**FINAL PROJECT - MILESTONE 1**

**PROBABILITY AND STATISTICS ASSIGNMENT**

By:

**AKASH RAJ,**

**HARSHIT GAUR,**

**VARUN SINGH**

MASTER OF PROFESSIONAL STUDIES IN ANALYTICS

ALY 6010: PROBABILITY THEORY AND INTRODUCTORY STATISTICS

NOVEMBER 13, 2021

To: **PROF. AMIN KARIMPOUR**

Table of Contents

[Introduction 1](#_Toc87735565)

[Data Analysis 2](#_Toc87735566)

[Data Cleaning 4](#_Toc87735567)

[Feature Engineering 7](#_Toc87735568)

[Exploratory Data Analysis 9](#_Toc87735569)

[Univariate Analysis 11](#_Toc87735570)

[Multivariate Analysis 23](#_Toc87735571)

[Conclusion 26](#_Toc87735572)

[Bibliography 26](#_Toc87735573)

[Appendix 27](#_Toc87735574)

# Introduction

Data analytics is the process of analyzing raw data and generating actionable insights. It comprises of the processes, tools and techniques of data analysis and management, including the collection, organization, and storage of data. Organizations use data analytics to gain competitive advantage by improving their performance and operational efficiency. Data analytics is performed on a variety of big data sets, like transactions, click streams, server logs, electronic health records, insurance claims, etc. Different analytical techniques and algorithms can be applied on these data sets to accomplish different objectives. These different types of analytical techniques are colloquially called:-

1. **Descriptive analytics** - Summarizing the data to understand past events and performance
2. **Diagnostic analytics** - Investigating the root cause of certain events
3. **Predictive analytics** - Predicting the future for planning
4. **Prescriptive analytics** - Recommending the optimal outcomes

Irrespective of the type of analytics being performed, the basis of every method or algorithm in data analytics is descriptive/inferential statistics and machine learning. In this analysis report, we will leverage descriptive statistics to generate insights from the data.

**Problems Statement:**

CarDekho.com is India’s leading car search venture that helps users buy cars that are right for them. It’s website and app carry rich automotive content such as expert reviews, detailed specs and prices, comparisons as well as videos and pictures of all car brands and models available in India.

We have the sales data of all the cars sold during the time frame of 1983 to 2020. We are going to analyse this data set in order to help them expand their business, gain and retain customers, and stand out the competitions they face.

The data set has 8128 data points with 13 features in it related to :

* **Car Details -** *Car name, transmission, fuel type, number of seats, year of manufacturing.*
* **Engine Details -** *Mileage, Engine type, Torque, Maximum power in BHP.*
* **Sale Details -** *Selling price, kilometers driven by the car.*

Later, we will implement learning algorithms and modelling techniques to understand the patterns and achieve high quality, consistent results targeting the following points:

1. Identify the right prospects at the right time
2. Build customer loyalty.
3. Promote efficiency across the departments. Marketing expenditures and supply chain management.

# Data Analysis

**Importing the libraries required for the analysis**

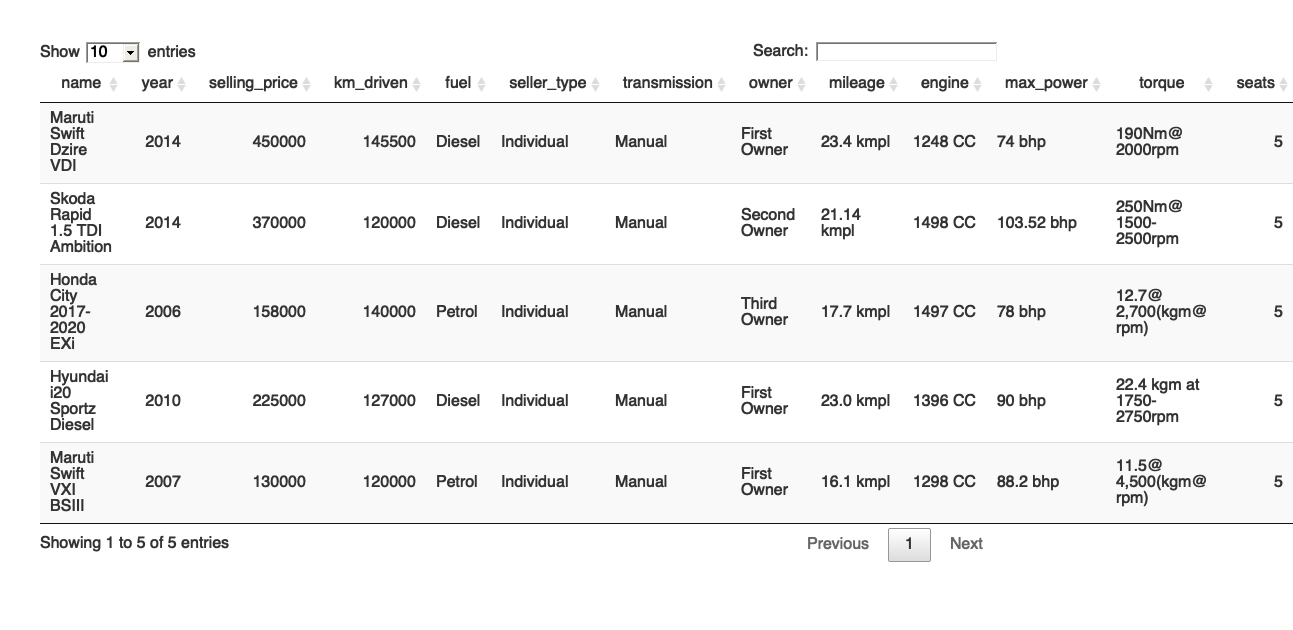
library(tidyverse)  
library(grid)  
library(gridExtra)  
library(dplyr)  
library(ggplot2)  
library(reshape2)  
library(DT)  
library(RColorBrewer)  
library(data.table)  
library(knitr)  
library(caret)  
library(stringr)  
library(RANN)  
library(data.table)  
library(vtable)  
library(scales)

