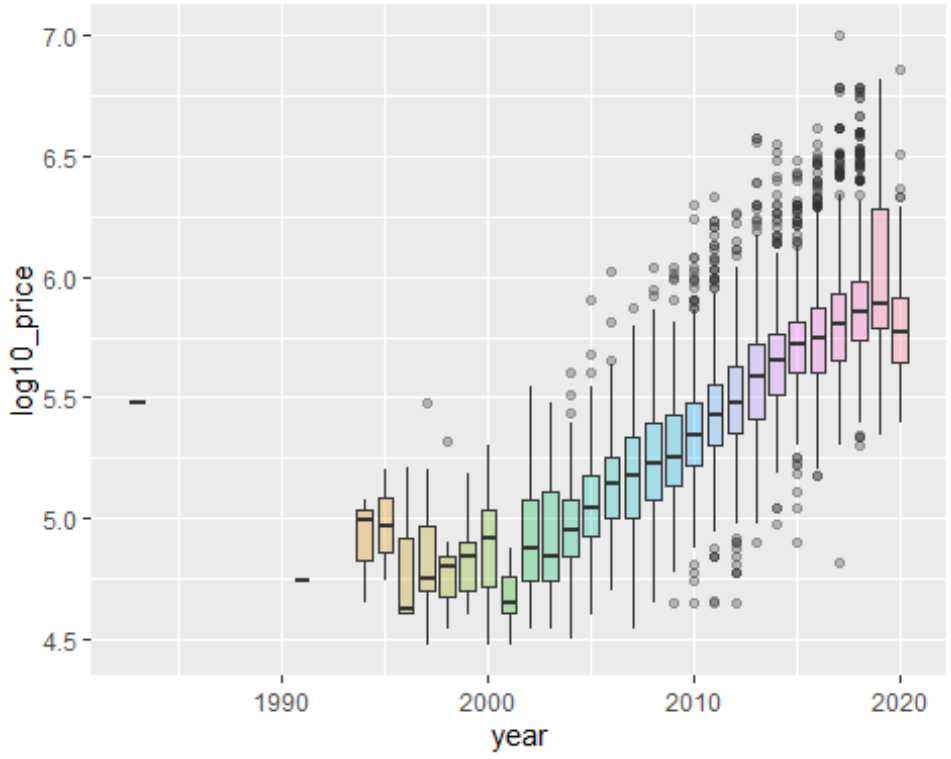
```
ggplot(car, aes(x=km_driven)) +
  geom_histogram(color="black", fill="blue", bins = 200)+
  labs(x='Km Driven ') + labs(title = "Histogram Graph of Km Driven") +
  scale_x_continuous(trans='log10')
```



Box plot of year

It can be seen from the box plot that the newer the cars, the more expensive they are which is reasonable.

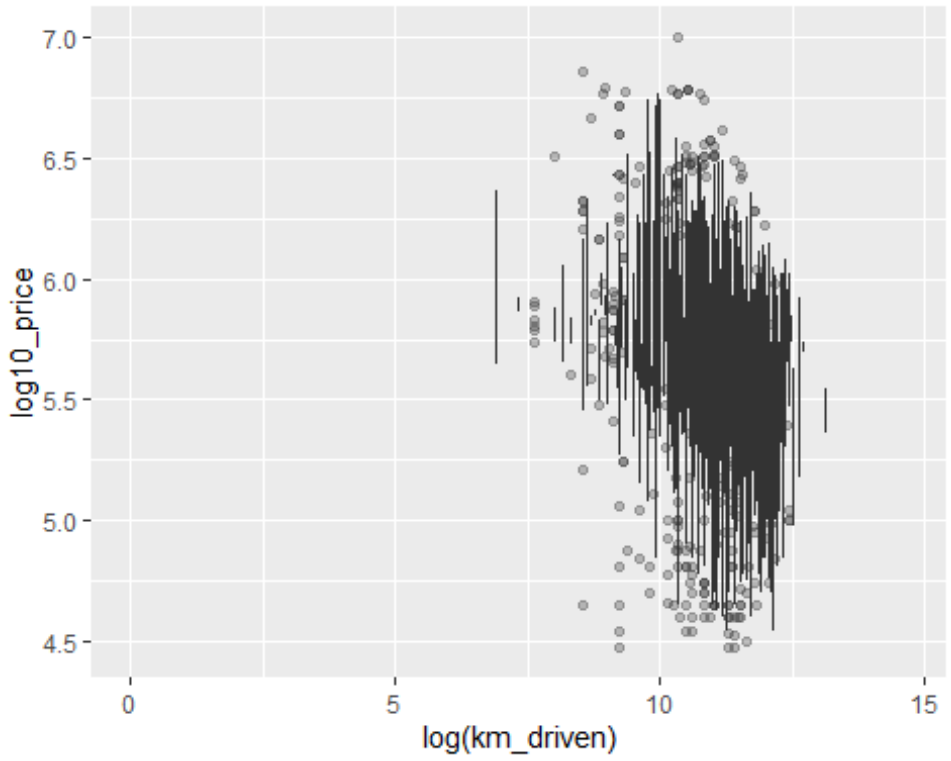
```
ggplot(car, aes(x=year, y=log10_price, fill=factor(year)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of Km Driven

It can be seen from the box plot that the greater number of Kilometers the car is driven, the lower the price which is reasonable again.

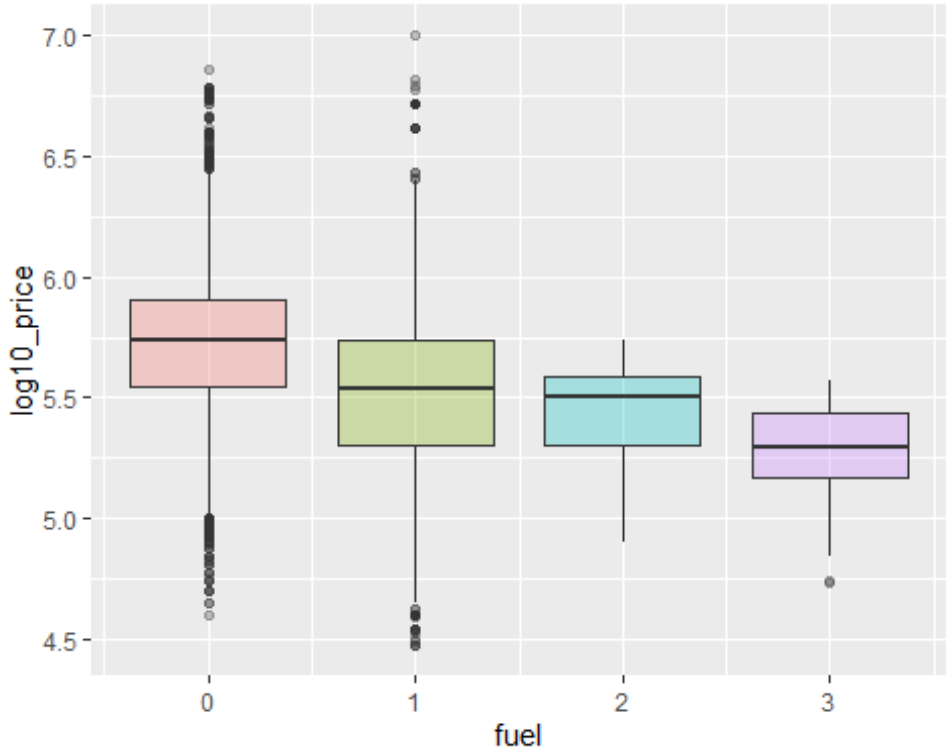
```
ggplot(car, aes(x= log(km_driven), y=log10_price, fill=factor(km_driven)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of fuel

The price of cars decreases as the type of fuel change from diesel to petrol, petrol to CNG and CNG to LPG, respectively.

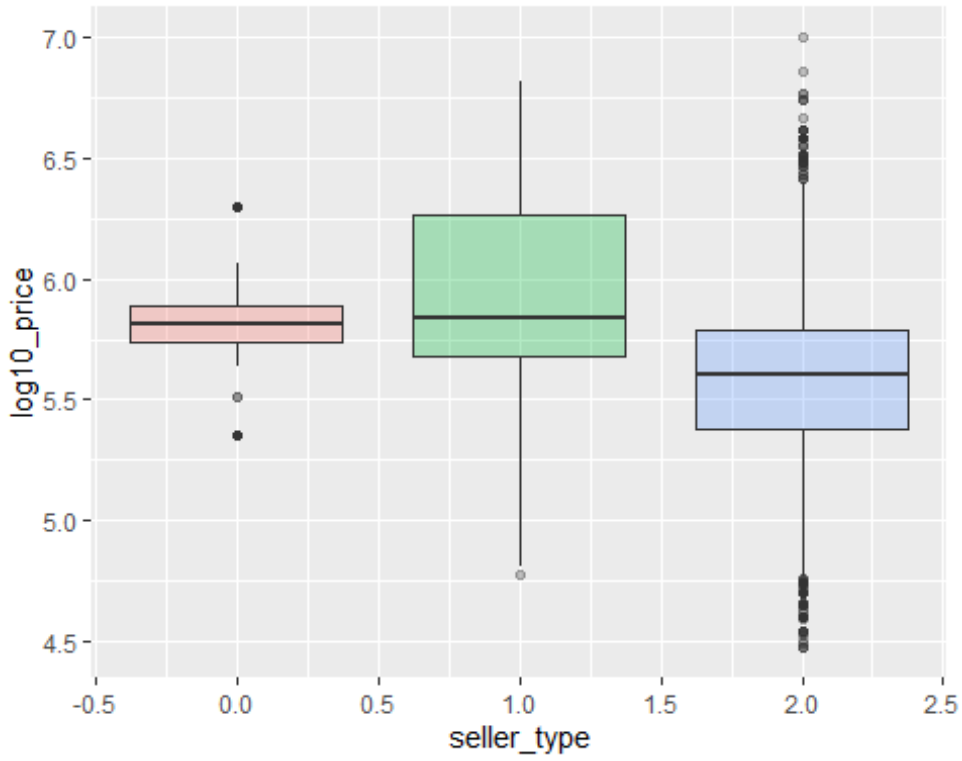
```
ggplot(car, aes(x= fuel, y=log10_price, fill=factor(fuel)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of seller type

TrustMark dealer and dealer type seem to have equal average selling price, but selling price of a car with warranty programm is more cetralized with less variance. price of cars selling by individual are the lowest.

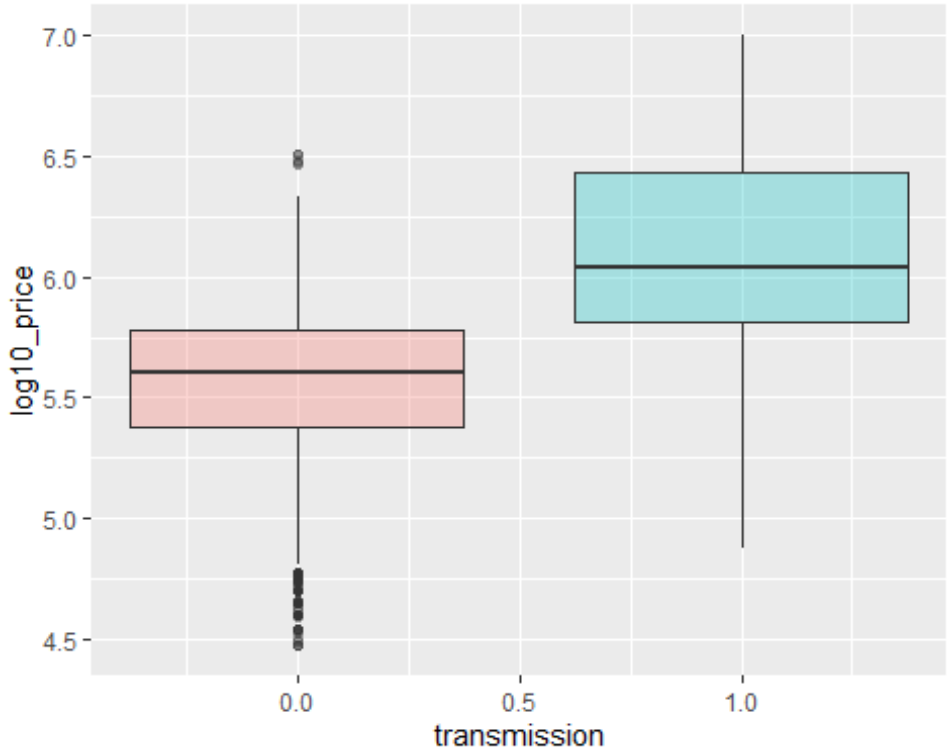
```
ggplot(car, aes(x= seller_type, y=log10_price, fill=factor(seller_type)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of transmission

It seems clear that price of an automatic car is higher than manual.

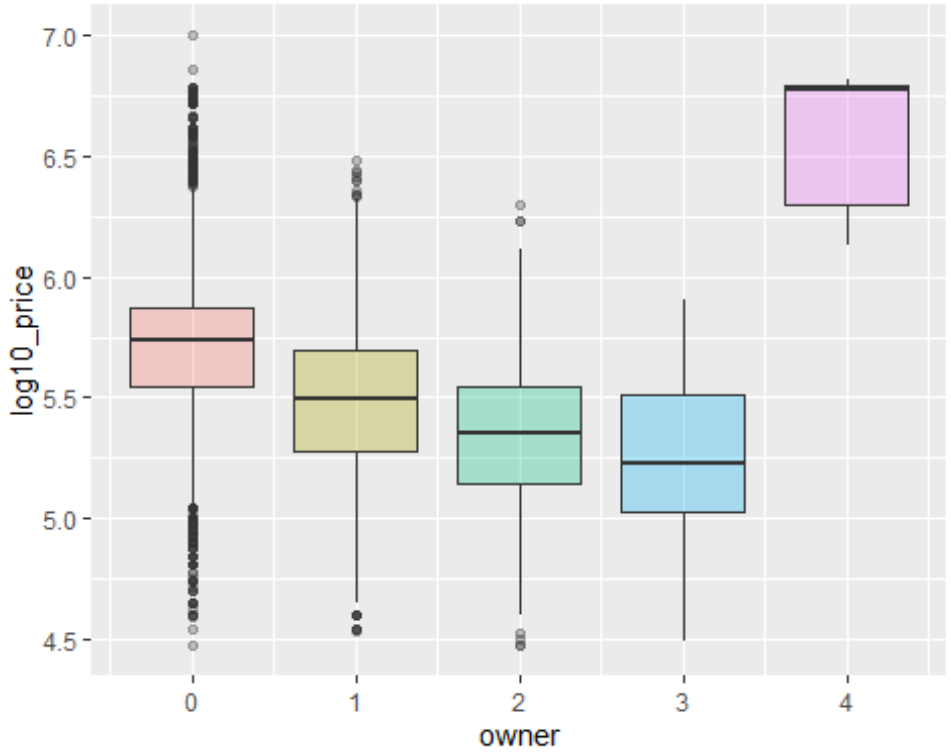
```
ggplot(car, aes(x= transmission, y=log10_price, fill=factor(transmission)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of owner

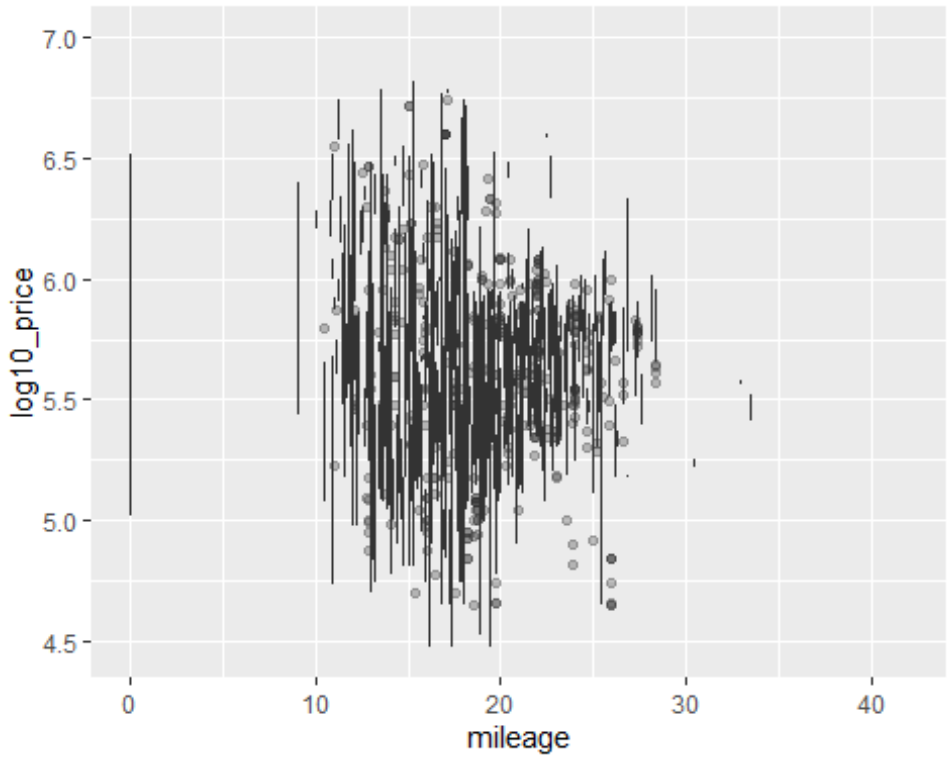
As the owner increases, the price decreases. The price of car increases a lot when the car is a test drive car.

```
ggplot(car, aes(x= owner, y=log10_price, fill=factor(owner)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of mileage

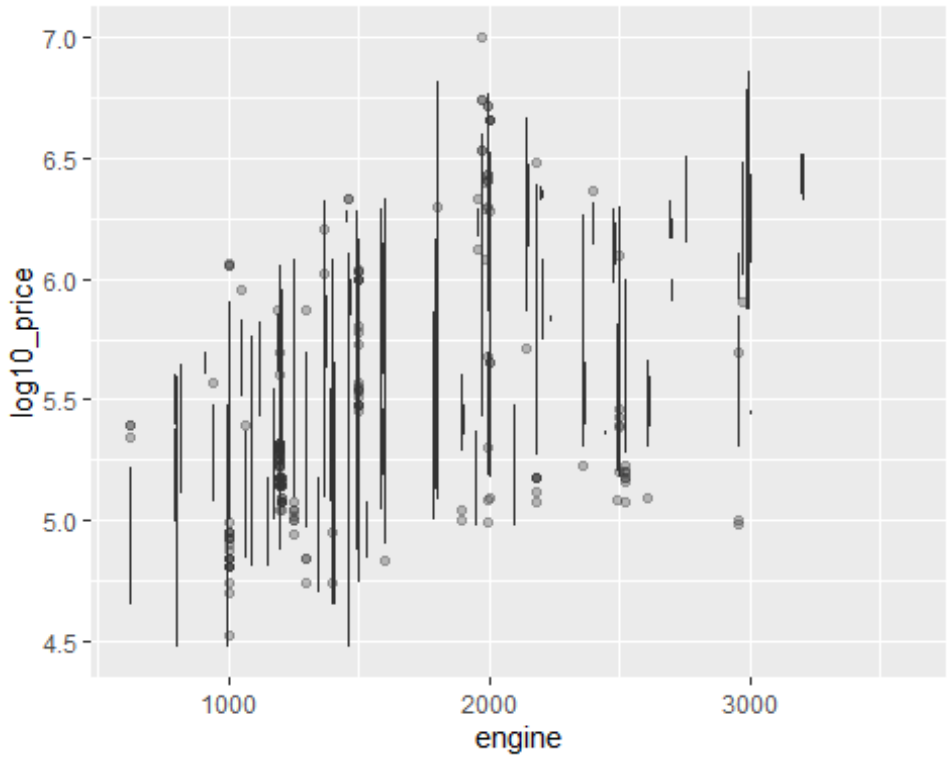
```
ggplot(car, aes(x= mileage, y=log10_price, fill=factor(mileage)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of engine

As the engine capacity of the car increases, the price increases as well.

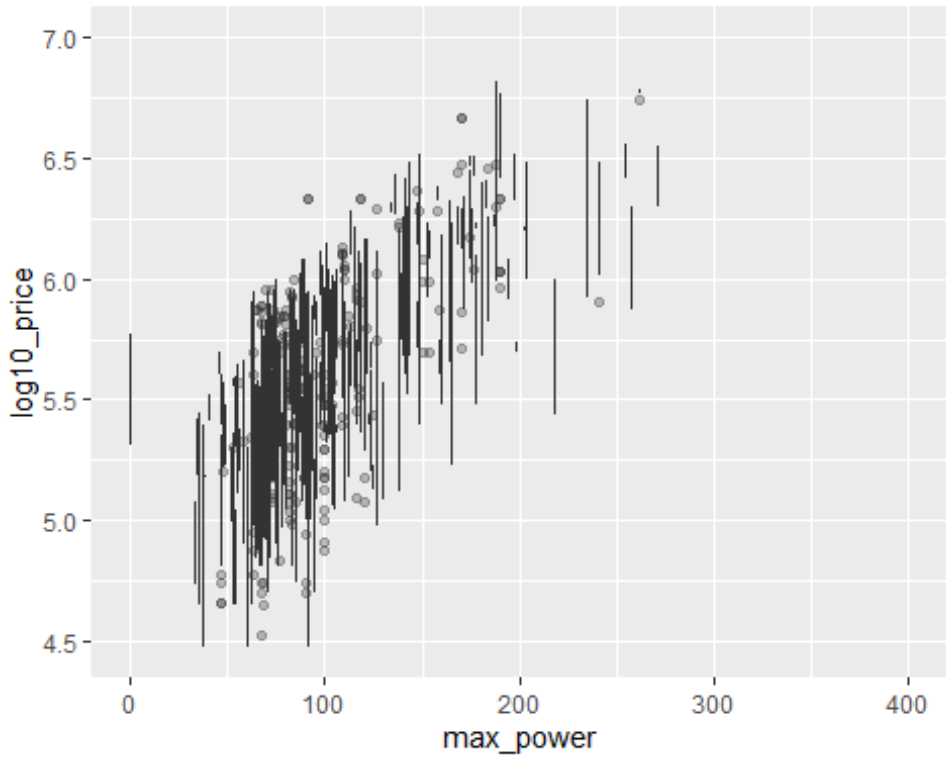
```
ggplot(car, aes(x= engine, y=log10_price, fill=factor(engine)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of max power

As the amount of power that a car's engine generates to move it increases, the price increases as well.

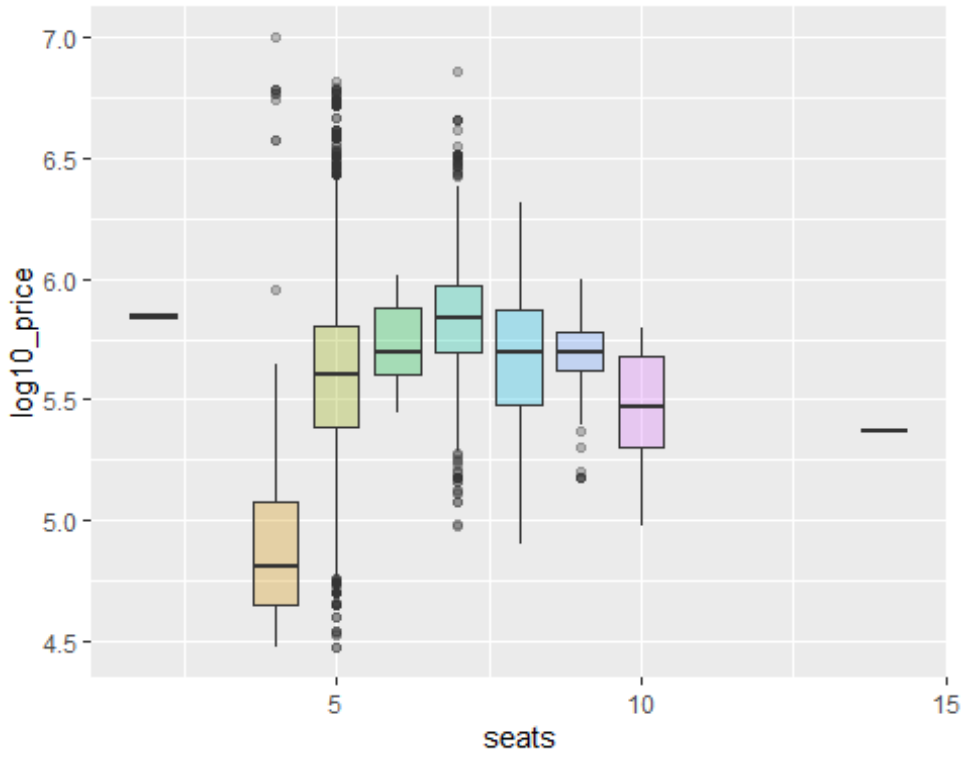
```
ggplot(car, aes(x= max_power, y=log10_price, fill=factor(max_power)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



Box plot of seats

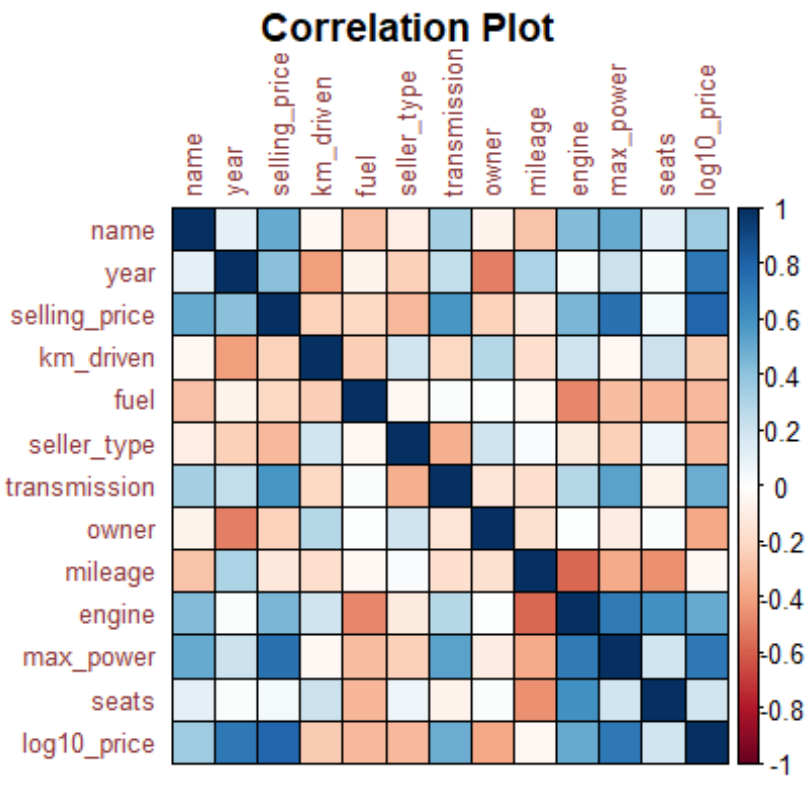
The box plot shows that cars with less than 4 seats have the lowest price. Also 5 seats or more than that does not change the price that much.

```
ggplot(car, aes(x= seats, y=log10_price, fill=factor(seats)))+
geom_boxplot(alpha=0.3)+
theme(legend.position="none")
```



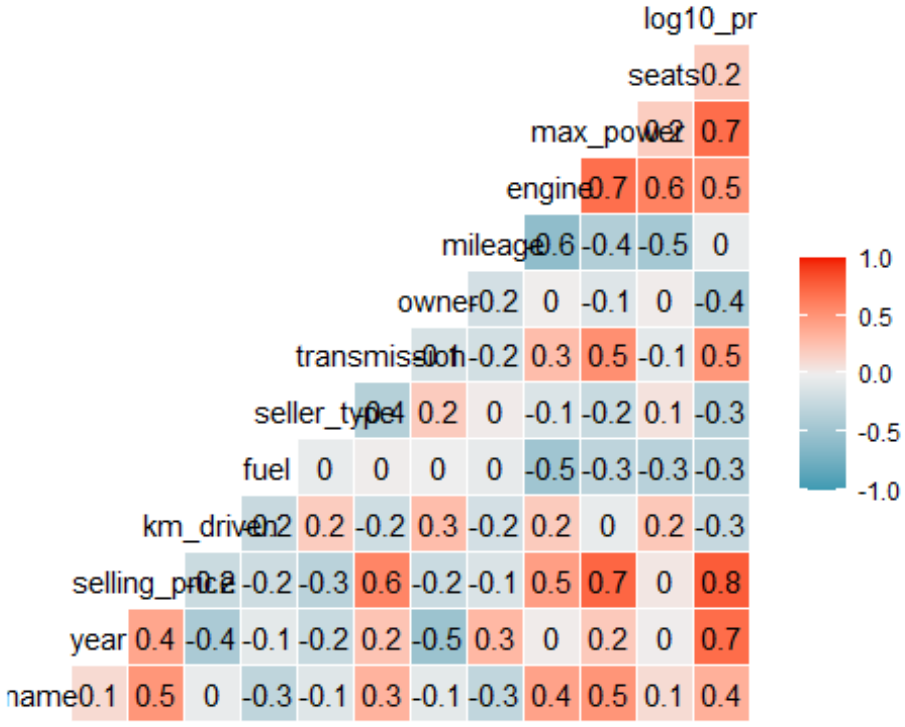
Now we plot correlation plot of features.

