Car Sales Project Group 11

Rahul Poojari Samyukta Nacham Maneesh Kotha

2023-04-28

LIBRARIES NEEDED FOR THE PROJECT

```
library(ggplot2)
library(tidyverse)
library(dplyr)
library(tigerstats)
library(MASS)
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.2.3
```

```
library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.2.3
```

```
library(reticulate)
library(scales)
```

UPLOADNG CSV

```
CarSalesData = read.csv("/Users/rahul/Downloads/Car_Dataset/car_sales_data.csv")
```

DATA PREPERATION

```
dim(CarSalesData)
```

```
## [1] 7770   12
```

> The Dataset CarSales contains 8128 observations and 12 features.

```
str(CarSalesData)
```

```
## 'data.frame':    7770 obs. of  12 variables:
##  $ name         : chr  "Maruti Swift Dzire VDI" "Skoda Rapid 1.5 TDI Ambition" "Honda City 20
17-2020 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" ...
##  $ seats        : num  5 5 5 5 5 5 5 4 5 5 ...
```

The Dataset CarSalesData contains:-

1) Four “Integer” Data types
2) Eight “Character” Data Types.

We can see that in the mileage, max power and engine attributes, the data stored is unnecessarily stored in the form of characters while might not let us study the trends in the engine capacities and mileages. So lets convert them into integer types.

After pre proccesing the data, we again read the new csv file.

```
CarSales = read.csv("/Users/rahul/Downloads/Car_Dataset/CarDetails.csv")
```

```
dim(CarSales)
```

```
## [1] 7770   12
```

```
str(CarSales)
```

```
## 'data.frame':    7770 obs. of  12 variables:
##  $ name         : chr  "Maruti Swift Dzire VDI" "Skoda Rapid 1.5 TDI Ambition" "Honda City 20
17-2020 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" ...
##  $ engine       : num  1248 1498 1497 1396 1298 ...
##  $ max_power    : num  74 103.5 78 90 88.2 ...
##  $ seats        : num  5 5 5 5 5 5 5 4 5 5 ...
##  $ mileage      : num  23.4 21.1 17.7 23 16.1 ...
```

Now the Dataset CarSales contains:-

1) Seven “Integer” or “Number” Data types
2) Five “Character” Data Types.

```
summary(CarSales)
```

```
##      name                year      selling_price       km_driven
##  Length:7770        Min.   :1983   Min.   :  29999   Min.   :      1
##  Class :character   1st Qu.:2011   1st Qu.: 250000   1st Qu.:  36000
##  Mode  :character   Median :2014   Median : 430000   Median :  65000
##                     Mean   :2014   Mean   : 490945   Mean   :  71961
##                     3rd Qu.:2017   3rd Qu.: 650000   3rd Qu.: 100000
##                     Max.   :2020   Max.   :2000000   Max.   :2360457
##
##      fuel           seller_type        transmission          owner
##  Length:7770        Length:7770        Length:7770        Length:7770
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      engine       max_power          seats           mileage
##  Min.   : 624   Min.   :  0.00   Min.   : 2.000   Min.   : 0.00
##  1st Qu.:1197   1st Qu.: 68.00   1st Qu.: 5.000   1st Qu.:16.50
##  Median :1248   Median : 81.86   Median : 5.000   Median :19.30
##  Mean   :1424   Mean   : 86.93   Mean   : 5.427   Mean   :18.98
##  3rd Qu.:1498   3rd Qu.: 99.00   3rd Qu.: 5.000   3rd Qu.:22.32
##  Max.   :3498   Max.   :272.00   Max.   :14.000   Max.   :33.44
##  NA's   :219    NA's   :214      NA's   :219
```

This gives the summary of the CarSales Dataset.

Understanding the Attributes

1 - Car Brand Name
2 - Year of Manufacture
3 - Selling Price
4 - Km Driven
5 - Fuel type : Diesel, Petrol, LPG or CNG
6 - Seller type : Individual or a Dealer
7 - Transmission : Manual or Automatic
8 - Owner : Is it the first, second or third owner
9 - Engine Capacity in CC
10 - Max Power in bhp
11 - Number of Seats in the car
12 - Mileage : in kmpl for petrol and diesel adn in km/kg for LPG and CNG

```
head(CarSales)
```

```
##                           name year selling_price km_driven   fuel seller_type
## 1        Maruti Swift Dzire VDI 2014        450000    145500 Diesel  Individual
## 2  Skoda Rapid 1.5 TDI Ambition 2014        370000    120000 Diesel  Individual
## 3      Honda City 2017-2020 EXi 2006        158000    140000 Petrol  Individual
## 4     Hyundai i20 Sportz Diesel 2010        225000    127000 Diesel  Individual
## 5        Maruti Swift VXI BSIII 2007        130000    120000 Petrol  Individual
## 6 Hyundai Xcent 1.2 VTVT E Plus 2017        440000     45000 Petrol  Individual
##   transmission        owner engine max_power seats mileage
## 1       Manual  First Owner   1248     74.00     5   23.40
## 2       Manual Second Owner   1498    103.52     5   21.14
## 3       Manual  Third Owner   1497     78.00     5   17.70
## 4       Manual  First Owner   1396     90.00     5   23.00
## 5       Manual  First Owner   1298     88.20     5   16.10
## 6       Manual  First Owner   1197     81.86     5   20.14
```

This gives the First Six roxs of the CarSales Dataset.

```
names(CarSales)
```

```
##  [1] "name"          "year"          "selling_price" "km_driven"
##  [5] "fuel"          "seller_type"   "transmission"  "owner"
##  [9] "engine"        "max_power"     "seats"         "mileage"
```

```
sum(is.na(CarSales$engine))
```

```
## [1] 219
```

We see that we have 221 rows with null values in engine and other attributes

```
CarSales = CarSales[complete.cases(CarSales), ]
```

```
dim(CarSales)
```

```
## [1] 7550   12
```

Rechecking the null values

```
sum(is.na(CarSales))
```

```
## [1] 0
```

Now we have 0 null values in the dataset

Checking for any duplicate values in the dataset

```
sum(duplicated(CarSales))
```

```
## [1] 940
```

We can use unique function to remove the duplicated rows

```
CarSales = unique(CarSales)
```

Rechecking the dimension of the datset and if any duplicated values

```
dim(CarSales)
```

```
## [1] 6610   12
```

```
sum(duplicated(CarSales))
```