**Importing the data set for analysis**

car\_sales <- read.csv("CarDekho Sales.csv")

**Head of data**

# View the head and summary of the data  
DT::datatable(head(car\_sales, 5), rownames = FALSE)



**Dimension and summary of data**

# View the dimension and summary of the data  
dim(car\_sales)

## [1] 8128 13

summary(car\_sales)

## name year selling\_price km\_driven   
## Length:8128 Min. :1983 Min. : 29999 Min. : 1   
## Class :character 1st Qu.:2011 1st Qu.: 254999 1st Qu.: 35000   
## Mode :character Median :2015 Median : 450000 Median : 60000   
## Mean :2014 Mean : 638272 Mean : 69820   
## 3rd Qu.:2017 3rd Qu.: 675000 3rd Qu.: 98000   
## Max. :2020 Max. :10000000 Max. :2360457   
##   
## fuel seller\_type transmission owner   
## Length:8128 Length:8128 Length:8128 Length:8128   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## mileage engine max\_power torque   
## Length:8128 Length:8128 Length:8128 Length:8128   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## seats   
## Min. : 2.000   
## 1st Qu.: 5.000   
## Median : 5.000   
## Mean : 5.417   
## 3rd Qu.: 5.000   
## Max. :14.000   
## NA's :221

Observation - There are 8128 observations with 13 features in the data set.

## DATA CLEANING

**Extract value from the ‘Mileage’ column**

The feature ‘mileage’ contains nominal categorical values suffixed with unit measurements of fuel efficiency. We need to clean this feature by removing the measurements (like kmpl, km/kg, etc.) and convert the data type of this feature from categorical to numeric for further utilization in analysis. We are using ‘gsub’ function of the ‘base’ package to perform this operation.

# Replace the 'kmpl, km/kg' string from data points in the 'mileage' feature. Convert to numeric data type.  
car\_sales$mileage <- as.numeric(gsub('[a-zA-Z/ ]', '', car\_sales$mileage))

**Extract value from the ‘Engine’ column**

The feature ‘engine’ contains nominal categorical values suffixed with unit measurements of capacity. We need to clean this feature by removing the measurements (like CC) and convert the data type of this feature from categorical to numeric for further utilization in analysis. We are using ‘gsub’ function of the ‘base’ package to perform this operation.

# Replace the 'CC' string from data points in the 'engine' feature. Convert to numeric data type.  
car\_sales$engine <- as.numeric(gsub('[a-zA-Z/ ]', '', car\_sales$engine))

**Extract value from the ‘Max Power’ column**

The feature ‘max\_power’ contains nominal categorical values suffixed with unit measurements of Brake Horse Power (BHP). We need to clean this feature by removing these measurements (like bhp) and convert the data type of this feature from categorical to numeric for further utilization in analysis. We used the ‘str\_remove’ function from the ‘stringr’ package to perform this operation.

car\_sales$max\_power <- as.numeric(str\_remove(car\_sales$max\_power, " [a-z]+"))

**Extract torque value from the ‘Torque’ column**

The feature ‘torque’ contains nominal categorical values suffixed with imperial measurements of Newton meters along with ‘Revolutions per minute’. We need to extract the torque values from this feature into another feature which will be further utilized in analysis and have the data type of this new feature as numeric. There were 2 values in the ‘torque’ feature from which we needed to extract the 1st part of the numeric value as ‘Torque’ and the latter part as ‘RPM’. The functions ‘str\_sub’ and ‘str\_locate’ form the package ‘stringr’ were used to perform this operation.

car\_sales$torque\_val <- as.numeric(str\_sub(car\_sales$torque, rep(1, nrow(car\_sales)),   
 str\_locate(car\_sales$torque, "\\D+")[,1]-1))

**Extract RPM value from the torque column**

The feature ‘torque’ contains nominal categorical values suffixed with imperial measurements of Newton meters along with ‘Revolutions per minute’. We need to extract the RPM values from this ‘torque’ feature into another feature which will be further utilized in analysis and have the data type of this new feature as numeric. We extracted the first part of torque as ‘Torque Value’ and needed to extract the second part as ‘RPM’ value. The functions of ‘str\_sub’ and ‘gsub’ were used to perform these operations.

num\_length <- str\_length(gsub("\\D+", "", car\_sales$torque))  
car\_sales$rpm <- as.numeric(str\_sub(as.numeric(gsub("\\D+", "", car\_sales$torque)),num\_length-3, num\_length))

**Extract brand from the can name column**

The feature ‘name’ contains the name of the car, its manufacturer, and the model. We will extract the Brand name from this feature into another feature which will be further utilized in analysis. Now, there are several data points where the brand names contain single word and multiple words. We formulated the extraction to take out the words corresponding to the brand names these cars belong to.