```
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.2.3

## corrplot 0.92 loaded

corrplot(cor(car), type="full",
         method ="color", title = "Correlation Plot",
         mar=c(0,0,1,0), tl.cex= 0.8, outline= T, tl.col="indianred4")
```



and also

```
ggcorr(car, label = T)
```




```
round(cor(car),2)
```

```
##                 name  year selling_price km_driven  fuel seller_type
## name            1.00  0.12          0.50     -0.03 -0.29       -0.10
## year            0.12  1.00          0.41     -0.42 -0.06       -0.23
## selling_price   0.50  0.41          1.00     -0.23 -0.21       -0.32
## km_driven      -0.03 -0.42         -0.23      1.00 -0.24        0.19
## fuel           -0.29 -0.06         -0.21     -0.24  1.00       -0.03
## seller_type    -0.10 -0.23         -0.32      0.19 -0.03        1.00
## transmission    0.34  0.24          0.59     -0.20  0.01       -0.36
## owner          -0.06 -0.50         -0.22      0.28  0.00        0.20
## mileage        -0.28  0.31         -0.13     -0.17 -0.04        0.02
## engine          0.43  0.02          0.45      0.20 -0.48       -0.12
## max_power       0.51  0.21          0.74     -0.04 -0.30       -0.24
## seats           0.11  0.01          0.05      0.22 -0.34        0.07
## log10_price     0.35  0.71          0.79     -0.25 -0.32       -0.33
##                transmission owner mileage engine max_power seats
log10_price
## name                   0.34 -0.06   -0.28   0.43      0.51  0.11
0.35
## year                   0.24 -0.50    0.31   0.02      0.21  0.01
0.71
## selling_price          0.59 -0.22   -0.13   0.45      0.74  0.05
0.79
## km_driven             -0.20  0.28   -0.17   0.20     -0.04  0.22       -
0.25
## fuel                   0.01  0.00   -0.04  -0.48     -0.30 -0.34       -
0.32
## seller_type           -0.36  0.20    0.02  -0.12     -0.24  0.07       -
0.33
## transmission           1.00 -0.14   -0.18   0.28      0.54 -0.07
0.50
## owner                 -0.14  1.00   -0.17   0.01     -0.10  0.02       -
0.39
## mileage               -0.18 -0.17    1.00  -0.58     -0.37 -0.45       -
0.03
## engine                 0.28  0.01   -0.58   1.00      0.70  0.61
0.50
## max_power              0.54 -0.10   -0.37   0.70      1.00  0.19
0.71
## seats                 -0.07  0.02   -0.45   0.61      0.19  1.00
0.20
## log10_price            0.50 -0.39   -0.03   0.50      0.71  0.20
1.00
```

We can see that selling price is highly correlated to max_power then transmission and name.

4. Modeling Data

After analyzing the behaviors of the dataset, we try to find the model that fits well.

4.1. Accuracy metrics

R-squared (R2): which is the proportion of variation in the outcome that is explained by the predictor variables. In multiple regression models, R2 corresponds to the squared correlation between the observed outcome values and the predicted values by the model. The Higher the R-squared, the better the model.

Root Mean Squared Error (RMSE): which measures the average error performed by the model in predicting the outcome for an observation. Mathematically, the RMSE is the square root of the mean squared error (MSE), which is the average squared difference between the observed actual outcome values and the values predicted by the model. So, MSE = mean((observeds - predicteds)^2) and RMSE = sqrt(MSE). The lower the RMSE, the better the model.

Residual Standard Error (RSE): also known as the model sigma, is a variant of the RMSE adjusted for the number of predictors in the model. The lower the RSE, the better the model. In practice, the difference between RMSE and RSE is very small, particularly for large multivariate data.

Akaike Information Criterion (AIC) : is a measure used to compare and select between competing statistical models. It provides a trade-off between the goodness of fit of the model and the complexity of the model. The AIC is based on the principle that a good model should fit the data well while minimizing the number of parameters used.

4.2. Splitting of train data and test data

Below we will partition the dataset into Train (70%) and Test (30%) set and we will train different ML algorithms to Training sets and apply to the testing set.

```
set.seed(5)
trainIndex <- createDataPartition(car$selling_price, p = .7,
                                  list = FALSE,
                                  times = 1)
train <- car[ trainIndex,]
test <- car[-trainIndex,]
```

4.3. Linear Regression 1

For start, we build the simplest model using all the available features.

```
m1_lr <- lm(selling_price ~
name+year+km_driven+seller_type+mileage+transmission+max_power+engine+fuel+ow
ner+seats, data = train)
summary(m1_lr)
```

```
##
## Call:
## lm(formula = selling_price ~ name + year + km_driven + seller_type +
##     mileage + transmission + max_power + engine + fuel + owner +
##     seats, data = train)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2464480  -211652    -5224   167360  3952046
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.854e+07  4.256e+06 -13.753  < 2e-16 ***
## name          2.444e+04  1.426e+03  17.134  < 2e-16 ***
## year          2.862e+04  2.127e+03  13.455  < 2e-16 ***
## km_driven    -1.585e+00  1.608e-01  -9.855  < 2e-16 ***
## seller_type  -9.833e+04  1.407e+04  -6.990 3.07e-12 ***
## mileage       2.360e+04  2.422e+03   9.744  < 2e-16 ***
## transmission  4.348e+05  2.307e+04  18.844  < 2e-16 ***
## max_power     1.243e+04  3.001e+02  41.427  < 2e-16 ***
## engine        9.283e+01  2.682e+01   3.461 0.000542 ***
## fuel          1.777e+04  1.532e+04   1.160 0.246009
## owner        -3.004e+03  9.705e+03  -0.310 0.756893
## seats        -1.293e+04  9.401e+03  -1.376 0.169011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 449100 on 5679 degrees of freedom
## Multiple R-squared:  0.696,  Adjusted R-squared:  0.6954
## F-statistic:  1182 on 11 and 5679 DF,  p-value: < 2.2e-16
```

```
m1_lr_pred <- predict(m1_lr, test)
radj <- summary(m1_lr)$adj.r.squared
rse <- sqrt(sum(residuals(m1_lr)^2) / m1_lr$df.residual )
rmse <- RMSE(m1_lr_pred, test$selling_price)
aic <- AIC(m1_lr)
m1_lr_reg <- cbind("Adjusted R sq"=radj, "RSE"=rse, "RMSE"=rmse, "AIC"=aic)
```

Checking the p_values of each explanatory variables, it is seen that features such as fuel, owner, and seats are not significant so we can remove them to improve the accuracy of the model.Moreover, we have an F-statistic of 1182 and a p-value almost equal to 0.

4.4. Linear Regression 2

Below we build the linear model without the features fuel, owner, and seats.

```
m2_lr <- lm(selling_price ~
name+year+km_driven+seller_type+mileage+transmission+max_power+engine, data =
train)
summary(m2_lr)
```

```
##
## Call:
## lm(formula = selling_price ~ name + year + km_driven + seller_type +
##     mileage + transmission + max_power + engine, data = train)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2451313  -212110    -4713   164808  3963632
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.752e+07  3.834e+06 -15.002  < 2e-16 ***
## name          2.430e+04  1.380e+03  17.606  < 2e-16 ***
## year          2.811e+04  1.913e+03  14.696  < 2e-16 ***
## km_driven    -1.648e+00  1.562e-01 -10.554  < 2e-16 ***
## seller_type  -1.015e+05  1.394e+04  -7.281 3.76e-13 ***
## mileage       2.314e+04  2.057e+03  11.250  < 2e-16 ***
## transmission  4.423e+05  2.274e+04  19.450  < 2e-16 ***
## max_power     1.256e+04  2.848e+02  44.114  < 2e-16 ***
## engine        6.006e+01  2.067e+01   2.905  0.00369 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 449200 on 5682 degrees of freedom
## Multiple R-squared:  0.6958, Adjusted R-squared:  0.6954
## F-statistic:  1625 on 8 and 5682 DF,  p-value: < 2.2e-16

m2_lr_pred <- predict(m2_lr, test)
radj <- summary(m2_lr)$adj.r.squared
rse <- sqrt(sum(residuals(m2_lr)^2) / m2_lr$df.residual )
rmse <- RMSE(m2_lr_pred, test$selling_price)
aic <- AIC(m2_lr)
m2_lr_reg <- cbind("Adjusted R sq"=radj, "RSE"=rse, "RMSE"=rmse, "AIC"=aic)
```

As it can be seen, we have 8 features, all of which are significant on selling price because of their p_values. We interpret from the coefficients that a unit change in transmission feature gives a bigger change in selling price than a unit change in other features give, given all other features are fixed and this is because transmission has the biggest coefficient of all features. In this case, given that all other features are fixed, a manual type car would have less price than an automatic one by $442,300.

4.5. Adequacy checking of Linear Regression

In this section, we check some assumptions of linear regression such as:

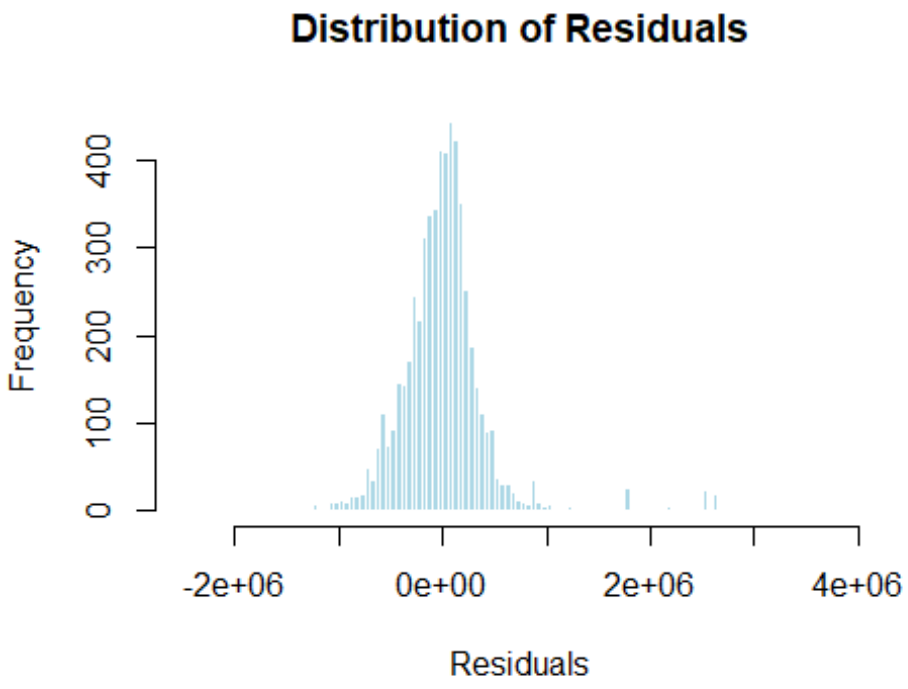
• Linearity of data using Residuals x Fitted plot

• Normality of residuals using Normal QQ plot

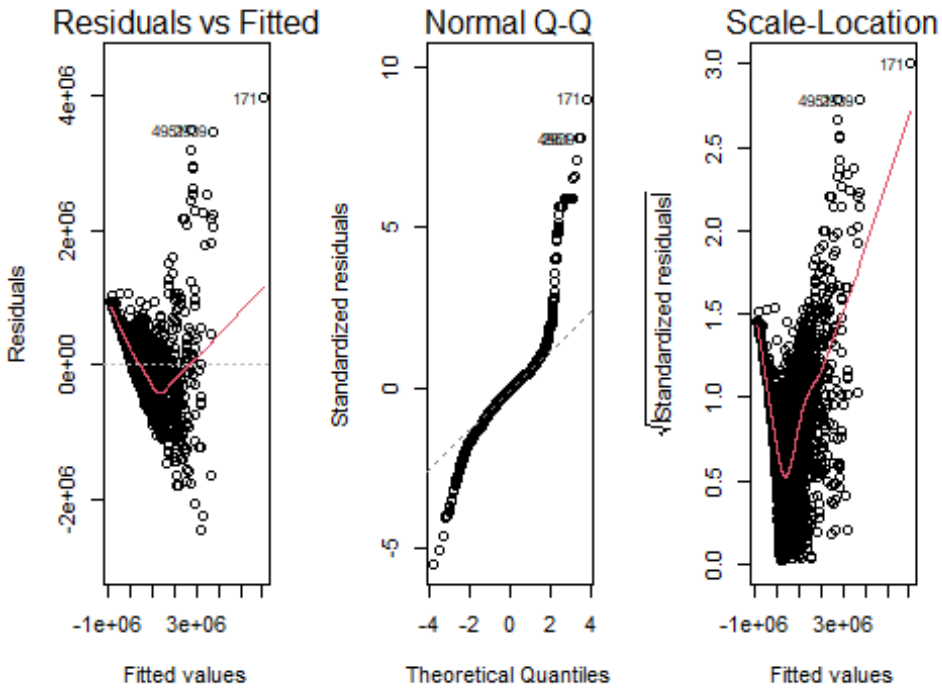
• Homogeneity of residuals variance, residuals with constant variance using scale location plot

```
residual1 <- residuals(m2_lr)

# Plot the histogram of residuals
```

```
hist(residual1, breaks = "FD", col = "lightblue", border = "white", main =
"Distribution of Residuals", xlab = "Residuals")
```




```
options(repr.plot.width=20, repr.plot.height=5)
par(mfrow=c(1,3))
plot(m2_lr, which=c(1,2,3))
```



The distribution of error terms looks normal seeing the histogram plot and skewed to the left. In the Residuals x Fitted plot, we have a non-horizontal line which may indicate a non-linear relationship. In the Normal QQ plot, it is seen that the residuals are not exactly on the straight line, indicating that they are not normally distributed. The distribution with a fat tail will have both the ends of the Q-Q plot to deviate from the straight line and its center follows a straight line, whereas a thin-tailed distribution will form a Q-Q plot with a very less or negligible deviation at the ends thus making it a perfect fit for the Normal Distribution. In the Scale-Location plot, we have a non straight line which indicates heteroscedasticity. Heteroscedasticity can indicate that the model is not appropriate for the data or that there is some other underlying issue.

4.6. Log Linear Regression

In order to correct the model inadequacies, we can try log-linear regression. Log-linear regression is a useful technique for modeling non-linear relationships between variables, particularly when the relationship can be transformed to be linear by taking the logarithm of one or more variables.

```
train$log_selling_price <- log(train$selling_price)
log_lr <- lm(log_selling_price ~
name+year+km_driven+seller_type+mileage+transmission+max_power+engine, data
=train)
summary(log_lr)

##
## Call:
```

```
## lm(formula = log_selling_price ~ name + year + km_driven + seller_type +
##     mileage + transmission + max_power + engine, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63890 -0.18819  0.01647  0.19385  1.91515
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.214e+02  2.652e+00 -83.484  < 2e-16 ***
## name         -5.487e-03  9.549e-04  -5.747 9.58e-09 ***
## year          1.155e-01  1.323e-03  87.324  < 2e-16 ***
## km_driven    -4.763e-07  1.080e-07  -4.409 1.06e-05 ***
## seller_type  -6.980e-02  9.641e-03  -7.240 5.09e-13 ***
## mileage       1.964e-02  1.422e-03  13.810  < 2e-16 ***
## transmission  1.859e-01  1.573e-02  11.817  < 2e-16 ***
## max_power     9.861e-03  1.970e-04  50.066  < 2e-16 ***
## engine        4.196e-04  1.430e-05  29.344  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3107 on 5682 degrees of freedom
## Multiple R-squared:  0.862,  Adjusted R-squared:  0.8618
## F-statistic:  4435 on 8 and 5682 DF,  p-value: < 2.2e-16
```

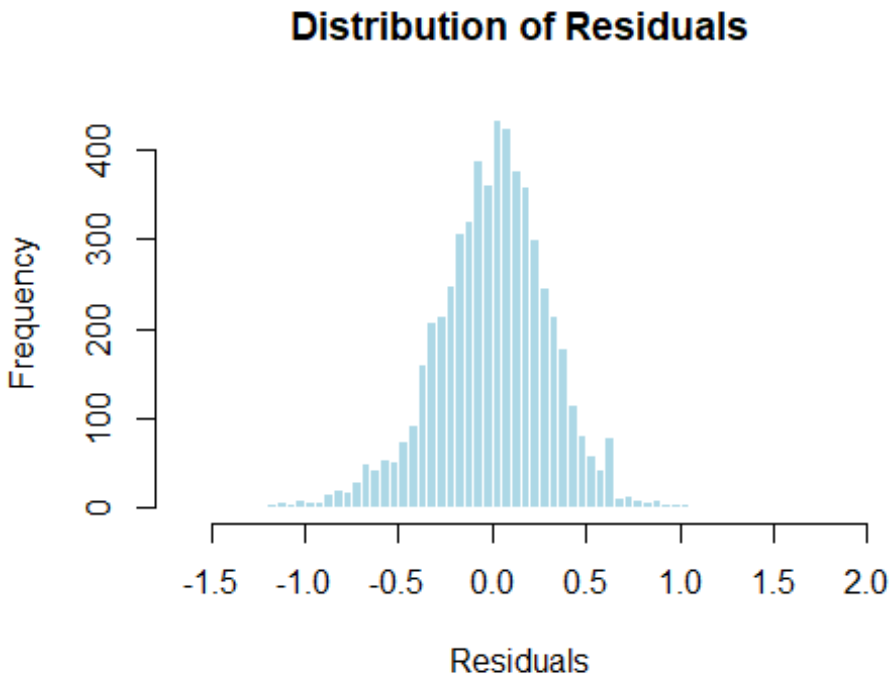
```
log_lr_pred <- predict(log_lr, test)
radj <- summary(log_lr)$adj.r.squared
rse <- sqrt(sum(residuals(log_lr)^2) / log_lr$df.residual )
rmse <- RMSE(log_lr_pred, log(test$selling_price))
aic <- AIC(log_lr)
log_lr_reg <- cbind("Adjusted R sq"=radj, "RSE"=rse, "RMSE"=rmse, "AIC"=aic)
```

In this model,the adjusted R-squared has improved. We have an adjusted R-squared near 1 which indicates that the independent variables in the model are good predictors of the dependent variable. It can interprate that we have a good model fit or overfitting.

4.7. Adequacy checking of Log Linear Regression

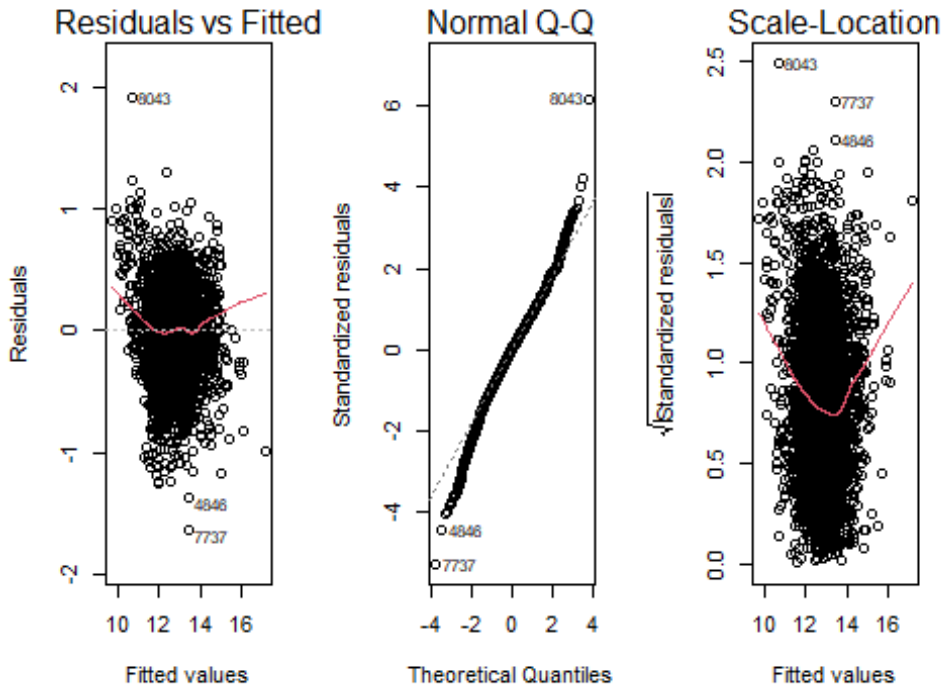
Now we check the adequacy of the log linear model using related plots.

```
residual2 <- residuals(log_lr)

# Plot the histogram of residuals
hist(residual2, breaks = "FD", col = "lightblue", border = "white", main =
"Distribution of Residuals", xlab = "Residuals")
```



Again, the distribution of error terms looks normal seeing the histogram plot.

```
par(mfrow=c(1,3))
plot(log_lr, which=c(1,2,3))
```



The evaluation has improved. In an ideal and unbiased regression, the red lines in the Residuals x Fitted and Scale-Location plots should be parallel to x axis. So, these two results are more or less acceptable. In the Normal QQ plot for ideal regression the points should lie on the diagonal line, our result is not that bad to reject the model.

4.8. Polynomial regression

In this section, we want to use polynomial regression in order to improve our model because we observed that residual plots exhibit parabolic trend, which provides an indication of non-linearity in the data. We tried to generate a new feature matrix consisting of all polynomial combinations of the features with degree 2. We don't want selling price to be included in the process of generating the polynomial combinations, so we take out selling price from train and test and save them as y_train and y_test, respectively.

```
y_train <- train$selling_price
y_test <- test$selling_price
```

From EDA, we know that feature seats has no correlation with selling price. Therefore, we can drop it.

```
X_train <- train %>%
select(-c(selling_price, seats, log_selling_price))
X_test <- test %>%
select(-c(selling_price, seats))
```

Now we use the formula below in order to apply polynomial combinations.

```
formula <- as.formula(
paste(' ~ .^2 + ', paste('poly(', colnames(X_train), ', 2, raw=TRUE)[, 2]',
collapse = ' + '))
)
formula
```

```
## ~.^2 + poly(name, 2, raw = TRUE)[, 2] + poly(year, 2, raw = TRUE)[,
##     2] + poly(km_driven, 2, raw = TRUE)[, 2] + poly(fuel, 2,
##     raw = TRUE)[, 2] + poly(seller_type, 2, raw = TRUE)[, 2] +
##     poly(transmission, 2, raw = TRUE)[, 2] + poly(owner, 2, raw = TRUE)[,
##     2] + poly(mileage, 2, raw = TRUE)[, 2] + poly(engine, 2,
##     raw = TRUE)[, 2] + poly(max_power, 2, raw = TRUE)[, 2] +
##     poly(log10_price, 2, raw = TRUE)[, 2]
```

In codes below, we insert y_train and y_test back to the new datasets.

```
train_poly <- as.data.frame(model.matrix(formula, data = X_train))
test_poly <- as.data.frame(model.matrix(formula, data = X_test))
train_poly$selling_price <- y_train
test_poly$selling_price <- y_test
colnames(train_poly)
```

```
##  [1] "(Intercept)"
##  [2] "name"
##  [3] "year"
##  [4] "km_driven"
##  [5] "fuel"
##  [6] "seller_type"
##  [7] "transmission"
##  [8] "owner"
##  [9] "mileage"
## [10] "engine"
## [11] "max_power"
## [12] "log10_price"
## [13] "poly(name, 2, raw = TRUE)[, 2]"
## [14] "poly(year, 2, raw = TRUE)[, 2]"
## [15] "poly(km_driven, 2, raw = TRUE)[, 2]"
## [16] "poly(fuel, 2, raw = TRUE)[, 2]"
## [17] "poly(seller_type, 2, raw = TRUE)[, 2]"
## [18] "poly(transmission, 2, raw = TRUE)[, 2]"
## [19] "poly(owner, 2, raw = TRUE)[, 2]"
## [20] "poly(mileage, 2, raw = TRUE)[, 2]"
## [21] "poly(engine, 2, raw = TRUE)[, 2]"
## [22] "poly(max_power, 2, raw = TRUE)[, 2]"
## [23] "poly(log10_price, 2, raw = TRUE)[, 2]"
## [24] "name:year"
## [25] "name:km_driven"
## [26] "name:fuel"
## [27] "name:seller_type"
## [28] "name:transmission"
## [29] "name:owner"
```

```
## [30] "name:mileage"
## [31] "name:engine"
## [32] "name:max_power"
## [33] "name:log10_price"
## [34] "year:km_driven"
## [35] "year:fuel"
## [36] "year:seller_type"
## [37] "year:transmission"
## [38] "year:owner"
## [39] "year:mileage"
## [40] "year:engine"
## [41] "year:max_power"
## [42] "year:log10_price"
## [43] "km_driven:fuel"
## [44] "km_driven:seller_type"
## [45] "km_driven:transmission"
## [46] "km_driven:owner"
## [47] "km_driven:mileage"
## [48] "km_driven:engine"
## [49] "km_driven:max_power"
## [50] "km_driven:log10_price"
## [51] "fuel:seller_type"
## [52] "fuel:transmission"
## [53] "fuel:owner"
## [54] "fuel:mileage"
## [55] "fuel:engine"
## [56] "fuel:max_power"
## [57] "fuel:log10_price"
## [58] "seller_type:transmission"
## [59] "seller_type:owner"
## [60] "seller_type:mileage"
## [61] "seller_type:engine"
## [62] "seller_type:max_power"
## [63] "seller_type:log10_price"
## [64] "transmission:owner"
## [65] "transmission:mileage"
## [66] "transmission:engine"
## [67] "transmission:max_power"
## [68] "transmission:log10_price"
## [69] "owner:mileage"
## [70] "owner:engine"
## [71] "owner:max_power"
## [72] "owner:log10_price"
## [73] "mileage:engine"
## [74] "mileage:max_power"
## [75] "mileage:log10_price"
## [76] "engine:max_power"
## [77] "engine:log10_price"
## [78] "max_power:log10_price"
## [79] "selling_price"
```

We can see that our new datasets train_poly and test_poly now have 133 columns. Then, in order to find the best model, we start with all features and use the backward elimination via the function step. The best model is the one with lowest AIC.