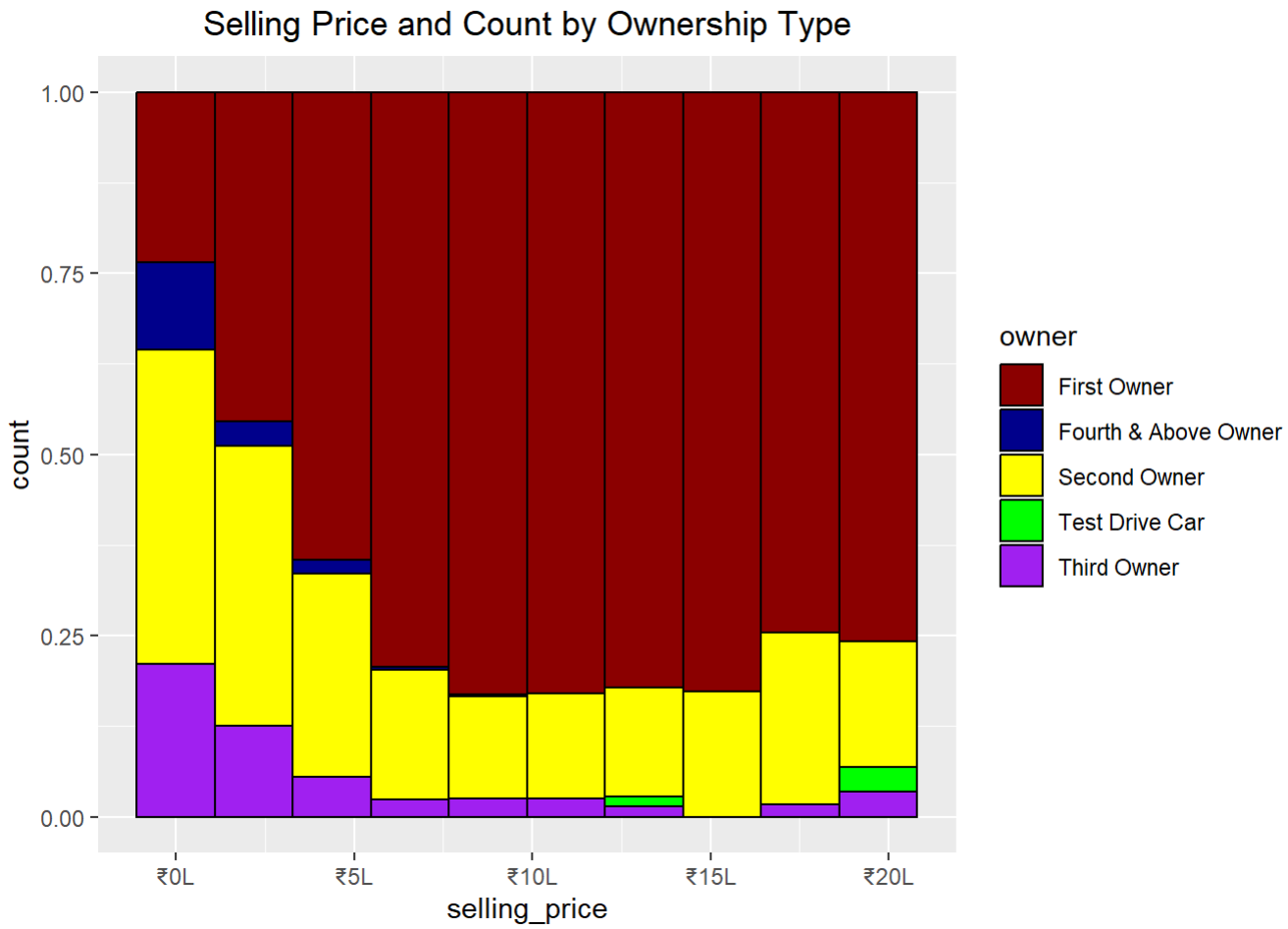
```
## [1] 0
```

> Finally after all the data preprocessing by datatype tranformations and data cleaning, Our dataset is ready to proceed with EDA

EXPLORATORY DATA ANALYSIS

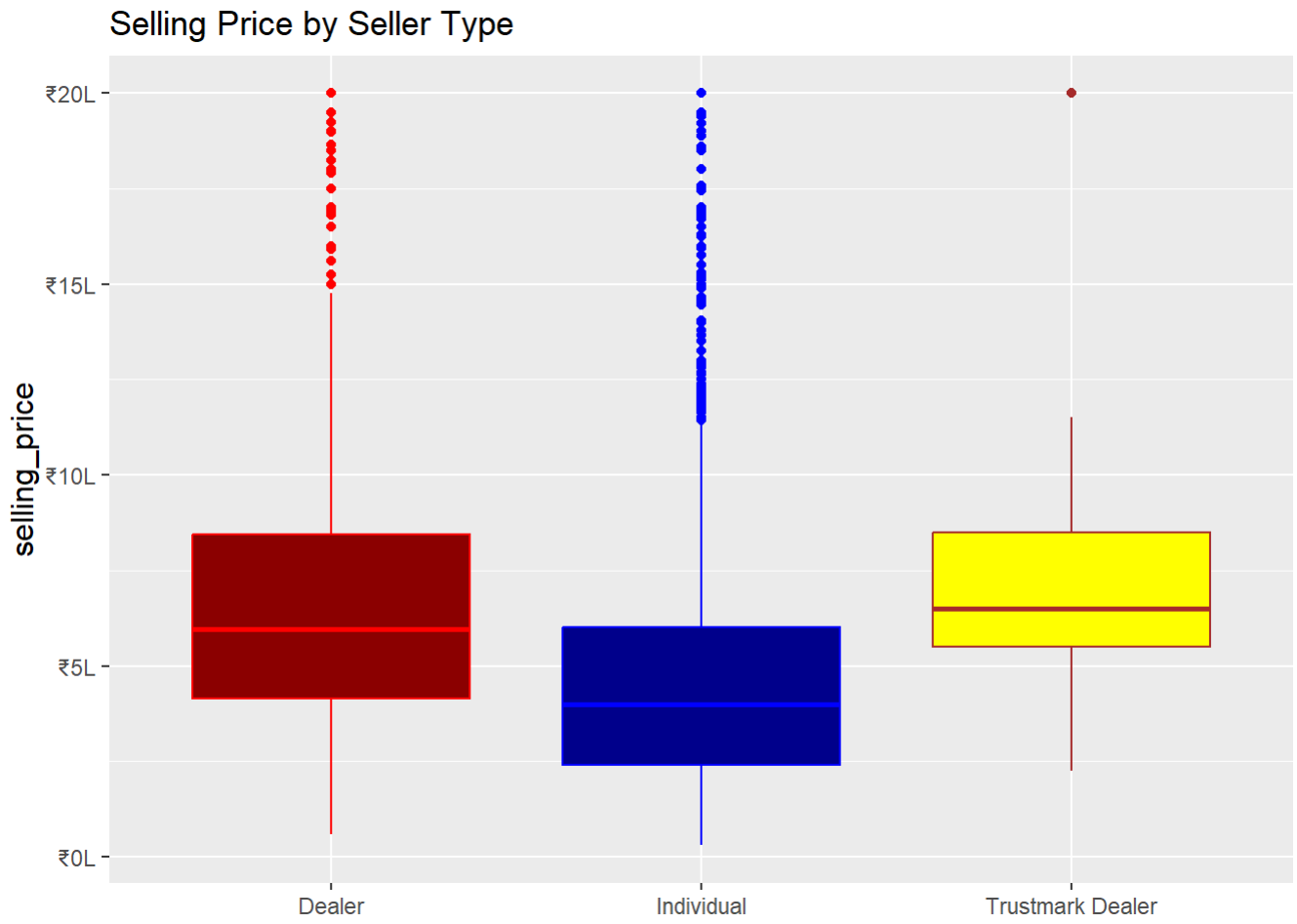
Histogram Plot of Selling Price by Owner type

```
h = ggplot(data = CarSales, aes(x = selling_price)) +
  geom_histogram(aes(fill = owner), bins = 10, color = "black", show.legend = TRUE, position =
"fill") +
  scale_x_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L")) +
  scale_color_manual(values = c("darkred", "darkblue", "yellow", "green", "purple")) +
  scale_fill_manual(values = c("darkred", "darkblue", "yellow", "green", "purple")) +labs(title
= "Selling Price and Count by Ownership Type") + theme(plot.title = element_text(hjust = 0.5))

plot(h)
```