# Extract the brand name (first word) into another feature from the 'name' feature.  
car\_sales$brand <- word(car\_sales$name, start = 1, end = 1)  
  
# Extract the brand name (first 2 words) into another feature from the 'name' feature.  
car\_sales$brand[car\_sales$brand == "Ashok"] <- word(car\_sales$name[car\_sales$brand == "Ashok"], start = 1, end = 2)  
car\_sales$brand[car\_sales$brand == "Land"] <- word(car\_sales$name[car\_sales$brand == "Land"], start = 1, end = 2)  
car\_sales$brand[car\_sales$brand == "Range"] <- word(car\_sales$name[car\_sales$brand == "Range"], start = 1, end = 2)

**Filter out the values for CNG and LPG**

The feature ‘fuel’ contains 4 different types namely Diesel, Petrol, CNG, LPG. We will eliminate those records which belong to CNG and LPG as their cumulative data points account for only 1.5% of the total records in the data set.

car\_sales <- filter(car\_sales, !fuel %in% c("LPG", "CNG"))

**Identify blank and NA values in the data**

We further checked the data set for NA and blank values in all of its features.

colSums(is.na(car\_sales))

## name year selling\_price km\_driven fuel   
## 0 0 0 0 0   
## seller\_type transmission owner mileage engine   
## 0 0 0 214 214   
## max\_power torque seats torque\_val rpm   
## 208 0 214 214 214   
## brand   
## 0

**Remove unnecessary columns in the dataframe**

We removed some of the unnecessary columns from the data set as either informative values were extracted from them or were of no use.

car\_sales$name <- NULL  
car\_sales$torque <- NULL  
names(car\_sales)[12] <- "torque"

Observation - We found 214 records with NA values in most of the numeric features. These features need to be either removed or imputed by mean, median, etc.

**Impute missing and NA values in the data using kNN**

The data set contained few data points which contained missing or NA values in some of their features. We implemented the kNN algorithm to impute these missing and NA values with means of the features. This way we wouldn’t need to eliminate these records and can utilize them as well for our analysis. The functions ‘preProcess’ and ‘predict’ from the ‘caret’ package were used to perform these operations.

preProcValues <- preProcess(car\_sales %>%   
 dplyr::select(year, selling\_price, km\_driven, fuel, seller\_type,   
 transmission, owner, mileage, engine, max\_power, seats, torque, rpm),  
 method = c("knnImpute"),  
 k = 20,  
 knnSummary = mean)  
  
impute\_cars\_info <- predict(preProcValues, car\_sales, na.action = na.pass)  
  
procNames <- data.frame(col = names(preProcValues$mean), mean = preProcValues$mean, sd = preProcValues$std)  
  
for(i in procNames$col){  
 car\_sales[i] <- impute\_cars\_info[i]\*preProcValues$std[i]+preProcValues$mean[i]   
}  
  
colSums(is.na(car\_sales))

## year selling\_price km\_driven fuel seller\_type   
## 0 0 0 0 0   
## transmission owner mileage engine max\_power   
## 0 0 0 0 0   
## seats torque rpm brand   
## 0 0 0 0

Observation - No NA or missing values were found after the kNN imputation.

## 

## FEATURE ENGINEERING

**Create categorical column to indicate the size of the car**

A new feature named ‘Car Type’ is created to indicate the size of the car. We formulated the feature ‘size’ to calculate the size of car based on the following rule set :

If number of seats are less than or equal to 5, the type of car is “Small”.

If number of seats are greater than 5 and less than equal to 8, the type of car is “Medium”.

If number of seats are greater than 8 and less than equal to 14, the type of car is “Large”.

If number of seats don’t lie in the above range, we give the type as “Others” to the cars.

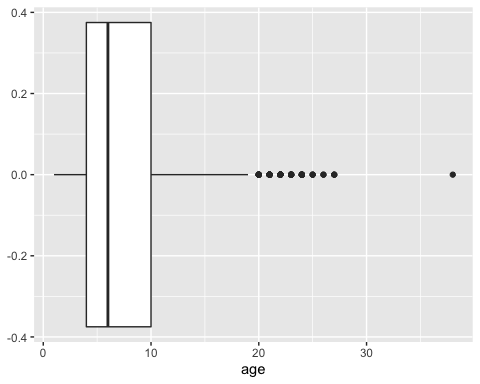
car\_sales$car\_type <- ifelse(car\_sales$seats <= 5, "Small", ifelse(car\_sales$seats <= 8, "Medium", ifelse(car\_sales$seats <= 14, "Large", "Others")))  
unique(car\_sales$car\_type)

## [1] "Small" "Medium" "Large"

**Create a numeric column to obtain the age of the car**

We created another feature named ‘age’ which signifies the age of the car. The feature ‘year’ was utilised to calculate the age of all the cars.

car\_sales$age <- as.numeric(format(Sys.Date(), "%Y")) - car\_sales$year  
ggplot(car\_sales) +  
 geom\_boxplot(mapping = aes(age))



**Unit Modification (Changing 1000s to 1s) of two features**

The values in the features ‘Selling Price’ and ‘KMs driven’ are in hundreds of thousands. We reduced these values from 1000s to Ones in order for proper analysis and better scaling of visualisations.

car\_sales$selling\_price <- round( (car\_sales$selling\_price / 1000), 2)  
car\_sales$km\_driven <- round( (car\_sales$km\_driven / 1000), 2)  
  
#car\_sales[sapply(car\_sales, as.character)] <- lapply(car\_sales[sapply(car\_sales, as.character)], as.factor)

**Changing Character data type to Factor data type**

The features belonging to character data types need to be changed into factor data type for efficient text manipulations.

