VISUALIZATION AND ANALYSIS OF INDIAN AUTOMOBILE DATASET USING MACHINE LEARNING IN R

FINAL REVIEW REPORT

Submitted by

Kollipara Venkata Naga Hemanth (18BCE0538)

Prepared For

Data Visualization (CSE3020) – PROJECT COMPONENT

Submitted To

Dr. Dilip Kumar Choubey

Assistant Professor (Senior)

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Table of Content

Abstract

The automobile industry today is the most profitable industry. Due to increase in the income in both rural and urban sector and availability of easy finance are the main drivers of high-volume car segments. Further competition is heating up with host of new players coming in and global manufacturers. This analysis and visualization of the automobile dataset will be helpful for the existing and new entrant car manufacturing companies in India to find out the customer expectations and the current analysis of various thousands of variants of vehicles that are running in the market currently. Indian Automobile car business is influenced by the presence of many national and multinational manufacturers which are covered in the dataset which consisted of several tens and hundreds of manufacturers from around the world. This project presents various levels of visualizations using barplots, histograms, scatter plots, boxplots, violinplots etc. And data analysis of consumer automobiles to get a proper understanding of consumer buying and pricing behavior of vehicles that are currently in market to predict prices of future cars based on their other attributes.

The objective of this project is to visualize and provide various insights from the considered Indian automobile dataset by performing data analysis that utilizing machine learning algorithms in R programming language. The considered dataset is of Indian cars that consists of various features such as model, manufacturer, year, transmission, engine, power etc. The insights that could be estimated from this dataset would be feature such as price of a specific car model that could be estimated using the other attributes of that particular car model using machine learning algorithms like Linear Regression. The objective also includes the study of various attributes of the considered Indian automobile dataset and finding the relationship or statistically, finding the correlation between them and visualizing the findings. The result of finding this relationship between various attributes of a vehicle will provide useful insights in building in a prediction model capable of predicting the price of a vehicle based on the other attributes. This kind of an analytics will help the consumers to decide the selling price of a vehicle without rough estimates which sometimes may underestimate the price of vehicles leading to loss of customer automobile value. Thus, this kind of analytics will certainly have a practical industry use case which might be useful to create end products to consumers which are capable of providing insights of various attributes of automobiles and also to look into analytics and knowing the segment of automobiles that are successful in the market.

Introduction

Background

The project aims to perform various visualizations and perform data analysis on the automobile dataset in order to determine the various relationships between different features of the vehicle. The visualization starts with univariate analysis, analyzing the data in perspective of a single attribute then with bivariate analysis and then with multivariate which deals with more than two attributes at the same time. In this project we are using the Indian automobile dataset and perform various analysis of the attributes like the capacity and power of the automobiles using R programming language. The insights that could be estimated from this dataset would be feature such as price of a specific car model that could be estimated using the other attributes of that particular car model using machine learning algorithms like linear regression or polynomial regression. Finally, we shall be building a machine learning model that is capable of predicting the price of a vehicle based on the other attributes of the automobile.

Objective

The primary objective of this project is to visualize and provide various insights from the considered Indian automobile dataset by performing data analysis that utilizing machine learning algorithms in R programming language. Also, to derive a prediction model that can appropriately estimate the pricing of various car models with their parameters like manufacturer, year, horsepower and so on. The considered dataset is of Indian cars that consists of various features such as model, manufacturer, year, transmission, engine, power etc. The insights that could be estimated from this dataset would be feature such as price of a specific car model that could be estimated using the other attributes of that particular car model using machine learning algorithms like linear regression. The objective also includes the study of various attributes of the considered Indian automobile dataset and attempts to consolidate the findings of the relationship between the attributes or statistically, finding the correlation between them and visualizing the findings. Of these features some of them might be a redundant and might be a good contributor to the prediction model and the task of eliminating such attributes also shall be considered.

Motivation

The reason for choosing this particular project was because of its practical applications involved in it. Many people often face the problem of pricing vehicles while they are selling it online. Thus, a prediction model capable of pricing of a particular model of a car can be useful when an owner wants to sell their vehicle. Also, with the help of some attributes of the car like manufacturer, engine capacity, horsepower, the price of an upcoming car can be closely estimated without its release. These kind of prediction models can be used in online websites to provide prediction to the website users, either for estimating price of an vehicle before its being revealed by the manufacturer using data analysis on the data in the dataset or the predictions will be really helpful while users are selling their vehicles.

Contributions of project

The data taken into consideration is taken from Kaggle website which hosts a variety of datasets from all over the world. The dataset contains 5975 rows and 14 columns, cars with their variants there are more than 1200 model car variants to study. The data concerns pricing of vehicles in rupees, to be predicted in terms of 5 multivalued discrete attributes - manufacturer, location, fuel type, transmission, ownership, ownership and 6 continuous attributes - year, km driven, engine, seats, horsepower. There is a variety of models which can be studied. Car prices ranges from few lacs to few crores. The dataset consisted of many missing values and some required attributed were wrongly recorded as zero values like mileage which can only be a non-zero value. Since the rows that consisted missing values only amount to less than one percent of the data, rows with missing values are deleted and some rows with zero values are imputed with the mode of that particular attribute.

Organization of project

The organization of the entire project is divided into two parts which are the visualization and the data analysis parts of the project. The visualization part consists of univariate analysis, analyzing the data in perspective of a single attribute then with bivariate analysis, analysis using two attributes and then with multivariate which deals with more than two attributes at the same time. The data analysis part deals with finding the relationship between various attributes and building a prediction model capable of predicting the price of a vehicle based on the other attributes.

Project Resource Requirements

Software Requirements

- R software environment for statistical computing
- R Studio IDE (Integrated Development Environment)
- Ggplot2 graphic data visualization package
- Plotly visualization library
- Dplyr data manipulation package
- MLBench package
- Caret machine learning package

Hardware Requirements

- Intel-compatible chipset
- 1GB RAM
- 20GB of free disk space
- Windows or Linux or Mac

LITERATURE SURVEY

Background

In this section literature survey of various papers on topics of linear and polynomial regression is performed and analyzed various methodologies of each paper and their respective advantages and drawbacks.

Literature Review

Authors	Method	Purpose	Advantages	Disadvantages
Dacheng Tao	Multivariate linear regression and its principal component regression, deal well with the situations of data having low-dimensional vector. When the dimension grows higher, it leads to the under-sample problem.	To find two low-dimensional coefficients so that the principal components selection problem is avoided.	Numerical experiments carefully discussed the influences of all the factors in models and showed that MMR is more effective in case of multilinear regression problems.	There are two types of error in PCR, which are resulted from small sample size and noise, respectively.
Wolfgang Tysiak	Regression Analysis of Intrinsic Linear Models with Automated Transformations of Monotone Predictors.	To detect several monotone transformations by means of logistic functions in the predictors.	The technique used to optimize the transformation and the OLS-criteria in one step can also be applied to discover the structure of distributed lags within the data.	due to the numerical optimization a relatively large sample size is needed.