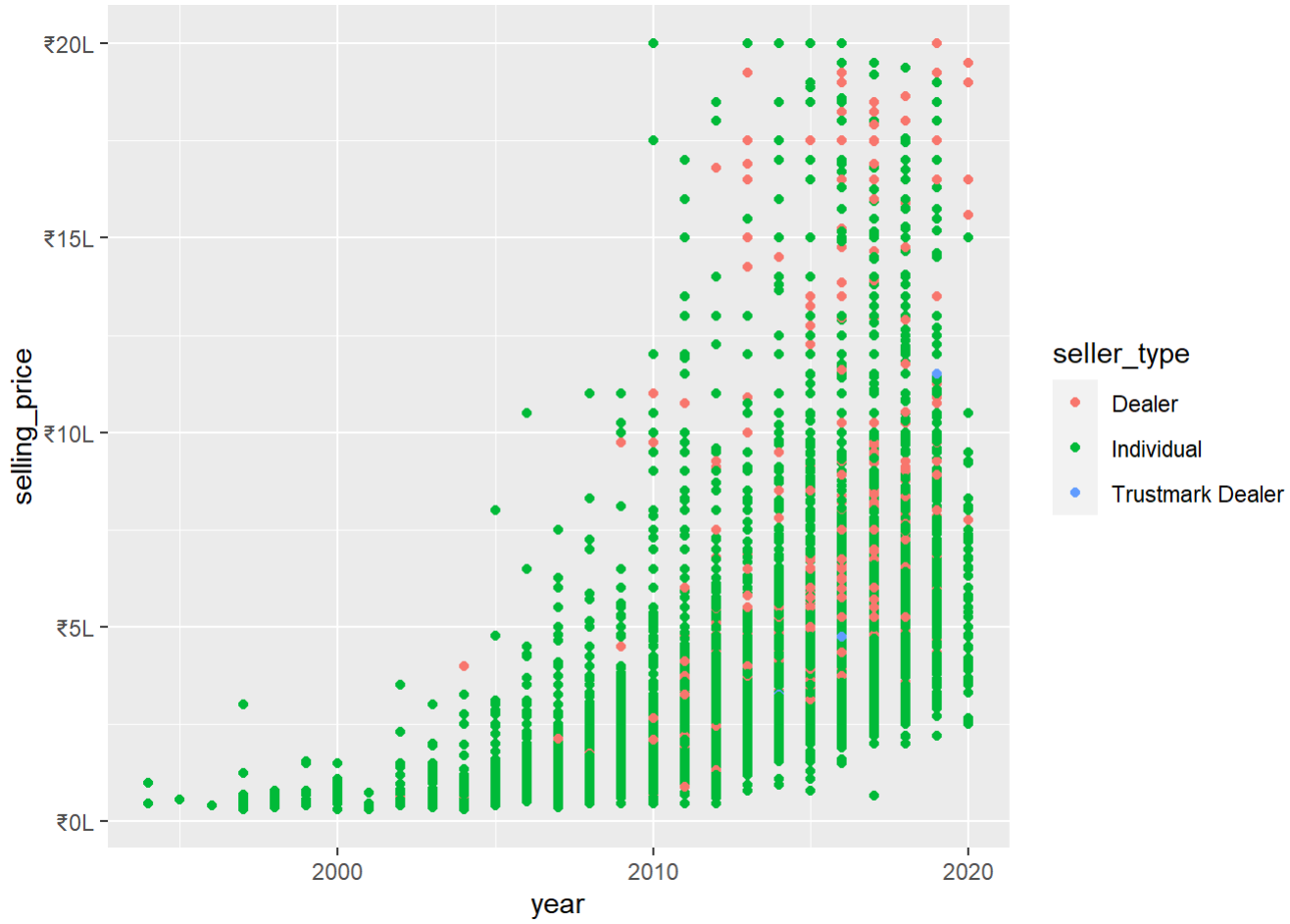
Boxplot of Selling Price by Seller Type

```
e <- ggplot(CarSales, aes(x = seller_type, y = selling_price)) +
  geom_boxplot(fill = c("darkred", "darkblue", "yellow"), col = c("red", "blue", "brown")) +
  scale_y_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L")) +
  theme(axis.title.x = element_blank(), axis.title.y = element_text(size = 12)) +
  labs(title = "Selling Price by Seller Type")
plot(e)
```