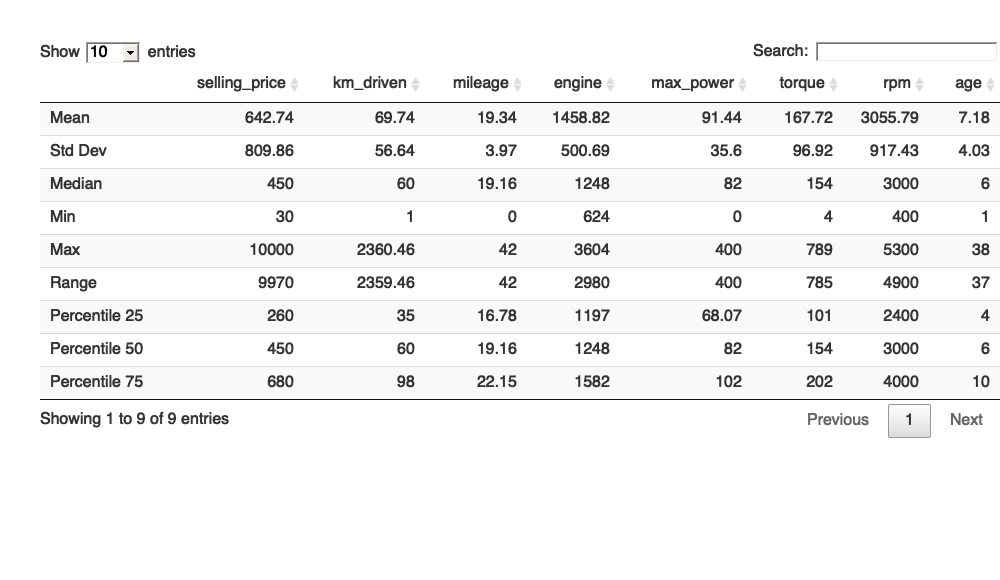
car\_sales$fuel <- as.factor(car\_sales$fuel)  
car\_sales$seller\_type <- as.factor(car\_sales$seller\_type)  
car\_sales$transmission <- as.factor(car\_sales$transmission)  
car\_sales$owner <- as.factor(car\_sales$owner)  
car\_sales$brand <- as.factor(car\_sales$brand)  
car\_sales$car\_type <- as.factor(car\_sales$car\_type)  
  
#car\_sales[sapply(car\_sales, as.character)] <- lapply(car\_sales[sapply(car\_sales, as.character)], as.factor)

## 

## EXPLORATORY DATA ANALYSIS

**Descriptive Statistics of the data set**

# Descriptive Statistics of the data set.  
mul\_fun <- function(x) {  
 c(mean(x), sd(x), median(x), min(x), max(x), max(x)-min(x), quantile(x, 0.25),   
 quantile(x, 0.5), quantile(x, 0.75))  
}  
  
var\_names <- c("Mean", "Std Dev", "Median", "Min", "Max", "Range", "Percentile 25", "Percentile 50",   
 "Percentile 75")  
num\_cols <- c("selling\_price", "km\_driven", "mileage", "engine", "max\_power", "torque", "rpm", "age")  
desc\_stats <- as.data.frame(round(sapply(car\_sales[num\_cols], mul\_fun), 2), row.names = var\_names)  
DT::datatable(desc\_stats)



Observation - From the statistics table, we found the below points :

The minimum value of mileage is **0**. On further analysis, we found that some of the cars contain *0.0 Kmpl* values as their mileage. We would need to impute these values in our second phase of analysis.

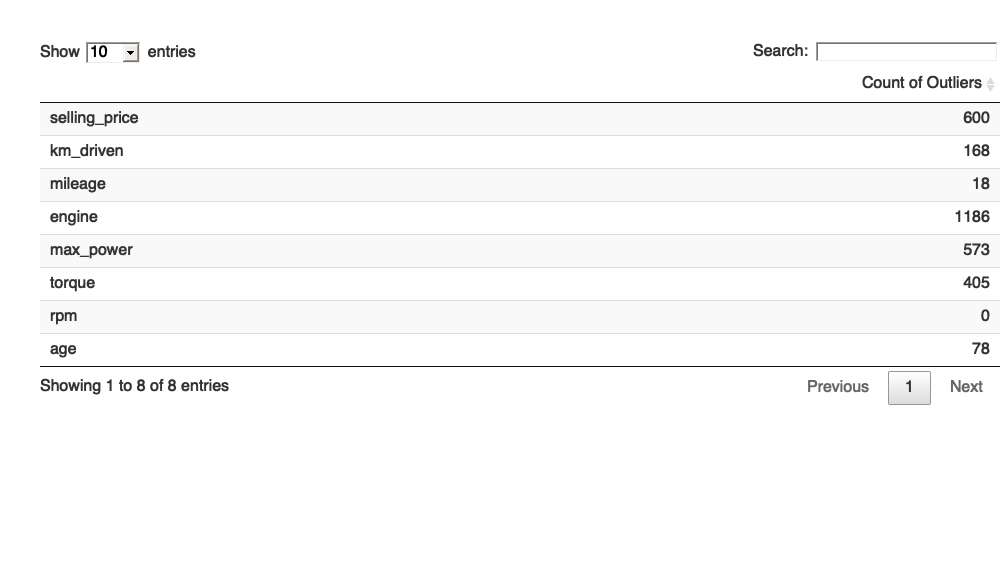
The minimum value of max power also came out to be **0**. We found that some of the cars contain *0 bhp* values as their max power. On further analysis, we would need to impute these values as well.