Jinfang Sheng	BTP Prediction model based on regression analysis	The prediction values and correlated variables are sent back to L1 system through BTP control model to realize a closed-loop control system.	prediction result is calculated based on both the prediction value from N-BTP-Prediction model and the compensation value from linear regression model.	prediction accuracy is a little bit low
H. Shakouri	Fuzzy linear regression models with absolute errors and optimum uncertainty	To find the parameters of a linear fuzzy regression. It is designed and solved, by which a minimum degree of acceptable uncertainty.	This approach, is much more accurate, compared to the other methods	This is idea of reducing distance between the output of the possibility model and the measured output, while trying to increase their Conjunction.
A. Martinez-Coll	Comparison of near infrared spectroscopy (NIRS) signal quantitation by multilinear regression	To compare signal quantitation by conventional multiple regression.	The best multilinear regressions had significant shortcomings with regards to underestimate flow values below the mean.	Other methods are more accurate than multilinear regression methods
X. Feng	Contact temperature prediction of high voltage switchgear based on multiple linear regression model	To analyse and process a monitoring point data, the regression model	regression in multiple linear regression has a high accuracy in	multiple linear regression uses big data.

		of temperature is established by using multiple linear regression method.	the long-term prediction of temperature.	
S. Yamamoto	Polynomial regression-based model-free predictive control for nonlinear systems	to predict the optimal control input it utilizes massive stored and observed input/output dataset.	it shows a satisfactory control performance when large datasets are available around the reference trajectory.	it is preferable to the previously maintained datasets.
Shunxin Wang	Parametric modelling of the coupling channel of conducted interference based on multi-linear regression model	It is necessary to understand the characteristic of the coupling channel, so that the EMI of the EUT can be effectively analyzed and restrained	modelling method provides an approach to analyses the internal interference coupling channel of the EUT that needs to pass the CE test standard.	parametric modelling method in frequency domain and its evaluation method based on the multi-linear regression model
Ahmed Karama	A Multi Linear Regression Approach for Handling Missing Values with Unknown Dependent Variable	To predict Missing values for a data set with Unknown Dependent variable.	Splitting the data set into training and test sets, finding the dependent variable from the dataset while performing training set, then using the model to predict missing	It outperforms BGA for all the missing value ratios.

			values in the test set	
M. C. Roziqin	A comparison of Monte Carlo linear and dynamic polynomial regression in predicting dengue fever case	comparing and calculating the deviation value of the predicted number of cases, as a result of the prediction, to the number of actual cases.	dynamic polynomial regression able to predict very well as compared to Monte Carlo linear regression method	the smallest MSE as compared to linear regression, exponential regression, and quadratic regression
S.Edebalı	Prediction of wastewater treatment plant performance using multilinear regression	To understand the effects of the tested parameters, regression function was developed in Multilinear regression method	High correlation coefficient of determination and lowest mean squared error value (MSE) between the measured and predicted output variables	coefficient of determination and mean squared error were not obtained with this model
J. Wu	Personalized Collaborative Filtering Recommendation Algorithm based on Linear Regression	to minimize the value of the linear regression cost function to obtain the item label	minimize the value of the cost function. The average deviation between the predicted score and the actual value is calculated. .	Prediction errors are present but smaller than other algorithms
S. Yamamoto	A model-free predictive control method based on polynomial regression	To propose a model-free predictive control method for nonlinear systems on the basis of polynomial	Estimating the coefficients of polynomial regression, an appropriate control input can be determined by	maintaining a rich dataset is important' that is, the dataset must contain input/output data that is near the

		regression.	containing the input/output data of the controlled system.	desired output.
Renato Monteiro	Dimension reduction and coefficient estimation in multivariate linear regression	To formulate the dimension reduction and coefficient estimation in the multivariate linear model.	this method is extended to a non-parametric model for predicting multiple responses	It can be unstable because of discrete nature of selecting the number of factors.
Anatolii V. Omelchenko	Polynomial regression coefficients estimation in finite differences space	To define i-th order regression coefficient with respect to equidistantly spaced samples are finite differences of the same order	allows us to create efficient methods to synthesize estimators for polynomial regression coefficients at presence of correlated disturbances.	roundoff errors on precision of regression coefficients calculation.

Summary

From the papers studies we have concluded that the automobile data needs a machine learning model using polynomial regression model would be the best predictive model for a dataset similar to this. Also, for RSME which is root mean square error will be used as the evaluation metric for quantifying the error parameter. A prediction model that can appropriately estimate the pricing of various car models with their parameters like manufacturer, year, horsepower. Considering a linear relationship might among the attributes in the dataset not always be suffice so, a polynomial model ensures that the attributes related in non linear could be correlated appropriately, making the model much more precise and reducing the error generated by the RSME parameter.

References

[1] W. Tysiak, "Regression Analysis of Intrinsic Linear Models with Automated Transformations of Monotone Predictors," 2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Sofia, 2005, pp. 620-623, doi: 10.1109/IDAACS.2005.283059.

[2] H. Shakouri, G. R. Nadimi and F. Ghaderi, "Fuzzy linear regression models with absolute errors and optimum uncertainty," 2007 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 2007, pp. 917-921, doi: 10.1109/IEEM.2007.4419325.

[3] A. E. Tümer and S. Edebalı, "Prediction of wastewater treatment plant performance using multilinear regression and artificial neural networks," *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, Madrid, 2015, pp. 1-5, doi: 10.1109/INISTA.2015.7276742.

[4] Y. Su, X. Gao, X. Li and D. Tao, "Multivariate Multilinear Regression," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 6, pp. 1560-1573, Dec. 2012, doi: 10.1109/TSMCB.2012.2195171.

[5] B. Wang, Y. Fang, J. Sheng, W. Gui and Y. Sun, "BTP Prediction Model Based on ANN and Regression Analysis," 2009 Second International Workshop on Knowledge Discovery and Data Mining, Moscow, 2009, pp. 108-111, doi: 10.1109/WKDD.2009.179.

[6] A. Martinez-Coll and H. T. Nguyen, "Comparison of near infrared spectroscopy (NIRS) signal quantitation by multilinear regression and neural networks," 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Istanbul, Turkey, 2001, pp. 1625-1628 vol.2, doi: 10.1109/IEMBS.2001.1020525.

[7] S. Wang, F. Dai and T. Zheng, "Parametric modeling of the coupling channel of conducted interference based on multi-linear regression model," 2016 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Beijing, 2016, pp. 1-2, doi: 10.1109/NEMO.2016.7561656.

[8] A. Karama, M. Farouk and A. Atiya, "A Multi Linear Regression Approach for Handling Missing Values with Unknown Dependent Variable (MLRMUD)," 2018 14th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 2018, pp. 195-201, doi: 10.1109/ICENCO.2018.8636126.

[9] H. Li and S. Yamamoto, "Polynomial regression-based model-free predictive control for nonlinear systems," *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Tsukuba, 2016, pp. 578-582, doi: 10.1109/SICE.2016.7749264.