```
temp <- lm(formula = selling_price ~ ., data = train_poly)
step(temp)
```

```
## Start:  AIC=133227.2
## selling_price ~ `(Intercept)` + name + year + km_driven + fuel +
##     seller_type + transmission + owner + mileage + engine + max_power +
##     log10_price + `poly(name, 2, raw = TRUE)[, 2]` + `poly(year, 2, raw =
TRUE)[, 2]` +
##     `poly(km_driven, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[,
2]` +
##     `poly(seller_type, 2, raw = TRUE)[, 2]` + `poly(transmission, 2, raw =
TRUE)[, 2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(mileage, 2, raw = TRUE)[,
2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:owner` + `fuel:mileage` + `fuel:engine` + `fuel:max_power` +
##     `fuel:log10_price` + `seller_type:transmission` + `seller_type:owner`
+
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:mileage` +
##     `transmission:engine` + `transmission:max_power` +
`transmission:log10_price` +
##     `owner:mileage` + `owner:engine` + `owner:max_power` +
`owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##
## Step:  AIC=133227.2
## selling_price ~ `(Intercept)` + name + year + km_driven + fuel +
##     seller_type + transmission + owner + mileage + engine + max_power +
```

```
##     log10_price + `poly(name, 2, raw = TRUE)[, 2]` + `poly(year, 2, raw =
TRUE)[, 2]` +
##     `poly(km_driven, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[,
2]` +
##     `poly(seller_type, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw =
TRUE)[, 2]` +
##     `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[,
2]` +
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:owner` +
##     `fuel:mileage` + `fuel:engine` + `fuel:max_power` + `fuel:log10_price`
+
##     `seller_type:transmission` + `seller_type:owner` +
`seller_type:mileage` +
##     `seller_type:engine` + `seller_type:max_power` +
`seller_type:log10_price` +
##     `transmission:owner` + `transmission:mileage` + `transmission:engine`
+
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##
## Step:  AIC=133227.2
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(name, 2, raw = TRUE)[, 2]` + `poly(year, 2, raw = TRUE)[, 2]` +
##     `poly(km_driven, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[,
2]` +
##     `poly(seller_type, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw =
TRUE)[, 2]` +
##     `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[,
2]` +
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
```

```
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:owner` +
##     `fuel:mileage` + `fuel:engine` + `fuel:max_power` + `fuel:log10_price`
+
##     `seller_type:transmission` + `seller_type:owner` +
`seller_type:mileage` +
##     `seller_type:engine` + `seller_type:max_power` +
`seller_type:log10_price` +
##     `transmission:owner` + `transmission:mileage` + `transmission:engine`
+
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                         Df  Sum of Sq        RSS    AIC
## - `transmission:mileage`                 1 2.1306e+05 8.1347e+13 133225
## - `poly(name, 2, raw = TRUE)[, 2]`       1 8.7175e+08 8.1348e+13 133225
## - `seller_type:owner`                    1 9.2742e+08 8.1348e+13 133225
## - `fuel:owner`                           1 1.4524e+09 8.1348e+13 133225
## - `fuel:max_power`                       1 2.4411e+09 8.1349e+13 133225
## - `km_driven:owner`                      1 3.0488e+09 8.1350e+13 133225
## - `mileage:log10_price`                  1 3.2902e+09 8.1350e+13 133225
## - `owner:mileage`                        1 3.3037e+09 8.1350e+13 133225
## - `poly(km_driven, 2, raw = TRUE)[, 2]`  1 3.4191e+09 8.1350e+13 133225
## - `poly(seller_type, 2, raw = TRUE)[, 2]` 1 3.4477e+09 8.1350e+13 133225
## - `poly(mileage, 2, raw = TRUE)[, 2]`    1 3.8605e+09 8.1351e+13 133225
## - `poly(owner, 2, raw = TRUE)[, 2]`      1 6.4179e+09 8.1353e+13 133226
## - max_power                              1 6.5083e+09 8.1353e+13 133226
## - `transmission:engine`                  1 7.1695e+09 8.1354e+13 133226
## - `poly(max_power, 2, raw = TRUE)[, 2]`  1 7.3312e+09 8.1354e+13 133226
## - `year:mileage`                         1 7.8287e+09 8.1354e+13 133226
## - `transmission:max_power`               1 8.2288e+09 8.1355e+13 133226
## - `name:engine`                          1 8.2368e+09 8.1355e+13 133226
## - mileage                                1 8.5778e+09 8.1355e+13 133226
## - `poly(fuel, 2, raw = TRUE)[, 2]`       1 8.8158e+09 8.1355e+13 133226
## - `fuel:engine`                          1 1.1406e+10 8.1358e+13 133226
## - `name:km_driven`                       1 1.3214e+10 8.1360e+13 133226
## - `year:max_power`                       1 1.8473e+10 8.1365e+13 133227
## - `transmission:owner`                   1 2.4986e+10 8.1372e+13 133227
## <none>                                                8.1347e+13 133227
## - `fuel:log10_price`                     1 3.0536e+10 8.1377e+13 133227
## - `name:owner`                           1 3.5604e+10 8.1382e+13 133228
## - `poly(engine, 2, raw = TRUE)[, 2]`     1 3.6670e+10 8.1383e+13 133228
## - `seller_type:max_power`                1 3.9953e+10 8.1387e+13 133228
## - `name:max_power`                       1 5.0335e+10 8.1397e+13 133229
```

```
## - `km_driven:transmission`                  1 5.8042e+10 8.1405e+13 133229
## - `mileage:engine`                          1 6.0388e+10 8.1407e+13 133229
## - `name:seller_type`                        1 6.7603e+10 8.1414e+13 133230
## - `km_driven:engine`                        1 7.8483e+10 8.1425e+13 133231
## - `mileage:max_power`                       1 9.7160e+10 8.1444e+13 133232
## - `seller_type:mileage`                     1 1.2133e+11 8.1468e+13 133234
## - `km_driven:seller_type`                   1 1.3208e+11 8.1479e+13 133234
## - `fuel:transmission`                       1 1.4370e+11 8.1490e+13 133235
## - `year:fuel`                               1 1.5968e+11 8.1506e+13 133236
## - fuel                                      1 1.5976e+11 8.1506e+13 133236
## - `fuel:seller_type`                        1 1.6036e+11 8.1507e+13 133236
## - engine                                    1 1.6125e+11 8.1508e+13 133236
## - km_driven                                 1 1.8183e+11 8.1528e+13 133238
## - `owner:engine`                            1 1.8673e+11 8.1533e+13 133238
## - `fuel:mileage`                            1 1.9288e+11 8.1540e+13 133239
## - `year:engine`                             1 1.9809e+11 8.1545e+13 133239
## - `engine:max_power`                        1 1.9843e+11 8.1545e+13 133239
## - `km_driven:max_power`                     1 1.9980e+11 8.1546e+13 133239
## - `year:km_driven`                          1 2.0982e+11 8.1556e+13 133240
## - `year:transmission`                       1 2.3365e+11 8.1580e+13 133242
## - `owner:log10_price`                       1 2.6587e+11 8.1613e+13 133244
## - owner                                     1 2.6888e+11 8.1616e+13 133244
## - `owner:max_power`                         1 2.6972e+11 8.1616e+13 133244
## - `year:owner`                              1 2.7391e+11 8.1621e+13 133244
## - `km_driven:fuel`                          1 2.7942e+11 8.1626e+13 133245
## - `km_driven:mileage`                       1 2.8493e+11 8.1632e+13 133245
## - transmission                              1 2.9396e+11 8.1641e+13 133246
## - year                                      1 3.2720e+11 8.1674e+13 133248
## - `poly(year, 2, raw = TRUE)[, 2]`          1 3.6996e+11 8.1717e+13 133251
## - `name:mileage`                            1 3.9925e+11 8.1746e+13 133253
## - seller_type                               1 5.6203e+11 8.1909e+13 133264
## - name                                      1 5.7283e+11 8.1919e+13 133265
## - `year:seller_type`                        1 5.8616e+11 8.1933e+13 133266
## - `name:year`                               1 6.5485e+11 8.2001e+13 133271
## - `km_driven:log10_price`                   1 7.5272e+11 8.2099e+13 133278
## - `seller_type:transmission`                1 8.1996e+11 8.2167e+13 133282
## - `transmission:log10_price`                1 8.7888e+11 8.2226e+13 133286
## - `seller_type:log10_price`                 1 1.2126e+12 8.2559e+13 133309
## - `seller_type:engine`                      1 1.2325e+12 8.2579e+13 133311
## - `name:fuel`                               1 1.2807e+12 8.2627e+13 133314
## - `max_power:log10_price`                   1 1.4829e+12 8.2830e+13 133328
## - `engine:log10_price`                      1 1.7501e+12 8.3097e+13 133346
## - log10_price                               1 1.7586e+12 8.3105e+13 133347
## - `year:log10_price`                        1 2.0464e+12 8.3393e+13 133367
## - `name:transmission`                       1 2.2989e+12 8.3646e+13 133384
## - `name:log10_price`                        1 2.5515e+12 8.3898e+13 133401
## - `poly(log10_price, 2, raw = TRUE)[, 2]`   1 1.2503e+13 9.3850e+13 134039
##
## Step:  AIC=133225.2
## selling_price ~ name + year + km_driven + fuel + seller_type +
```

```
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(name, 2, raw = TRUE)[, 2]` + `poly(year, 2, raw = TRUE)[, 2]` +
##     `poly(km_driven, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[,
2]` +
##     `poly(seller_type, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw =
TRUE)[, 2]` +
##     `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[,
2]` +
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:owner` +
##     `fuel:mileage` + `fuel:engine` + `fuel:max_power` + `fuel:log10_price`
+
##     `seller_type:transmission` + `seller_type:owner` +
`seller_type:mileage` +
##     `seller_type:engine` + `seller_type:max_power` +
`seller_type:log10_price` +
##     `transmission:owner` + `transmission:engine` +
`transmission:max_power` +
##     `transmission:log10_price` + `owner:mileage` + `owner:engine` +
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `mileage:log10_price` + `engine:max_power` +
##     `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `poly(name, 2, raw = TRUE)[, 2]`        1 8.7166e+08 8.1348e+13 133223
## - `seller_type:owner`                     1 9.2724e+08 8.1348e+13 133223
## - `fuel:owner`                            1 1.4587e+09 8.1348e+13 133223
## - `fuel:max_power`                        1 2.4431e+09 8.1349e+13 133223
## - `km_driven:owner`                       1 3.0489e+09 8.1350e+13 133223
## - `owner:mileage`                         1 3.3137e+09 8.1350e+13 133223
## - `poly(km_driven, 2, raw = TRUE)[, 2]`   1 3.4213e+09 8.1350e+13 133223
## - `poly(seller_type, 2, raw = TRUE)[, 2]` 1 3.4654e+09 8.1350e+13 133223
## - `mileage:log10_price`                   1 3.5117e+09 8.1350e+13 133223
## - `poly(mileage, 2, raw = TRUE)[, 2]`     1 4.0725e+09 8.1351e+13 133224
## - `poly(owner, 2, raw = TRUE)[, 2]`       1 6.4221e+09 8.1353e+13 133224
## - max_power                               1 6.5201e+09 8.1353e+13 133224
## - `poly(max_power, 2, raw = TRUE)[, 2]`   1 7.3312e+09 8.1354e+13 133224
## - `year:mileage`                          1 7.9154e+09 8.1355e+13 133224
## - `name:engine`                           1 8.2649e+09 8.1355e+13 133224
## - `transmission:max_power`                1 8.2761e+09 8.1355e+13 133224
## - mileage                                 1 8.6642e+09 8.1355e+13 133224
```

```
## - `poly(fuel, 2, raw = TRUE)[, 2]`          1 8.8543e+09 8.1356e+13 133224
## - `transmission:engine`                     1 1.0297e+10 8.1357e+13 133224
## - `fuel:engine`                             1 1.1472e+10 8.1358e+13 133224
## - `name:km_driven`                          1 1.3219e+10 8.1360e+13 133224
## - `year:max_power`                          1 1.8502e+10 8.1365e+13 133225
## - `transmission:owner`                      1 2.5037e+10 8.1372e+13 133225
## <none>                                                   8.1347e+13 133225
## - `fuel:log10_price`                        1 3.1190e+10 8.1378e+13 133225
## - `name:owner`                              1 3.5609e+10 8.1382e+13 133226
## - `poly(engine, 2, raw = TRUE)[, 2]`        1 3.6729e+10 8.1383e+13 133226
## - `seller_type:max_power`                   1 3.9954e+10 8.1387e+13 133226
## - `name:max_power`                          1 5.0828e+10 8.1397e+13 133227
## - `km_driven:transmission`                  1 5.8106e+10 8.1405e+13 133227
## - `mileage:engine`                          1 6.1102e+10 8.1408e+13 133227
## - `name:seller_type`                        1 6.7815e+10 8.1414e+13 133228
## - `km_driven:engine`                        1 7.8513e+10 8.1425e+13 133229
## - `mileage:max_power`                       1 9.8364e+10 8.1445e+13 133230
## - `km_driven:seller_type`                   1 1.3317e+11 8.1480e+13 133233
## - `seller_type:mileage`                     1 1.3593e+11 8.1483e+13 133233
## - fuel                                      1 1.6051e+11 8.1507e+13 133234
## - `year:fuel`                               1 1.6054e+11 8.1507e+13 133234
## - engine                                    1 1.6158e+11 8.1508e+13 133235
## - `fuel:seller_type`                        1 1.7009e+11 8.1517e+13 133235
## - km_driven                                 1 1.8183e+11 8.1528e+13 133236
## - `owner:engine`                            1 1.8700e+11 8.1534e+13 133236
## - `fuel:mileage`                            1 1.9551e+11 8.1542e+13 133237
## - `year:engine`                             1 1.9856e+11 8.1545e+13 133237
## - `km_driven:max_power`                     1 2.0058e+11 8.1547e+13 133237
## - `engine:max_power`                        1 2.0161e+11 8.1548e+13 133237
## - `fuel:transmission`                       1 2.0914e+11 8.1556e+13 133238
## - `year:km_driven`                          1 2.0983e+11 8.1556e+13 133238
## - `year:transmission`                       1 2.4006e+11 8.1587e+13 133240
## - `owner:log10_price`                       1 2.6597e+11 8.1613e+13 133242
## - owner                                     1 2.6914e+11 8.1616e+13 133242
## - `owner:max_power`                         1 2.6973e+11 8.1616e+13 133242
## - `year:owner`                              1 2.7419e+11 8.1621e+13 133242
## - `km_driven:fuel`                          1 2.7942e+11 8.1626e+13 133243
## - `km_driven:mileage`                       1 2.8592e+11 8.1633e+13 133243
## - transmission                              1 3.0114e+11 8.1648e+13 133244
## - year                                      1 3.2767e+11 8.1674e+13 133246
## - `poly(year, 2, raw = TRUE)[, 2]`          1 3.7053e+11 8.1717e+13 133249
## - `name:mileage`                            1 4.0998e+11 8.1757e+13 133252
## - seller_type                               1 5.6212e+11 8.1909e+13 133262
## - name                                      1 5.7457e+11 8.1921e+13 133263
## - `year:seller_type`                        1 5.8632e+11 8.1933e+13 133264
## - `name:year`                               1 6.5695e+11 8.2004e+13 133269
## - `km_driven:log10_price`                   1 7.5291e+11 8.2100e+13 133276
## - `seller_type:transmission`                1 8.1998e+11 8.2167e+13 133280
## - `transmission:log10_price`                1 8.7922e+11 8.2226e+13 133284
## - `seller_type:log10_price`                 1 1.2137e+12 8.2560e+13 133308
```

```
## - `seller_type:engine`                       1 1.2889e+12 8.2636e+13 133313
## - `name:fuel`                                1 1.3001e+12 8.2647e+13 133313
## - `max_power:log10_price`                    1 1.4841e+12 8.2831e+13 133326
## - log10_price                                1 1.7607e+12 8.3107e+13 133345
## - `engine:log10_price`                       1 1.7927e+12 8.3139e+13 133347
## - `year:log10_price`                         1 2.0492e+12 8.3396e+13 133365
## - `name:transmission`                        1 2.3817e+12 8.3728e+13 133387
## - `name:log10_price`                         1 2.5546e+12 8.3901e+13 133399
## - `poly(log10_price, 2, raw = TRUE)[, 2]`    1 1.2510e+13 9.3856e+13 134037
##
## Step:  AIC=133223.3
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(seller_type, 2, raw = TRUE)[,
2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(mileage, 2, raw = TRUE)[,
2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:owner` + `fuel:mileage` + `fuel:engine` + `fuel:max_power` +
##     `fuel:log10_price` + `seller_type:transmission` + `seller_type:owner`
+
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `seller_type:owner`                     1 9.4840e+08 8.1348e+13 133221
## - `fuel:owner`                            1 1.3801e+09 8.1349e+13 133221
## - `fuel:max_power`                        1 2.3523e+09 8.1350e+13 133221
```

```
## - `km_driven:owner`                            1 3.1043e+09 8.1351e+13 133221
## - `poly(seller_type, 2, raw = TRUE)[, 2]`      1 3.2438e+09 8.1351e+13 133222
## - `owner:mileage`                              1 3.3285e+09 8.1351e+13 133222
## - `mileage:log10_price`                        1 3.4211e+09 8.1351e+13 133222
## - `poly(km_driven, 2, raw = TRUE)[, 2]`        1 3.5518e+09 8.1351e+13 133222
## - `poly(mileage, 2, raw = TRUE)[, 2]`          1 4.0787e+09 8.1352e+13 133222
## - `poly(owner, 2, raw = TRUE)[, 2]`            1 6.4688e+09 8.1354e+13 133222
## - max_power                                    1 6.5060e+09 8.1354e+13 133222
## - `poly(max_power, 2, raw = TRUE)[, 2]`        1 7.7238e+09 8.1355e+13 133222
## - `name:engine`                                1 7.7738e+09 8.1355e+13 133222
## - `year:mileage`                               1 8.0347e+09 8.1356e+13 133222
## - `transmission:max_power`                     1 8.1593e+09 8.1356e+13 133222
## - mileage                                      1 8.7882e+09 8.1356e+13 133222
## - `poly(fuel, 2, raw = TRUE)[, 2]`             1 9.3668e+09 8.1357e+13 133222
## - `transmission:engine`                        1 1.0142e+10 8.1358e+13 133222
## - `fuel:engine`                                1 1.1804e+10 8.1359e+13 133222
## - `name:km_driven`                             1 1.4048e+10 8.1362e+13 133222
## - `year:max_power`                             1 1.8484e+10 8.1366e+13 133223
## - `transmission:owner`                         1 2.4910e+10 8.1372e+13 133223
## <none>                                                      8.1348e+13 133223
## - `fuel:log10_price`                           1 3.0589e+10 8.1378e+13 133223
## - `name:owner`                                 1 3.5330e+10 8.1383e+13 133224
## - `poly(engine, 2, raw = TRUE)[, 2]`           1 3.6630e+10 8.1384e+13 133224
## - `seller_type:max_power`                      1 3.9468e+10 8.1387e+13 133224
## - `name:max_power`                             1 5.0483e+10 8.1398e+13 133225
## - `km_driven:transmission`                     1 5.7664e+10 8.1405e+13 133225
## - `mileage:engine`                             1 6.2952e+10 8.1410e+13 133226
## - `name:seller_type`                           1 6.7745e+10 8.1415e+13 133226
## - `km_driven:engine`                           1 7.8250e+10 8.1426e+13 133227
## - `mileage:max_power`                          1 1.0160e+11 8.1449e+13 133228
## - `km_driven:seller_type`                      1 1.3323e+11 8.1481e+13 133231
## - `seller_type:mileage`                        1 1.3664e+11 8.1484e+13 133231
## - fuel                                         1 1.5966e+11 8.1507e+13 133232
## - `year:fuel`                                  1 1.5968e+11 8.1507e+13 133232
## - engine                                       1 1.6267e+11 8.1510e+13 133233
## - `fuel:seller_type`                           1 1.6986e+11 8.1517e+13 133233
## - km_driven                                    1 1.8121e+11 8.1529e+13 133234
## - `owner:engine`                               1 1.8735e+11 8.1535e+13 133234
## - `fuel:mileage`                               1 1.9468e+11 8.1542e+13 133235
## - `year:engine`                                1 1.9976e+11 8.1547e+13 133235
## - `km_driven:max_power`                        1 1.9980e+11 8.1547e+13 133235
## - `engine:max_power`                           1 2.0272e+11 8.1550e+13 133235
## - `fuel:transmission`                          1 2.0903e+11 8.1557e+13 133236
## - `year:km_driven`                             1 2.0920e+11 8.1557e+13 133236
## - `year:transmission`                          1 2.4387e+11 8.1591e+13 133238
## - `owner:log10_price`                          1 2.6583e+11 8.1613e+13 133240
## - owner                                        1 2.6886e+11 8.1616e+13 133240
## - `owner:max_power`                            1 2.7030e+11 8.1618e+13 133240
## - `year:owner`                                 1 2.7390e+11 8.1621e+13 133240
## - `km_driven:fuel`                             1 2.8009e+11 8.1628e+13 133241
```

```
## - `km_driven:mileage`                         1 2.8558e+11 8.1633e+13 133241
## - transmission                                1 3.0556e+11 8.1653e+13 133243
## - year                                        1 3.2680e+11 8.1674e+13 133244
## - `poly(year, 2, raw = TRUE)[, 2]`            1 3.6966e+11 8.1717e+13 133247
## - `name:mileage`                              1 4.1348e+11 8.1761e+13 133250
## - seller_type                                 1 5.6137e+11 8.1909e+13 133260
## - `year:seller_type`                          1 5.8560e+11 8.1933e+13 133262
## - name                                        1 5.9015e+11 8.1938e+13 133262
## - `name:year`                                 1 6.7305e+11 8.2021e+13 133268
## - `km_driven:log10_price`                     1 7.5359e+11 8.2101e+13 133274
## - `seller_type:transmission`                  1 8.1931e+11 8.2167e+13 133278
## - `transmission:log10_price`                  1 8.7958e+11 8.2227e+13 133282
## - `seller_type:log10_price`                   1 1.2137e+12 8.2561e+13 133306
## - `seller_type:engine`                        1 1.2922e+12 8.2640e+13 133311
## - `name:fuel`                                 1 1.3419e+12 8.2689e+13 133314
## - `max_power:log10_price`                     1 1.4836e+12 8.2831e+13 133324
## - log10_price                                 1 1.7610e+12 8.3108e+13 133343
## - `engine:log10_price`                        1 1.7923e+12 8.3140e+13 133345
## - `year:log10_price`                          1 2.0498e+12 8.3397e+13 133363
## - `name:transmission`                         1 2.4275e+12 8.3775e+13 133389
## - `name:log10_price`                          1 2.5628e+12 8.3910e+13 133398
## - `poly(log10_price, 2, raw = TRUE)[, 2]`     1 1.2510e+13 9.3858e+13 134035
##
## Step:  AIC=133221.4
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(seller_type, 2, raw = TRUE)[,
2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(mileage, 2, raw = TRUE)[,
2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:owner` + `fuel:mileage` + `fuel:engine` + `fuel:max_power` +
##     `fuel:log10_price` + `seller_type:transmission` +
`seller_type:mileage` +
##     `seller_type:engine` + `seller_type:max_power` +
```

```
`seller_type:log10_price` +
##     `transmission:owner` + `transmission:engine` +
`transmission:max_power` +
##     `transmission:log10_price` + `owner:mileage` + `owner:engine` +
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `mileage:log10_price` + `engine:max_power` +
##     `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `fuel:owner`                            1 1.2055e+09 8.1350e+13 133219
## - `fuel:max_power`                        1 2.5305e+09 8.1351e+13 133220
## - `km_driven:owner`                       1 2.6855e+09 8.1351e+13 133220
## - `poly(seller_type, 2, raw = TRUE)[, 2]` 1 3.0357e+09 8.1352e+13 133220
## - `owner:mileage`                         1 3.3554e+09 8.1352e+13 133220
## - `mileage:log10_price`                   1 3.5987e+09 8.1352e+13 133220
## - `poly(km_driven, 2, raw = TRUE)[, 2]`   1 3.6695e+09 8.1352e+13 133220
## - `poly(mileage, 2, raw = TRUE)[, 2]`     1 4.1142e+09 8.1353e+13 133220
## - max_power                               1 6.6376e+09 8.1355e+13 133220
## - `poly(owner, 2, raw = TRUE)[, 2]`       1 6.7893e+09 8.1355e+13 133220
## - `poly(max_power, 2, raw = TRUE)[, 2]`   1 7.5924e+09 8.1356e+13 133220
## - `name:engine`                           1 7.7871e+09 8.1356e+13 133220
## - `year:mileage`                          1 7.8209e+09 8.1356e+13 133220
## - mileage                                 1 8.5688e+09 8.1357e+13 133220
## - `transmission:max_power`                1 8.6050e+09 8.1357e+13 133220
## - `poly(fuel, 2, raw = TRUE)[, 2]`        1 9.4786e+09 8.1358e+13 133220
## - `transmission:engine`                   1 9.9291e+09 8.1358e+13 133220
## - `fuel:engine`                           1 1.2530e+10 8.1361e+13 133220
## - `name:km_driven`                        1 1.3960e+10 8.1362e+13 133220
## - `year:max_power`                        1 1.8696e+10 8.1367e+13 133221
## - `transmission:owner`                    1 2.4660e+10 8.1373e+13 133221
## <none>                                                 8.1348e+13 133221
## - `fuel:log10_price`                      1 3.0115e+10 8.1379e+13 133221
## - `name:owner`                            1 3.4492e+10 8.1383e+13 133222
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 3.6377e+10 8.1385e+13 133222
## - `seller_type:max_power`                 1 4.3775e+10 8.1392e+13 133222
## - `name:max_power`                        1 5.0310e+10 8.1399e+13 133223
## - `km_driven:transmission`                1 5.7300e+10 8.1406e+13 133223
## - `mileage:engine`                        1 6.2595e+10 8.1411e+13 133224
## - `name:seller_type`                      1 6.7515e+10 8.1416e+13 133224
## - `km_driven:engine`                      1 7.8033e+10 8.1427e+13 133225
## - `mileage:max_power`                     1 1.0291e+11 8.1451e+13 133227
## - `km_driven:seller_type`                 1 1.3244e+11 8.1481e+13 133229
## - `seller_type:mileage`                   1 1.3576e+11 8.1484e+13 133229
## - `year:fuel`                             1 1.5920e+11 8.1508e+13 133230
## - fuel                                    1 1.5922e+11 8.1508e+13 133230
## - engine                                  1 1.6385e+11 8.1512e+13 133231
## - `fuel:seller_type`                      1 1.6942e+11 8.1518e+13 133231
## - km_driven                               1 1.8096e+11 8.1529e+13 133232
## - `owner:engine`                          1 1.8668e+11 8.1535e+13 133232
## - `fuel:mileage`                          1 1.9524e+11 8.1544e+13 133233
```

```
## - `km_driven:max_power`                      1 1.9976e+11 8.1548e+13 133233
## - `year:engine`                              1 2.0102e+11 8.1549e+13 133233
## - `engine:max_power`                         1 2.0328e+11 8.1552e+13 133234
## - `fuel:transmission`                        1 2.0875e+11 8.1557e+13 133234
## - `year:km_driven`                           1 2.0890e+11 8.1557e+13 133234
## - `year:transmission`                        1 2.4298e+11 8.1591e+13 133236
## - owner                                      1 2.6814e+11 8.1617e+13 133238
## - `owner:max_power`                          1 2.6999e+11 8.1618e+13 133238
## - `year:owner`                               1 2.7355e+11 8.1622e+13 133238
## - `owner:log10_price`                        1 2.7356e+11 8.1622e+13 133238
## - `km_driven:fuel`                           1 2.7937e+11 8.1628e+13 133239
## - `km_driven:mileage`                        1 2.8482e+11 8.1633e+13 133239
## - transmission                               1 3.0476e+11 8.1653e+13 133241
## - year                                       1 3.2719e+11 8.1676e+13 133242
## - `poly(year, 2, raw = TRUE)[, 2]`           1 3.7015e+11 8.1719e+13 133245
## - `name:mileage`                             1 4.1264e+11 8.1761e+13 133248
## - seller_type                                1 5.8797e+11 8.1936e+13 133260
## - name                                       1 5.8954e+11 8.1938e+13 133260
## - `year:seller_type`                         1 6.1225e+11 8.1961e+13 133262
## - `name:year`                                1 6.7236e+11 8.2021e+13 133266
## - `km_driven:log10_price`                    1 7.5265e+11 8.2101e+13 133272
## - `seller_type:transmission`                 1 8.2062e+11 8.2169e+13 133276
## - `transmission:log10_price`                 1 8.8604e+11 8.2235e+13 133281
## - `seller_type:log10_price`                  1 1.2165e+12 8.2565e+13 133304
## - `seller_type:engine`                       1 1.3344e+12 8.2683e+13 133312
## - `name:fuel`                                1 1.3411e+12 8.2690e+13 133312
## - `max_power:log10_price`                    1 1.4835e+12 8.2832e+13 133322
## - log10_price                                1 1.7626e+12 8.3111e+13 133341
## - `engine:log10_price`                       1 1.7948e+12 8.3143e+13 133344
## - `year:log10_price`                         1 2.0512e+12 8.3400e+13 133361
## - `name:transmission`                        1 2.4276e+12 8.3776e+13 133387
## - `name:log10_price`                         1 2.5641e+12 8.3913e+13 133396
## - `poly(log10_price, 2, raw = TRUE)[, 2]`    1 1.2517e+13 9.3865e+13 134034
##
## Step:  AIC=133219.4
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(seller_type, 2, raw = TRUE)[,
2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(mileage, 2, raw = TRUE)[,
2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
```

```
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:mileage` + `fuel:engine` + `fuel:max_power` + `fuel:log10_price`
+
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:max_power` + `seller_type:log10_price` +
`transmission:owner` +
##     `transmission:engine` + `transmission:max_power` +
`transmission:log10_price` +
##     `owner:mileage` + `owner:engine` + `owner:max_power` +
`owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                           Df  Sum of Sq        RSS    AIC
## - `fuel:max_power`                         1 2.2374e+09 8.1352e+13 133218
## - `poly(seller_type, 2, raw = TRUE)[, 2]`  1 3.0068e+09 8.1353e+13 133218
## - `poly(km_driven, 2, raw = TRUE)[, 2]`    1 3.5342e+09 8.1353e+13 133218
## - `km_driven:owner`                        1 3.8184e+09 8.1353e+13 133218
## - `mileage:log10_price`                    1 4.0110e+09 8.1354e+13 133218
## - `poly(mileage, 2, raw = TRUE)[, 2]`      1 4.1739e+09 8.1354e+13 133218
## - `poly(owner, 2, raw = TRUE)[, 2]`        1 6.5621e+09 8.1356e+13 133218
## - max_power                                1 6.6018e+09 8.1356e+13 133218
## - `owner:mileage`                          1 6.7424e+09 8.1356e+13 133218
## - `name:engine`                            1 7.8456e+09 8.1358e+13 133218
## - `poly(max_power, 2, raw = TRUE)[, 2]`    1 7.9796e+09 8.1358e+13 133218
## - `year:mileage`                           1 8.1257e+09 8.1358e+13 133218
## - `transmission:max_power`                 1 8.6513e+09 8.1358e+13 133218
## - mileage                                  1 8.9252e+09 8.1359e+13 133218
## - `poly(fuel, 2, raw = TRUE)[, 2]`         1 9.4538e+09 8.1359e+13 133218
## - `transmission:engine`                    1 9.9691e+09 8.1360e+13 133218
## - `fuel:engine`                            1 1.2388e+10 8.1362e+13 133218
## - `name:km_driven`                         1 1.3954e+10 8.1364e+13 133218
## - `year:max_power`                         1 1.8663e+10 8.1368e+13 133219
## - `transmission:owner`                     1 2.6187e+10 8.1376e+13 133219
## <none>                                                  8.1350e+13 133219
## - `fuel:log10_price`                       1 2.8932e+10 8.1379e+13 133219
## - `name:owner`                             1 3.3296e+10 8.1383e+13 133220
## - `poly(engine, 2, raw = TRUE)[, 2]`       1 3.6200e+10 8.1386e+13 133220
## - `seller_type:max_power`                  1 4.3909e+10 8.1394e+13 133221
## - `name:max_power`                         1 4.9823e+10 8.1399e+13 133221
## - `km_driven:transmission`                 1 5.7180e+10 8.1407e+13 133221
## - `mileage:engine`                         1 6.1891e+10 8.1412e+13 133222
## - `name:seller_type`                       1 6.7841e+10 8.1418e+13 133222
```