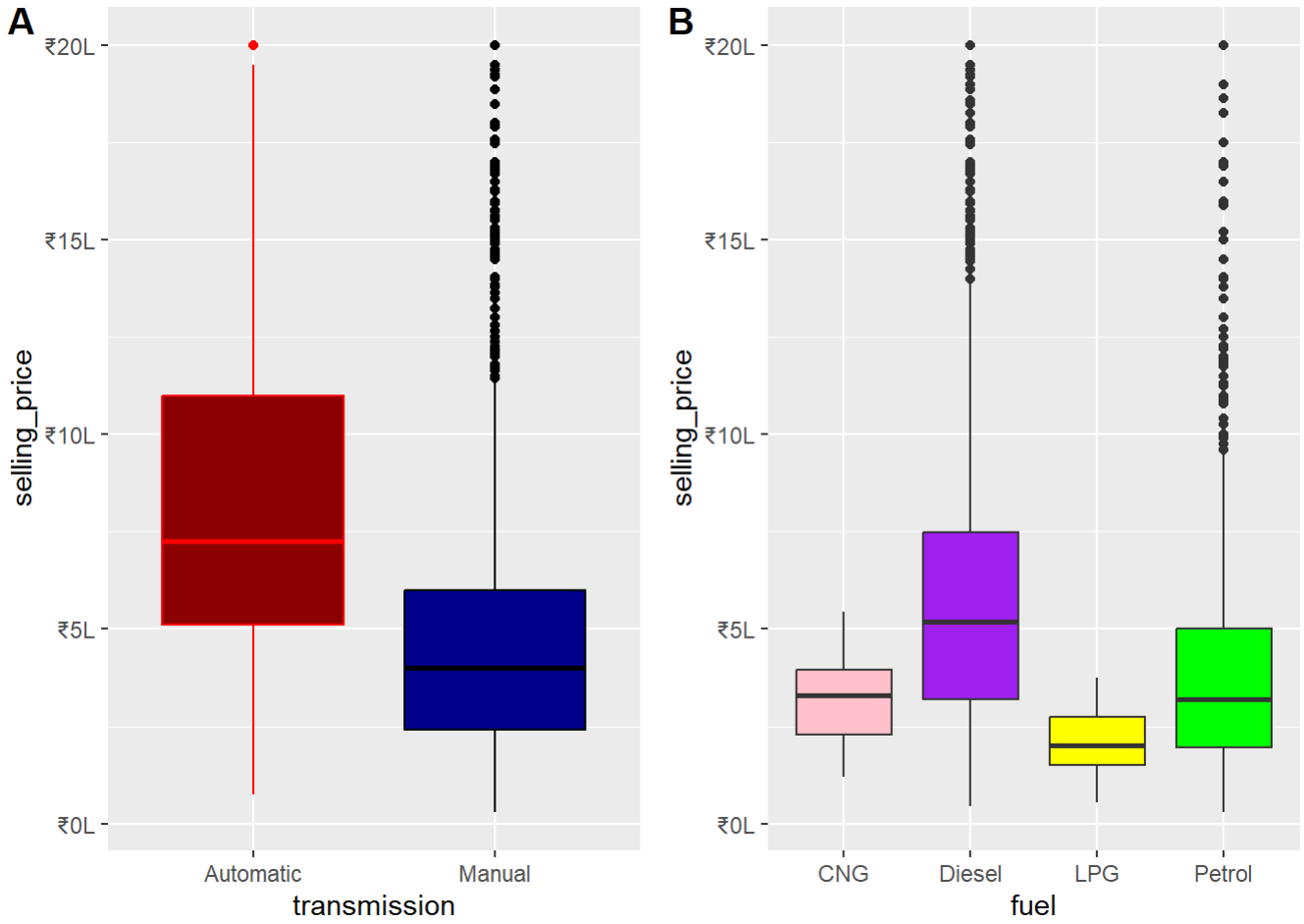
Scatterplot of Selling Price vs Year of Manufacture for Seller Type

```
ggplot(data = CarSales) +
  geom_point(mapping = aes(x = year, y = selling_price, colour = seller_type)) +
  scale_y_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L"))
```



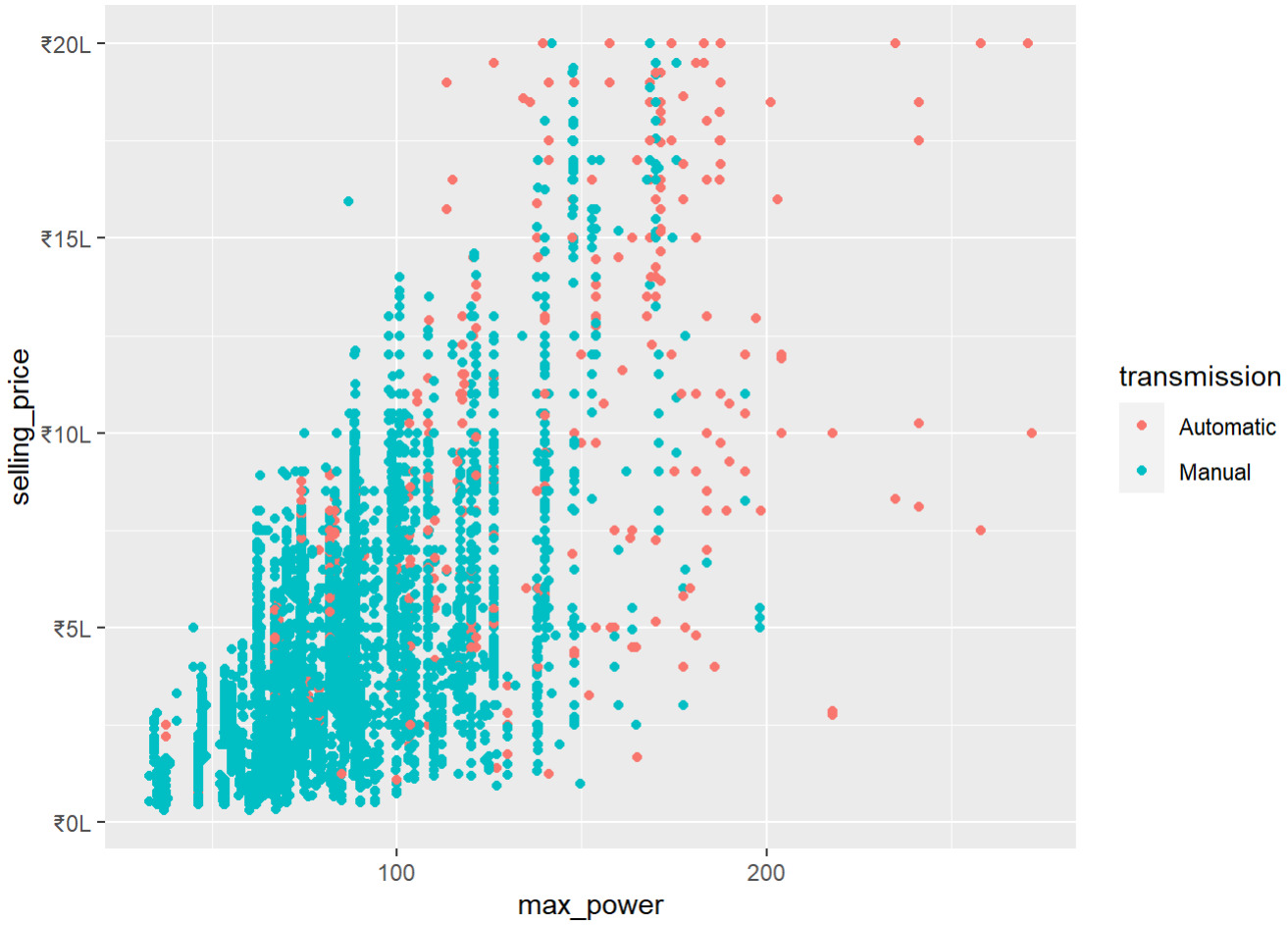
Boxplot of Selling Price by Transmission type and Fuel Type

```
e=ggplot(CarSales, aes(x = transmission, y = selling_price)) +
  geom_boxplot(fill=c("darkred","darkblue"), col=c("red","black"))+
  scale_y_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L"))

h=ggplot(CarSales, aes(x = fuel, y = selling_price)) +
  geom_boxplot(fill=c("pink","purple", "yellow", "green"))+
  scale_y_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L"))

plot_grid(e, h,labels = "AUTO")
```




```
ggplot(data = CarSales) +
  geom_point(mapping = aes(x = engine, y = selling_price, colour = transmission)) +
  scale_y_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L"))
```


```
ggplot(data = CarSales) +
  geom_point(mapping = aes(x = max_power, y = selling_price, colour = transmission)) +
  scale_y_continuous(labels = comma_format(prefix = "\u20B9", scale = 1e-5, suffix = "L"))
```