[10] Yuan, M., Ekici, A., Lu, Z. and Monteiro, R. (2007), Dimension reduction and coefficient estimation in multivariate linear regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 69: 329-346. doi:10.1111/j.1467-9868.2007.00591.x

[11] A. V. Omelchenko and O. V. Fedorov, "Polynomial regression coefficients estimation in finite differences space," 2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, 2015, pp. 257-260, doi: 10.1109/RADIOELEK.2015.7129024.

[12] M. C. Roziqin, A. Basuki and T. Harsono, "A comparison of Montecarlo linear and dynamic polynomial regression in predicting dengue fever case," 2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC), Manado, 2016, pp. 213-218, doi: 10.1109/KCIC.2016.7883649.

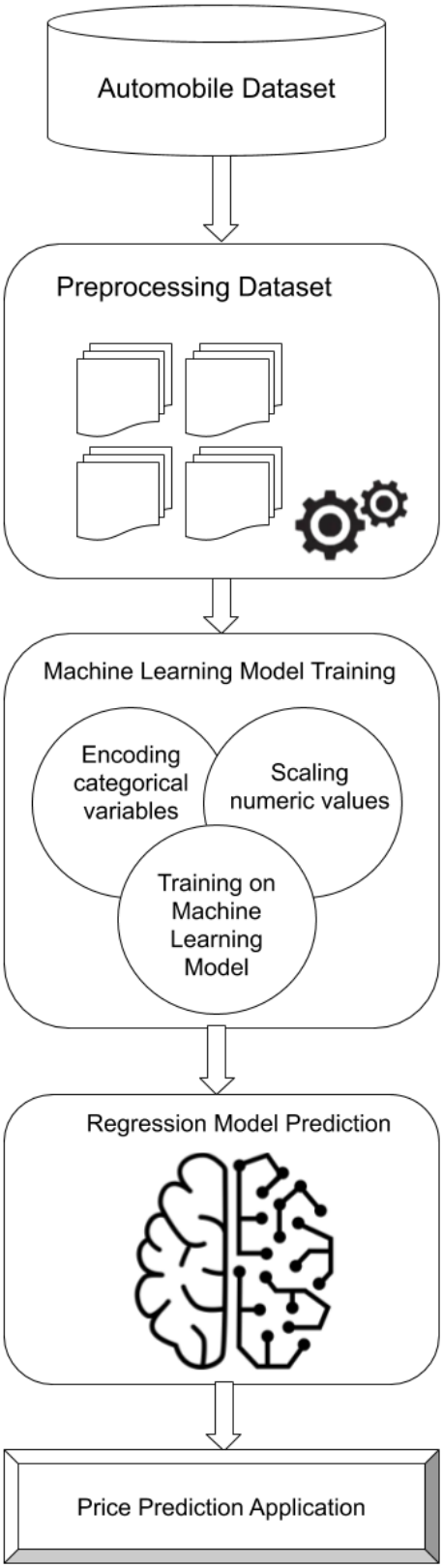
[13] H. Li and S. Yamamoto, "A model-free predictive control method based on polynomial regression," 2016 SICE International Symposium on Control Systems (ISCS), Nagoya, 2016, pp. 1-6, doi: 10.1109/SICEISCS.2016.7470167.

[14] X. Feng, Y. Zhou, T. Hua, Y. Zou and J. Xiao, "Contact temperature prediction of high voltage switchgear based on multiple linear regression model," 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Hefei, 2017, pp. 277-280, doi: 10.1109/YAC.2017.7967419.

[15] B. Leonardi and V. Ajjarapu, "Development of Multilinear Regression Models for Online Voltage Stability Margin Estimation," in IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 374-383, Feb. 2011, doi: 10.1109/TPWRS.2010.2050155.

Proposed Methodology

Proposed Architecture



Methodology

The architecture of the entire project is divided into two parts which are the visualization and the data analysis parts of the project. The visualization part of the project deals with the various plotting of attributes while the data analysis part of the project deals with finding the relationship between various attributes in the dataset.

First the dataset if taken into preprocessing where the data is cleaned of missing and nan values. Also, the data imputation takes place in this step. The dataset consists of many missing values and some required attributed that were false recorded as zero values like mileage which can only be a non-zero value. Since the rows that consisted missing values only amount to less than one percent of the data, rows with missing values are deleted and some rows with zero values are imputed with the mode of that particular attribute.

The visualization part consists of univariate analysis, analyzing the data in perspective of a single attribute then with bivariate analysis, analysis using two attributes and then with multivariate which deals with more than two attributes at the same time. Here the attribute's distributions are visualized using count plots, barplots, histograms, etc. The bivariate analysis is done using scatter plots, box plots, violin plots and so on. Similar plots are used in multivariate analysis but the third or more dimensions are represented on two dimensions by adding colors or size to the plot attributes.

The data analysis is performed on the automobile dataset utilizing machine learning algorithms in order to study the various relationships between attributes of the considered Indian automobile dataset and attempts to consolidate the findings of the relationship between the attributes or statistically, finding the correlation between them and visualizing the findings. Of these features some of them might be a redundant and might be a good contributor to the prediction model and the task of eliminating such attributes also shall be considered. The result of finding this relationship between various attributes of a vehicle will provide useful insights in building in a prediction model capable of predicting the price of a vehicle based on the other parameters like manufacturer, year, horsepower and so on.

Implementation Details and User Manuals

Introduction

The data taken into consideration is taken from Kaggle website which hosts a variety of datasets from all over the world. The dataset contains 5975 rows and 14 columns, cars with their variants. The data concerns pricing of vehicles in rupees, to be predicted in terms of 5 multivalued discrete attributes - manufacturer, location, fuel type, transmission, ownership, ownership and 6 continuous attributes - year, km driven, engine, seats, horsepower.

Implementation Details

DATASET AND PACKAGES

We have imported all the packages and libraries we will be using for the exploration of data. First the data using read csv function and the path for the location of the dataset csv file is given as argument. Exploration and visualization using ggplot and plotly packages in R.

PREPROCESSING

The dataset consists of many missing values and some required attributed that were false recorded as zero values like mileage which can only be a non-zero value. Since the rows that consisted missing values only amount to less than one percent of the data, rows with missing values are deleted and some rows with zero values are imputed with the mode of that particular attribute. Also, the engine capacity and engine power attributes of the data had units appended at the end of the data like cc and hp which are needed to be removed in order to convert the attributes to numerical from object datatype. Now the attributes are ordered according to their datatype. This completes the basic dataset processing.

VISUALIZATION

The visualizations of data are performed which starts with univariate analysis, analyzing the data in perspective of a single attribute then with bivariate analysis and then with multivariate which deals with more than two attributes at the same time. Here the attribute's distributions are visualized using count plots, barplots, histograms, etc. Before performing the bivariate analysis, the values of both the dimensions are scaled in order for the visual plots to appear appropriately. The bivariate analysis is done using scatter plots, box plots, violin plots and so on. Similar plots are used in multivariate analysis but the third or more dimensions are represented on two dimensions by adding colors or size to the plot attributes. Now the data is split into train and test data to perform the model building, training and testing.