```
## - `km_driven:engine`                          1 7.8554e+10 8.1428e+13 133223
## - `mileage:max_power`                         1 1.0425e+11 8.1454e+13 133225
## - `km_driven:seller_type`                     1 1.3229e+11 8.1482e+13 133227
## - `seller_type:mileage`                       1 1.3685e+11 8.1487e+13 133227
## - engine                                      1 1.6273e+11 8.1512e+13 133229
## - `year:fuel`                                 1 1.6531e+11 8.1515e+13 133229
## - fuel                                        1 1.6629e+11 8.1516e+13 133229
## - `fuel:seller_type`                          1 1.7112e+11 8.1521e+13 133229
## - km_driven                                   1 1.8023e+11 8.1530e+13 133230
## - `fuel:mileage`                              1 1.9405e+11 8.1544e+13 133231
## - `km_driven:max_power`                       1 1.9878e+11 8.1548e+13 133231
## - `year:engine`                               1 1.9990e+11 8.1550e+13 133231
## - `engine:max_power`                          1 2.0651e+11 8.1556e+13 133232
## - `year:km_driven`                            1 2.0818e+11 8.1558e+13 133232
## - `fuel:transmission`                         1 2.1051e+11 8.1560e+13 133232
## - `year:transmission`                         1 2.4314e+11 8.1593e+13 133234
## - `owner:engine`                              1 2.5000e+11 8.1600e+13 133235
## - owner                                       1 2.6696e+11 8.1617e+13 133236
## - `year:owner`                                1 2.7247e+11 8.1622e+13 133236
## - `owner:log10_price`                         1 2.7341e+11 8.1623e+13 133237
## - `owner:max_power`                           1 2.7437e+11 8.1624e+13 133237
## - `km_driven:fuel`                            1 2.8070e+11 8.1630e+13 133237
## - `km_driven:mileage`                         1 2.8700e+11 8.1637e+13 133237
## - transmission                                1 3.0500e+11 8.1655e+13 133239
## - year                                        1 3.2742e+11 8.1677e+13 133240
## - `poly(year, 2, raw = TRUE)[, 2]`            1 3.7042e+11 8.1720e+13 133243
## - `name:mileage`                              1 4.1254e+11 8.1762e+13 133246
## - seller_type                                 1 5.8781e+11 8.1937e+13 133258
## - name                                        1 5.9228e+11 8.1942e+13 133259
## - `year:seller_type`                          1 6.1209e+11 8.1962e+13 133260
## - `name:year`                                 1 6.7506e+11 8.2025e+13 133264
## - `km_driven:log10_price`                     1 7.5519e+11 8.2105e+13 133270
## - `seller_type:transmission`                  1 8.2008e+11 8.2170e+13 133275
## - `transmission:log10_price`                  1 8.8885e+11 8.2239e+13 133279
## - `seller_type:log10_price`                   1 1.2173e+12 8.2567e+13 133302
## - `seller_type:engine`                        1 1.3396e+12 8.2689e+13 133310
## - `name:fuel`                                 1 1.3400e+12 8.2690e+13 133310
## - `max_power:log10_price`                     1 1.4904e+12 8.2840e+13 133321
## - log10_price                                 1 1.7626e+12 8.3112e+13 133339
## - `engine:log10_price`                        1 1.8156e+12 8.3165e+13 133343
## - `year:log10_price`                          1 2.0509e+12 8.3401e+13 133359
## - `name:transmission`                         1 2.4279e+12 8.3778e+13 133385
## - `name:log10_price`                          1 2.5630e+12 8.3913e+13 133394
## - `poly(log10_price, 2, raw = TRUE)[, 2]`     1 1.2530e+13 9.3880e+13 134033
##
## Step:  AIC=133217.6
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
```

```
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(seller_type, 2, raw = TRUE)[,
2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(mileage, 2, raw = TRUE)[,
2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:mileage` + `fuel:engine` + `fuel:log10_price` +
`seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                           Df  Sum of Sq        RSS    AIC
## - `poly(seller_type, 2, raw = TRUE)[, 2]`  1 3.3056e+09 8.1355e+13 133216
## - `poly(km_driven, 2, raw = TRUE)[, 2]`    1 3.5400e+09 8.1355e+13 133216
## - `poly(mileage, 2, raw = TRUE)[, 2]`      1 3.6117e+09 8.1356e+13 133216
## - `km_driven:owner`                        1 3.7820e+09 8.1356e+13 133216
## - `mileage:log10_price`                    1 5.1221e+09 8.1357e+13 133216
## - `owner:mileage`                          1 6.5949e+09 8.1359e+13 133216
## - `poly(owner, 2, raw = TRUE)[, 2]`        1 6.6069e+09 8.1359e+13 133216
## - `year:mileage`                           1 7.0104e+09 8.1359e+13 133216
## - max_power                                1 7.5962e+09 8.1360e+13 133216
## - mileage                                  1 7.7770e+09 8.1360e+13 133216
## - `poly(fuel, 2, raw = TRUE)[, 2]`         1 8.7382e+09 8.1361e+13 133216
## - `transmission:max_power`                 1 9.2925e+09 8.1361e+13 133216
## - `transmission:engine`                    1 9.5088e+09 8.1361e+13 133216
## - `name:engine`                            1 9.6738e+09 8.1362e+13 133216
## - `poly(max_power, 2, raw = TRUE)[, 2]`    1 1.2313e+10 8.1364e+13 133216
## - `fuel:engine`                            1 1.2563e+10 8.1364e+13 133216
## - `name:km_driven`                         1 1.4237e+10 8.1366e+13 133217
## - `year:max_power`                         1 2.0593e+10 8.1373e+13 133217
## - `transmission:owner`                     1 2.5781e+10 8.1378e+13 133217
```

```
## <none>                                                  8.1352e+13 133218
## - `fuel:log10_price`                          1 2.9445e+10 8.1381e+13 133218
## - `name:owner`                                1 3.3146e+10 8.1385e+13 133218
## - `poly(engine, 2, raw = TRUE)[, 2]`          1 4.2036e+10 8.1394e+13 133219
## - `seller_type:max_power`                     1 4.3659e+10 8.1396e+13 133219
## - `name:max_power`                            1 4.7592e+10 8.1400e+13 133219
## - `km_driven:transmission`                    1 5.6813e+10 8.1409e+13 133220
## - `mileage:engine`                            1 6.0423e+10 8.1412e+13 133220
## - `name:seller_type`                          1 6.7081e+10 8.1419e+13 133220
## - `km_driven:engine`                          1 7.8050e+10 8.1430e+13 133221
## - `mileage:max_power`                         1 1.0942e+11 8.1461e+13 133223
## - `km_driven:seller_type`                     1 1.3231e+11 8.1484e+13 133225
## - `seller_type:mileage`                       1 1.3461e+11 8.1487e+13 133225
## - `fuel:seller_type`                          1 1.6897e+11 8.1521e+13 133227
## - fuel                                        1 1.7699e+11 8.1529e+13 133228
## - `year:fuel`                                 1 1.7729e+11 8.1529e+13 133228
## - km_driven                                   1 1.8012e+11 8.1532e+13 133228
## - engine                                      1 1.8018e+11 8.1532e+13 133228
## - `fuel:mileage`                              1 1.9213e+11 8.1544e+13 133229
## - `km_driven:max_power`                       1 1.9655e+11 8.1548e+13 133229
## - `year:km_driven`                            1 2.0807e+11 8.1560e+13 133230
## - `fuel:transmission`                         1 2.0856e+11 8.1560e+13 133230
## - `year:engine`                               1 2.2118e+11 8.1573e+13 133231
## - `year:transmission`                         1 2.4132e+11 8.1593e+13 133232
## - `owner:engine`                              1 2.5024e+11 8.1602e+13 133233
## - owner                                       1 2.6636e+11 8.1618e+13 133234
## - `year:owner`                                1 2.7182e+11 8.1624e+13 133235
## - `owner:log10_price`                         1 2.7183e+11 8.1624e+13 133235
## - `owner:max_power`                           1 2.7742e+11 8.1629e+13 133235
## - `engine:max_power`                          1 2.7763e+11 8.1630e+13 133235
## - `km_driven:fuel`                            1 2.7968e+11 8.1632e+13 133235
## - `km_driven:mileage`                         1 2.8492e+11 8.1637e+13 133235
## - transmission                                1 3.0306e+11 8.1655e+13 133237
## - year                                        1 3.2608e+11 8.1678e+13 133238
## - `poly(year, 2, raw = TRUE)[, 2]`            1 3.6889e+11 8.1721e+13 133241
## - `name:mileage`                              1 4.1525e+11 8.1767e+13 133245
## - seller_type                                 1 5.8982e+11 8.1942e+13 133257
## - name                                        1 5.9157e+11 8.1943e+13 133257
## - `year:seller_type`                          1 6.1434e+11 8.1966e+13 133258
## - `name:year`                                 1 6.7526e+11 8.2027e+13 133263
## - `km_driven:log10_price`                     1 7.5892e+11 8.2111e+13 133268
## - `seller_type:transmission`                  1 8.2061e+11 8.2173e+13 133273
## - `transmission:log10_price`                  1 8.9309e+11 8.2245e+13 133278
## - `seller_type:log10_price`                   1 1.2233e+12 8.2575e+13 133301
## - `seller_type:engine`                        1 1.3408e+12 8.2693e+13 133309
## - `name:fuel`                                 1 1.4053e+12 8.2757e+13 133313
## - `max_power:log10_price`                     1 1.6015e+12 8.2953e+13 133327
## - log10_price                                 1 1.7605e+12 8.3112e+13 133337
## - `engine:log10_price`                        1 2.0130e+12 8.3365e+13 133355
## - `year:log10_price`                          1 2.0493e+12 8.3401e+13 133357
```

```
## - `name:transmission`                       1 2.4352e+12 8.3787e+13 133383
## - `name:log10_price`                        1 2.6028e+12 8.3955e+13 133395
## - `poly(log10_price, 2, raw = TRUE)[, 2]`   1 1.2672e+13 9.4024e+13 134039
##
## Step:  AIC=133215.8
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw = TRUE)[, 2]` +
##     `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[,
2]` +
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `poly(mileage, 2, raw = TRUE)[, 2]`     1 3.3374e+09 8.1359e+13 133214
## - `poly(km_driven, 2, raw = TRUE)[, 2]`   1 3.6437e+09 8.1359e+13 133214
## - `km_driven:owner`                       1 4.0020e+09 8.1359e+13 133214
## - `mileage:log10_price`                   1 5.1935e+09 8.1360e+13 133214
## - `owner:mileage`                         1 6.7914e+09 8.1362e+13 133214
## - `poly(owner, 2, raw = TRUE)[, 2]`       1 6.7936e+09 8.1362e+13 133214
## - `year:mileage`                          1 6.9430e+09 8.1362e+13 133214
## - mileage                                 1 7.7104e+09 8.1363e+13 133214
## - max_power                               1 7.7223e+09 8.1363e+13 133214
## - `transmission:max_power`                1 8.5380e+09 8.1364e+13 133214
## - `poly(fuel, 2, raw = TRUE)[, 2]`        1 8.7722e+09 8.1364e+13 133214
## - `name:engine`                           1 1.0153e+10 8.1365e+13 133215
## - `transmission:engine`                   1 1.0318e+10 8.1366e+13 133215
## - `poly(max_power, 2, raw = TRUE)[, 2]`   1 1.2754e+10 8.1368e+13 133215
## - `fuel:engine`                           1 1.2865e+10 8.1368e+13 133215
```

```
## - `name:km_driven`                          1 1.4095e+10 8.1369e+13 133215
## - `year:max_power`                          1 2.0833e+10 8.1376e+13 133215
## - `transmission:owner`                      1 2.6402e+10 8.1382e+13 133216
## <none>                                                   8.1355e+13 133216
## - `fuel:log10_price`                        1 2.8752e+10 8.1384e+13 133216
## - `name:owner`                              1 3.2524e+10 8.1388e+13 133216
## - `seller_type:max_power`                   1 4.0365e+10 8.1396e+13 133217
## - `poly(engine, 2, raw = TRUE)[, 2]`        1 4.1416e+10 8.1397e+13 133217
## - `name:max_power`                          1 4.7315e+10 8.1403e+13 133217
## - `km_driven:transmission`                  1 5.4844e+10 8.1410e+13 133218
## - `mileage:engine`                          1 6.1057e+10 8.1416e+13 133218
## - `name:seller_type`                        1 7.2957e+10 8.1428e+13 133219
## - `km_driven:engine`                        1 7.8571e+10 8.1434e+13 133219
## - `mileage:max_power`                       1 1.1124e+11 8.1466e+13 133222
## - `km_driven:seller_type`                   1 1.3080e+11 8.1486e+13 133223
## - `seller_type:mileage`                     1 1.3527e+11 8.1490e+13 133223
## - fuel                                      1 1.7553e+11 8.1531e+13 133226
## - `year:fuel`                               1 1.7570e+11 8.1531e+13 133226
## - km_driven                                 1 1.7974e+11 8.1535e+13 133226
## - engine                                    1 1.8060e+11 8.1536e+13 133226
## - `fuel:seller_type`                        1 1.8103e+11 8.1536e+13 133226
## - `fuel:mileage`                            1 1.9152e+11 8.1547e+13 133227
## - `km_driven:max_power`                     1 1.9553e+11 8.1551e+13 133227
## - `fuel:transmission`                       1 2.0540e+11 8.1561e+13 133228
## - `year:km_driven`                          1 2.0772e+11 8.1563e+13 133228
## - `year:engine`                             1 2.2170e+11 8.1577e+13 133229
## - `year:transmission`                       1 2.4175e+11 8.1597e+13 133231
## - `owner:engine`                            1 2.4838e+11 8.1604e+13 133231
## - owner                                     1 2.6593e+11 8.1621e+13 133232
## - `year:owner`                              1 2.7139e+11 8.1627e+13 133233
## - `owner:log10_price`                       1 2.7199e+11 8.1627e+13 133233
## - `owner:max_power`                         1 2.7556e+11 8.1631e+13 133233
## - `engine:max_power`                        1 2.7806e+11 8.1633e+13 133233
## - `km_driven:fuel`                          1 2.7870e+11 8.1634e+13 133233
## - `km_driven:mileage`                       1 2.8645e+11 8.1642e+13 133234
## - transmission                              1 3.0341e+11 8.1659e+13 133235
## - year                                      1 3.2670e+11 8.1682e+13 133237
## - `poly(year, 2, raw = TRUE)[, 2]`          1 3.6956e+11 8.1725e+13 133240
## - `name:mileage`                            1 4.1225e+11 8.1767e+13 133243
## - seller_type                               1 5.8715e+11 8.1942e+13 133255
## - name                                      1 5.9063e+11 8.1946e+13 133255
## - `year:seller_type`                        1 6.1206e+11 8.1967e+13 133256
## - `name:year`                               1 6.7427e+11 8.2029e+13 133261
## - `km_driven:log10_price`                   1 7.5965e+11 8.2115e+13 133267
## - `seller_type:transmission`                1 8.1752e+11 8.2173e+13 133271
## - `transmission:log10_price`                1 8.9008e+11 8.2245e+13 133276
## - `seller_type:log10_price`                 1 1.2505e+12 8.2606e+13 133301
## - `name:fuel`                               1 1.4026e+12 8.2758e+13 133311
## - `seller_type:engine`                      1 1.4055e+12 8.2761e+13 133311
## - `max_power:log10_price`                   1 1.6033e+12 8.2959e+13 133325
```

```
## - log10_price                                1 1.7615e+12 8.3117e+13 133336
## - `engine:log10_price`                       1 2.0121e+12 8.3367e+13 133353
## - `year:log10_price`                         1 2.0502e+12 8.3405e+13 133355
## - `name:transmission`                        1 2.4328e+12 8.3788e+13 133382
## - `name:log10_price`                         1 2.6014e+12 8.3957e+13 133393
## - `poly(log10_price, 2, raw = TRUE)[, 2]`    1 1.2669e+13 9.4025e+13 134037
##
## Step:  AIC=133214
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:mileage` + `fuel:engine` + `fuel:log10_price` +
`seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                           Df  Sum of Sq        RSS    AIC
## - `poly(km_driven, 2, raw = TRUE)[, 2]`    1 3.4903e+09 8.1362e+13 133212
## - `km_driven:owner`                        1 4.0426e+09 8.1363e+13 133212
## - `mileage:log10_price`                    1 4.6736e+09 8.1363e+13 133212
## - `year:mileage`                           1 5.0471e+09 8.1364e+13 133212
## - mileage                                  1 5.7186e+09 8.1364e+13 133212
## - `owner:mileage`                          1 6.0476e+09 8.1365e+13 133212
## - `poly(owner, 2, raw = TRUE)[, 2]`        1 6.8275e+09 8.1365e+13 133213
## - `poly(fuel, 2, raw = TRUE)[, 2]`         1 7.9086e+09 8.1366e+13 133213
## - max_power                                1 8.6152e+09 8.1367e+13 133213
## - `transmission:max_power`                 1 9.3987e+09 8.1368e+13 133213
```

```
## - `transmission:engine`                        1 9.5895e+09 8.1368e+13 133213
## - `poly(max_power, 2, raw = TRUE)[, 2]`        1 1.1907e+10 8.1370e+13 133213
## - `fuel:engine`                                1 1.2025e+10 8.1371e+13 133213
## - `name:engine`                                1 1.2427e+10 8.1371e+13 133213
## - `name:km_driven`                             1 1.3712e+10 8.1372e+13 133213
## - `year:max_power`                             1 2.2359e+10 8.1381e+13 133214
## - `transmission:owner`                         1 2.6830e+10 8.1385e+13 133214
## <none>                                                      8.1359e+13 133214
## - `fuel:log10_price`                           1 3.0381e+10 8.1389e+13 133214
## - `name:owner`                                 1 3.2755e+10 8.1391e+13 133214
## - `seller_type:max_power`                      1 4.1435e+10 8.1400e+13 133215
## - `name:max_power`                             1 4.6121e+10 8.1405e+13 133215
## - `poly(engine, 2, raw = TRUE)[, 2]`           1 5.0353e+10 8.1409e+13 133216
## - `km_driven:transmission`                     1 5.5557e+10 8.1414e+13 133216
## - `name:seller_type`                           1 7.2744e+10 8.1431e+13 133217
## - `km_driven:engine`                           1 7.7672e+10 8.1436e+13 133217
## - `mileage:engine`                             1 1.1067e+11 8.1469e+13 133220
## - `mileage:max_power`                          1 1.2451e+11 8.1483e+13 133221
## - `km_driven:seller_type`                      1 1.3144e+11 8.1490e+13 133221
## - `seller_type:mileage`                        1 1.3849e+11 8.1497e+13 133222
## - fuel                                         1 1.7475e+11 8.1533e+13 133224
## - `year:fuel`                                  1 1.7500e+11 8.1534e+13 133224
## - km_driven                                    1 1.7817e+11 8.1537e+13 133225
## - `fuel:seller_type`                           1 1.8327e+11 8.1542e+13 133225
## - `fuel:mileage`                               1 1.9186e+11 8.1550e+13 133225
## - engine                                       1 1.9297e+11 8.1552e+13 133226
## - `km_driven:max_power`                        1 1.9456e+11 8.1553e+13 133226
## - `year:km_driven`                             1 2.0599e+11 8.1565e+13 133226
## - `fuel:transmission`                          1 2.1052e+11 8.1569e+13 133227
## - `year:engine`                                1 2.3594e+11 8.1594e+13 133229
## - `year:transmission`                          1 2.4077e+11 8.1599e+13 133229
## - `owner:engine`                               1 2.4573e+11 8.1604e+13 133229
## - owner                                        1 2.6581e+11 8.1624e+13 133231
## - `owner:log10_price`                          1 2.7089e+11 8.1629e+13 133231
## - `year:owner`                                 1 2.7129e+11 8.1630e+13 133231
## - `owner:max_power`                            1 2.7501e+11 8.1634e+13 133231
## - `km_driven:fuel`                             1 2.7634e+11 8.1635e+13 133231
## - `engine:max_power`                           1 2.7764e+11 8.1636e+13 133231
## - `km_driven:mileage`                          1 2.8316e+11 8.1642e+13 133232
## - transmission                                 1 3.0241e+11 8.1661e+13 133233
## - year                                         1 3.2490e+11 8.1683e+13 133235
## - `poly(year, 2, raw = TRUE)[, 2]`             1 3.6761e+11 8.1726e+13 133238
## - `name:mileage`                               1 4.2382e+11 8.1782e+13 133242
## - name                                         1 5.8751e+11 8.1946e+13 133253
## - seller_type                                  1 5.8767e+11 8.1946e+13 133253
## - `year:seller_type`                           1 6.1250e+11 8.1971e+13 133255
## - `name:year`                                  1 6.7104e+11 8.2030e+13 133259
## - `km_driven:log10_price`                      1 7.5745e+11 8.2116e+13 133265
## - `seller_type:transmission`                   1 8.1845e+11 8.2177e+13 133269
## - `transmission:log10_price`                   1 8.9709e+11 8.2256e+13 133274
```

```
## - `seller_type:log10_price`                1 1.2495e+12 8.2608e+13 133299
## - `name:fuel`                              1 1.3993e+12 8.2758e+13 133309
## - `seller_type:engine`                     1 1.4190e+12 8.2778e+13 133310
## - `max_power:log10_price`                  1 1.6128e+12 8.2971e+13 133324
## - log10_price                              1 1.7612e+12 8.3120e+13 133334
## - `engine:log10_price`                     1 2.0088e+12 8.3367e+13 133351
## - `year:log10_price`                       1 2.0499e+12 8.3408e+13 133354
## - `name:transmission`                      1 2.4320e+12 8.3791e+13 133380
## - `name:log10_price`                       1 2.5999e+12 8.3958e+13 133391
## - `poly(log10_price, 2, raw = TRUE)[, 2]`  1 1.2667e+13 9.4026e+13 134036
##
## Step:  AIC=133212.3
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[, 2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `mileage:log10_price` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `mileage:log10_price`                   1 4.7401e+09 8.1367e+13 133211
## - `year:mileage`                          1 4.7619e+09 8.1367e+13 133211
## - mileage                                 1 5.4089e+09 8.1367e+13 133211
## - `owner:mileage`                         1 5.6499e+09 8.1368e+13 133211
## - `km_driven:owner`                       1 5.8699e+09 8.1368e+13 133211
## - `poly(owner, 2, raw = TRUE)[, 2]`       1 6.7691e+09 8.1369e+13 133211
## - `poly(fuel, 2, raw = TRUE)[, 2]`        1 7.4811e+09 8.1370e+13 133211
## - max_power                               1 8.5746e+09 8.1371e+13 133211
## - `transmission:max_power`                1 9.4932e+09 8.1372e+13 133211
## - `transmission:engine`                   1 9.7646e+09 8.1372e+13 133211
```

```
## - `poly(max_power, 2, raw = TRUE)[, 2]`    1 1.1810e+10 8.1374e+13 133211
## - `fuel:engine`                            1 1.1852e+10 8.1374e+13 133211
## - `name:km_driven`                         1 1.2537e+10 8.1375e+13 133211
## - `name:engine`                            1 1.2543e+10 8.1375e+13 133211
## - `year:max_power`                         1 2.2296e+10 8.1384e+13 133212
## - `transmission:owner`                     1 2.6602e+10 8.1389e+13 133212
## <none>                                                  8.1362e+13 133212
## - `fuel:log10_price`                       1 2.9691e+10 8.1392e+13 133212
## - `name:owner`                             1 3.2470e+10 8.1395e+13 133213
## - `seller_type:max_power`                  1 4.1485e+10 8.1404e+13 133213
## - `name:max_power`                         1 4.5873e+10 8.1408e+13 133214
## - `poly(engine, 2, raw = TRUE)[, 2]`       1 5.0114e+10 8.1412e+13 133214
## - `km_driven:transmission`                 1 6.0401e+10 8.1422e+13 133215
## - `name:seller_type`                       1 7.2606e+10 8.1435e+13 133215
## - `km_driven:engine`                       1 8.8571e+10 8.1451e+13 133216
## - `mileage:engine`                         1 1.1118e+11 8.1473e+13 133218
## - `mileage:max_power`                      1 1.2468e+11 8.1487e+13 133219
## - `km_driven:seller_type`                  1 1.2843e+11 8.1490e+13 133219
## - `seller_type:mileage`                    1 1.3745e+11 8.1499e+13 133220
## - fuel                                     1 1.7412e+11 8.1536e+13 133222
## - `year:fuel`                              1 1.7433e+11 8.1536e+13 133222
## - `fuel:seller_type`                       1 1.8118e+11 8.1543e+13 133223
## - engine                                   1 1.9089e+11 8.1553e+13 133224
## - `km_driven:max_power`                    1 1.9226e+11 8.1554e+13 133224
## - `fuel:mileage`                           1 1.9239e+11 8.1554e+13 133224
## - km_driven                                1 1.9483e+11 8.1557e+13 133224
## - `fuel:transmission`                      1 2.1277e+11 8.1575e+13 133225
## - `year:km_driven`                         1 2.2367e+11 8.1586e+13 133226
## - `year:engine`                            1 2.3375e+11 8.1596e+13 133227
## - `year:transmission`                      1 2.4263e+11 8.1605e+13 133227
## - `owner:engine`                           1 2.4267e+11 8.1605e+13 133227
## - owner                                    1 2.7134e+11 8.1633e+13 133229
## - `owner:log10_price`                      1 2.7226e+11 8.1634e+13 133229
## - `owner:max_power`                        1 2.7307e+11 8.1635e+13 133229
## - `km_driven:fuel`                         1 2.7573e+11 8.1638e+13 133230
## - `engine:max_power`                       1 2.7661e+11 8.1639e+13 133230
## - `year:owner`                             1 2.7679e+11 8.1639e+13 133230
## - `km_driven:mileage`                      1 2.8033e+11 8.1642e+13 133230
## - transmission                             1 3.0467e+11 8.1667e+13 133232
## - year                                     1 3.2434e+11 8.1686e+13 133233
## - `poly(year, 2, raw = TRUE)[, 2]`         1 3.6703e+11 8.1729e+13 133236
## - `name:mileage`                           1 4.2531e+11 8.1787e+13 133240
## - seller_type                              1 5.8591e+11 8.1948e+13 133251
## - name                                     1 5.8594e+11 8.1948e+13 133251
## - `year:seller_type`                       1 6.1083e+11 8.1973e+13 133253
## - `name:year`                              1 6.6950e+11 8.2032e+13 133257
## - `km_driven:log10_price`                  1 7.6093e+11 8.2123e+13 133263
## - `seller_type:transmission`               1 8.1784e+11 8.2180e+13 133267
## - `transmission:log10_price`               1 8.9963e+11 8.2262e+13 133273
## - `seller_type:log10_price`                1 1.2534e+12 8.2615e+13 133297
```