Since, the cars are manufactured in the time frame of 1983 to 2020, the attributes (mileage, max power, engine, torque, and rpm) are very dispersed.

Mileage has the least skewed distribution as its mean and median are almost same for it.

**Identify the Outliers in numeric variables**

num\_cols <- c("selling\_price", "km\_driven", "mileage", "engine", "max\_power", "torque", "rpm", "age")  
outlier\_val <- sapply(car\_sales[num\_cols], function(x) boxplot.stats(x)$out)  
min\_outlier <- data.frame('Count of Outliers' = sapply(outlier\_val, length), check.names = FALSE)  
DT::datatable(min\_outlier)



Observation - The count of outliers corresponding to their features have been tabulated in the above table. The count of outliers in the feature ‘Selling Type’ is 600 and ‘engine’ has outliers in north of 1000. We found that ‘RPM’ has no outliers in the data set.

These outliers will need to be dealt with before building the regression models.

### 

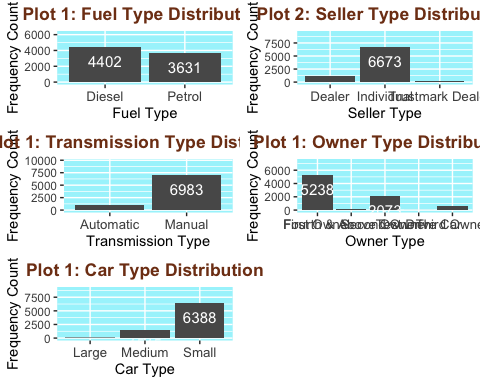
### Univariate Analysis

**Frequency plot of the categorical variables**

We have various categorical features which can provide insightful information about the data set. The features for which we have formulated the frequency tables are :

* Fuel Type
* Seller Type
* Transmission Type
* Owner Type
* Car Type

# Frequency Table of Brands  
freqTable\_fuel <- count(car\_sales, car\_sales$fuel)  
colnames(freqTable\_fuel) <- c("Fuel Type", "Frequency")  
  
freqTable\_sellerType <- count(car\_sales, car\_sales$seller\_type)  
colnames(freqTable\_sellerType) <- c("Seller Type", "Frequency")  
  
freqTable\_transmission <- count(car\_sales, car\_sales$transmission)  
colnames(freqTable\_transmission) <- c("Transmission", "Frequency")  
  
freqTable\_owner <- count(car\_sales, car\_sales$owner)  
colnames(freqTable\_owner) <- c("Owner Type", "Frequency")  
  
freqTable\_carType <- count(car\_sales, car\_sales$car\_type)  
colnames(freqTable\_carType) <- c("Car Type", "Frequency")  
  
bar\_fuel <- ggplot(freqTable\_fuel)+  
 geom\_bar(  
 mapping = aes(  
 x = `Fuel Type`,   
 y = Frequency  
 ),  
 stat="identity"  
 ) +  
 theme(panel.background = element\_rect("#a7f4fc")) +  
 labs(  
 title = "Plot 1: Fuel Type Distribution",  
 x = "Fuel Type",  
 y = "Frequency Count"  
 ) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, colour = "#7F3D17", face = "bold"),  
 axis.text.x = element\_text(vjust = 1, size = 10)  
 ) +  
 scale\_y\_continuous(labels = comma) +  
 ylim(0, max(freqTable\_fuel$Frequency) \* 1.4) +  
 geom\_text(aes(x = `Fuel Type`, y = Frequency, label = Frequency), vjust = 1.8, color = "WHITE", size = 4)  
  
bar\_sellerType <- ggplot(freqTable\_sellerType)+  
 geom\_bar(  
 mapping = aes(  
 x = `Seller Type`,   
 y = Frequency  
 ),  
 stat="identity"  
 ) +  
 theme(panel.background = element\_rect("#a7f4fc")) +  
 labs(  
 title = "Plot 2: Seller Type Distribution",  
 x = "Seller Type",  
 y = "Frequency Count"  
 ) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, colour = "#7F3D17", face = "bold"),  
 axis.text.x = element\_text(vjust = 1, size = 10)  
 ) +  
 scale\_y\_continuous(labels = comma) +  
 ylim(0, max(freqTable\_sellerType$Frequency) \* 1.4) +  
 geom\_text(aes(x = `Seller Type`, y = Frequency, label = Frequency), vjust = 1.8, color = "WHITE", size = 4)  
  
bar\_transmission <- ggplot(freqTable\_transmission)+  
 geom\_bar(  
 mapping = aes(  
 x = Transmission,   
 y = Frequency  
 ),  
 stat="identity"  
 ) +  
 theme(panel.background = element\_rect("#a7f4fc")) +  
 labs(  
 title = "Plot 1: Transmission Type Distribution",  
 x = "Transmission Type",  
 y = "Frequency Count"  
 ) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, colour = "#7F3D17", face = "bold"),  
 axis.text.x = element\_text(vjust = 1, size = 10)  
 ) +  
 scale\_y\_continuous(labels = comma) +  
 ylim(0, max(freqTable\_transmission$Frequency) \* 1.4) +  
 geom\_text(aes(x = Transmission, y = Frequency, label = Frequency), vjust = 1.8, color = "WHITE", size = 4)  
  