FITTING DATA TO REGRESSION MODEL

Now that we know what our data looks good, we use some machine learning models to predict the value of prices of vehicles given the values of the other attributes. We will use caret package to train test and tune various regression models on our data and compare the results. Building evaluating and tuning different regression models using caret machine learning algorithms package. Before that categorical attributes with number of levels in them are identified since the categorical variables cannot be directly trained in the model. Instead we create dummy variables to represent each level in a categorical variable sort of like one hot encoding to represent the category of a particular attribute. The numerical values are then scaled to mean zero and variance one making them scale to the range of zero to one. This scaled data is now used for training the machine learning model. The model used to predict the price of automobile is multivariate regression model and this machine learning model is considered since we need a model capable of handling more than two attributes and therefore multiple regression is used. Multiple Regression is performed using the dummy encoded variables and then trained. Also, there might a probability of a need to considering polynomial regression also, since we the relationship between dependent and independent variables might not always be a linear one. The Price attribute of the data is regressed on the remaining numerical and categorical attributes to create a regression model.

User Manual

The entire project is divided into two parts which are the visualization and the data analysis parts of the project. The visualization part consists of univariate analysis, analyzing the data in perspective of a single attribute then with bivariate analysis, analysis using two attributes and then with multivariate which deals with more than two attributes at the same time. The data analysis part deals with finding the relationship between various attributes and building a prediction model capable of predicting the price of a vehicle based on the other attributes.

Experiment Results and Analysis

Introduction

The visualization of various attributes of the dataset has been done highlighting the various relationships between the attributes of the data. After fitting the model to the data, price prediction can be performed and regression plots are plotted to identify the extent of correlation the attribute has with the independent variable that is price.