```
## - `name:fuel`                                1 1.4100e+12 8.2772e+13 133308
## - `seller_type:engine`                       1 1.4213e+12 8.2783e+13 133309
## - `max_power:log10_price`                    1 1.6128e+12 8.2975e+13 133322
## - log10_price                                1 1.7622e+12 8.3124e+13 133332
## - `engine:log10_price`                       1 2.0084e+12 8.3370e+13 133349
## - `year:log10_price`                         1 2.0507e+12 8.3413e+13 133352
## - `name:transmission`                        1 2.4299e+12 8.3792e+13 133378
## - `name:log10_price`                         1 2.6024e+12 8.3964e+13 133389
## - `poly(log10_price, 2, raw = TRUE)[, 2]`    1 1.2664e+13 9.4026e+13 134034
##
## Step:  AIC=133210.6
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[, 2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` +
`owner:mileage` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `engine:max_power` +
##     `engine:log10_price` + `max_power:log10_price`
##
##                                             Df  Sum of Sq        RSS    AIC
## - `owner:mileage`                            1 4.5610e+09 8.1371e+13 133209
## - `km_driven:owner`                          1 6.1932e+09 8.1373e+13 133209
## - `poly(owner, 2, raw = TRUE)[, 2]`          1 6.5703e+09 8.1373e+13 133209
## - `poly(fuel, 2, raw = TRUE)[, 2]`           1 7.7659e+09 8.1375e+13 133209
## - max_power                                  1 8.4166e+09 8.1375e+13 133209
## - `transmission:engine`                      1 9.6537e+09 8.1376e+13 133209
## - `fuel:engine`                              1 1.0053e+10 8.1377e+13 133209
## - `transmission:max_power`                   1 1.0164e+10 8.1377e+13 133209
## - `poly(max_power, 2, raw = TRUE)[, 2]`      1 1.1657e+10 8.1378e+13 133209
## - `name:engine`                              1 1.3398e+10 8.1380e+13 133210
## - `name:km_driven`                           1 1.3469e+10 8.1380e+13 133210
```

```
## - `year:max_power`                          1 2.2019e+10 8.1389e+13 133210
## - `transmission:owner`                      1 2.7386e+10 8.1394e+13 133211
## <none>                                                   8.1367e+13 133211
## - `name:owner`                              1 3.2396e+10 8.1399e+13 133211
## - mileage                                   1 3.9471e+10 8.1406e+13 133211
## - `fuel:log10_price`                        1 3.9831e+10 8.1407e+13 133211
## - `year:mileage`                            1 3.9901e+10 8.1407e+13 133211
## - `seller_type:max_power`                   1 4.0114e+10 8.1407e+13 133211
## - `name:max_power`                          1 4.7828e+10 8.1415e+13 133212
## - `poly(engine, 2, raw = TRUE)[, 2]`        1 5.5244e+10 8.1422e+13 133212
## - `km_driven:transmission`                  1 6.0426e+10 8.1427e+13 133213
## - `name:seller_type`                        1 7.3220e+10 8.1440e+13 133214
## - `km_driven:engine`                        1 8.7262e+10 8.1454e+13 133215
## - `km_driven:seller_type`                   1 1.2801e+11 8.1495e+13 133218
## - `mileage:engine`                          1 1.3324e+11 8.1500e+13 133218
## - `seller_type:mileage`                     1 1.5562e+11 8.1522e+13 133220
## - `mileage:max_power`                       1 1.7260e+11 8.1539e+13 133221
## - `fuel:seller_type`                        1 1.9142e+11 8.1558e+13 133222
## - `km_driven:max_power`                     1 1.9362e+11 8.1560e+13 133222
## - km_driven                                 1 1.9386e+11 8.1561e+13 133222
## - engine                                    1 1.9830e+11 8.1565e+13 133222
## - fuel                                      1 1.9960e+11 8.1566e+13 133223
## - `year:fuel`                               1 2.0068e+11 8.1567e+13 133223
## - `fuel:mileage`                            1 2.0285e+11 8.1570e+13 133223
## - `fuel:transmission`                       1 2.1299e+11 8.1580e+13 133224
## - `year:km_driven`                          1 2.2265e+11 8.1589e+13 133224
## - `owner:engine`                            1 2.3875e+11 8.1606e+13 133225
## - `year:transmission`                       1 2.4021e+11 8.1607e+13 133225
## - `year:engine`                             1 2.4721e+11 8.1614e+13 133226
## - `owner:log10_price`                       1 2.7129e+11 8.1638e+13 133228
## - owner                                     1 2.7268e+11 8.1639e+13 133228
## - `owner:max_power`                         1 2.7389e+11 8.1641e+13 133228
## - `year:owner`                              1 2.7818e+11 8.1645e+13 133228
## - `km_driven:mileage`                       1 2.7881e+11 8.1646e+13 133228
## - `km_driven:fuel`                          1 2.7948e+11 8.1646e+13 133228
## - `engine:max_power`                        1 3.0054e+11 8.1667e+13 133230
## - transmission                              1 3.0204e+11 8.1669e+13 133230
## - year                                      1 3.4739e+11 8.1714e+13 133233
## - `poly(year, 2, raw = TRUE)[, 2]`          1 3.9328e+11 8.1760e+13 133236
## - `name:mileage`                            1 4.2583e+11 8.1793e+13 133238
## - seller_type                               1 5.8289e+11 8.1950e+13 133249
## - name                                      1 5.8661e+11 8.1953e+13 133250
## - `year:seller_type`                        1 6.0766e+11 8.1974e+13 133251
## - `name:year`                               1 6.7027e+11 8.2037e+13 133255
## - `km_driven:log10_price`                   1 7.6023e+11 8.2127e+13 133262
## - `seller_type:transmission`                1 8.1350e+11 8.2180e+13 133265
## - `transmission:log10_price`                1 9.0757e+11 8.2274e+13 133272
## - `seller_type:log10_price`                 1 1.2508e+12 8.2618e+13 133295
## - `name:fuel`                               1 1.4110e+12 8.2778e+13 133306
## - `seller_type:engine`                      1 1.4454e+12 8.2812e+13 133309
```

```
## - `max_power:log10_price`                    1 1.6517e+12 8.3018e+13 133323
## - log10_price                                1 1.7765e+12 8.3143e+13 133332
## - `year:log10_price`                         1 2.0678e+12 8.3435e+13 133351
## - `name:transmission`                        1 2.4282e+12 8.3795e+13 133376
## - `name:log10_price`                         1 2.6136e+12 8.3980e+13 133389
## - `engine:log10_price`                       1 2.7561e+12 8.4123e+13 133398
## - `poly(log10_price, 2, raw = TRUE)[, 2]`    1 1.2794e+13 9.4160e+13 134040
##
## Step:  AIC=133209
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[, 2]` +
##     `poly(owner, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` + `owner:engine`
+
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power` + `engine:log10_price` +
##     `max_power:log10_price`
##
##                                           Df  Sum of Sq        RSS    AIC
## - `poly(owner, 2, raw = TRUE)[, 2]`        1 6.2871e+09 8.1378e+13 133207
## - `km_driven:owner`                        1 6.8658e+09 8.1378e+13 133207
## - `poly(fuel, 2, raw = TRUE)[, 2]`         1 7.6551e+09 8.1379e+13 133207
## - max_power                                1 8.3210e+09 8.1380e+13 133208
## - `transmission:engine`                    1 9.6101e+09 8.1381e+13 133208
## - `fuel:engine`                            1 9.8771e+09 8.1381e+13 133208
## - `transmission:max_power`                 1 1.0208e+10 8.1382e+13 133208
## - `poly(max_power, 2, raw = TRUE)[, 2]`    1 1.1512e+10 8.1383e+13 133208
## - `name:engine`                            1 1.3131e+10 8.1384e+13 133208
## - `name:km_driven`                         1 1.3452e+10 8.1385e+13 133208
## - `year:max_power`                         1 2.1885e+10 8.1393e+13 133208
## <none>                                                  8.1371e+13 133209
## - `name:owner`                             1 3.2748e+10 8.1404e+13 133209
```

```
## - `transmission:owner`                      1 3.2895e+10 8.1404e+13 133209
## - mileage                                   1 3.4913e+10 8.1406e+13 133209
## - `year:mileage`                            1 3.5345e+10 8.1407e+13 133209
## - `fuel:log10_price`                        1 3.9070e+10 8.1410e+13 133210
## - `seller_type:max_power`                   1 3.9903e+10 8.1411e+13 133210
## - `name:max_power`                          1 4.8261e+10 8.1420e+13 133210
## - `poly(engine, 2, raw = TRUE)[, 2]`        1 5.7305e+10 8.1429e+13 133211
## - `km_driven:transmission`                  1 6.0408e+10 8.1432e+13 133211
## - `name:seller_type`                        1 7.2739e+10 8.1444e+13 133212
## - `km_driven:engine`                        1 8.8149e+10 8.1459e+13 133213
## - `km_driven:seller_type`                   1 1.2750e+11 8.1499e+13 133216
## - `mileage:engine`                          1 1.3870e+11 8.1510e+13 133217
## - `seller_type:mileage`                     1 1.5165e+11 8.1523e+13 133218
## - `mileage:max_power`                       1 1.7391e+11 8.1545e+13 133219
## - `fuel:seller_type`                        1 1.9000e+11 8.1561e+13 133220
## - `km_driven:max_power`                     1 1.9424e+11 8.1566e+13 133221
## - km_driven                                 1 1.9643e+11 8.1568e+13 133221
## - fuel                                      1 1.9912e+11 8.1570e+13 133221
## - `year:fuel`                               1 2.0011e+11 8.1571e+13 133221
## - `fuel:mileage`                            1 2.0103e+11 8.1572e+13 133221
## - engine                                    1 2.0474e+11 8.1576e+13 133221
## - `fuel:transmission`                       1 2.1418e+11 8.1586e+13 133222
## - `year:km_driven`                          1 2.2548e+11 8.1597e+13 133223
## - `year:transmission`                       1 2.4416e+11 8.1616e+13 133224
## - `year:engine`                             1 2.5442e+11 8.1626e+13 133225
## - `owner:log10_price`                       1 2.6687e+11 8.1638e+13 133226
## - `owner:engine`                            1 2.6936e+11 8.1641e+13 133226
## - `km_driven:fuel`                          1 2.7864e+11 8.1650e+13 133226
## - `owner:max_power`                         1 2.7890e+11 8.1650e+13 133226
## - `km_driven:mileage`                       1 2.8453e+11 8.1656e+13 133227
## - `engine:max_power`                        1 2.9932e+11 8.1671e+13 133228
## - owner                                     1 3.0492e+11 8.1676e+13 133228
## - transmission                              1 3.0679e+11 8.1678e+13 133228
## - `year:owner`                              1 3.1110e+11 8.1682e+13 133229
## - year                                      1 3.4320e+11 8.1715e+13 133231
## - `poly(year, 2, raw = TRUE)[, 2]`          1 3.8900e+11 8.1760e+13 133234
## - `name:mileage`                            1 4.2707e+11 8.1798e+13 133237
## - seller_type                               1 5.8344e+11 8.1955e+13 133248
## - name                                      1 5.8622e+11 8.1958e+13 133248
## - `year:seller_type`                        1 6.0826e+11 8.1980e+13 133249
## - `name:year`                               1 6.6991e+11 8.2041e+13 133254
## - `km_driven:log10_price`                   1 7.6317e+11 8.2135e+13 133260
## - `seller_type:transmission`                1 8.1172e+11 8.2183e+13 133263
## - `transmission:log10_price`                1 9.0962e+11 8.2281e+13 133270
## - `seller_type:log10_price`                 1 1.2496e+12 8.2621e+13 133294
## - `name:fuel`                               1 1.4153e+12 8.2787e+13 133305
## - `seller_type:engine`                      1 1.4413e+12 8.2813e+13 133307
## - `max_power:log10_price`                   1 1.6547e+12 8.3026e+13 133322
## - log10_price                               1 1.7786e+12 8.3150e+13 133330
## - `year:log10_price`                        1 2.0698e+12 8.3441e+13 133350
```

```
## - `name:transmission`                       1 2.4280e+12 8.3799e+13 133374
## - `name:log10_price`                        1 2.6147e+12 8.3986e+13 133387
## - `engine:log10_price`                      1 2.7533e+12 8.4125e+13 133396
## - `poly(log10_price, 2, raw = TRUE)[, 2]`   1 1.2789e+13 9.4161e+13 134038
##
## Step:  AIC=133207.4
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(fuel, 2, raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` + `km_driven:owner`
+
##     `km_driven:mileage` + `km_driven:engine` + `km_driven:max_power` +
##     `km_driven:log10_price` + `fuel:seller_type` + `fuel:transmission` +
##     `fuel:mileage` + `fuel:engine` + `fuel:log10_price` +
`seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` + `owner:engine`
+
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power` + `engine:log10_price` +
##     `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `poly(fuel, 2, raw = TRUE)[, 2]`        1 7.7874e+09 8.1385e+13 133206
## - `km_driven:owner`                       1 7.9974e+09 8.1386e+13 133206
## - max_power                               1 8.8143e+09 8.1386e+13 133206
## - `transmission:engine`                   1 9.5854e+09 8.1387e+13 133206
## - `fuel:engine`                           1 1.0170e+10 8.1388e+13 133206
## - `transmission:max_power`                1 1.0426e+10 8.1388e+13 133206
## - `poly(max_power, 2, raw = TRUE)[, 2]`   1 1.1313e+10 8.1389e+13 133206
## - `name:engine`                           1 1.2577e+10 8.1390e+13 133206
## - `name:km_driven`                        1 1.3208e+10 8.1391e+13 133206
## - `year:max_power`                        1 2.2663e+10 8.1400e+13 133207
## <none>                                                 8.1378e+13 133207
## - `name:owner`                            1 3.0597e+10 8.1408e+13 133208
## - `transmission:owner`                    1 3.1297e+10 8.1409e+13 133208
## - mileage                                 1 3.5311e+10 8.1413e+13 133208
```

```
## - `year:mileage`                          1 3.5750e+10 8.1413e+13 133208
## - `fuel:log10_price`                      1 3.8419e+10 8.1416e+13 133208
## - `seller_type:max_power`                 1 3.9303e+10 8.1417e+13 133208
## - `name:max_power`                        1 4.8603e+10 8.1426e+13 133209
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 5.8907e+10 8.1437e+13 133210
## - `km_driven:transmission`                1 6.0719e+10 8.1438e+13 133210
## - `name:seller_type`                      1 7.3853e+10 8.1451e+13 133211
## - `km_driven:engine`                      1 8.8947e+10 8.1467e+13 133212
## - `km_driven:seller_type`                 1 1.2855e+11 8.1506e+13 133214
## - `mileage:engine`                        1 1.3921e+11 8.1517e+13 133215
## - `seller_type:mileage`                   1 1.5218e+11 8.1530e+13 133216
## - `mileage:max_power`                     1 1.7515e+11 8.1553e+13 133218
## - `fuel:seller_type`                      1 1.9113e+11 8.1569e+13 133219
## - `km_driven:max_power`                   1 1.9409e+11 8.1572e+13 133219
## - km_driven                               1 1.9637e+11 8.1574e+13 133219
## - fuel                                    1 1.9773e+11 8.1575e+13 133219
## - `year:fuel`                             1 1.9871e+11 8.1576e+13 133219
## - `fuel:mileage`                          1 2.0158e+11 8.1579e+13 133219
## - engine                                  1 2.0331e+11 8.1581e+13 133220
## - `fuel:transmission`                     1 2.1384e+11 8.1591e+13 133220
## - `year:km_driven`                        1 2.2549e+11 8.1603e+13 133221
## - `year:transmission`                     1 2.4213e+11 8.1620e+13 133222
## - `year:engine`                           1 2.5278e+11 8.1630e+13 133223
## - `owner:engine`                          1 2.6345e+11 8.1641e+13 133224
## - `owner:log10_price`                     1 2.6937e+11 8.1647e+13 133224
## - `owner:max_power`                       1 2.7287e+11 8.1651e+13 133224
## - `km_driven:fuel`                        1 2.7787e+11 8.1656e+13 133225
## - `km_driven:mileage`                     1 2.8548e+11 8.1663e+13 133225
## - `engine:max_power`                      1 2.9911e+11 8.1677e+13 133226
## - owner                                   1 3.0045e+11 8.1678e+13 133226
## - transmission                            1 3.0453e+11 8.1682e+13 133227
## - `year:owner`                            1 3.0630e+11 8.1684e+13 133227
## - year                                    1 3.4504e+11 8.1723e+13 133229
## - `poly(year, 2, raw = TRUE)[, 2]`        1 3.9075e+11 8.1768e+13 133233
## - `name:mileage`                          1 4.2720e+11 8.1805e+13 133235
## - seller_type                             1 5.8213e+11 8.1960e+13 133246
## - name                                    1 5.8676e+11 8.1964e+13 133246
## - `year:seller_type`                      1 6.0694e+11 8.1985e+13 133248
## - `name:year`                             1 6.7040e+11 8.2048e+13 133252
## - `km_driven:log10_price`                 1 7.6582e+11 8.2143e+13 133259
## - `seller_type:transmission`              1 8.1248e+11 8.2190e+13 133262
## - `transmission:log10_price`              1 9.1226e+11 8.2290e+13 133269
## - `seller_type:log10_price`               1 1.2490e+12 8.2627e+13 133292
## - `name:fuel`                             1 1.4155e+12 8.2793e+13 133304
## - `seller_type:engine`                    1 1.4373e+12 8.2815e+13 133305
## - `max_power:log10_price`                 1 1.6537e+12 8.3031e+13 133320
## - log10_price                             1 1.7737e+12 8.3151e+13 133328
## - `year:log10_price`                      1 2.0646e+12 8.3442e+13 133348
## - `name:transmission`                     1 2.4230e+12 8.3801e+13 133372
## - `name:log10_price`                      1 2.6113e+12 8.3989e+13 133385
```

```
## - `engine:log10_price`                      1 2.7470e+12 8.4125e+13 133394
## - `poly(log10_price, 2, raw = TRUE)[, 2]`  1 1.2783e+13 9.4161e+13 134036
##
## Step:  AIC=133205.9
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:owner` + `km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` + `owner:engine`
+
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power` + `engine:log10_price` +
##     `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `km_driven:owner`                       1 7.6204e+09 8.1393e+13 133204
## - max_power                               1 8.5651e+09 8.1394e+13 133205
## - `transmission:max_power`                1 1.0305e+10 8.1396e+13 133205
## - `name:engine`                           1 1.0851e+10 8.1396e+13 133205
## - `poly(max_power, 2, raw = TRUE)[, 2]`   1 1.0995e+10 8.1396e+13 133205
## - `transmission:engine`                   1 1.1111e+10 8.1397e+13 133205
## - `name:km_driven`                        1 1.2348e+10 8.1398e+13 133205
## - `fuel:engine`                           1 1.8523e+10 8.1404e+13 133205
## - `year:max_power`                        1 2.2204e+10 8.1408e+13 133205
## <none>                                                 8.1385e+13 133206
## - `name:owner`                            1 3.0974e+10 8.1416e+13 133206
## - `transmission:owner`                    1 3.1165e+10 8.1417e+13 133206
## - mileage                                 1 3.3030e+10 8.1418e+13 133206
## - `year:mileage`                          1 3.3505e+10 8.1419e+13 133206
## - `fuel:log10_price`                      1 3.8369e+10 8.1424e+13 133207
## - `seller_type:max_power`                 1 3.8571e+10 8.1424e+13 133207
## - `name:max_power`                        1 5.0213e+10 8.1436e+13 133207
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 5.1696e+10 8.1437e+13 133208
## - `km_driven:transmission`                1 6.0251e+10 8.1446e+13 133208
```

```
## - `name:seller_type`                          1 7.5539e+10 8.1461e+13 133209
## - `km_driven:engine`                          1 8.4166e+10 8.1470e+13 133210
## - `km_driven:seller_type`                     1 1.2755e+11 8.1513e+13 133213
## - `mileage:engine`                            1 1.3470e+11 8.1520e+13 133213
## - `seller_type:mileage`                       1 1.5501e+11 8.1540e+13 133215
## - `mileage:max_power`                         1 1.7229e+11 8.1558e+13 133216
## - fuel                                        1 1.9056e+11 8.1576e+13 133217
## - `year:fuel`                                 1 1.9179e+11 8.1577e+13 133217
## - `fuel:seller_type`                          1 1.9239e+11 8.1578e+13 133217
## - `km_driven:max_power`                       1 1.9288e+11 8.1578e+13 133217
## - km_driven                                   1 1.9681e+11 8.1582e+13 133218
## - engine                                      1 2.0347e+11 8.1589e+13 133218
## - `fuel:transmission`                         1 2.0701e+11 8.1592e+13 133218
## - `year:km_driven`                            1 2.2586e+11 8.1611e+13 133220
## - `fuel:mileage`                              1 2.2599e+11 8.1611e+13 133220
## - `year:transmission`                         1 2.3873e+11 8.1624e+13 133221
## - `year:engine`                               1 2.5219e+11 8.1638e+13 133222
## - `owner:engine`                              1 2.6458e+11 8.1650e+13 133222
## - `owner:log10_price`                         1 2.6913e+11 8.1655e+13 133223
## - `km_driven:fuel`                            1 2.7349e+11 8.1659e+13 133223
## - `owner:max_power`                           1 2.7370e+11 8.1659e+13 133223
## - `km_driven:mileage`                         1 2.8053e+11 8.1666e+13 133224
## - `engine:max_power`                          1 2.9132e+11 8.1677e+13 133224
## - owner                                       1 2.9998e+11 8.1685e+13 133225
## - transmission                                1 3.0064e+11 8.1686e+13 133225
## - `year:owner`                                1 3.0585e+11 8.1691e+13 133225
## - year                                        1 3.4378e+11 8.1729e+13 133228
## - `poly(year, 2, raw = TRUE)[, 2]`            1 3.8929e+11 8.1775e+13 133231
## - `name:mileage`                              1 4.3936e+11 8.1825e+13 133235
## - seller_type                                 1 5.8413e+11 8.1970e+13 133245
## - name                                        1 5.9467e+11 8.1980e+13 133245
## - `year:seller_type`                          1 6.0900e+11 8.1994e+13 133246
## - `name:year`                                 1 6.7978e+11 8.2065e+13 133251
## - `km_driven:log10_price`                     1 7.6814e+11 8.2154e+13 133257
## - `seller_type:transmission`                  1 8.1010e+11 8.2196e+13 133260
## - `transmission:log10_price`                  1 9.1331e+11 8.2299e+13 133267
## - `seller_type:log10_price`                   1 1.2544e+12 8.2640e+13 133291
## - `seller_type:engine`                        1 1.4453e+12 8.2831e+13 133304
## - `name:fuel`                                 1 1.4990e+12 8.2884e+13 133308
## - `max_power:log10_price`                     1 1.6475e+12 8.3033e+13 133318
## - log10_price                                 1 1.7683e+12 8.3154e+13 133326
## - `year:log10_price`                          1 2.0590e+12 8.3444e+13 133346
## - `name:transmission`                         1 2.4414e+12 8.3827e+13 133372
## - `name:log10_price`                          1 2.6554e+12 8.4041e+13 133387
## - `engine:log10_price`                        1 2.7512e+12 8.4137e+13 133393
## - `poly(log10_price, 2, raw = TRUE)[, 2]`     1 1.2775e+13 9.4161e+13 134034
##
## Step:  AIC=133204.5
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + log10_price +
```

```
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `km_driven:log10_price` + `fuel:seller_type` +
##     `fuel:transmission` + `fuel:mileage` + `fuel:engine` +
`fuel:log10_price` +
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:max_power` + `seller_type:log10_price` +
`transmission:owner` +
##     `transmission:engine` + `transmission:max_power` +
`transmission:log10_price` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `engine:max_power` +
##     `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - max_power                               1 8.6429e+09 8.1402e+13 133203
## - `transmission:max_power`                1 1.0711e+10 8.1404e+13 133203
## - `name:km_driven`                        1 1.0879e+10 8.1404e+13 133203
## - `transmission:engine`                   1 1.1210e+10 8.1404e+13 133203
## - `name:engine`                           1 1.1290e+10 8.1404e+13 133203
## - `poly(max_power, 2, raw = TRUE)[, 2]`   1 1.1396e+10 8.1404e+13 133203
## - `fuel:engine`                           1 1.8009e+10 8.1411e+13 133204
## - `year:max_power`                        1 2.2342e+10 8.1415e+13 133204
## <none>                                                 8.1393e+13 133204
## - mileage                                 1 3.0797e+10 8.1424e+13 133205
## - `year:mileage`                          1 3.1263e+10 8.1424e+13 133205
## - `name:owner`                            1 3.3635e+10 8.1427e+13 133205
## - `transmission:owner`                    1 3.6421e+10 8.1429e+13 133205
## - `seller_type:max_power`                 1 3.7031e+10 8.1430e+13 133205
## - `fuel:log10_price`                      1 3.8550e+10 8.1432e+13 133205
## - `name:max_power`                        1 5.0084e+10 8.1443e+13 133206
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 5.1622e+10 8.1445e+13 133206
## - `km_driven:transmission`                1 6.0362e+10 8.1453e+13 133207
## - `name:seller_type`                      1 7.4597e+10 8.1468e+13 133208
## - `km_driven:engine`                      1 8.3140e+10 8.1476e+13 133208
## - `km_driven:seller_type`                 1 1.2413e+11 8.1517e+13 133211
## - `mileage:engine`                        1 1.3564e+11 8.1529e+13 133212
## - `seller_type:mileage`                   1 1.5579e+11 8.1549e+13 133213
## - `mileage:max_power`                     1 1.7233e+11 8.1565e+13 133215
## - fuel                                    1 1.9073e+11 8.1584e+13 133216
```

```
## - `year:fuel`                                  1 1.9198e+11 8.1585e+13 133216
## - `fuel:seller_type`                           1 1.9248e+11 8.1586e+13 133216
## - engine                                       1 1.9981e+11 8.1593e+13 133216
## - `km_driven:max_power`                        1 2.0096e+11 8.1594e+13 133216
## - `fuel:transmission`                          1 2.0876e+11 8.1602e+13 133217
## - km_driven                                    1 2.1630e+11 8.1609e+13 133218
## - `fuel:mileage`                               1 2.2857e+11 8.1622e+13 133218
## - `year:transmission`                          1 2.4029e+11 8.1633e+13 133219
## - `year:km_driven`                             1 2.4527e+11 8.1638e+13 133220
## - `year:engine`                                1 2.4840e+11 8.1641e+13 133220
## - `km_driven:fuel`                             1 2.6814e+11 8.1661e+13 133221
## - `owner:log10_price`                          1 2.6821e+11 8.1661e+13 133221
## - `km_driven:mileage`                          1 2.7356e+11 8.1667e+13 133222
## - `owner:max_power`                            1 2.8277e+11 8.1676e+13 133222
## - owner                                        1 2.9254e+11 8.1686e+13 133223
## - `engine:max_power`                           1 2.9493e+11 8.1688e+13 133223
## - `year:owner`                                 1 2.9834e+11 8.1691e+13 133223
## - transmission                                 1 3.0267e+11 8.1696e+13 133224
## - year                                         1 3.4770e+11 8.1741e+13 133227
## - `owner:engine`                               1 3.4904e+11 8.1742e+13 133227
## - `poly(year, 2, raw = TRUE)[, 2]`             1 3.9311e+11 8.1786e+13 133230
## - `name:mileage`                               1 4.4216e+11 8.1835e+13 133233
## - seller_type                                  1 5.8372e+11 8.1977e+13 133243
## - name                                         1 5.9261e+11 8.1986e+13 133244
## - `year:seller_type`                           1 6.0876e+11 8.2002e+13 133245
## - `name:year`                                  1 6.7780e+11 8.2071e+13 133250
## - `km_driven:log10_price`                      1 7.6067e+11 8.2154e+13 133255
## - `seller_type:transmission`                   1 8.0850e+11 8.2202e+13 133259
## - `transmission:log10_price`                   1 9.2241e+11 8.2315e+13 133267
## - `seller_type:log10_price`                    1 1.2630e+12 8.2656e+13 133290
## - `seller_type:engine`                         1 1.4466e+12 8.2840e+13 133303
## - `name:fuel`                                  1 1.5079e+12 8.2901e+13 133307
## - `max_power:log10_price`                      1 1.6500e+12 8.3043e+13 133317
## - log10_price                                  1 1.7635e+12 8.3157e+13 133324
## - `year:log10_price`                           1 2.0540e+12 8.3447e+13 133344
## - `name:transmission`                          1 2.4383e+12 8.3831e+13 133370
## - `name:log10_price`                           1 2.6626e+12 8.4056e+13 133386
## - `engine:log10_price`                         1 2.7703e+12 8.4163e+13 133393
## - `poly(log10_price, 2, raw = TRUE)[, 2]`      1 1.2771e+13 9.4164e+13 134032
##
## Step:  AIC=133203.1
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(max_power, 2, raw = TRUE)[,
2]` +
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` +
`name:km_driven` +
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
```