HYPOTHESIS TESTING

```
DealerC =filter(CarSales, seller_type=="Dealer")
head(DealerC)
```

```
##                       name year selling_price km_driven   fuel seller_type
## 1      Honda City i VTEC VX 2018        925000     28900 Petrol      Dealer
## 2           Honda City V MT 2013        425000     86300 Petrol      Dealer
## 3 Maruti Swift Dzire VXi AT 2018        675000     23300 Petrol      Dealer
## 4  Maruti Vitara Brezza VDi 2018        819999     32600 Diesel      Dealer
## 5       Maruti Alto K10 VXI 2018        390000     10300 Petrol      Dealer
## 6    Toyota Fortuner 4x4 MT 2014       1500000     77000 Diesel      Dealer
##   transmission       owner engine max_power seats mileage
## 1       Manual First Owner   1497    117.30     5   17.80
## 2       Manual First Owner   1497    116.30     5   16.80
## 3    Automatic First Owner   1197     83.14     5   18.50
## 4       Manual First Owner   1248     88.50     5   24.30
## 5       Manual First Owner    998     67.05     5   23.95
## 6       Manual First Owner   2982    168.50     7   12.55
```

```
summary(DealerC)
```

```
##      name                year      selling_price       km_driven
##  Length:609         Min.   :2002   Min.   :  60000   Min.   :  1303
##  Class :character   1st Qu.:2013   1st Qu.: 415000   1st Qu.: 31000
##  Mode  :character   Median :2016   Median : 595000   Median : 50800
##                     Mean   :2015   Mean   : 680535   Mean   : 54545
##                     3rd Qu.:2017   3rd Qu.: 844999   3rd Qu.: 70195
##                     Max.   :2020   Max.   :2000000   Max.   :221889
##      fuel           seller_type        transmission          owner
##  Length:609         Length:609         Length:609         Length:609
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      engine       max_power          seats         mileage
##  Min.   : 624   Min.   : 34.20   Min.   :4.00   Min.   : 0.00
##  1st Qu.:1197   1st Qu.: 74.96   1st Qu.:5.00   1st Qu.:17.00
##  Median :1396   Median : 88.70   Median :5.00   Median :19.70
##  Mean   :1479   Mean   : 98.42   Mean   :5.35   Mean   :19.80
##  3rd Qu.:1498   3rd Qu.:115.00   3rd Qu.:5.00   3rd Qu.:22.77
##  Max.   :2999   Max.   :204.00   Max.   :9.00   Max.   :28.40
```

```
IndC =filter(CarSales, seller_type=="Individual")
head(IndC)
```

```
##                           name year selling_price km_driven   fuel seller_type
## 1        Maruti Swift Dzire VDI 2014        450000    145500 Diesel  Individual
## 2  Skoda Rapid 1.5 TDI Ambition 2014        370000    120000 Diesel  Individual
## 3      Honda City 2017-2020 EXi 2006        158000    140000 Petrol  Individual
## 4     Hyundai i20 Sportz Diesel 2010        225000    127000 Diesel  Individual
## 5        Maruti Swift VXI BSIII 2007        130000    120000 Petrol  Individual
## 6 Hyundai Xcent 1.2 VTVT E Plus 2017        440000     45000 Petrol  Individual
##   transmission        owner engine max_power seats mileage
## 1       Manual  First Owner   1248     74.00     5   23.40
## 2       Manual Second Owner   1498    103.52     5   21.14
## 3       Manual  Third Owner   1497     78.00     5   17.70
## 4       Manual  First Owner   1396     90.00     5   23.00
## 5       Manual  First Owner   1298     88.20     5   16.10
## 6       Manual  First Owner   1197     81.86     5   20.14
```

```
summary(IndC)
```

```
##      name                year      selling_price       km_driven
##  Length:5974        Min.   :1994   Min.   :  29999   Min.   :      1
##  Class :character   1st Qu.:2011   1st Qu.: 240000   1st Qu.:  40000
##  Mode  :character   Median :2014   Median : 400000   Median :  70000
##                     Mean   :2013   Mean   : 456708   Mean   :  76179
##                     3rd Qu.:2017   3rd Qu.: 600000   3rd Qu.: 100000
##                     Max.   :2020   Max.   :2000000   Max.   :2360457
##      fuel           seller_type        transmission          owner
##  Length:5974        Length:5974        Length:5974        Length:5974
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      engine       max_power          seats           mileage
##  Min.   : 624   Min.   : 32.80   Min.   : 2.000   Min.   : 0.00
##  1st Qu.:1196   1st Qu.: 67.10   1st Qu.: 5.000   1st Qu.:16.81
##  Median :1248   Median : 81.80   Median : 5.000   Median :19.64
##  Mean   :1410   Mean   : 84.76   Mean   : 5.443   Mean   :19.50
##  3rd Qu.:1498   3rd Qu.: 98.60   3rd Qu.: 5.000   3rd Qu.:22.50
##  Max.   :3498   Max.   :272.00   Max.   :14.000   Max.   :33.44
```

Consider the hypothesis as given below,

Null Hypothesis(Ho) : σ (DealerC$selling_price) = σ (IndC$selling_price)

Alternate Hypothesis(H1) : σ (DealerC$selling_price) ≠ σ (IndC$selling_price)

```
var.test(DealerC$selling_price, IndC$selling_price, alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
## data:  DealerC$selling_price and IndC$selling_price
## F = 1.7921, num df = 608, denom df = 5973, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.597057 2.022495
## sample estimates:
## ratio of variances
##           1.792064
```

The variances are not equal because the p value is much lesser than significance level and the fvalue doesn't lie between f1 and f2, hence we reject the hypothesis.

For unknown mean and unequal variances

Consider the hypothesis as given below,

Null Hypothesis(Ho) : μ (DealerC$selling_price) - μ (IndC$selling_price) = 0

Alternate Hypothesis(H1) : μ (DealerC$selling_price) - μ (IndC$selling_price) ≠ 0

```
t.test(DealerC$selling_price, IndC$selling_price, var.equal = FALSE, conf.level = 0.95, alternat
ive= "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  DealerC$selling_price and IndC$selling_price
## t = 13.283, df = 678.92, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  190740.6 256913.5
## sample estimates:
## mean of x mean of y
##  680535.2  456708.2
```