Results

UNIVARIATE ANALYSIS

HISTOGRAMS

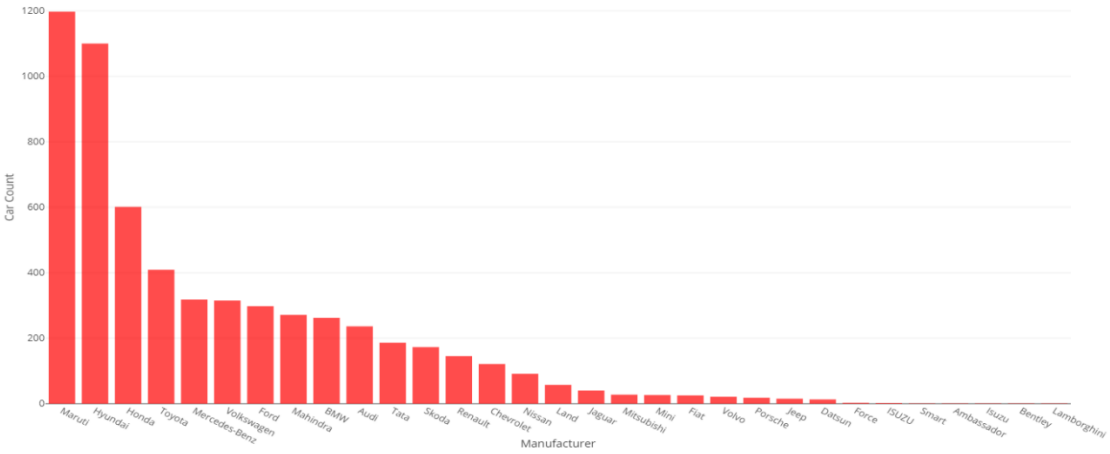


Fig 1.1

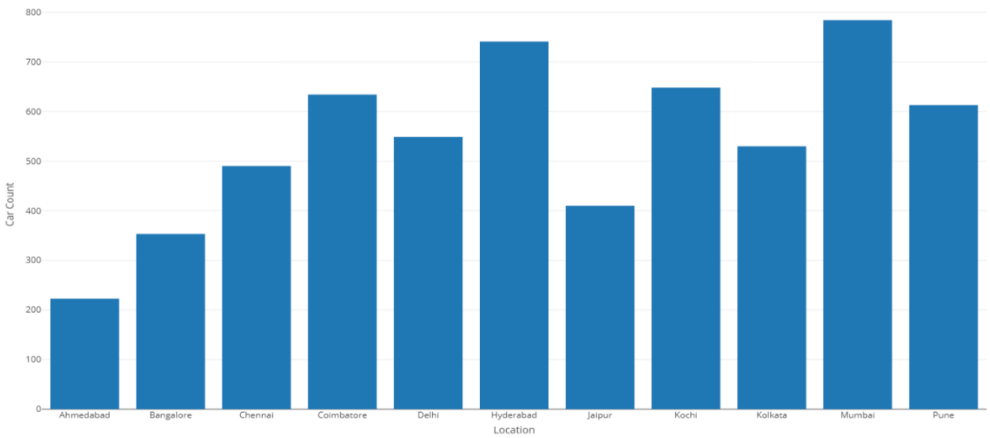


Fig 1.2

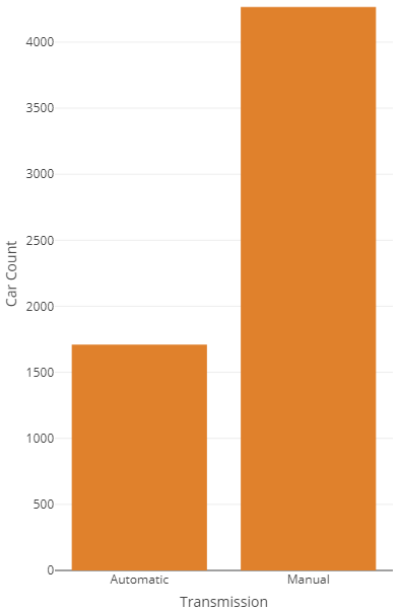


Fig 1.3

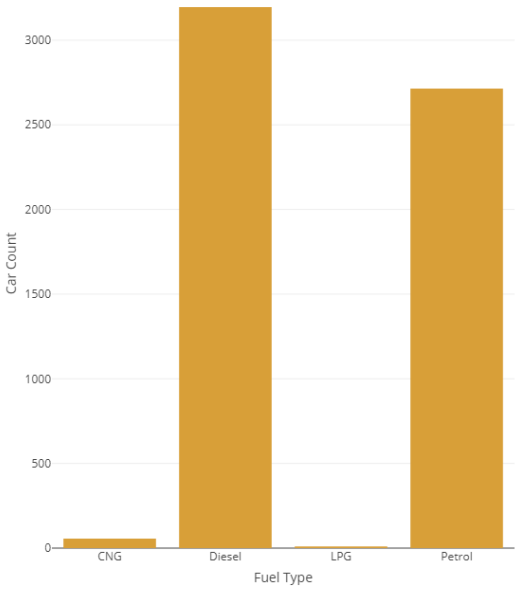


Fig 1.4

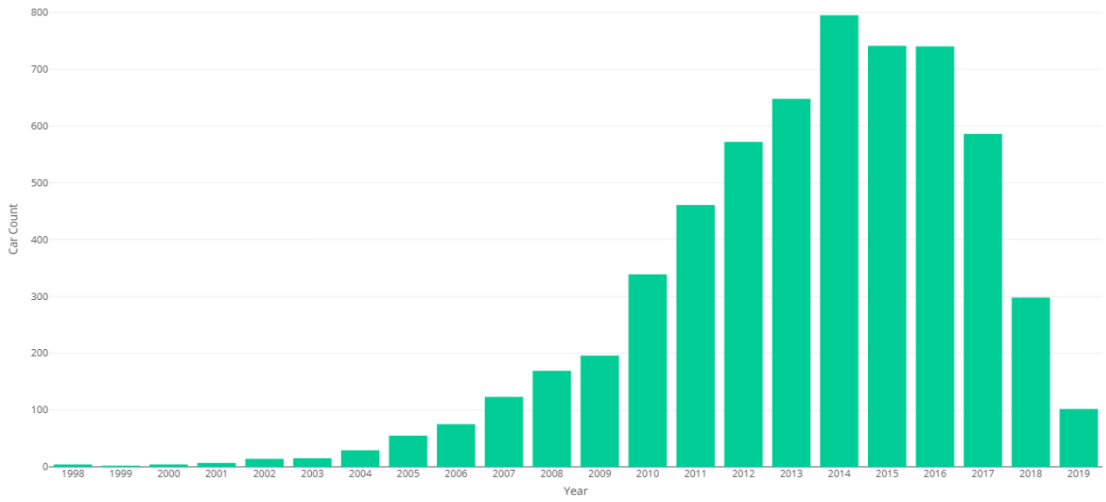