```
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` +
`km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:engine` +
##     `transmission:max_power` + `transmission:log10_price` + `owner:engine`
+
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power` + `engine:log10_price` +
##     `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `poly(max_power, 2, raw = TRUE)[, 2]`    1 5.0952e+09 8.1407e+13 133201
## - `transmission:max_power`                 1 9.2222e+09 8.1411e+13 133202
## - `name:km_driven`                         1 1.0875e+10 8.1413e+13 133202
## - `name:engine`                            1 1.1385e+10 8.1413e+13 133202
## - `transmission:engine`                    1 1.3493e+10 8.1415e+13 133202
## - `fuel:engine`                            1 1.7657e+10 8.1419e+13 133202
## <none>                                                  8.1402e+13 133203
## - mileage                                  1 3.1296e+10 8.1433e+13 133203
## - `year:mileage`                           1 3.1771e+10 8.1433e+13 133203
## - `name:owner`                             1 3.2995e+10 8.1435e+13 133203
## - `transmission:owner`                     1 3.4050e+10 8.1436e+13 133203
## - `seller_type:max_power`                  1 3.7559e+10 8.1439e+13 133204
## - `fuel:log10_price`                       1 4.0183e+10 8.1442e+13 133204
## - `name:max_power`                         1 5.1753e+10 8.1453e+13 133205
## - `poly(engine, 2, raw = TRUE)[, 2]`       1 5.1965e+10 8.1454e+13 133205
## - `km_driven:transmission`                 1 5.9736e+10 8.1461e+13 133205
## - `name:seller_type`                       1 7.4707e+10 8.1476e+13 133206
## - `km_driven:engine`                       1 8.5346e+10 8.1487e+13 133207
## - `km_driven:seller_type`                  1 1.2489e+11 8.1527e+13 133210
## - `mileage:engine`                         1 1.3115e+11 8.1533e+13 133210
## - `seller_type:mileage`                    1 1.5388e+11 8.1556e+13 133212
## - `mileage:max_power`                      1 1.6747e+11 8.1569e+13 133213
## - `fuel:seller_type`                       1 1.9103e+11 8.1593e+13 133214
## - `km_driven:max_power`                    1 1.9305e+11 8.1595e+13 133215
## - engine                                   1 1.9868e+11 8.1600e+13 133215
## - fuel                                     1 2.0462e+11 8.1606e+13 133215
## - `year:fuel`                              1 2.0567e+11 8.1607e+13 133215
## - `fuel:transmission`                      1 2.0569e+11 8.1607e+13 133215
## - km_driven                                1 2.2320e+11 8.1625e+13 133217
## - `year:transmission`                      1 2.3235e+11 8.1634e+13 133217
```

```
## - `fuel:mileage`                                1 2.3372e+11 8.1635e+13 133217
## - `year:engine`                                 1 2.5139e+11 8.1653e+13 133219
## - `year:km_driven`                              1 2.5284e+11 8.1655e+13 133219
## - `owner:log10_price`                           1 2.6474e+11 8.1666e+13 133220
## - `km_driven:mileage`                           1 2.7121e+11 8.1673e+13 133220
## - `km_driven:fuel`                              1 2.7240e+11 8.1674e+13 133220
## - transmission                                  1 2.9432e+11 8.1696e+13 133222
## - `engine:max_power`                            1 2.9548e+11 8.1697e+13 133222
## - owner                                         1 2.9663e+11 8.1698e+13 133222
## - `year:owner`                                  1 3.0221e+11 8.1704e+13 133222
## - `owner:max_power`                             1 3.0778e+11 8.1709e+13 133223
## - `owner:engine`                                1 3.5839e+11 8.1760e+13 133226
## - year                                          1 4.3910e+11 8.1841e+13 133232
## - `name:mileage`                                1 4.4719e+11 8.1849e+13 133232
## - `poly(year, 2, raw = TRUE)[, 2]`              1 4.9757e+11 8.1899e+13 133236
## - seller_type                                   1 5.8510e+11 8.1987e+13 133242
## - `year:seller_type`                            1 6.1018e+11 8.2012e+13 133244
## - name                                          1 6.4403e+11 8.2046e+13 133246
## - `name:year`                                   1 7.3462e+11 8.2136e+13 133252
## - `km_driven:log10_price`                       1 7.8278e+11 8.2184e+13 133256
## - `seller_type:transmission`                    1 8.0971e+11 8.2211e+13 133257
## - `transmission:log10_price`                    1 9.3222e+11 8.2334e+13 133266
## - `seller_type:log10_price`                     1 1.2631e+12 8.2665e+13 133289
## - `seller_type:engine`                          1 1.4445e+12 8.2846e+13 133301
## - `name:fuel`                                   1 1.5194e+12 8.2921e+13 133306
## - `name:transmission`                           1 2.4639e+12 8.3866e+13 133371
## - log10_price                                   1 2.5341e+12 8.3936e+13 133376
## - `name:log10_price`                            1 2.7684e+12 8.4170e+13 133391
## - `engine:log10_price`                          1 2.8468e+12 8.4249e+13 133397
## - `year:log10_price`                            1 2.9363e+12 8.4338e+13 133403
## - `max_power:log10_price`                       1 3.2768e+12 8.4678e+13 133426
## - `year:max_power`                              1 3.9210e+12 8.5323e+13 133469
## - `poly(log10_price, 2, raw = TRUE)[, 2]`       1 1.6440e+13 9.7842e+13 134248
##
## Step:  AIC=133201.4
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `km_driven:log10_price` + `fuel:seller_type` +
##     `fuel:transmission` + `fuel:mileage` + `fuel:engine` +
`fuel:log10_price` +
```

```
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:max_power` + `seller_type:log10_price` +
`transmission:owner` +
##     `transmission:engine` + `transmission:max_power` +
`transmission:log10_price` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `engine:max_power` +
##     `engine:log10_price` + `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `transmission:engine`                   1 1.0051e+10 8.1417e+13 133200
## - `name:engine`                           1 1.0371e+10 8.1417e+13 133200
## - `name:km_driven`                        1 1.0556e+10 8.1417e+13 133200
## - `fuel:engine`                           1 1.4823e+10 8.1422e+13 133200
## - `transmission:max_power`                1 2.0837e+10 8.1428e+13 133201
## <none>                                                 8.1407e+13 133201
## - mileage                                 1 3.1000e+10 8.1438e+13 133202
## - `year:mileage`                          1 3.1414e+10 8.1438e+13 133202
## - `name:owner`                            1 3.4186e+10 8.1441e+13 133202
## - `transmission:owner`                    1 3.5217e+10 8.1442e+13 133202
## - `seller_type:max_power`                 1 3.9287e+10 8.1446e+13 133202
## - `fuel:log10_price`                      1 4.0215e+10 8.1447e+13 133202
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 5.4208e+10 8.1461e+13 133203
## - `name:max_power`                        1 6.0220e+10 8.1467e+13 133204
## - `km_driven:transmission`                1 6.2467e+10 8.1469e+13 133204
## - `name:seller_type`                      1 7.2667e+10 8.1479e+13 133205
## - `km_driven:engine`                      1 8.3635e+10 8.1490e+13 133205
## - `km_driven:seller_type`                 1 1.2568e+11 8.1532e+13 133208
## - `mileage:engine`                        1 1.2742e+11 8.1534e+13 133208
## - `seller_type:mileage`                   1 1.4879e+11 8.1556e+13 133210
## - `mileage:max_power`                     1 1.6840e+11 8.1575e+13 133211
## - `fuel:seller_type`                      1 1.8627e+11 8.1593e+13 133212
## - engine                                  1 1.9582e+11 8.1603e+13 133213
## - fuel                                    1 2.0288e+11 8.1610e+13 133214
## - `km_driven:max_power`                   1 2.0355e+11 8.1610e+13 133214
## - `year:fuel`                             1 2.0386e+11 8.1611e+13 133214
## - `fuel:transmission`                     1 2.1318e+11 8.1620e+13 133214
## - km_driven                               1 2.2198e+11 8.1629e+13 133215
## - `year:transmission`                     1 2.3013e+11 8.1637e+13 133215
## - `fuel:mileage`                          1 2.3077e+11 8.1638e+13 133216
## - `year:engine`                           1 2.4804e+11 8.1655e+13 133217
## - `year:km_driven`                        1 2.5155e+11 8.1658e+13 133217
## - `km_driven:mileage`                     1 2.7135e+11 8.1678e+13 133218
## - `owner:log10_price`                     1 2.7291e+11 8.1680e+13 133218
## - `km_driven:fuel`                        1 2.7375e+11 8.1681e+13 133219
## - transmission                            1 2.9225e+11 8.1699e+13 133220
## - owner                                   1 2.9797e+11 8.1705e+13 133220
## - `year:owner`                            1 3.0376e+11 8.1711e+13 133221
## - `owner:max_power`                       1 3.0583e+11 8.1713e+13 133221
```

```
## - `engine:max_power`                          1 3.3592e+11 8.1743e+13 133223
## - `owner:engine`                              1 3.5333e+11 8.1760e+13 133224
## - year                                        1 4.3466e+11 8.1841e+13 133230
## - `name:mileage`                              1 4.5683e+11 8.1864e+13 133231
## - `poly(year, 2, raw = TRUE)[, 2]`            1 4.9299e+11 8.1900e+13 133234
## - seller_type                                 1 5.8405e+11 8.1991e+13 133240
## - `year:seller_type`                          1 6.0917e+11 8.2016e+13 133242
## - name                                        1 6.4380e+11 8.2051e+13 133244
## - `name:year`                                 1 7.3468e+11 8.2141e+13 133251
## - `km_driven:log10_price`                     1 7.8077e+11 8.2188e+13 133254
## - `seller_type:transmission`                  1 8.1885e+11 8.2226e+13 133256
## - `transmission:log10_price`                  1 9.9399e+11 8.2401e+13 133268
## - `seller_type:log10_price`                   1 1.2590e+12 8.2666e+13 133287
## - `seller_type:engine`                        1 1.4493e+12 8.2856e+13 133300
## - `name:fuel`                                 1 1.5293e+12 8.2936e+13 133305
## - `name:transmission`                         1 2.4993e+12 8.3906e+13 133372
## - log10_price                                 1 2.5299e+12 8.3937e+13 133374
## - `name:log10_price`                          1 2.7920e+12 8.4199e+13 133391
## - `engine:log10_price`                        1 2.9310e+12 8.4338e+13 133401
## - `year:log10_price`                          1 2.9320e+12 8.4339e+13 133401
## - `max_power:log10_price`                     1 4.2242e+12 8.5631e+13 133487
## - `year:max_power`                            1 5.1037e+12 8.6511e+13 133545
## - `poly(log10_price, 2, raw = TRUE)[, 2]`     1 1.6590e+13 9.7996e+13 134255
##
## Step:  AIC=133200.1
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:km_driven` + `name:fuel` + `name:seller_type` +
##     `name:transmission` + `name:owner` + `name:mileage` + `name:engine` +
##     `name:max_power` + `name:log10_price` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `km_driven:log10_price` + `fuel:seller_type` +
##     `fuel:transmission` + `fuel:mileage` + `fuel:engine` +
`fuel:log10_price` +
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:max_power` + `seller_type:log10_price` +
`transmission:owner` +
##     `transmission:max_power` + `transmission:log10_price` + `owner:engine`
+
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power` + `engine:log10_price` +
##     `max_power:log10_price`
##
```

```
##                                          Df  Sum of Sq        RSS    AIC
## - `name:km_driven`                        1 1.0296e+10 8.1427e+13 133199
## - `fuel:engine`                           1 1.1967e+10 8.1429e+13 133199
## - `name:engine`                           1 1.6113e+10 8.1433e+13 133199
## <none>                                                 8.1417e+13 133200
## - mileage                                 1 3.0481e+10 8.1447e+13 133200
## - `year:mileage`                          1 3.0924e+10 8.1448e+13 133200
## - `transmission:owner`                    1 3.2391e+10 8.1449e+13 133200
## - `name:owner`                            1 3.2917e+10 8.1450e+13 133200
## - `seller_type:max_power`                 1 3.9654e+10 8.1456e+13 133201
## - `fuel:log10_price`                      1 4.0487e+10 8.1457e+13 133201
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 5.0042e+10 8.1467e+13 133202
## - `name:max_power`                        1 5.1112e+10 8.1468e+13 133202
## - `transmission:max_power`                1 5.7436e+10 8.1474e+13 133202
## - `km_driven:transmission`                1 5.7567e+10 8.1474e+13 133202
## - `name:seller_type`                      1 7.2526e+10 8.1489e+13 133203
## - `km_driven:engine`                      1 8.5879e+10 8.1503e+13 133204
## - `mileage:engine`                        1 1.2089e+11 8.1538e+13 133207
## - `km_driven:seller_type`                 1 1.2672e+11 8.1544e+13 133207
## - `mileage:max_power`                     1 1.5913e+11 8.1576e+13 133209
## - `seller_type:mileage`                   1 1.5918e+11 8.1576e+13 133209
## - `fuel:seller_type`                      1 1.9805e+11 8.1615e+13 133212
## - `km_driven:max_power`                   1 1.9839e+11 8.1615e+13 133212
## - engine                                  1 2.0083e+11 8.1618e+13 133212
## - fuel                                    1 2.0988e+11 8.1627e+13 133213
## - `year:fuel`                             1 2.1068e+11 8.1628e+13 133213
## - km_driven                               1 2.2404e+11 8.1641e+13 133214
## - `fuel:mileage`                          1 2.3371e+11 8.1651e+13 133214
## - `year:km_driven`                        1 2.5362e+11 8.1670e+13 133216
## - `year:engine`                           1 2.5480e+11 8.1672e+13 133216
## - `fuel:transmission`                     1 2.5530e+11 8.1672e+13 133216
## - `year:transmission`                     1 2.6685e+11 8.1684e+13 133217
## - `owner:log10_price`                     1 2.7193e+11 8.1689e+13 133217
## - `km_driven:fuel`                        1 2.7392e+11 8.1691e+13 133217
## - `km_driven:mileage`                     1 2.7711e+11 8.1694e+13 133217
## - owner                                   1 3.0062e+11 8.1717e+13 133219
## - `year:owner`                            1 3.0634e+11 8.1723e+13 133219
## - `owner:max_power`                       1 3.2222e+11 8.1739e+13 133221
## - `engine:max_power`                      1 3.3333e+11 8.1750e+13 133221
## - transmission                            1 3.3462e+11 8.1751e+13 133221
## - `owner:engine`                          1 3.6347e+11 8.1780e+13 133223
## - year                                    1 4.5915e+11 8.1876e+13 133230
## - `name:mileage`                          1 4.6769e+11 8.1885e+13 133231
## - `poly(year, 2, raw = TRUE)[, 2]`        1 5.2036e+11 8.1937e+13 133234
## - seller_type                             1 5.9190e+11 8.2009e+13 133239
## - `year:seller_type`                      1 6.1709e+11 8.2034e+13 133241
## - name                                    1 6.7086e+11 8.2088e+13 133245
## - `name:year`                             1 7.6399e+11 8.2181e+13 133251
## - `km_driven:log10_price`                 1 7.7832e+11 8.2195e+13 133252
## - `seller_type:transmission`              1 8.2253e+11 8.2239e+13 133255
```

```
## - `transmission:log10_price`                1 1.0183e+12 8.2435e+13 133269
## - `seller_type:log10_price`                 1 1.2728e+12 8.2690e+13 133286
## - `seller_type:engine`                      1 1.5203e+12 8.2937e+13 133303
## - `name:fuel`                               1 1.5328e+12 8.2950e+13 133304
## - `name:transmission`                       1 2.4949e+12 8.3912e+13 133370
## - log10_price                               1 2.6758e+12 8.4093e+13 133382
## - `name:log10_price`                        1 2.8188e+12 8.4236e+13 133392
## - `year:log10_price`                        1 3.0950e+12 8.4512e+13 133410
## - `engine:log10_price`                      1 3.2405e+12 8.4657e+13 133420
## - `max_power:log10_price`                   1 4.8115e+12 8.6228e+13 133525
## - `year:max_power`                          1 5.5629e+12 8.6980e+13 133574
## - `poly(log10_price, 2, raw = TRUE)[, 2]`   1 1.7026e+13 9.8443e+13 134279
##
## Step:  AIC=133198.9
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` +
`km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:engine` + `fuel:log10_price` + `seller_type:transmission` +
##     `seller_type:mileage` + `seller_type:engine` + `seller_type:max_power`
+
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:max_power` +
##     `transmission:log10_price` + `owner:engine` + `owner:max_power` +
##     `owner:log10_price` + `mileage:engine` + `mileage:max_power` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                           Df  Sum of Sq        RSS    AIC
## - `fuel:engine`                            1 1.1140e+10 8.1438e+13 133198
## - `name:engine`                            1 2.0425e+10 8.1448e+13 133198
## <none>                                                  8.1427e+13 133199
## - mileage                                  1 2.9927e+10 8.1457e+13 133199
## - `year:mileage`                           1 3.0368e+10 8.1457e+13 133199
## - `name:owner`                             1 3.0985e+10 8.1458e+13 133199
## - `transmission:owner`                     1 3.2585e+10 8.1460e+13 133199
## - `seller_type:max_power`                  1 3.7477e+10 8.1465e+13 133199
## - `fuel:log10_price`                       1 4.2324e+10 8.1469e+13 133200
## - `poly(engine, 2, raw = TRUE)[, 2]`       1 4.8271e+10 8.1475e+13 133200
## - `km_driven:transmission`                 1 5.1570e+10 8.1479e+13 133200
```

```
## - `name:max_power`                              1 5.3097e+10 8.1480e+13 133201
## - `transmission:max_power`                      1 5.6252e+10 8.1483e+13 133201
## - `name:seller_type`                            1 6.6281e+10 8.1493e+13 133201
## - `km_driven:engine`                            1 9.2467e+10 8.1520e+13 133203
## - `mileage:engine`                              1 1.2094e+11 8.1548e+13 133205
## - `km_driven:seller_type`                       1 1.2698e+11 8.1554e+13 133206
## - `seller_type:mileage`                         1 1.5757e+11 8.1585e+13 133208
## - `mileage:max_power`                           1 1.6108e+11 8.1588e+13 133208
## - `fuel:seller_type`                            1 1.9440e+11 8.1622e+13 133210
## - engine                                        1 2.0270e+11 8.1630e+13 133211
## - fuel                                          1 2.0567e+11 8.1633e+13 133211
## - `year:fuel`                                   1 2.0675e+11 8.1634e+13 133211
## - km_driven                                     1 2.1375e+11 8.1641e+13 133212
## - `km_driven:max_power`                         1 2.2109e+11 8.1648e+13 133212
## - `fuel:mileage`                                1 2.3722e+11 8.1664e+13 133213
## - `year:km_driven`                              1 2.4336e+11 8.1670e+13 133214
## - `fuel:transmission`                           1 2.4969e+11 8.1677e+13 133214
## - `year:engine`                                 1 2.5712e+11 8.1684e+13 133215
## - `km_driven:fuel`                              1 2.6694e+11 8.1694e+13 133215
## - `year:transmission`                           1 2.6716e+11 8.1694e+13 133215
## - `km_driven:mileage`                           1 2.7006e+11 8.1697e+13 133216
## - `owner:log10_price`                           1 2.7044e+11 8.1698e+13 133216
## - owner                                         1 2.9904e+11 8.1726e+13 133218
## - `year:owner`                                  1 3.0472e+11 8.1732e+13 133218
## - `owner:max_power`                             1 3.2176e+11 8.1749e+13 133219
## - transmission                                  1 3.3476e+11 8.1762e+13 133220
## - `engine:max_power`                            1 3.4113e+11 8.1768e+13 133221
## - `owner:engine`                                1 3.6228e+11 8.1789e+13 133222
## - year                                          1 4.5891e+11 8.1886e+13 133229
## - `name:mileage`                                1 4.5973e+11 8.1887e+13 133229
## - `poly(year, 2, raw = TRUE)[, 2]`              1 5.2028e+11 8.1947e+13 133233
## - seller_type                                   1 5.9559e+11 8.2023e+13 133238
## - `year:seller_type`                            1 6.2096e+11 8.2048e+13 133240
## - name                                          1 6.6081e+11 8.2088e+13 133243
## - `name:year`                                   1 7.5385e+11 8.2181e+13 133249
## - `km_driven:log10_price`                       1 7.8081e+11 8.2208e+13 133251
## - `seller_type:transmission`                    1 8.2044e+11 8.2248e+13 133254
## - `transmission:log10_price`                    1 1.0119e+12 8.2439e+13 133267
## - `seller_type:log10_price`                     1 1.2801e+12 8.2707e+13 133286
## - `seller_type:engine`                          1 1.5181e+12 8.2945e+13 133302
## - `name:fuel`                                   1 1.6917e+12 8.3119e+13 133314
## - `name:transmission`                           1 2.4847e+12 8.3912e+13 133368
## - log10_price                                   1 2.6881e+12 8.4115e+13 133382
## - `name:log10_price`                            1 2.9628e+12 8.4390e+13 133400
## - `year:log10_price`                            1 3.1098e+12 8.4537e+13 133410
## - `engine:log10_price`                          1 3.2745e+12 8.4702e+13 133421
## - `max_power:log10_price`                       1 4.8063e+12 8.6233e+13 133523
## - `year:max_power`                              1 5.5559e+12 8.6983e+13 133572
## - `poly(log10_price, 2, raw = TRUE)[, 2]`       1 1.7189e+13 9.8616e+13 134287
##
```

```
## Step:  AIC=133197.6
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:engine` + `name:max_power` +
##     `name:log10_price` + `year:km_driven` + `year:fuel` +
`year:seller_type` +
##     `year:transmission` + `year:owner` + `year:mileage` + `year:engine` +
##     `year:max_power` + `year:log10_price` + `km_driven:fuel` +
##     `km_driven:seller_type` + `km_driven:transmission` +
`km_driven:mileage` +
##     `km_driven:engine` + `km_driven:max_power` + `km_driven:log10_price` +
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:log10_price` + `seller_type:transmission` +
`seller_type:mileage` +
##     `seller_type:engine` + `seller_type:max_power` +
`seller_type:log10_price` +
##     `transmission:owner` + `transmission:max_power` +
`transmission:log10_price` +
##     `owner:engine` + `owner:max_power` + `owner:log10_price` +
##     `mileage:engine` + `mileage:max_power` + `engine:max_power` +
##     `engine:log10_price` + `max_power:log10_price`
##
##                                         Df  Sum of Sq        RSS    AIC
## - `name:engine`                          1 1.9680e+10 8.1458e+13 133197
## <none>                                               8.1438e+13 133198
## - `name:owner`                           1 3.0185e+10 8.1468e+13 133198
## - `transmission:owner`                   1 3.0898e+10 8.1469e+13 133198
## - `seller_type:max_power`                1 3.3729e+10 8.1472e+13 133198
## - mileage                                1 3.4664e+10 8.1473e+13 133198
## - `year:mileage`                         1 3.4983e+10 8.1473e+13 133198
## - `transmission:max_power`               1 4.7943e+10 8.1486e+13 133199
## - `km_driven:transmission`               1 5.1435e+10 8.1490e+13 133199
## - `name:seller_type`                     1 6.4638e+10 8.1503e+13 133200
## - `name:max_power`                       1 6.4807e+10 8.1503e+13 133200
## - `poly(engine, 2, raw = TRUE)[, 2]`     1 7.8636e+10 8.1517e+13 133201
## - `fuel:log10_price`                     1 9.4820e+10 8.1533e+13 133202
## - `km_driven:engine`                     1 9.9432e+10 8.1538e+13 133203
## - `km_driven:seller_type`                1 1.2563e+11 8.1564e+13 133204
## - `mileage:engine`                       1 1.3578e+11 8.1574e+13 133205
## - `seller_type:mileage`                  1 1.4831e+11 8.1587e+13 133206
## - `mileage:max_power`                    1 1.5941e+11 8.1598e+13 133207
## - `fuel:seller_type`                     1 1.8375e+11 8.1622e+13 133208
## - engine                                 1 1.9664e+11 8.1635e+13 133209
## - km_driven                              1 2.0874e+11 8.1647e+13 133210
## - `fuel:mileage`                         1 2.3424e+11 8.1673e+13 133212
## - `year:km_driven`                       1 2.3794e+11 8.1676e+13 133212
```

```
## - `km_driven:max_power`                      1 2.3890e+11 8.1677e+13 133212
## - `year:engine`                              1 2.5103e+11 8.1689e+13 133213
## - `km_driven:fuel`                           1 2.6793e+11 8.1706e+13 133214
## - `owner:log10_price`                        1 2.6809e+11 8.1706e+13 133214
## - `year:transmission`                        1 2.7331e+11 8.1712e+13 133215
## - `fuel:transmission`                        1 2.7369e+11 8.1712e+13 133215
## - `km_driven:mileage`                        1 2.7919e+11 8.1717e+13 133215
## - fuel                                       1 2.8976e+11 8.1728e+13 133216
## - `year:fuel`                                1 2.9132e+11 8.1730e+13 133216
## - owner                                      1 2.9898e+11 8.1737e+13 133216
## - `year:owner`                               1 3.0456e+11 8.1743e+13 133217
## - `owner:max_power`                          1 3.2263e+11 8.1761e+13 133218
## - transmission                               1 3.4163e+11 8.1780e+13 133219
## - `owner:engine`                             1 3.6306e+11 8.1801e+13 133221
## - `engine:max_power`                         1 3.6985e+11 8.1808e+13 133221
## - year                                       1 4.6480e+11 8.1903e+13 133228
## - `name:mileage`                             1 4.6731e+11 8.1906e+13 133228
## - `poly(year, 2, raw = TRUE)[, 2]`           1 5.2694e+11 8.1965e+13 133232
## - seller_type                                1 6.0645e+11 8.2045e+13 133238
## - `year:seller_type`                         1 6.3268e+11 8.2071e+13 133240
## - name                                       1 6.9192e+11 8.2130e+13 133244
## - `km_driven:log10_price`                    1 7.6975e+11 8.2208e+13 133249
## - `name:year`                                1 7.8905e+11 8.2227e+13 133250
## - `seller_type:transmission`                 1 8.2136e+11 8.2260e+13 133253
## - `transmission:log10_price`                 1 1.0015e+12 8.2440e+13 133265
## - `seller_type:log10_price`                  1 1.3024e+12 8.2741e+13 133286
## - `seller_type:engine`                       1 1.5146e+12 8.2953e+13 133300
## - `name:fuel`                                1 1.8015e+12 8.3240e+13 133320
## - `name:transmission`                        1 2.4763e+12 8.3915e+13 133366
## - log10_price                                1 2.7079e+12 8.4146e+13 133382
## - `name:log10_price`                         1 3.1240e+12 8.4562e+13 133410
## - `year:log10_price`                         1 3.1347e+12 8.4573e+13 133411
## - `engine:log10_price`                       1 3.2729e+12 8.4711e+13 133420
## - `max_power:log10_price`                    1 4.8555e+12 8.6294e+13 133525
## - `year:max_power`                           1 5.5897e+12 8.7028e+13 133573
## - `poly(log10_price, 2, raw = TRUE)[, 2]`    1 1.7607e+13 9.9045e+13 134310
##
## Step:  AIC=133197
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:max_power` + `name:log10_price`
+
##     `year:km_driven` + `year:fuel` + `year:seller_type` +
`year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
```

```
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `km_driven:log10_price` + `fuel:seller_type` +
##     `fuel:transmission` + `fuel:mileage` + `fuel:log10_price` +
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:max_power` + `seller_type:log10_price` +
`transmission:owner` +
##     `transmission:max_power` + `transmission:log10_price` + `owner:engine`
+
##     `owner:max_power` + `owner:log10_price` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power` + `engine:log10_price` +
##     `max_power:log10_price`
##
##                                          Df  Sum of Sq        RSS    AIC
## - `seller_type:max_power`                 1 2.8359e+10 8.1486e+13 133197
## - `name:owner`                            1 2.8471e+10 8.1486e+13 133197
## <none>                                                 8.1458e+13 133197
## - `transmission:owner`                    1 2.9250e+10 8.1487e+13 133197
## - mileage                                 1 3.6064e+10 8.1494e+13 133198
## - `year:mileage`                          1 3.6376e+10 8.1494e+13 133198
## - `transmission:max_power`                1 4.5514e+10 8.1503e+13 133198
## - `km_driven:transmission`                1 4.9139e+10 8.1507e+13 133198
## - `name:seller_type`                      1 7.0245e+10 8.1528e+13 133200
## - `fuel:log10_price`                      1 8.4308e+10 8.1542e+13 133201
## - `km_driven:engine`                      1 8.9780e+10 8.1548e+13 133201
## - `poly(engine, 2, raw = TRUE)[, 2]`      1 1.1153e+11 8.1569e+13 133203
## - `name:max_power`                        1 1.2097e+11 8.1579e+13 133203
## - `km_driven:seller_type`                 1 1.2266e+11 8.1581e+13 133204
## - `mileage:max_power`                     1 1.5318e+11 8.1611e+13 133206
## - `seller_type:mileage`                   1 1.5404e+11 8.1612e+13 133206
## - `mileage:engine`                        1 1.6395e+11 8.1622e+13 133206
## - `fuel:seller_type`                      1 1.8535e+11 8.1643e+13 133208
## - engine                                  1 1.8818e+11 8.1646e+13 133208
## - km_driven                               1 2.1070e+11 8.1669e+13 133210
## - `km_driven:max_power`                   1 2.3478e+11 8.1693e+13 133211
## - `fuel:mileage`                          1 2.3646e+11 8.1694e+13 133211
## - `year:km_driven`                        1 2.3993e+11 8.1698e+13 133212
## - `year:engine`                           1 2.4203e+11 8.1700e+13 133212
## - `km_driven:fuel`                        1 2.5779e+11 8.1716e+13 133213
## - `owner:log10_price`                     1 2.6582e+11 8.1724e+13 133214
## - `km_driven:mileage`                     1 2.6680e+11 8.1725e+13 133214
## - `fuel:transmission`                     1 2.6939e+11 8.1727e+13 133214
## - `year:transmission`                     1 2.7234e+11 8.1730e+13 133214
## - fuel                                    1 2.8260e+11 8.1741e+13 133215
## - `year:fuel`                             1 2.8374e+11 8.1742e+13 133215
## - owner                                   1 2.9526e+11 8.1753e+13 133216
## - `year:owner`                            1 3.0083e+11 8.1759e+13 133216
## - `owner:max_power`                       1 3.2614e+11 8.1784e+13 133218
## - transmission                            1 3.4041e+11 8.1798e+13 133219
## - `engine:max_power`                      1 3.5518e+11 8.1813e+13 133220
```