LINEAR REGRESSION

```
summary(CarSales)
```

```
##      name                year      selling_price       km_driven
##  Length:6610        Min.   :1994   Min.   :  29999   Min.   :      1
##  Class :character   1st Qu.:2011   1st Qu.: 250000   1st Qu.:  40000
##  Mode  :character   Median :2014   Median : 409999   Median :  70000
##                     Mean   :2014   Mean   : 478398   Mean   :  74023
##                     3rd Qu.:2017   3rd Qu.: 625000   3rd Qu.: 100000
##                     Max.   :2020   Max.   :2000000   Max.   :2360457
##      fuel           seller_type        transmission          owner
##  Length:6610        Length:6610        Length:6610        Length:6610
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      engine       max_power          seats           mileage
##  Min.   : 624   Min.   : 32.80   Min.   : 2.000   Min.   : 0.00
##  1st Qu.:1197   1st Qu.: 67.10   1st Qu.: 5.000   1st Qu.:16.90
##  Median :1248   Median : 81.83   Median : 5.000   Median :19.67
##  Mean   :1416   Mean   : 86.07   Mean   : 5.433   Mean   :19.52
##  3rd Qu.:1498   3rd Qu.: 98.96   3rd Qu.: 5.000   3rd Qu.:22.54
##  Max.   :3498   Max.   :272.00   Max.   :14.000   Max.   :33.44
```

```
numeric_CarSales = select_if(CarSales, is.numeric)
i = sample(2,nrow(numeric_CarSales),replace =TRUE,prob =c(0.8,0.2))
CarSalesTraining = numeric_CarSales[i==1,]
summary(CarSalesTraining)
```

```
##       year      selling_price       km_driven           engine
##  Min.   :1994   Min.   :  29999   Min.   :      1   Min.   : 624
##  1st Qu.:2011   1st Qu.: 250000   1st Qu.:  39000   1st Qu.:1197
##  Median :2014   Median : 415000   Median :  70000   Median :1248
##  Mean   :2014   Mean   : 479098   Mean   :  73852   Mean   :1414
##  3rd Qu.:2017   3rd Qu.: 625000   3rd Qu.: 100000   3rd Qu.:1498
##  Max.   :2020   Max.   :2000000   Max.   :2360457   Max.   :3498
##    max_power          seats           mileage
##  Min.   : 32.80   Min.   : 2.000   Min.   : 0.00
##  1st Qu.: 67.10   1st Qu.: 5.000   1st Qu.:16.90
##  Median : 81.83   Median : 5.000   Median :19.67
##  Mean   : 86.00   Mean   : 5.427   Mean   :19.54
##  3rd Qu.: 98.96   3rd Qu.: 5.000   3rd Qu.:22.54
##  Max.   :272.00   Max.   :14.000   Max.   :33.44
```

```
CarSalesTest = numeric_CarSales[i==2,]
summary(CarSalesTest)
```

```
##       year      selling_price       km_driven          engine
##  Min.   :1996   Min.   :  40000   Min.   :  1000   Min.   : 624
##  1st Qu.:2011   1st Qu.: 250000   1st Qu.: 40000   1st Qu.:1196
##  Median :2014   Median : 400000   Median : 70000   Median :1248
##  Mean   :2013   Mean   : 475638   Mean   : 74699   Mean   :1423
##  3rd Qu.:2016   3rd Qu.: 630000   3rd Qu.:100000   3rd Qu.:1498
##  Max.   :2020   Max.   :2000000   Max.   :375000   Max.   :2997
##    max_power          seats           mileage
##  Min.   : 32.80   Min.   : 4.000   Min.   : 0.00
##  1st Qu.: 67.10   1st Qu.: 5.000   1st Qu.:16.95
##  Median : 81.83   Median : 5.000   Median :19.30
##  Mean   : 86.32   Mean   : 5.455   Mean   :19.46
##  3rd Qu.: 98.97   3rd Qu.: 5.000   3rd Qu.:22.32
##  Max.   :218.00   Max.   :10.000   Max.   :33.44
```

```
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.2.3
```

```
##
## Attaching package: 'DescTools'
```

```
## The following objects are masked from 'package:MLmetrics':
##
##     AUC, Gini, MAE, MAPE, MSE, RMSE
```

```
## The following object is masked from 'package:mosaic':
##
##     MAD
```

```
#summary(CarSales)
#numeric_CarSales = select_if(CarSales, is.numeric)
#q_range <- quantile(CarSales, probs=c(0.05, 0.95))
#Clean_CarSales<- CarSales[CarSales >= q_range[1] & CarSales <= q_range[2]]
#Car_Clean <- Winsorize(CarSales, probs = c(0.05, 0.95))
#data_winsorized <- apply(numeric_CarSales, 2, winsorize, probs = probs)
#data_win_seat <- subset(data_winsorized, select = -seats)
#data_win_seat <- as.data.frame(data_win_seat)
#i = sample(2,nrow(data_win_seat),replace =TRUE,prob =c(0.8,0.2))
#CarSalesTraining = data_win_seat[i==1,]
#CarSalesTest = data_win_seat[i==2,]
```

```
library(robustHD)
```

```
## Warning: package 'robustHD' was built under R version 4.2.3
```

```
## Loading required package: perry
```

```
## Warning: package 'perry' was built under R version 4.2.3
```

```
## Loading required package: parallel
```

```
## Loading required package: robustbase
```

```
## Warning: package 'robustbase' was built under R version 4.2.3
```

```
##
## Attaching package: 'robustbase'
```

```