Fig 1.5

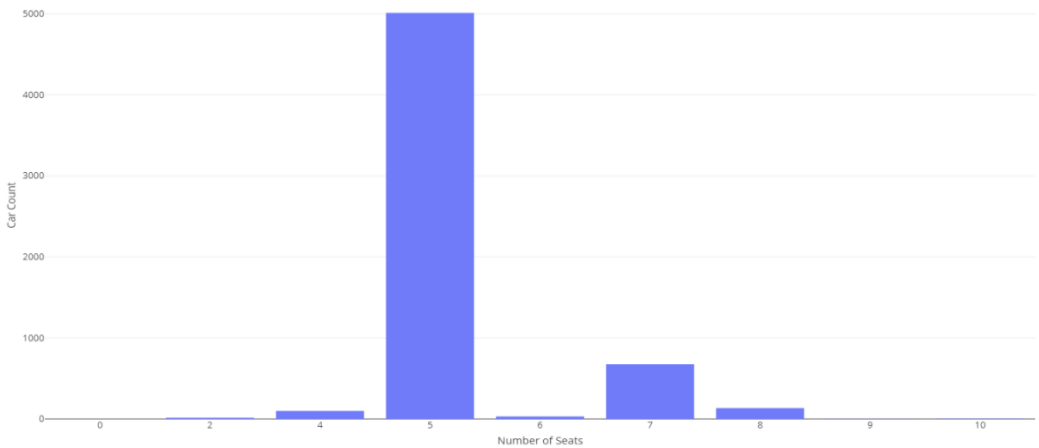


Fig 1.6

BOXPLOTS

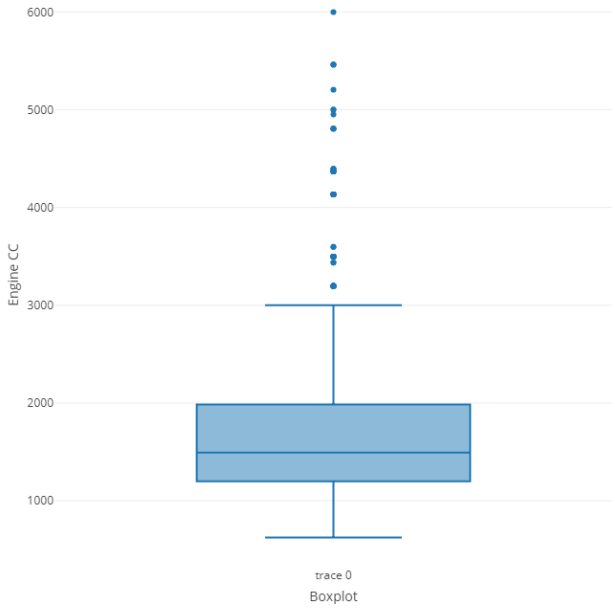


Fig 1.7

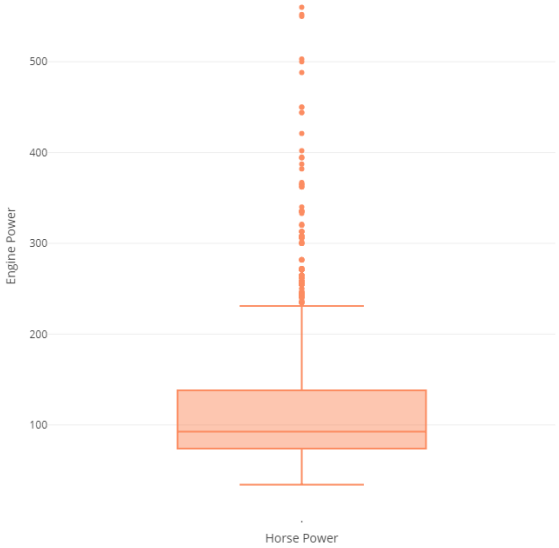


Fig.18

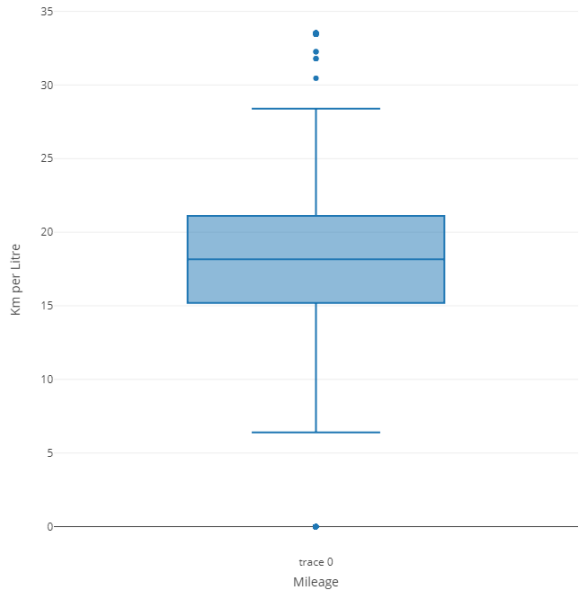


Fig 1.9

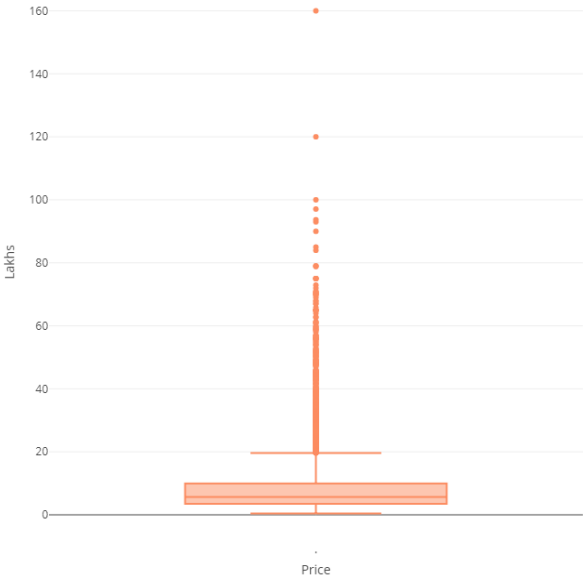


Fig 1.10

BIVARIATE ANALYSIS