```
## - `owner:engine`                                1 3.6246e+11 8.1820e+13 133220
## - year                                          1 4.7176e+11 8.1930e+13 133228
## - `poly(year, 2, raw = TRUE)[, 2]`              1 5.3424e+11 8.1992e+13 133232
## - seller_type                                   1 6.2635e+11 8.2084e+13 133239
## - `year:seller_type`                            1 6.5332e+11 8.2111e+13 133240
## - name                                          1 6.7355e+11 8.2131e+13 133242
## - `name:mileage`                                1 6.8694e+11 8.2145e+13 133243
## - `name:year`                                   1 7.7053e+11 8.2228e+13 133249
## - `km_driven:log10_price`                       1 7.7912e+11 8.2237e+13 133249
## - `seller_type:transmission`                    1 8.0628e+11 8.2264e+13 133251
## - `transmission:log10_price`                    1 9.9822e+11 8.2456e+13 133264
## - `seller_type:log10_price`                     1 1.3407e+12 8.2799e+13 133288
## - `seller_type:engine`                          1 1.5092e+12 8.2967e+13 133299
## - `name:fuel`                                   1 2.2349e+12 8.3693e+13 133349
## - `name:transmission`                           1 2.4589e+12 8.3917e+13 133364
## - log10_price                                   1 2.7183e+12 8.4176e+13 133382
## - `name:log10_price`                            1 3.1247e+12 8.4583e+13 133409
## - `year:log10_price`                            1 3.1455e+12 8.4603e+13 133411
## - `engine:log10_price`                          1 3.2663e+12 8.4724e+13 133419
## - `max_power:log10_price`                       1 5.0377e+12 8.6496e+13 133537
## - `year:max_power`                              1 5.7376e+12 8.7196e+13 133582
## - `poly(log10_price, 2, raw = TRUE)[, 2]`       1 1.7623e+13 9.9081e+13 134310
##
## Step:  AIC=133197
## selling_price ~ name + year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + log10_price + `poly(year, 2,
raw = TRUE)[, 2]` +
##     `poly(engine, 2, raw = TRUE)[, 2]` + `poly(log10_price, 2, raw =
TRUE)[, 2]` +
##     `name:year` + `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:mileage` + `name:max_power` + `name:log10_price`
+
##     `year:km_driven` + `year:fuel` + `year:seller_type` +
`year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `km_driven:log10_price` + `fuel:seller_type` +
##     `fuel:transmission` + `fuel:mileage` + `fuel:log10_price` +
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:max_power` +
##     `transmission:log10_price` + `owner:engine` + `owner:max_power` +
##     `owner:log10_price` + `mileage:engine` + `mileage:max_power` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`
##
##                                            Df  Sum of Sq        RSS    AIC
## <none>                                                     8.1486e+13 133197
## - `transmission:owner`                      1 3.0256e+10 8.1517e+13 133197
```

```
## - `name:owner`                                1 3.1123e+10 8.1517e+13 133197
## - mileage                                     1 3.5337e+10 8.1522e+13 133197
## - `year:mileage`                              1 3.5570e+10 8.1522e+13 133197
## - `transmission:max_power`                    1 3.5731e+10 8.1522e+13 133197
## - `km_driven:transmission`                    1 4.6600e+10 8.1533e+13 133198
## - `name:seller_type`                          1 4.7248e+10 8.1534e+13 133198
## - `fuel:log10_price`                          1 8.0489e+10 8.1567e+13 133201
## - `km_driven:engine`                          1 8.9572e+10 8.1576e+13 133201
## - `poly(engine, 2, raw = TRUE)[, 2]`          1 1.0141e+11 8.1588e+13 133202
## - `km_driven:seller_type`                     1 1.1436e+11 8.1601e+13 133203
## - `name:max_power`                            1 1.2645e+11 8.1613e+13 133204
## - `mileage:max_power`                         1 1.3414e+11 8.1620e+13 133204
## - `seller_type:mileage`                       1 1.4036e+11 8.1627e+13 133205
## - `mileage:engine`                            1 1.5301e+11 8.1639e+13 133206
## - `fuel:seller_type`                          1 1.5858e+11 8.1645e+13 133206
## - engine                                      1 1.9282e+11 8.1679e+13 133208
## - km_driven                                   1 2.0378e+11 8.1690e+13 133209
## - `year:km_driven`                            1 2.3253e+11 8.1719e+13 133211
## - `fuel:mileage`                              1 2.4121e+11 8.1728e+13 133212
## - `year:engine`                               1 2.4760e+11 8.1734e+13 133212
## - `km_driven:max_power`                       1 2.5042e+11 8.1737e+13 133212
## - `km_driven:fuel`                            1 2.5904e+11 8.1745e+13 133213
## - `fuel:transmission`                         1 2.6201e+11 8.1748e+13 133213
## - `owner:log10_price`                         1 2.7026e+11 8.1757e+13 133214
## - `km_driven:mileage`                         1 2.7125e+11 8.1758e+13 133214
## - fuel                                        1 2.8317e+11 8.1769e+13 133215
## - `year:fuel`                                 1 2.8366e+11 8.1770e+13 133215
## - `year:transmission`                         1 2.8662e+11 8.1773e+13 133215
## - owner                                       1 3.0146e+11 8.1788e+13 133216
## - `year:owner`                                1 3.0703e+11 8.1793e+13 133216
## - `owner:max_power`                           1 3.1634e+11 8.1803e+13 133217
## - `engine:max_power`                          1 3.3225e+11 8.1819e+13 133218
## - transmission                                1 3.5589e+11 8.1842e+13 133220
## - `owner:engine`                              1 3.6865e+11 8.1855e+13 133221
## - year                                        1 4.7367e+11 8.1960e+13 133228
## - `poly(year, 2, raw = TRUE)[, 2]`            1 5.3520e+11 8.2022e+13 133232
## - name                                        1 6.9153e+11 8.2178e+13 133243
## - `name:mileage`                              1 7.0668e+11 8.2193e+13 133244
## - `km_driven:log10_price`                     1 7.6166e+11 8.2248e+13 133248
## - `seller_type:transmission`                  1 7.7877e+11 8.2265e+13 133249
## - `name:year`                                 1 7.8864e+11 8.2275e+13 133250
## - seller_type                                 1 9.1743e+11 8.2404e+13 133259
## - `year:seller_type`                          1 9.7022e+11 8.2457e+13 133262
## - `transmission:log10_price`                  1 9.7333e+11 8.2460e+13 133263
## - `seller_type:engine`                        1 1.4843e+12 8.2971e+13 133298
## - `name:fuel`                                 1 2.2401e+12 8.3726e+13 133349
## - `name:transmission`                         1 2.4887e+12 8.3975e+13 133366
## - log10_price                                 1 2.7151e+12 8.4201e+13 133382
## - `seller_type:log10_price`                   1 2.7703e+12 8.4257e+13 133385
## - `name:log10_price`                          1 3.1317e+12 8.4618e+13 133410
```

```
## - `year:log10_price`                        1 3.1380e+12 8.4624e+13 133410
## - `engine:log10_price`                      1 3.3088e+12 8.4795e+13 133422
## - `max_power:log10_price`                   1 5.3494e+12 8.6836e+13 133557
## - `year:max_power`                          1 6.8333e+12 8.8320e+13 133653
## - `poly(log10_price, 2, raw = TRUE)[, 2]`   1 1.7614e+13 9.9100e+13 134309

##
## Call:
## lm(formula = selling_price ~ name + year + km_driven + fuel +
##     seller_type + transmission + owner + mileage + engine + log10_price +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
##     `poly(log10_price, 2, raw = TRUE)[, 2]` + `name:year` + `name:fuel` +
##     `name:seller_type` + `name:transmission` + `name:owner` +
##     `name:mileage` + `name:max_power` + `name:log10_price` +
##     `year:km_driven` + `year:fuel` + `year:seller_type` +
`year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `year:log10_price` + `km_driven:fuel` + `km_driven:seller_type` +
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `km_driven:log10_price` + `fuel:seller_type` +
##     `fuel:transmission` + `fuel:mileage` + `fuel:log10_price` +
##     `seller_type:transmission` + `seller_type:mileage` +
`seller_type:engine` +
##     `seller_type:log10_price` + `transmission:owner` +
`transmission:max_power` +
##     `transmission:log10_price` + `owner:engine` + `owner:max_power` +
##     `owner:log10_price` + `mileage:engine` + `mileage:max_power` +
##     `engine:max_power` + `engine:log10_price` + `max_power:log10_price`,
##     data = train_poly)
##
## Coefficients:
##                               (Intercept)
##                                 3.847e+09
##                                      name
##                                 2.610e+06
##                                      year
##                                -4.187e+06
##                                 km_driven
##                                -1.339e+02
##                                      fuel
##                                 1.424e+07
##                               seller_type
##                                -4.297e+07
##                              transmission
##                                -3.656e+07
##                                     owner
##                                 9.727e+06
##                                   mileage
##                                 5.778e+05
```

```
##                                   engine
##                                -1.904e+04
##                               log10_price
##                                 1.489e+08
##        `poly(year, 2, raw = TRUE)[, 2]`
##                                 1.142e+03
##      `poly(engine, 2, raw = TRUE)[, 2]`
##                                -3.985e-02
## `poly(log10_price, 2, raw = TRUE)[, 2]`
##                                 1.774e+06
##                               `name:year`
##                                -1.436e+03
##                               `name:fuel`
##                                 1.182e+04
##                        `name:seller_type`
##                                 1.884e+03
##                       `name:transmission`
##                                -1.890e+04
##                             `name:owner`
##                                 9.485e+02
##                            `name:mileage`
##                                 1.002e+03
##                          `name:max_power`
##                                -6.780e+01
##                        `name:log10_price`
##                                 4.639e+04
##                          `year:km_driven`
##                                 7.322e-02
##                              `year:fuel`
##                                -7.293e+03
##                        `year:seller_type`
##                                 2.240e+04
##                       `year:transmission`
##                                 1.675e+04
##                             `year:owner`
##                                -5.029e+03
##                            `year:mileage`
##                                -2.871e+02
##                             `year:engine`
##                                 1.100e+01
##                          `year:max_power`
##                                -2.334e+01
##                        `year:log10_price`
##                                -8.277e+04
##                          `km_driven:fuel`
##                                -4.380e-01
##                 `km_driven:seller_type`
##                                 4.998e-01
##                `km_driven:transmission`
##                                 3.963e-01
```

```
##                     `km_driven:mileage`
##                              -7.601e-02
##                      `km_driven:engine`
##                              -3.827e-04
##                   `km_driven:max_power`
##                              -8.738e-03
##                 `km_driven:log10_price`
##                              -2.062e+00
##                      `fuel:seller_type`
##                               3.820e+04
##                     `fuel:transmission`
##                               7.942e+04
##                          `fuel:mileage`
##                               4.474e+03
##                       `fuel:log10_price`
##                               5.449e+04
##               `seller_type:transmission`
##                               8.337e+04
##                    `seller_type:mileage`
##                               4.747e+03
##                     `seller_type:engine`
##                               1.761e+02
##                `seller_type:log10_price`
##                              -4.487e+05
##                    `transmission:owner`
##                               1.864e+04
##                 `transmission:max_power`
##                              -5.401e+02
##               `transmission:log10_price`
##                               4.610e+05
##                          `owner:engine`
##                              -3.860e+01
##                       `owner:max_power`
##                               6.742e+02
##                    `owner:log10_price`
##                               7.071e+04
##                       `mileage:engine`
##                              -7.008e+00
##                    `mileage:max_power`
##                               6.552e+01
##                     `engine:max_power`
##                               1.171e+00
##                   `engine:log10_price`
##                              -5.701e+02
##                  `max_power:log10_price`
##                               7.765e+03
```

We save the best model as l_p, and after that we predict and calculate the metrics.

```
l_p <- lm(formula = selling_price ~ year + km_driven + fuel + seller_type +
    transmission + owner + mileage + engine + max_power + `poly(name, 2, raw
= TRUE)[, 2]` +
    `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[, 2]`
+
    `poly(max_power, 2, raw = TRUE)[, 2]` + `name:year` + `name:km_driven` +
    `name:fuel` + `name:seller_type` + `name:transmission` +
    `name:owner` + `name:engine` + `name:max_power` + `year:km_driven` +
    `year:fuel` + `year:seller_type` + `year:transmission` +
    `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
    `km_driven:transmission` + `km_driven:mileage` + `km_driven:max_power` +
    `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
    `fuel:max_power` + `seller_type:transmission` + `seller_type:owner` +
    `seller_type:mileage` + `seller_type:max_power` + `transmission:owner` +
    `transmission:engine` + `transmission:max_power` + `owner:engine` +
    `owner:max_power`, data = train_poly)
summary(l_p)

##
## Call:
## lm(formula = selling_price ~ year + km_driven + fuel + seller_type +
##     transmission + owner + mileage + engine + max_power + `poly(name, 2,
raw = TRUE)[, 2]` +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `name:year` + `name:km_driven`
+
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:owner` + `name:engine` + `name:max_power` + `year:km_driven` +
##     `year:fuel` + `year:seller_type` + `year:transmission` +
##     `year:owner` + `year:mileage` + `year:engine` + `year:max_power` +
##     `km_driven:transmission` + `km_driven:mileage` + `km_driven:max_power`
+
##     `fuel:seller_type` + `fuel:transmission` + `fuel:mileage` +
##     `fuel:max_power` + `seller_type:transmission` + `seller_type:owner` +
##     `seller_type:mileage` + `seller_type:max_power` + `transmission:owner`
+
##     `transmission:engine` + `transmission:max_power` + `owner:engine` +
##     `owner:max_power`, data = train_poly)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1857447   -81240    -2194    77208  1997530
##
## Coefficients:
##                                       Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                          2.214e+09  7.669e+08   2.887
0.003902
## year                                -2.239e+06  7.639e+05  -2.931
```

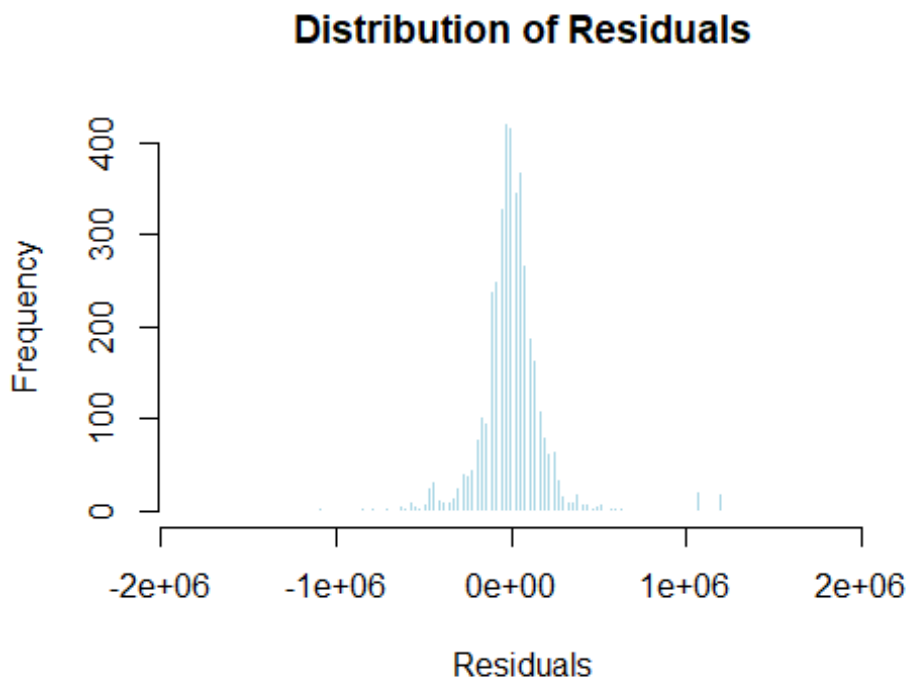
```
0.003392
## km_driven                                1.180e+02  5.057e+01   2.334
0.019629
## fuel                                     3.548e+07  3.980e+06   8.914  < 2e-
16
## seller_type                              3.876e+07  6.566e+06   5.902 3.79e-
09
## transmission                            -9.723e+07  1.014e+07  -9.590  < 2e-
16
## owner                                    1.044e+07  2.552e+06   4.089 4.39e-
05
## mileage                                  1.103e+06  6.746e+05   1.635
0.102100
## engine                                   4.329e+04  7.255e+03   5.967 2.56e-
09
## max_power                               -2.463e+06  1.057e+05 -23.302  < 2e-
16
## `poly(name, 2, raw = TRUE)[, 2]`         6.777e+02  1.170e+02   5.792 7.32e-
09
## `poly(year, 2, raw = TRUE)[, 2]`         5.653e+02  1.903e+02   2.971
0.002980
## `poly(km_driven, 2, raw = TRUE)[, 2]`    1.190e-06  7.658e-07   1.554
0.120129
## `poly(max_power, 2, raw = TRUE)[, 2]`    1.504e+01  2.705e+00   5.560 2.82e-
08
## `name:year`                              7.931e+00  2.749e+00   2.885
0.003934
## `name:km_driven`                        -7.379e-02  1.905e-02  -3.874
0.000108
## `name:fuel`                              8.227e+03  1.725e+03   4.769 1.90e-
06
## `name:seller_type`                      -1.594e+04  2.040e+03  -7.812 6.65e-
15
## `name:transmission`                      6.348e+03  2.768e+03   2.293
0.021876
## `name:owner`                             5.597e+03  1.157e+03   4.836 1.36e-
06
## `name:engine`                           -3.563e+01  3.063e+00 -11.633  < 2e-
16
## `name:max_power`                         4.949e+02  4.228e+01  11.706  < 2e-
16
## `year:km_driven`                        -5.671e-02  2.523e-02  -2.248
0.024617
## `year:fuel`                             -1.761e+04  1.984e+03  -8.878  < 2e-
16
## `year:seller_type`                      -1.873e+04  3.261e+03  -5.743 9.78e-
09
## `year:transmission`                      4.797e+04  5.018e+03   9.560  < 2e-
16
## `year:owner`                            -5.099e+03  1.267e+03  -4.025 5.76e-
```

```
05
## `year:mileage`                        -5.245e+02  3.343e+02  -1.569
0.116765
## `year:engine`                         -2.129e+01  3.603e+00  -5.908 3.66e-
09
## `year:max_power`                       1.229e+03  5.242e+01  23.443  < 2e-
16
## `km_driven:transmission`              -1.687e+00  4.068e-01  -4.148 3.41e-
05
## `km_driven:mileage`                   -8.148e-02  2.566e-02  -3.175
0.001507
## `km_driven:max_power`                 -3.068e-02  3.378e-03  -9.083  < 2e-
16
## `fuel:seller_type`                    -1.041e+05  1.808e+04  -5.757 9.04e-
09
## `fuel:transmission`                   -8.566e+04  3.349e+04  -2.558
0.010560
## `fuel:mileage`                         8.376e+03  1.958e+03   4.277 1.93e-
05
## `fuel:max_power`                      -8.773e+02  2.841e+02  -3.088
0.002024
## `seller_type:transmission`             9.943e+04  2.160e+04   4.603 4.25e-
06
## `seller_type:owner`                   -1.120e+05  2.153e+04  -5.202 2.04e-
07
## `seller_type:mileage`                 -2.197e+04  2.311e+03  -9.506  < 2e-
16
## `seller_type:max_power`               -5.327e+03  3.264e+02 -16.320  < 2e-
16
## `transmission:owner`                   1.486e+05  2.461e+04   6.038 1.66e-
09
## `transmission:engine`                 -3.746e+02  4.234e+01  -8.848  < 2e-
16
## `transmission:max_power`               1.028e+04  6.884e+02  14.930  < 2e-
16
## `owner:engine`                        -5.136e+01  1.434e+01  -3.582
0.000344
## `owner:max_power`                      9.229e+02  2.594e+02   3.558
0.000377
##
## (Intercept)                          **
## year                                 **
## km_driven                            *
## fuel                                 ***
## seller_type                          ***
## transmission                         ***
## owner                                ***
## mileage
## engine                               ***
## max_power                            ***
```

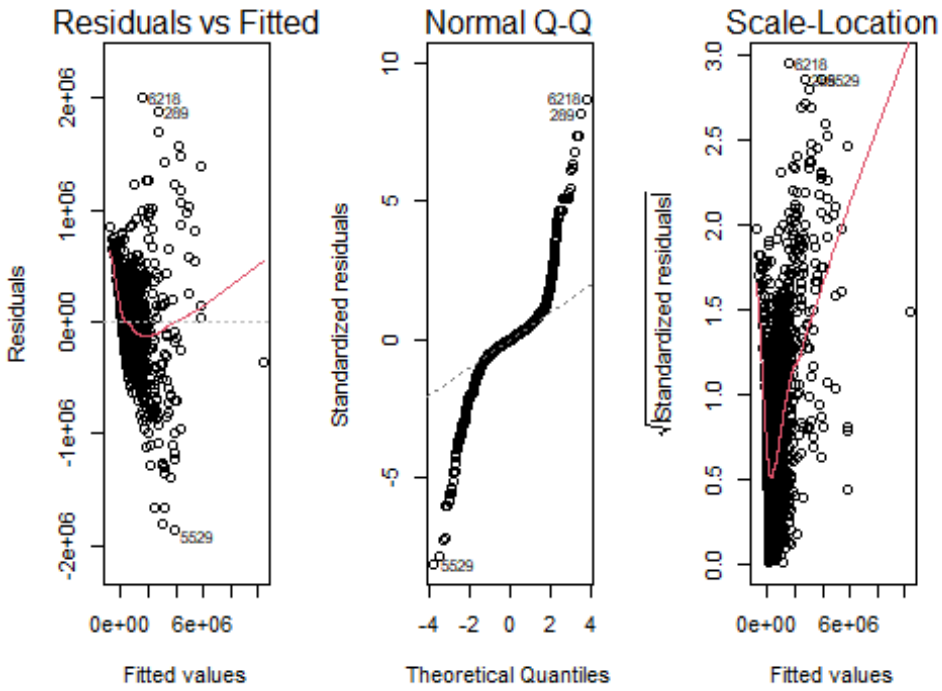
```
## `poly(name, 2, raw = TRUE)[, 2]`       ***
## `poly(year, 2, raw = TRUE)[, 2]`       **
## `poly(km_driven, 2, raw = TRUE)[, 2]`
## `poly(max_power, 2, raw = TRUE)[, 2]` ***
## `name:year`                            **
## `name:km_driven`                       ***
## `name:fuel`                            ***
## `name:seller_type`                     ***
## `name:transmission`                    *
## `name:owner`                           ***
## `name:engine`                          ***
## `name:max_power`                       ***
## `year:km_driven`                       *
## `year:fuel`                            ***
## `year:seller_type`                     ***
## `year:transmission`                    ***
## `year:owner`                           ***
## `year:mileage`
## `year:engine`                          ***
## `year:max_power`                       ***
## `km_driven:transmission`               ***
## `km_driven:mileage`                    **
## `km_driven:max_power`                  ***
## `fuel:seller_type`                     ***
## `fuel:transmission`                    *
## `fuel:mileage`                         ***
## `fuel:max_power`                       **
## `seller_type:transmission`             ***
## `seller_type:owner`                    ***
## `seller_type:mileage`                  ***
## `seller_type:max_power`                ***
## `transmission:owner`                   ***
## `transmission:engine`                  ***
## `transmission:max_power`               ***
## `owner:engine`                         ***
## `owner:max_power`                      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 232800 on 5645 degrees of freedom
## Multiple R-squared:  0.9188, Adjusted R-squared:  0.9182
## F-statistic:  1420 on 45 and 5645 DF,  p-value: < 2.2e-16

l_p_pred <- predict(l_p, test_poly)
radj <- summary(l_p)$adj.r.squared
rse <- sqrt(sum(residuals(l_p)^2) / l_p$df.residual )
rmse <- RMSE(l_p_pred, test$selling_price)
aic <- AIC(l_p)
l_p_reg <- cbind("Adjusted R sq"=radj, "RSE"=rse, "RMSE"=rmse, "AIC"=aic)
```

we have 42 features that are significant on selling price and 3 features that are not. these 3 features are mileage, km_driven * km_driven and year * mileage. Intercept has the biggest coefficient of all features, which may indicate that the model is not properly centered around the data, and may be a sign of overfitting in this model. Additionally, it may suggest that the higher-order terms in the polynomial equation are not providing significant additional explanatory power to the model.

```
residual3 <- residuals(l_p)

# Plot the histogram of residuals
hist(residual3, breaks = "FD", col = "lightblue", border = "white", main =
"Distribution of Residuals", xlab = "Residuals")
```




```
options(repr.plot.width=21, repr.plot.height=6)
par(mfrow=c(1,3))
plot(l_p, which=c(1,2,3))
```



In the Residuals x Fitted plot, again we have a non horizontal line which indicates a non-linear relationship. IN the Normal QQ plot, we see that the residuals are not exactly on the straight line, Showing that they are not normally distributed. The non straight line in the Scale-Location plot indicates heteroscedasticity.

4.9. Polynomial Regression 2

In this section, we decide to repeat the steps for polynomial regression in section 4.9. but this time we use the logarithm of selling price as target variable. We do this because of several reasons:

• The relationship between our target variable and independent variables seems to be non-linear as it discussed in previous sections. Taking the logarithm of the target variable can often help to linearize this relationship, making it easier to model using polynomial independent variables.

• In scale_location plot of our previous model, we observed heteroscedasticity. The logarithm of a variable often has the effect of stabilizing its variance, which can make it easier to model. This can be especially useful if the target variable exhibits heteroscedasticity, meaning that its variance is not constant across its range.

• The logarithm of a variable can also help to mitigate the impact of outliers, which can have a disproportionately large effect on the predicted values if the target variable is modeled directly.