## The following object is masked from 'package:tigerstats':
##
##     alcohol
```

```
# Specify the proportion of outliers to be removed from each attribute
probs <- c(0.05, 0.95)  # remove the bottom and top 5% of values

# Winsorize each column of the dataset separately
data_winsorized <- apply(numeric_CarSales, 2, winsorize, probs = probs)
data_winsorized <- as.data.frame(data_winsorized)
# View the original and winsorized datasets side-by-side
#cbind(my_data, my_data_winsorized)
```

```
summary(data_winsorized)
```

```
##       year      selling_price      km_driven          engine
##  Min.   :2005   Min.   : 29999   Min.   :     1   Min.   : 624
##  1st Qu.:2011   1st Qu.:250000   1st Qu.: 40000   1st Qu.:1197
##  Median :2014   Median :409999   Median : 70000   Median :1248
##  Mean   :2014   Mean   :455345   Mean   : 71042   Mean   :1353
##  3rd Qu.:2017   3rd Qu.:625000   3rd Qu.:100000   3rd Qu.:1498
##  Max.   :2020   Max.   :970425   Max.   :158956   Max.   :1974
##
##    max_power          seats         mileage
##  Min.   : 38.00   Min.   : NA    Min.   :11.16
##  1st Qu.: 67.10   1st Qu.: NA    1st Qu.:16.90
##  Median : 81.83   Median : NA    Median :19.67
##  Mean   : 83.80   Mean   :NaN    Mean   :19.55
##  3rd Qu.: 98.96   3rd Qu.: NA    3rd Qu.:22.54
##  Max.   :125.66   Max.   : NA    Max.   :28.18
##                   NA's   :6610
```

```
data_win_seat <- subset(data_winsorized, select = -seats)
sum(is.na(data_win_seat))
```

```
## [1] 0
```

```
dim(data_win_seat)
```

```
## [1] 6610    6
```

```
str(data_win_seat)
```

```
## 'data.frame':    6610 obs. of  6 variables:
##  $ year         : num  2014 2014 2006 2010 2007 ...
##  $ selling_price: num  450000 370000 158000 225000 130000 440000 96000 45000 350000 200000
...
##  $ km_driven    : num  145500 120000 140000 127000 120000 ...
##  $ engine       : num  1248 1498 1497 1396 1298 ...
##  $ max_power    : num  74 103.5 78 90 88.2 ...
##  $ mileage      : num  23.4 21.1 17.7 23 16.1 ...
```

```
str(data_winsorized)
```

```
## 'data.frame':    6610 obs. of  7 variables:
##  $ year         : num  2014 2014 2006 2010 2007 ...
##  $ selling_price: num  450000 370000 158000 225000 130000 440000 96000 45000 350000 200000
...
##  $ km_driven    : num  145500 120000 140000 127000 120000 ...
##  $ engine       : num  1248 1498 1497 1396 1298 ...
##  $ max_power    : num  74 103.5 78 90 88.2 ...
##  $ seats        : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
##  $ mileage      : num  23.4 21.1 17.7 23 16.1 ...
```

```
summary(data_win_seat)
```

```
##       year      selling_price      km_driven          engine
##  Min.   :2005   Min.   : 29999   Min.   :     1   Min.   : 624
##  1st Qu.:2011   1st Qu.:250000   1st Qu.: 40000   1st Qu.:1197
##  Median :2014   Median :409999   Median : 70000   Median :1248
##  Mean   :2014   Mean   :455345   Mean   : 71042   Mean   :1353
##  3rd Qu.:2017   3rd Qu.:625000   3rd Qu.:100000   3rd Qu.:1498
##  Max.   :2020   Max.   :970425   Max.   :158956   Max.   :1974
##    max_power         mileage
##  Min.   : 38.00   Min.   :11.16
##  1st Qu.: 67.10   1st Qu.:16.90
##  Median : 81.83   Median :19.67
##  Mean   : 83.80   Mean   :19.55
##  3rd Qu.: 98.96   3rd Qu.:22.54
##  Max.   :125.66   Max.   :28.18
```

```
dim(data_win_seat)
```

```
## [1] 6610    6
```

```
summary(CarSales)
```

```
##      name                year      selling_price       km_driven
##  Length:6610        Min.   :1994   Min.   :  29999   Min.   :      1
##  Class :character   1st Qu.:2011   1st Qu.: 250000   1st Qu.:  40000
##  Mode  :character   Median :2014   Median : 409999   Median :  70000
##                     Mean   :2014   Mean   : 478398   Mean   :  74023
##                     3rd Qu.:2017   3rd Qu.: 625000   3rd Qu.: 100000
##                     Max.   :2020   Max.   :2000000   Max.   :2360457
##      fuel           seller_type        transmission          owner
##  Length:6610        Length:6610        Length:6610        Length:6610
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      engine       max_power          seats           mileage
##  Min.   : 624   Min.   : 32.80   Min.   : 2.000   Min.   : 0.00
##  1st Qu.:1197   1st Qu.: 67.10   1st Qu.: 5.000   1st Qu.:16.90
##  Median :1248   Median : 81.83   Median : 5.000   Median :19.67
##  Mean   :1416   Mean   : 86.07   Mean   : 5.433   Mean   :19.52
##  3rd Qu.:1498   3rd Qu.: 98.96   3rd Qu.: 5.000   3rd Qu.:22.54
##  Max.   :3498   Max.   :272.00   Max.   :14.000   Max.   :33.44
```