SCATTERPLOTS

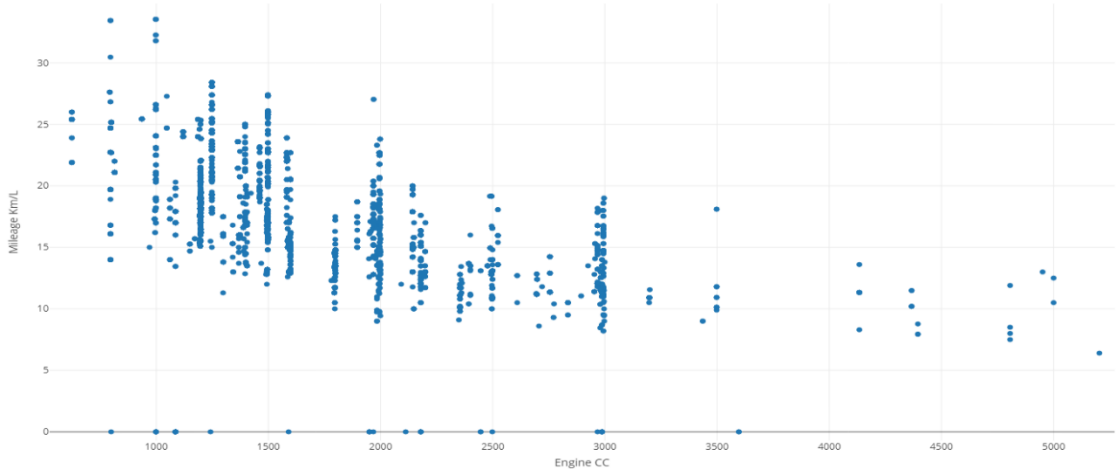


Fig 2.1

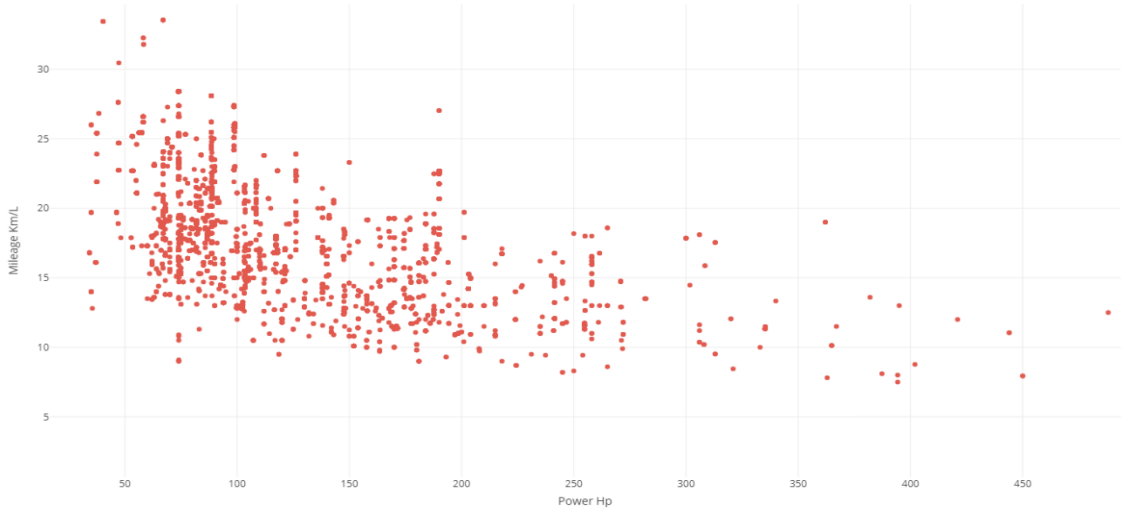


Fig 2.2

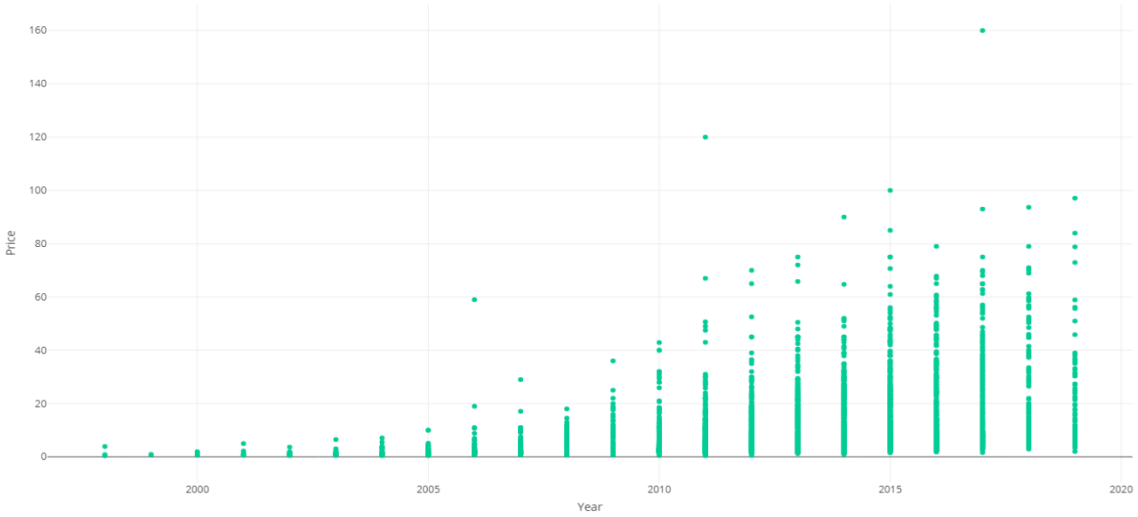


Fig 2.3

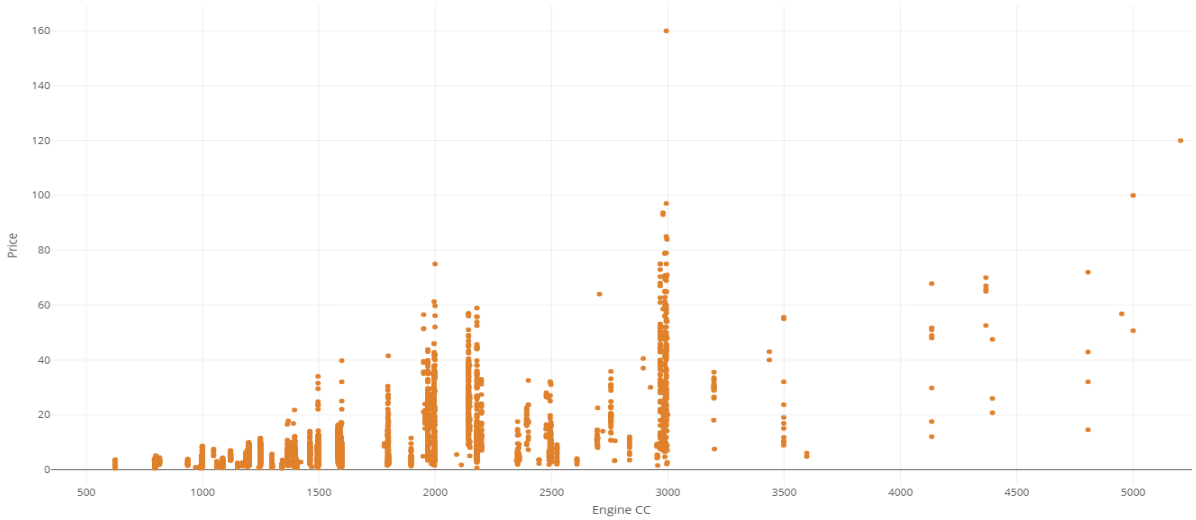


Fig 2.4

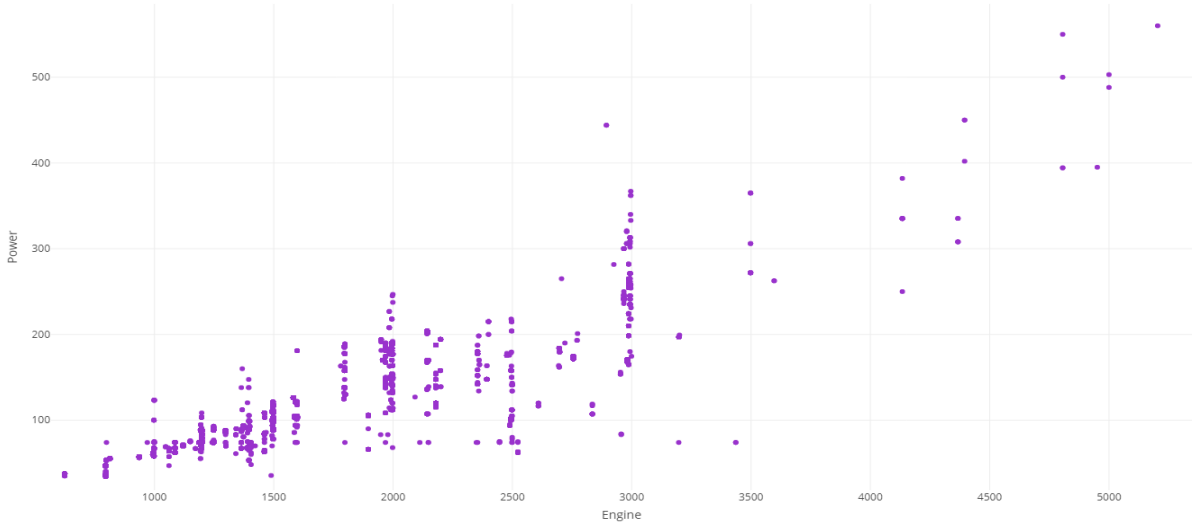


Fig 2.5