bar\_owner <- ggplot(freqTable\_owner)+  
 geom\_bar(  
 mapping = aes(  
 x = `Owner Type`,   
 y = Frequency  
 ),  
 stat="identity"  
 ) +  
 theme(panel.background = element\_rect("#a7f4fc")) +  
 labs(  
 title = "Plot 1: Owner Type Distribution",  
 x = "Owner Type",  
 y = "Frequency Count"  
 ) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, colour = "#7F3D17", face = "bold"),  
 axis.text.x = element\_text(vjust = 1, size = 10, angle = 0)  
 ) +  
 scale\_y\_continuous(labels = comma) +  
 ylim(0, max(freqTable\_owner$Frequency) \* 1.4) +  
 geom\_text(aes(x = `Owner Type`, y = Frequency, label = Frequency), vjust = 1.8, color = "WHITE", size = 4)  
  
bar\_carType <- ggplot(freqTable\_carType)+  
 geom\_bar(  
 mapping = aes(  
 x = `Car Type`,   
 y = Frequency  
 ),  
 stat="identity"  
 ) +  
 theme(panel.background = element\_rect("#a7f4fc")) +  
 labs(  
 title = "Plot 1: Car Type Distribution",  
 x = "Car Type",  
 y = "Frequency Count"  
 ) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, colour = "#7F3D17", face = "bold"),  
 axis.text.x = element\_text(vjust = 1, size = 10)  
 ) +  
 scale\_y\_continuous(labels = comma) +  
 ylim(0, max(freqTable\_carType$Frequency) \* 1.4) +  
 geom\_text(aes(x = `Car Type`, y = Frequency, label = Frequency), vjust = 1.8, color = "WHITE", size = 4)  
  
grid.arrange(bar\_fuel, bar\_sellerType, bar\_transmission, bar\_owner, bar\_carType)



Observation - From the above graphs, we can figure out the following points :

There is not much difference in between the frequency of diesel and petrol cars.

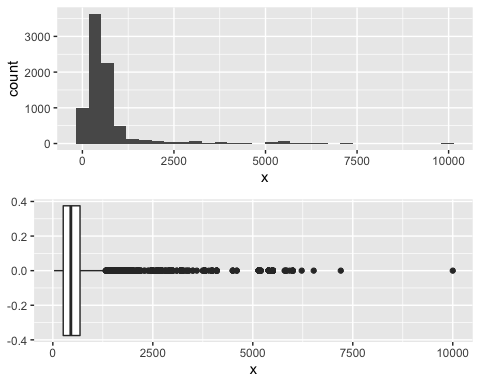
Individual seller type accounts for the most percentage of the cars in the data set. 75% of the cars on CarDekho.com belong to the ‘Individual’ seller type.

Majority of customers prefer cars of ‘Manual’ transmission type over ‘Automatic’ and of ‘Small’ size.

**Histogram and Box plot of numeric variables**

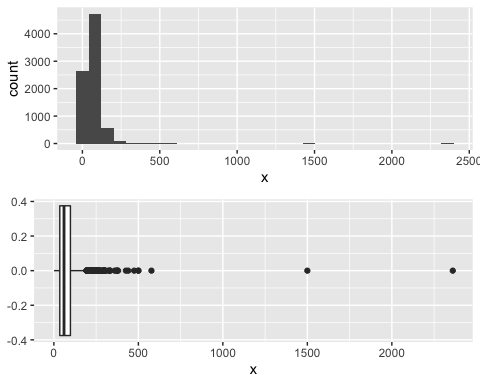
In the following section, we’re plotting the histograms and boxplots to understand the uni-variate distribution of the numeric variables.

# Function to plot graph  
univariate\_plot <- function(x){  
 grid.arrange(ggplot(data = car\_sales) + geom\_histogram(mapping = aes(x)),  
 ggplot(data = car\_sales) + geom\_boxplot(mapping = aes(x)))  
}  
  
#Selling Price  
univariate\_plot(car\_sales$selling\_price)



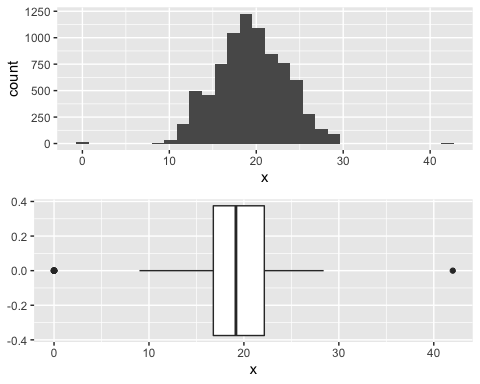
Observation - Most of the data is concentrated in the range of 0 to 1300 and any value above 1325 are outliers.