```
y_train <- log(train$selling_price)
y_test <- log(test$selling_price)
```

```r
X_train <- train %>%
  select(-c(selling_price, seats, log_selling_price))
X_test <- test %>%
  select(-c(selling_price, seats))

formula <- as.formula(
  paste(' ~ .^2 + ', paste('poly(', colnames(X_train), ', 2, raw=TRUE)[, 2]',
collapse = ' + '))
)
formula
```

```
## ~.^2 + poly(name, 2, raw = TRUE)[, 2] + poly(year, 2, raw = TRUE)[,
##     2] + poly(km_driven, 2, raw = TRUE)[, 2] + poly(fuel, 2,
##     raw = TRUE)[, 2] + poly(seller_type, 2, raw = TRUE)[, 2] +
##     poly(transmission, 2, raw = TRUE)[, 2] + poly(owner, 2, raw = TRUE)[,
##     2] + poly(mileage, 2, raw = TRUE)[, 2] + poly(engine, 2,
##     raw = TRUE)[, 2] + poly(max_power, 2, raw = TRUE)[, 2] +
##     poly(log10_price, 2, raw = TRUE)[, 2]
```

```r
train_poly <- as.data.frame(model.matrix(formula, data = X_train))
test_poly <- as.data.frame(model.matrix(formula, data = X_test))
train_poly$lg_selling_price <- y_train
test_poly$lg_selling_price <- y_test
colnames(train_poly)
```

```
##  [1] "(Intercept)"
##  [2] "name"
##  [3] "year"
##  [4] "km_driven"
##  [5] "fuel"
##  [6] "seller_type"
##  [7] "transmission"
##  [8] "owner"
##  [9] "mileage"
## [10] "engine"
## [11] "max_power"
## [12] "log10_price"
## [13] "poly(name, 2, raw = TRUE)[, 2]"
## [14] "poly(year, 2, raw = TRUE)[, 2]"
## [15] "poly(km_driven, 2, raw = TRUE)[, 2]"
## [16] "poly(fuel, 2, raw = TRUE)[, 2]"
## [17] "poly(seller_type, 2, raw = TRUE)[, 2]"
## [18] "poly(transmission, 2, raw = TRUE)[, 2]"
## [19] "poly(owner, 2, raw = TRUE)[, 2]"
## [20] "poly(mileage, 2, raw = TRUE)[, 2]"
## [21] "poly(engine, 2, raw = TRUE)[, 2]"
## [22] "poly(max_power, 2, raw = TRUE)[, 2]"
## [23] "poly(log10_price, 2, raw = TRUE)[, 2]"
## [24] "name:year"
## [25] "name:km_driven"
## [26] "name:fuel"
```

```
## [27] "name:seller_type"
## [28] "name:transmission"
## [29] "name:owner"
## [30] "name:mileage"
## [31] "name:engine"
## [32] "name:max_power"
## [33] "name:log10_price"
## [34] "year:km_driven"
## [35] "year:fuel"
## [36] "year:seller_type"
## [37] "year:transmission"
## [38] "year:owner"
## [39] "year:mileage"
## [40] "year:engine"
## [41] "year:max_power"
## [42] "year:log10_price"
## [43] "km_driven:fuel"
## [44] "km_driven:seller_type"
## [45] "km_driven:transmission"
## [46] "km_driven:owner"
## [47] "km_driven:mileage"
## [48] "km_driven:engine"
## [49] "km_driven:max_power"
## [50] "km_driven:log10_price"
## [51] "fuel:seller_type"
## [52] "fuel:transmission"
## [53] "fuel:owner"
## [54] "fuel:mileage"
## [55] "fuel:engine"
## [56] "fuel:max_power"
## [57] "fuel:log10_price"
## [58] "seller_type:transmission"
## [59] "seller_type:owner"
## [60] "seller_type:mileage"
## [61] "seller_type:engine"
## [62] "seller_type:max_power"
## [63] "seller_type:log10_price"
## [64] "transmission:owner"
## [65] "transmission:mileage"
## [66] "transmission:engine"
## [67] "transmission:max_power"
## [68] "transmission:log10_price"
## [69] "owner:mileage"
## [70] "owner:engine"
## [71] "owner:max_power"
## [72] "owner:log10_price"
## [73] "mileage:engine"
## [74] "mileage:max_power"
## [75] "mileage:log10_price"
## [76] "engine:max_power"
```

```
## [77] "engine:log10_price"
## [78] "max_power:log10_price"
## [79] "lg_selling_price"
```

Our datasets train_poly and test_poly have 133 columns.

Since the output of below code contain comparing so many models, we did not put the output in this document and the best of these models is shown in the next step.

```
temp <- lm(formula = lg_selling_price ~ ., data = train_poly)
step(temp)
```

Save the best model as l_p_log, then predict. After that, calculate the metrics.

```
l_p_log <- lm(formula = lg_selling_price ~ name + year + km_driven +
seller_type + transmission +
    owner + engine + max_power + `poly(name, 2, raw = TRUE)[, 2]` +
    `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[, 2]`
+
    `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw = TRUE)[, 2]` +
    `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
    `poly(max_power, 2, raw = TRUE)[, 2]` + `name:year` + `name:km_driven` +
    `name:fuel` + `name:seller_type` + `name:transmission` +
    `name:engine` + `name:max_power` + `year:seller_type` +
`year:transmission` +
    `year:owner` + `year:mileage` + `year:max_power` +
`km_driven:seller_type` +
    `km_driven:owner` + `km_driven:mileage` + `km_driven:engine` +
    `km_driven:max_power` + `fuel:seller_type` + `fuel:mileage` +
    `fuel:engine` + `seller_type:owner` + `seller_type:mileage` +
    `seller_type:engine` + `seller_type:max_power` + `transmission:owner` +
    `transmission:mileage` + `transmission:max_power` + `mileage:engine` +
    `mileage:max_power` + `engine:max_power`, data = train_poly)
summary(l_p_log)

##
## Call:
## lm(formula = lg_selling_price ~ name + year + km_driven + seller_type +
##     transmission + owner + engine + max_power + `poly(name, 2, raw =
TRUE)[, 2]` +
##     `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[,
2]` +
##     `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw = TRUE)[, 2]` +
##     `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[,
2]` +
##     `poly(max_power, 2, raw = TRUE)[, 2]` + `name:year` + `name:km_driven`
+
##     `name:fuel` + `name:seller_type` + `name:transmission` +
##     `name:engine` + `name:max_power` + `year:seller_type` +
`year:transmission` +
```

```
##     `year:owner` + `year:mileage` + `year:max_power` +
`km_driven:seller_type` +
##     `km_driven:owner` + `km_driven:mileage` + `km_driven:engine` +
##     `km_driven:max_power` + `fuel:seller_type` + `fuel:mileage` +
##     `fuel:engine` + `seller_type:owner` + `seller_type:mileage` +
##     `seller_type:engine` + `seller_type:max_power` + `transmission:owner`
+
##     `transmission:mileage` + `transmission:max_power` + `mileage:engine` +
##     `mileage:max_power` + `engine:max_power`, data = train_poly)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47833 -0.14068  0.01663  0.15690  1.68031
##
## Coefficients:
##                                         Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                            4.039e+03  7.622e+02   5.300 1.20e-
07
## name                                   1.133e+00  4.819e-01   2.351
0.018765
## year                                  -4.108e+00  7.570e-01  -5.426 5.99e-
08
## km_driven                             -2.447e-06  1.155e-06  -2.118
0.034228
## seller_type                            2.301e+01  8.546e+00   2.692
0.007119
## transmission                           4.226e+01  9.532e+00   4.433 9.45e-
06
## owner                                 -7.478e+00  3.066e+00  -2.439
0.014749
## engine                                 7.749e-04  1.653e-04   4.688 2.83e-
06
## max_power                             -8.107e-01  8.475e-02  -9.565  < 2e-
16
## `poly(name, 2, raw = TRUE)[, 2]`       2.028e-03  1.348e-04  15.042  < 2e-
16
## `poly(year, 2, raw = TRUE)[, 2]`       1.046e-03  1.880e-04   5.566 2.73e-
08
## `poly(km_driven, 2, raw = TRUE)[, 2]`  3.349e-12  8.679e-13   3.859
0.000115
## `poly(fuel, 2, raw = TRUE)[, 2]`       1.204e-01  9.423e-03  12.780  < 2e-
16
## `poly(owner, 2, raw = TRUE)[, 2]`      1.232e-02  5.738e-03   2.147
0.031800
## `poly(mileage, 2, raw = TRUE)[, 2]`   -7.186e-04  1.996e-04  -3.601
0.000320
## `poly(engine, 2, raw = TRUE)[, 2]`     6.938e-08  3.362e-08   2.064
0.039110
## `poly(max_power, 2, raw = TRUE)[, 2]` -1.505e-05  4.285e-06  -3.511
```

```
0.000450
## `name:year`                          -5.804e-04  2.386e-04  -2.432
0.015041
## `name:km_driven`                     -6.808e-08  2.330e-08  -2.922
0.003496
## `name:fuel`                           1.059e-02  1.924e-03   5.505 3.84e-
08
## `name:seller_type`                   -7.173e-03  2.321e-03  -3.090
0.002011
## `name:transmission`                   1.210e-02  2.976e-03   4.065 4.87e-
05
## `name:engine`                        -2.027e-05  3.571e-06  -5.676 1.45e-
08
## `name:max_power`                      2.591e-04  4.848e-05   5.345 9.38e-
08
## `year:seller_type`                   -1.122e-02  4.244e-03  -2.645
0.008200
## `year:transmission`                  -2.124e-02  4.740e-03  -4.480 7.60e-
06
## `year:owner`                          3.728e-03  1.517e-03   2.457
0.014037
## `year:mileage`                        2.950e-05  6.516e-06   4.527 6.11e-
06
## `year:max_power`                      4.125e-04  4.209e-05   9.801  < 2e-
16
## `km_driven:seller_type`              -9.598e-07  3.642e-07  -2.635
0.008438
## `km_driven:owner`                     3.029e-07  1.162e-07   2.607
0.009168
## `km_driven:mileage`                   9.566e-08  3.020e-08   3.167
0.001547
## `km_driven:engine`                    1.177e-09  2.482e-10   4.744 2.15e-
06
## `km_driven:max_power`                -6.971e-09  3.739e-09  -1.865
0.062287
## `fuel:seller_type`                   -8.917e-02  1.828e-02  -4.878 1.10e-
06
## `fuel:mileage`                       -8.591e-03  1.581e-03  -5.436 5.69e-
08
## `fuel:engine`                        -5.560e-05  1.948e-05  -2.854
0.004338
## `seller_type:owner`                  -6.486e-02  2.422e-02  -2.678
0.007423
## `seller_type:mileage`                -6.205e-03  3.141e-03  -1.975
0.048294
## `seller_type:engine`                 -6.301e-05  3.362e-05  -1.874
0.060973
## `seller_type:max_power`              -8.068e-04  4.538e-04  -1.778
0.075496
## `transmission:owner`                  6.113e-02  2.404e-02   2.543
```

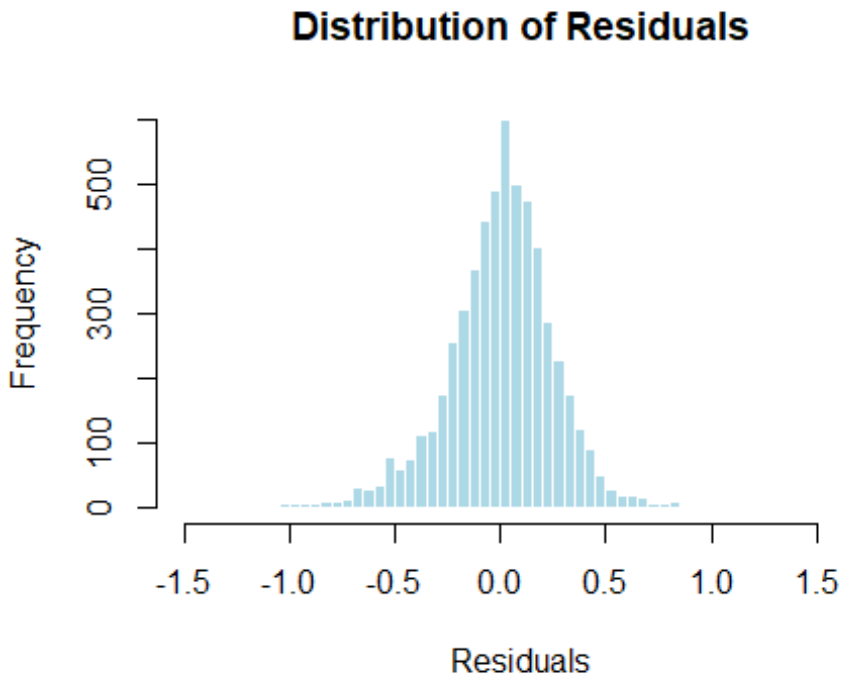
```
0.011029
## `transmission:mileage`                1.462e-02  4.702e-03   3.110
0.001879
## `transmission:max_power`              2.837e-03  5.667e-04   5.006 5.72e-
07
## `mileage:engine`                     -1.437e-05  5.065e-06  -2.837
0.004577
## `mileage:max_power`                  -8.141e-05  4.650e-05  -1.751
0.080034
## `engine:max_power`                   -3.441e-06  4.999e-07  -6.884 6.47e-
12
##
## (Intercept)                         ***
## name                                *
## year                                ***
## km_driven                           *
## seller_type                         **
## transmission                        ***
## owner                               *
## engine                              ***
## max_power                           ***
## `poly(name, 2, raw = TRUE)[, 2]`    ***
## `poly(year, 2, raw = TRUE)[, 2]`    ***
## `poly(km_driven, 2, raw = TRUE)[, 2]` ***
## `poly(fuel, 2, raw = TRUE)[, 2]`    ***
## `poly(owner, 2, raw = TRUE)[, 2]`   *
## `poly(mileage, 2, raw = TRUE)[, 2]` ***
## `poly(engine, 2, raw = TRUE)[, 2]`  *
## `poly(max_power, 2, raw = TRUE)[, 2]` ***
## `name:year`                         *
## `name:km_driven`                    **
## `name:fuel`                         ***
## `name:seller_type`                  **
## `name:transmission`                 ***
## `name:engine`                       ***
## `name:max_power`                    ***
## `year:seller_type`                  **
## `year:transmission`                 ***
## `year:owner`                        *
## `year:mileage`                      ***
## `year:max_power`                    ***
## `km_driven:seller_type`             **
## `km_driven:owner`                   **
## `km_driven:mileage`                 **
## `km_driven:engine`                  ***
## `km_driven:max_power`               .
## `fuel:seller_type`                  ***
## `fuel:mileage`                      ***
## `fuel:engine`                       **
## `seller_type:owner`                 **
```

```
## `seller_type:mileage`                  *
## `seller_type:engine`                   .
## `seller_type:max_power`                .
## `transmission:owner`                   *
## `transmission:mileage`                 **
## `transmission:max_power`               ***
## `mileage:engine`                       **
## `mileage:max_power`                    .
## `engine:max_power`                     ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2589 on 5644 degrees of freedom
## Multiple R-squared:  0.9048, Adjusted R-squared:  0.904
## F-statistic:  1166 on 46 and 5644 DF,  p-value: < 2.2e-16

l_p_pred_log <- predict(l_p_log, test_poly)
radj <- summary(l_p_log)$adj.r.squared
rse <- sqrt(sum(residuals(l_p_log)^2) / l_p_log$df.residual )
rmse <- RMSE(l_p_pred_log, log(test$selling_price))
aic <- AIC(l_p_log)
l_p_reg_log <- cbind("Adjusted R sq"=radj, "RSE"=rse, "RMSE"=rmse, "AIC"=aic)
```

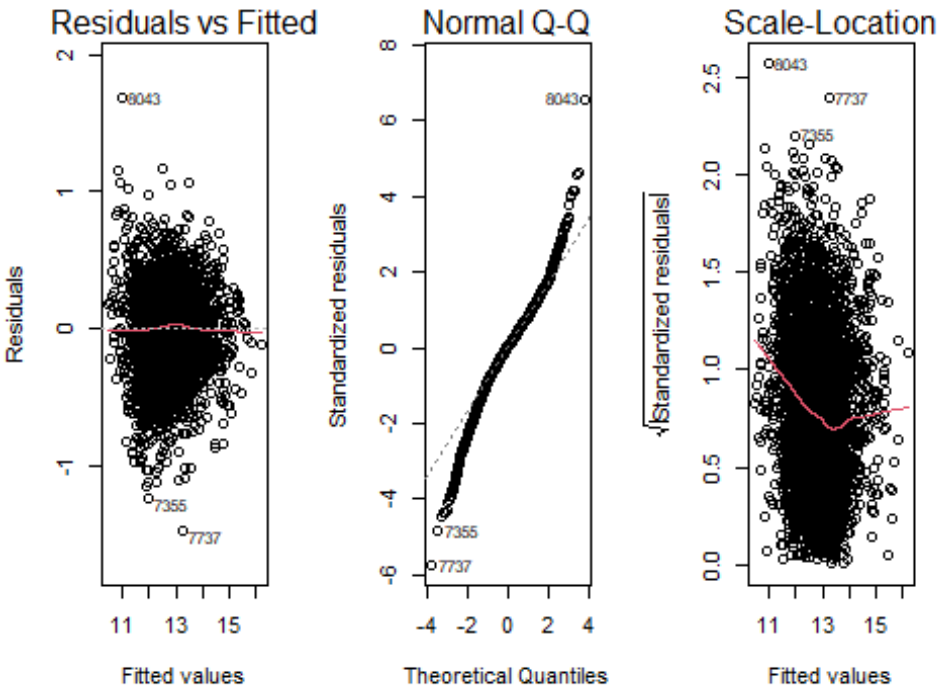
We have 46 features, all of which are significant on logarithm of selling price, except for km_driven * max_power, seller_type * engine, seller_type * max_power and mileage * max_power. Intercept has the biggest coefficient of all features, that means a unit change in intercept gives a bigger change in selling price than a unit change in other features give, given all other features are fixed. In this case, considering all other features to be zero, selling price will be equal to 4039$.

Now we plot our model to see the results.

```
residual4 <- residuals(l_p_log)

# Plot the histogram of residuals
hist(residual4, breaks = "FD", col = "lightblue", border = "white", main =
"Distribution of Residuals", xlab = "Residuals")
```




```
options(repr.plot.width=21, repr.plot.height=6)
par(mfrow=c(1,3))
plot(l_p_log, which=c(1,2,3))
```



The improved results can be seen in the plots. The red line in the Residuals x Fitted plot is almost parallel to x axis which shows a linear relationship. The red line in the Scale-Location plot is also more flat in compare to previous plots. This can be interpret that the variance of the residuals is more constant across all levels of the predictor variable. In the Normal QQ plot, more points are on the diagonal line which can indicate a normal distribution for residuals.

4.10. Models Evaluation

Now we are going to compare the metrics between all the implemented models.

```
result <- rbind(m1_lr_reg, m2_lr_reg,log_lr_reg, l_p_reg, l_p_reg_log)
rownames(result) <- c("Linear Regression 1", "Linear Regression 2", "Log
Linear Regression", "Polynomial Regression", "Polynomial Regression 2")
result

##                          Adjusted R sq          RSE         RMSE
AIC
## Linear Regression 1         0.6954480 4.491287e+05 4.590137e+05
164301.8136
## Linear Regression 2         0.6954036 4.491614e+05 4.592924e+05
164299.6481
## Log Linear Regression       0.8617610 3.106861e-01 3.249474e-01
2856.1085
## Polynomial Regression       0.9181999 2.327647e+05 2.877652e+05
156854.4604
## Polynomial Regression 2     0.9039901 2.589193e-01 3.997758e-01
819.3611
```

Linear Regression 1 and Linear Regression 2: Adjusted R squared of these models are about 0.6954, which means that 69.54% of the variation in selling price can be explained by the independent variables we took in consideration. They might be good models, but we should also consider the other metrics used to compare their complexity with how well models fits the data.

Log Linear Regression: This model has a very good Adjusted R squared and AIC values compared to our linear regression models

Polynomial Regression: This model is the best in case of Adjusted R squared and better in case of RMSE which means model has a better fit and is better at making predictions.

Polynomial Regression 2: The metrics indicate that this can the best model. We used a log transform for this model so the scale of its RMSE is different. The model was very complex so maybe it is susceptible to over fitting. It has a very good Adjusted R squared value which is almost near 1 and the lowest AIC value which can be interpret as the best fitting model.

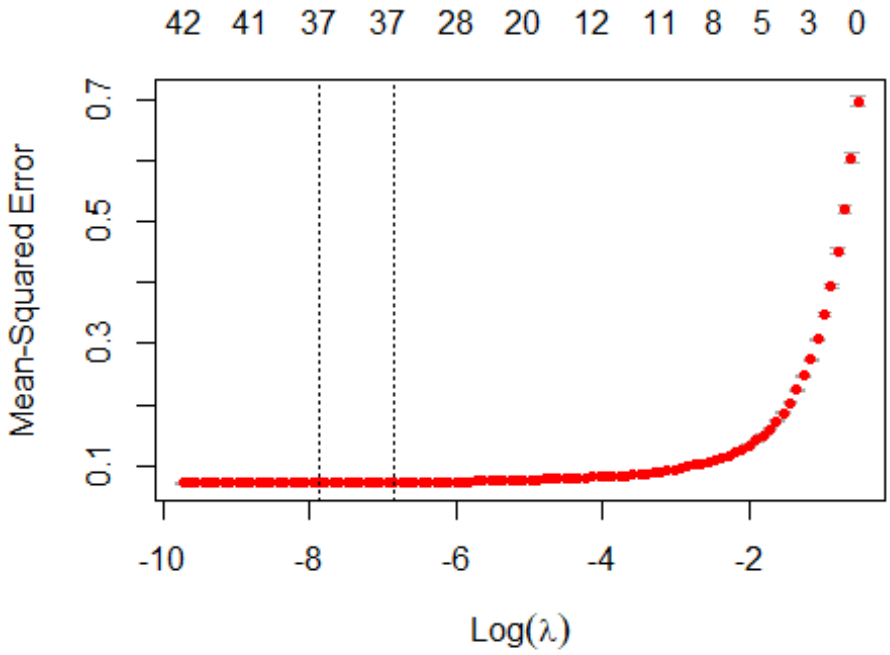
4.11. Lasso Model

Our second polynomial regression model seems to be one of our best model we obtained but since it is a complex model that has many features, in the next step, we are going to

introduce a lasso model which its response variable and its features are the same as our second Polynomial Regression model. We used the cross validation to determine the optimal lambda.

```
# Create the design matrix for the training set
X <- model.matrix(lg_selling_price ~ name + year + km_driven + seller_type +
transmission +
    owner + engine + max_power + `poly(name, 2, raw = TRUE)[, 2]` +
    `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[, 2]`
+
    `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw = TRUE)[, 2]` +
    `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
    `poly(max_power, 2, raw = TRUE)[, 2]` + `name:year` + `name:km_driven` +
    `name:fuel` + `name:seller_type` + `name:transmission` +
    `name:engine` + `name:max_power` + `year:seller_type` +
`year:transmission` +
    `year:owner` + `year:mileage` + `year:max_power` +
`km_driven:seller_type` +
    `km_driven:owner` + `km_driven:mileage` + `km_driven:engine` +
    `km_driven:max_power` + `fuel:seller_type` + `fuel:mileage` +
    `fuel:engine` + `seller_type:owner` + `seller_type:mileage` +
    `seller_type:engine` + `seller_type:max_power` + `transmission:owner` +
    `transmission:mileage` + `transmission:max_power` + `mileage:engine` +
    `mileage:max_power` + `engine:max_power`, data = train_poly)

# Remove the first column relative to the intercept
X <- X[, -1]

# Vector of responses for the training set
y <- train_poly$lg_selling_price

# Create the design matrix for the test set
X_test <- model.matrix(lg_selling_price ~ name + year + km_driven +
seller_type + transmission +
    owner + engine + max_power + `poly(name, 2, raw = TRUE)[, 2]` +
    `poly(year, 2, raw = TRUE)[, 2]` + `poly(km_driven, 2, raw = TRUE)[, 2]`
+
    `poly(fuel, 2, raw = TRUE)[, 2]` + `poly(owner, 2, raw = TRUE)[, 2]` +
    `poly(mileage, 2, raw = TRUE)[, 2]` + `poly(engine, 2, raw = TRUE)[, 2]`
+
    `poly(max_power, 2, raw = TRUE)[, 2]` + `name:year` + `name:km_driven` +
    `name:fuel` + `name:seller_type` + `name:transmission` +
    `name:engine` + `name:max_power` + `year:seller_type` +
`year:transmission` +
    `year:owner` + `year:mileage` + `year:max_power` +
`km_driven:seller_type` +
    `km_driven:owner` + `km_driven:mileage` + `km_driven:engine` +
    `km_driven:max_power` + `fuel:seller_type` + `fuel:mileage` +
    `fuel:engine` + `seller_type:owner` + `seller_type:mileage` +
```

```
    `seller_type:engine` + `seller_type:max_power` + `transmission:owner` +
    `transmission:mileage` + `transmission:max_power` + `mileage:engine` +
    `mileage:max_power` + `engine:max_power`, data = test_poly)

# Remove the first column relative to the intercept
X_test <- X_test[, -1]

# Vector of responses for the test set
y_test <- test_poly$lg_selling_price

# Apply lasso to the training set
lasso.mod <- glmnet(X, y, alpha = 1)

# Determine the optimal lambda value using cross-validation
cv.out.lasso <- cv.glmnet(X, y, alpha = 1)
plot(cv.out.lasso)
```




```
# Get the best lambda value
bestlam <- cv.out.lasso$lambda.min

# Compute the predictions on the test set
lasso.pred <- predict(lasso.mod, s = bestlam, newx = X_test)

lasso.rmse <- RMSE((lasso.pred), (y_test))
lasso.rmse
```

```
## [1] 0.427447
```

The amount of RMSE for our lasso model is a bit higher than our second model (almost the same). It suggests that the Lasso model is not performing better than the second polynomial model in terms of predictive accuracy. Lasso regularization can shrink coefficients to zero, effectively performing feature selection and reducing model complexity. We want to see how much restrictive is our Lasso model in selecting features or controlling the coefficient values. Then, through the following code, we extract the coefficient estimates of the Lasso model based on the optimal lambda.The aim is to see how many variables does the lasso model considered to be zero.

```
lasso.coef <- predict(lasso.mod,type="coefficients",s=bestlam)
lasso.coef
```

```
## 47 x 1 sparse Matrix of class "dgCMatrix"
##                                                      s1
## (Intercept)                                -2.077489e+02
## name                                       -3.464620e-02
## year                                        1.052993e-01
## km_driven                                  -1.045429e-07
## seller_type                                 1.690686e-01
## transmission                                .
## owner                                       .
## engine                                      3.703098e-04
## max_power                                   6.647270e-03
## `poly(name, 2, raw = TRUE)[, 2]`            2.082710e-03
## `poly(year, 2, raw = TRUE)[, 2]`            1.758682e-06
## `poly(km_driven, 2, raw = TRUE)[, 2]`       3.684102e-12
## `poly(fuel, 2, raw = TRUE)[, 2]`            1.026540e-01
## `poly(owner, 2, raw = TRUE)[, 2]`           1.303347e-02
## `poly(mileage, 2, raw = TRUE)[, 2]`        -2.374764e-04
## `poly(engine, 2, raw = TRUE)[, 2]`          1.178987e-08
## `poly(max_power, 2, raw = TRUE)[, 2]`      -1.267247e-05
## `name:year`                                 .
## `name:km_driven`                           -6.745005e-08
## `name:fuel`                                 9.161378e-03
## `name:seller_type`                         -6.468074e-03
## `name:transmission`                         1.439145e-02
## `name:engine`                              -1.564323e-05
## `name:max_power`                            1.565065e-04
## `year:seller_type`                          .
## `year:transmission`                        -6.618869e-05
## `year:owner`                                .
## `year:mileage`                              1.017017e-06
## `year:max_power`                            4.459580e-06
## `km_driven:seller_type`                    -4.229226e-07
## `km_driven:owner`                           1.215062e-07
## `km_driven:mileage`                         .
## `km_driven:engine`                          9.432413e-10
## `km_driven:max_power`                      -2.212117e-08
```

```
## `fuel:seller_type`                    -5.656841e-02
## `fuel:mileage`                        -6.929705e-03
## `fuel:engine`                         -1.040327e-04
## `seller_type:owner`                   -4.976914e-02
## `seller_type:mileage`                  .
## `seller_type:engine`                   .
## `seller_type:max_power`               -1.378377e-03
## `transmission:owner`                   8.059285e-02
## `transmission:mileage`                 .
## `transmission:max_power`               1.584422e-03
## `mileage:engine`                       2.956839e-06
## `mileage:max_power`                    3.515478e-05
## `engine:max_power`                    -1.023301e-06
```

```
lasso.coef[lasso.coef!=0]
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```
##  [1] -2.077489e+02 -3.464620e-02  1.052993e-01 -1.045429e-07  1.690686e-01
##  [6]  3.703098e-04  6.647270e-03  2.082710e-03  1.758682e-06  3.684102e-12
## [11]  1.026540e-01  1.303347e-02 -2.374764e-04  1.178987e-08 -1.267247e-05
## [16] -6.745005e-08  9.161378e-03 -6.468074e-03  1.439145e-02 -1.564323e-05
## [21]  1.565065e-04 -6.618869e-05  1.017017e-06  4.459580e-06 -4.229226e-07
## [26]  1.215062e-07  9.432413e-10 -2.212117e-08 -5.656841e-02 -6.929705e-03
## [31] -1.040327e-04 -4.976914e-02 -1.378377e-03  8.059285e-02  1.584422e-03
## [36]  2.956839e-06  3.515478e-05 -1.023301e-06
```

Our lasso model has considered the coefficient of only 4 variables out of our total 47 variable equal to zero. These variables are owner, name:year, year:seller_type and year:owner.

5. Conclusion

In This section, we express the result obtained during analyzing and modeling data. The number of seats seems to have no or very low correlation with price, saying that it is not that it is not an important factor in deciding the price of the car. The rest of the factors had a relative correlation with the price which among them, Max Power (the amount of power that a car's engine generates to move it) had the highest correlation value. There was not that much high positive or negative correlation among features themselves suggesting that features do not have strong impact on each other. The impact of each feature on price seems to have a logical and reasonable trend. By fitting models to our data, we observed that the most important thing to have better results is applying a log transformation to our target variable price. Our log linear model without the features fuel, owner, and seats was a good model in case of Adjusted R2 and AIC and our polynomial regression model which considered the log of price had even a bit better results in case of these metrics too. Furthermore, since these models have a high Adjusted R2, we have an indicator of an accurate prediction for future values. Since the polynomial regression model was complex,

we used a lasso model to consider a regularization. The result of lasso model has not changed that much and only a few coefficient were considered to be zero.