We have constructed a simple linear regression of `selling_price` by `seller_type` using `carSalesTraining`.

```
summary(CarSales)
```

```
##      name                year      selling_price       km_driven
##  Length:6610        Min.   :1994   Min.   :  29999   Min.   :      1
##  Class :character   1st Qu.:2011   1st Qu.: 250000   1st Qu.:  40000
##  Mode  :character   Median :2014   Median : 409999   Median :  70000
##                     Mean   :2014   Mean   : 478398   Mean   :  74023
##                     3rd Qu.:2017   3rd Qu.: 625000   3rd Qu.: 100000
##                     Max.   :2020   Max.   :2000000   Max.   :2360457
##      fuel           seller_type        transmission          owner
##  Length:6610        Length:6610        Length:6610        Length:6610
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      engine       max_power          seats           mileage
##  Min.   : 624   Min.   : 32.80   Min.   : 2.000   Min.   : 0.00
##  1st Qu.:1197   1st Qu.: 67.10   1st Qu.: 5.000   1st Qu.:16.90
##  Median :1248   Median : 81.83   Median : 5.000   Median :19.67
##  Mean   :1416   Mean   : 86.07   Mean   : 5.433   Mean   :19.52
##  3rd Qu.:1498   3rd Qu.: 98.96   3rd Qu.: 5.000   3rd Qu.:22.54
##  Max.   :3498   Max.   :272.00   Max.   :14.000   Max.   :33.44
```

```
numeric_training_data <- select_if(CarSalesTraining, is.numeric)
numeric_testing_data <- select_if(CarSalesTest, is.numeric)
```

```
slr_sale <- lm(selling_price ~ max_power, data = CarSalesTraining)
summary(slr_sale)
```

```
##
## Call:
## lm(formula = selling_price ~ max_power, data = CarSalesTraining)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1013108  -147300   -10164   137264  1170822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -162946.0    10448.4  -15.60   <2e-16 ***
## max_power      7465.3      115.3   64.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 238800 on 5271 degrees of freedom
## Multiple R-squared:  0.4429, Adjusted R-squared:  0.4428
## F-statistic:  4191 on 1 and 5271 DF,  p-value: < 2.2e-16
```

```
mlr_sale <- lm(selling_price ~., data = numeric_training_data)
summary(mlr_sale)
```

```
##
## Call:
## lm(formula = selling_price ~ ., data = numeric_training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -658363 -101323  -16516   81407 1182555
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.657e+07  1.607e+06 -47.660  < 2e-16 ***
## year         3.791e+04  8.046e+02  47.112  < 2e-16 ***
## km_driven   -3.527e-01  4.594e-02  -7.678 1.91e-14 ***
## engine       1.269e+02  1.019e+01  12.456  < 2e-16 ***
## max_power    5.606e+03  1.279e+02  43.838  < 2e-16 ***
## seats        2.767e+03  3.737e+03   0.741 0.459001
## mileage      3.293e+03  8.667e+02   3.799 0.000147 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 179500 on 5266 degrees of freedom
## Multiple R-squared:  0.6855, Adjusted R-squared:  0.6851
## F-statistic:  1913 on 6 and 5266 DF,  p-value: < 2.2e-16
```

```
library(MASS)
# Creating a null model
intercept_only <- lm(selling_price ~ 1, data=numeric_training_data)
# Creating a full model
all <- lm(selling_price~., data=numeric_training_data)
forward <- stepAIC (intercept_only, direction='forward',scope = formula(all))
```

```
## Start:  AIC=133678.8
## selling_price ~ 1
##
##             Df  Sum of Sq        RSS    AIC
## + max_power  1 2.3893e+14 3.0050e+14 130596
## + year       1 1.6584e+14 3.7359e+14 131744
## + engine     1 1.1457e+14 4.2486e+14 132422
## + seats      1 3.4556e+13 5.0487e+14 133332
## + km_driven  1 1.2681e+13 5.2675e+14 133555
## + mileage    1 2.3297e+12 5.3710e+14 133658
## <none>                    5.3943e+14 133679
##
## Step:  AIC=130595.7
## selling_price ~ max_power
##
##             Df  Sum of Sq        RSS    AIC
## + year       1 1.2167e+14 1.7883e+14 127861
## + km_driven  1 2.3983e+13 2.7652e+14 130159
## + mileage    1 2.1061e+13 2.7944e+14 130215
## + seats      1 2.0931e+12 2.9841e+14 130561
## + engine     1 2.4734e+11 3.0025e+14 130593
## <none>                    3.0050e+14 130596
##
## Step:  AIC=127861
## selling_price ~ max_power + year
##
##             Df  Sum of Sq        RSS    AIC
## + engine     1 7.0127e+12 1.7182e+14 127652
## + seats      1 3.1778e+12 1.7565e+14 127768
## + mileage    1 5.0380e+11 1.7833e+14 127848
## + km_driven  1 3.7379e+11 1.7846e+14 127852
## <none>                    1.7883e+14 127861
##
## Step:  AIC=127652.1
## selling_price ~ max_power + year + engine
##
##             Df  Sum of Sq        RSS    AIC
## + km_driven  1 1.6912e+12 1.7013e+14 127602
## + mileage    1 2.5525e+11 1.7156e+14 127646
## <none>                    1.7182e+14 127652
## + seats      1 4.0809e+09 1.7181e+14 127654
##
## Step:  AIC=127601.9
## selling_price ~ max_power + year + engine + km_driven
##
##           Df  Sum of Sq        RSS    AIC
## + mileage  1 4.4758e+11 1.6968e+14 127590
## <none>                  1.7013e+14 127602
## + seats    1 1.8716e+08 1.7013e+14 127604
##
## Step:  AIC=127590
## selling_price ~ max_power + year + engine + km_driven + mileage
```

```
##
##         Df  Sum of Sq        RSS    AIC
## <none>                1.6968e+14 127590
## + seats  1 1.7669e+10 1.6966e+14 127591
```

```
# view results of forward stepwise regression
forward$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## selling_price ~ 1
##
## Final Model:
## selling_price ~ max_power + year + engine + km_driven + mileage
##
##
##          Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1                                  5272 5.394280e+14 133678.7
## 2 + max_power  1 2.389293e+14      5271 3.004987e+14 130595.7
## 3      + year  1 1.216681e+14      5270 1.788306e+14 127861.0
## 4    + engine  1 7.012738e+12      5269 1.718179e+14 127652.0
## 5 + km_driven  1 1.691170e+12      5268 1.701267e+14 127601.9
## 6   + mileage  1 4.475774e+11      5267 1.696791e+14 127590.0
```

```
# view final model
summary(forward)
```

```
##
## Call:
## lm(formula = selling_price ~ max_power + year + engine + km_driven +
##     mileage, data = numeric_training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -656322 -101292  -16446   81904 1177841
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.682e+07  1.568e+06 -48.981  < 2e-16 ***
## max_power    5.574e+03  1.201e+02  46.397  < 2e-16 ***
## year         3.804e+04  7.838e+02  48.538  < 2e-16 ***
## engine       1.314e+02  8.199e+00  16.025  < 2e-16 ***
## km_driven   -3.505e-01  4.584e-02  -7.646 2.44e-14 ***
## mileage      3.155e+03  8.465e+02   3.727 0.000196 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 179500 on 5267 degrees of freedom
## Multiple R-squared:  0.6854, Adjusted R-squared:  0.6851
## F-statistic:  2295 on 5 and 5267 DF,  p-value: < 2.2e-16
```

```
ypredict_forward <- predict(object = forward, newdata = numeric_testing_data)
MAE(numeric_testing_data$selling_price,ypredict_forward)
```

```
## [1] 128838.3
```

```
MSE(numeric_testing_data$selling_price,ypredict_forward)
```

```
## [1] 32892433134
```

```
backward <- stepAIC(all, direction = 'backward')
```

```
## Start:  AIC=127591.4
## selling_price ~ year + km_driven + engine + max_power + seats +
##     mileage
##
##             Df  Sum of Sq        RSS    AIC
## - seats      1 1.7669e+10 1.6968e+14 127590
## <none>                    1.6966e+14 127591
## - mileage    1 4.6506e+11 1.7013e+14 127604
## - km_driven  1 1.8994e+12 1.7156e+14 127648
## - engine     1 4.9990e+12 1.7466e+14 127743
## - max_power  1 6.1915e+13 2.3158e+14 129230
## - year       1 7.1510e+13 2.4117e+14 129444
##
## Step:  AIC=127590
## selling_price ~ year + km_driven + engine + max_power + mileage
##
##             Df  Sum of Sq        RSS    AIC
## <none>                    1.6968e+14 127590
## - mileage    1 4.4758e+11 1.7013e+14 127602
## - km_driven  1 1.8835e+12 1.7156e+14 127646
## - engine     1 8.2727e+12 1.7795e+14 127839
## - max_power  1 6.9349e+13 2.3903e+14 129395
## - year       1 7.5899e+13 2.4558e+14 129537
```

```
backward$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## selling_price ~ year + km_driven + engine + max_power + seats +
##     mileage
##
## Final Model:
## selling_price ~ year + km_driven + engine + max_power + mileage
##
##
##      Step Df    Deviance Resid. Df   Resid. Dev      AIC
## 1                             5266 1.696614e+14 127591.4
## 2 - seats  1 17668795764      5267 1.696791e+14 127590.0
```

```
summary(backward)
```

```
##
## Call:
## lm(formula = selling_price ~ year + km_driven + engine + max_power +
##     mileage, data = numeric_training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -656322 -101292  -16446   81904 1177841
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.682e+07  1.568e+06 -48.981  < 2e-16 ***
## year         3.804e+04  7.838e+02  48.538  < 2e-16 ***
## km_driven   -3.505e-01  4.584e-02  -7.646 2.44e-14 ***
## engine       1.314e+02  8.199e+00  16.025  < 2e-16 ***
## max_power    5.574e+03  1.201e+02  46.397  < 2e-16 ***
## mileage      3.155e+03  8.465e+02   3.727 0.000196 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 179500 on 5267 degrees of freedom
## Multiple R-squared:  0.6854, Adjusted R-squared:  0.6851
## F-statistic:  2295 on 5 and 5267 DF,  p-value: < 2.2e-16
```

```
ypredict_bckwrd <- predict(object = backward, newdata = numeric_testing_data)
MAE(numeric_testing_data$selling_price,ypredict_bckwrd)
```

```
## [1] 128838.3
```

```
MSE(numeric_testing_data$selling_price,ypredict_bckwrd)
```

```
## [1] 32892433134
```