#Kilometer Driven  
univariate\_plot(car\_sales$km\_driven)



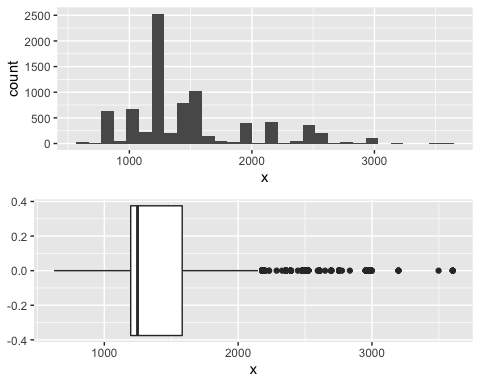
Observation - Mahindra XUV500 W6 2WD and Hyundai i20 Asta 1.2 are outliers to the data set as they have been driven over 150,000 kms.

#Car mileage  
univariate\_plot(car\_sales$mileage)



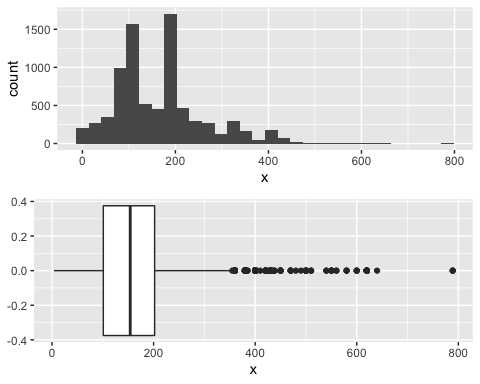
Observation - There are 2 outliers in both of the extremes. Otherwise, the data seems to be normally distributed.

#Engine BHP  
univariate\_plot(car\_sales$engine)



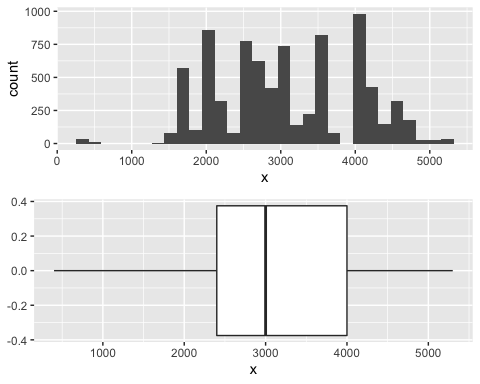
Observation - A fair share of cars belong to the engine types of 1200 CC. Records in north of 2200 CC are outliers to the data set.

# Torque  
univariate\_plot(car\_sales$torque)



Observation - The data set seems to be normally distributed for the feature ‘torque’ with the presence of some outliers.

# RPM  
univariate\_plot(car\_sales$rpm)



Observation - There are no outliers in the data set for the feature ‘rpm’ and no legible distribution can be figured out from the data.

**Check for the normality of numeric variables**

Q-Q Plot is used to visualise the normality of the numeric data. If the ‘Sample’ and ‘Theoretical’ quantiles fall in the same line, those features can be considered normally distributed.

# Check Normality using Shapiro-Wilks Test  
#shapiro.test(car\_sales$selling\_price)  
  
# Function to plot graph  
qq\_plot <- function(numeric\_feature, mainTitle) {  
 qqnorm(numeric\_feature, pch = 5, frame = TRUE, main = mainTitle)  
 qqline(numeric\_feature, col = "#52ffbf", lwd = 2)  
}  
  
# Changing Plot Matrix Size to 3x2.  
par(mfrow = c(3,3))  
  
# Check Normality using Q-Q Plot of 'Selling Price' Feature.  
qq\_plot(car\_sales$selling\_price, "Selling Price")  
  
# Check Normality using Q-Q Plot of 'KMs Driven' Feature.  
qq\_plot(car\_sales$km\_driven, "KMs Driven")  
  
# Check Normality using Q-Q Plot of 'Mileage' Feature.  
qq\_plot(car\_sales$mileage, "Mileage")  
  
# Check Normality using Q-Q Plot of 'Engine' Feature.  
qq\_plot(car\_sales$engine, "Engine")  
  
# Check Normality using Q-Q Plot of 'Max Power' Feature.  
qq\_plot(car\_sales$max\_power, "Max Power")  
  
# Check Normality using Q-Q Plot of 'Torque' Feature.  
qq\_plot(car\_sales$torque, "Torque")  
  
# Check Normality using Q-Q Plot of 'RPM' Feature.  
qq\_plot(car\_sales$rpm, "RPM")  
  
# Resetting Plot Matrix Size to 1x1.  
par(mfrow = c(1,1))

Diagram, engineering drawing

Description automatically generated

Observation - The features ‘mileage’ and ‘KMs driven’ can be considered as normally distributed and other features will need to be treated to make them normally distributed.

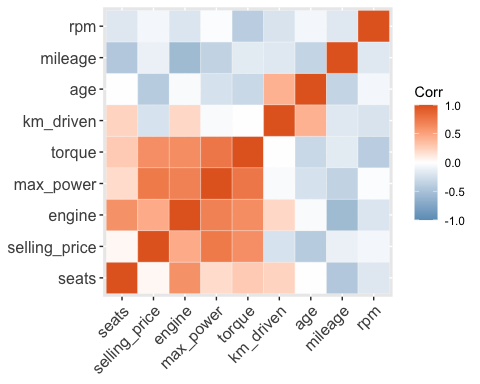
### 

### Multivariate Analysis

**Correlation matrix of numeric variables**

Understanding the correlation between the numeric features of the data set.

#install.packages("ggcorrplot")  
library(ggcorrplot)  
  
data\_corr <- car\_sales[, c("selling\_price", "km\_driven", "mileage", "engine",   
 "max\_power", "seats", "torque", "rpm", "age")]  
corr <- round(cor(data\_corr), 2)  
  
ggcorrplot(corr, hc.order = TRUE, outline.col = "white",   
 ggtheme = ggplot2::theme\_gray, colors = c("#6D9EC1", "white", "#E46726"))

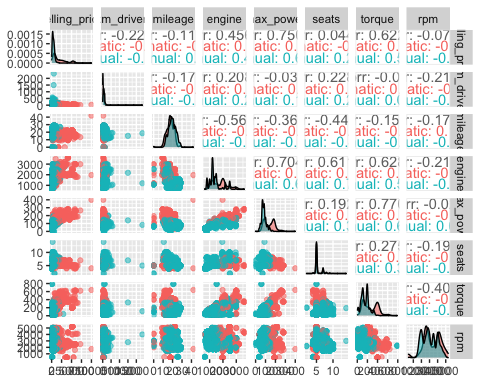


Observation - ‘Max Power’ seems to be highly correlated with ‘Torque’ and ‘Selling Price’.

**Pairplot of all the numeric variables**

Using pairplots, we can figure out the correlations (Positive, Negative, No Correlation) between the attributes (features) of the data set.

#install.packages("GGally")  
library(GGally)  
ggpairs(car\_sales, columns = c(2, 3, 8:13), aes(color = transmission, alpha = 0.5))

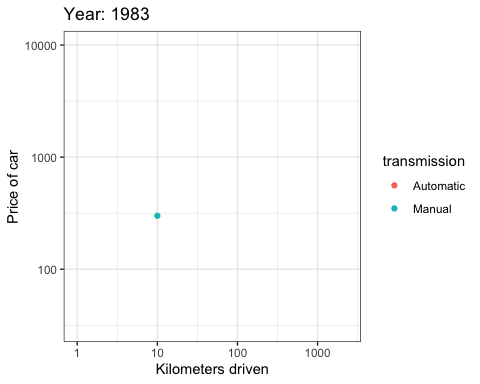


Observation - There are positive correlations between ‘Max Power’ and ‘Engine’. No correlations can be found between ‘Kms Driven’ and other attributes.

**Yearly trend of price**

Scatterplot between Kms driven and Selling price over the year. The plot has been shown in the form of GIF instead of static plot.

#install.packages("gganimate")  
#install.packages("gifski")  
#install.packages("av")  
library(ggplot2)  
library(gganimate)  
  
# Make a ggplot, but add frame=year: one image per year  
gif1 <- ggplot(car\_sales, aes(km\_driven, selling\_price, color = transmission)) +  
 geom\_point() +  
 scale\_x\_log10() +  
 scale\_y\_log10() +  
 theme\_bw() +  
 # gganimate specific bits:  
 labs(title = 'Year: {frame\_time}', x = 'Kilometers driven', y = 'Price of car') +  
 transition\_time(as.integer(year)) +  
 ease\_aes('linear')  
  
# Save as GIF:  
animate(gif1, nframes = 100, fps = 5, end\_pause = 20, renderer=gifski\_renderer("test.gif"))



Observation - 1. The number of cars sold from 1983 till 2000 were very less and most of them were manual cars. 2. After 2005, the demand of automatic cars increased which peaked in 2015 overtaking manual cars.

# CONCLUSION

The data set of car sales by CarDekho.com has provided various insights about the types of cars sold in the car industry and the patterns between them. The data set contains 8128 data points along with 13 features related to car details, engine details, and sale details.

Data cleaning and data pre-processing are important to prepare the data in the correct format before building the regression model. We extracted numerical values from text fields like engine, mileage, torque, along with brand information from car name.

Unnecessary data that skew the results were also filtered out.

To ensure that the numerical variables are not skewed, it’s crucial to remove or impute the missing values and outliers. We imputed the missing values using kNN imputation method.

Feature engineering and exploratory data analysis were performed to gather more meaningful information from the data.

Derived variables were created using the existing features and skewed variables were scaled.

Apart from this, various data visualization, like box plot, frequency plot, histogram, pair plot, correlation matrix and scatter plot were created to understand the uni-variate distribution and multi-variate relationship of the data.

# BIBLIOGRAPHY

1. Vehicle dataset. (2020, October 24). Kaggle. <https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho>
2. Xie, Y. C. D. (2021, October 7). R Markdown Cookbook. R Markdown. Retrieved October 30, 2021, from <https://bookdown.org/yihui/rmarkdown-cookbook/>
3. Bluman, A. (2017). Elementary Statistics: A Step By Step Approach (10th ed.). McGraw-Hill Education.
4. Kabacoff, R., I. (2022). R in Action, Third Edition. Manning.
5. CarDekho. (2021). About Us | CarDekho.com. <https://www.cardekho.com/info/about_us>
6. F. (2021, April 2). tidyverse in r – Complete Tutorial. R-Bloggers. Retrieved October 30, 2021, from <https://www.r-bloggers.com/2021/04/tidyverse-in-r-complete-tutorial/>
7. D. (2021a, March 22). Data Analytics for Car Dealers. Automated Metrics. <https://www.automatedmetrics.io/data-analytics-for-car-dealers/>

# 

# APPENDIX

The RMD file of the analysis is included with the analysis report.



Dataset: -



Link to report:-

<https://rpubs.com/akash_raj/834927>