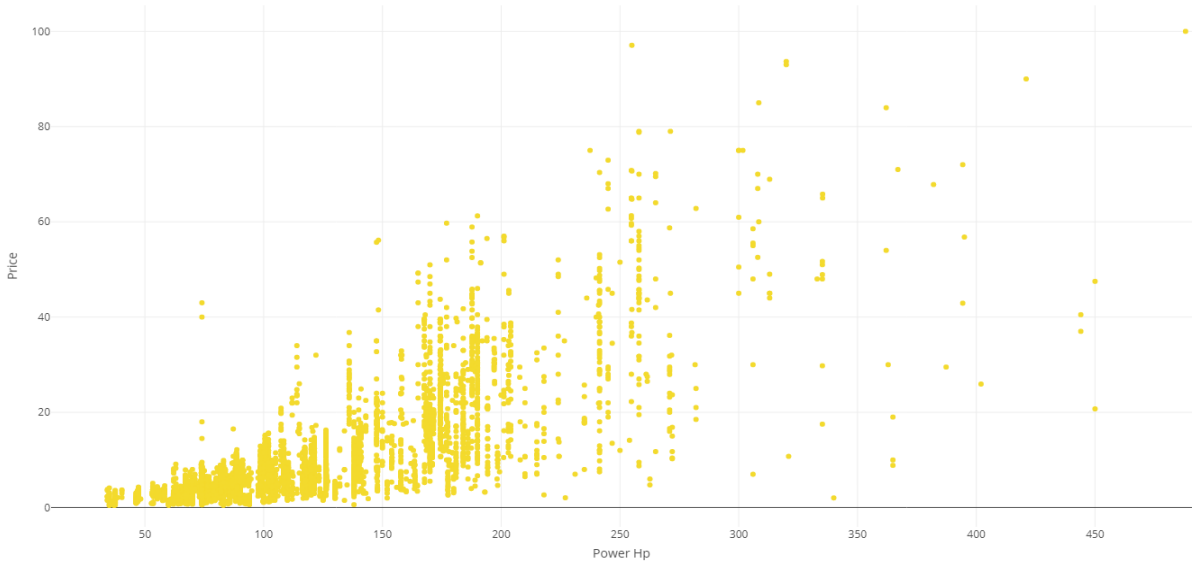


Fig 2.6

BOXPLOTS

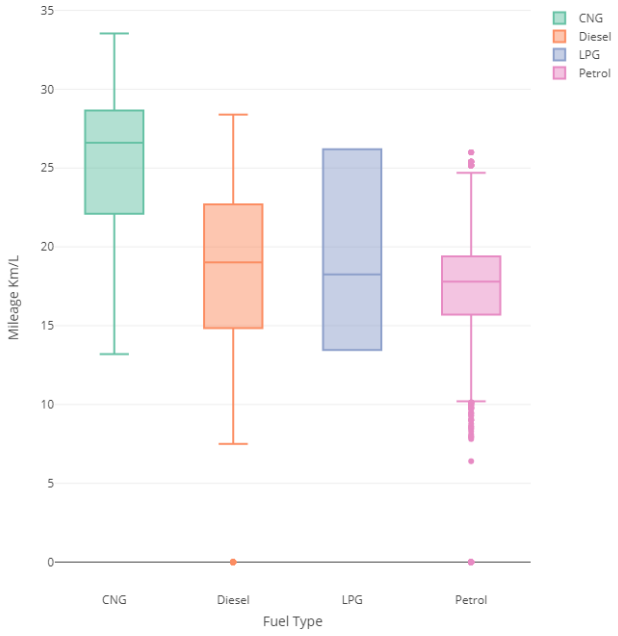


Fig 2.7

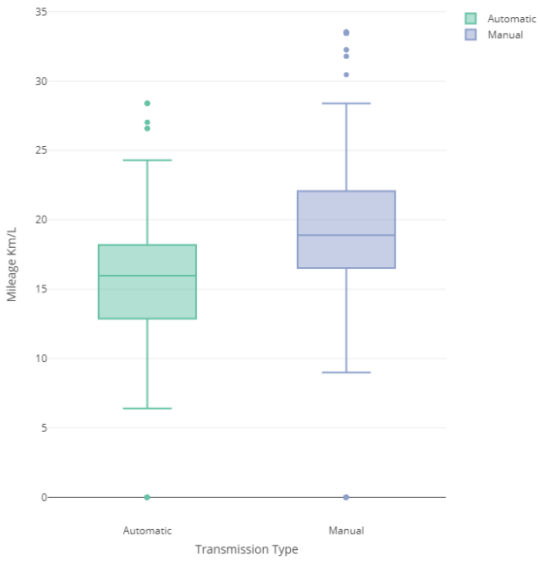


Fig 2.8

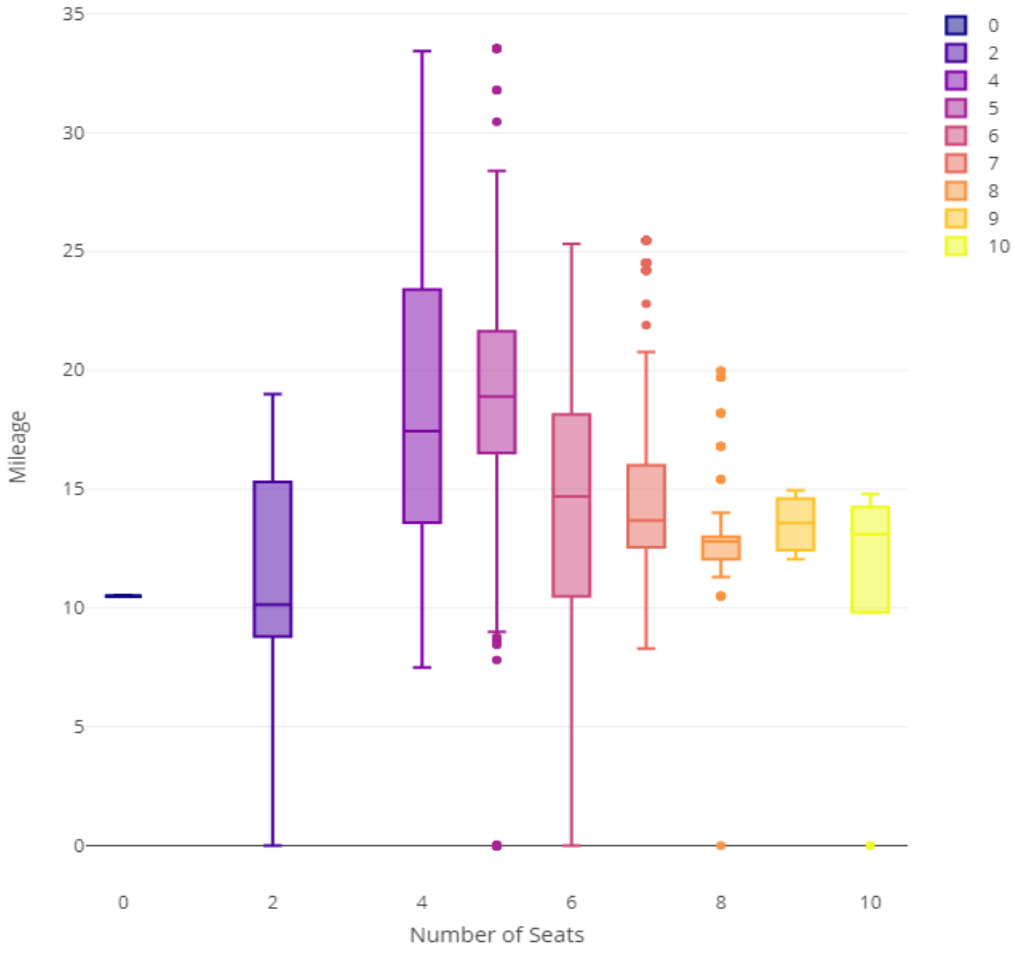
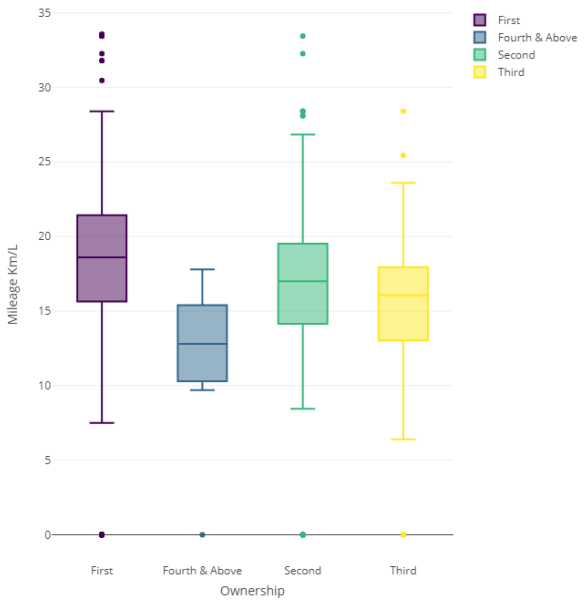
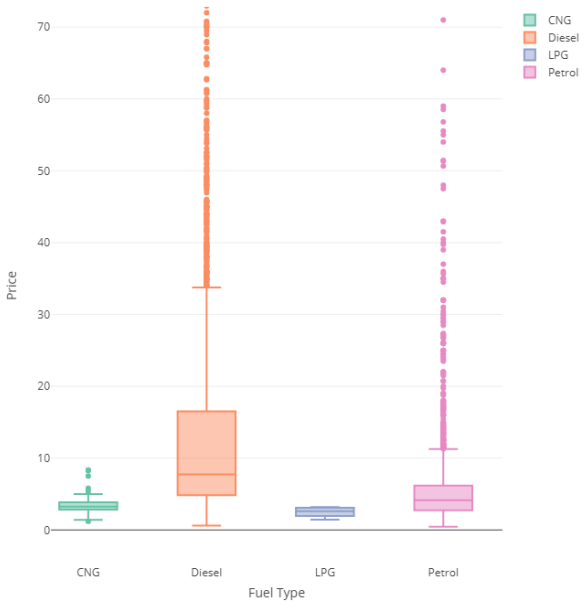


Fig 2.9



2.10



2.11

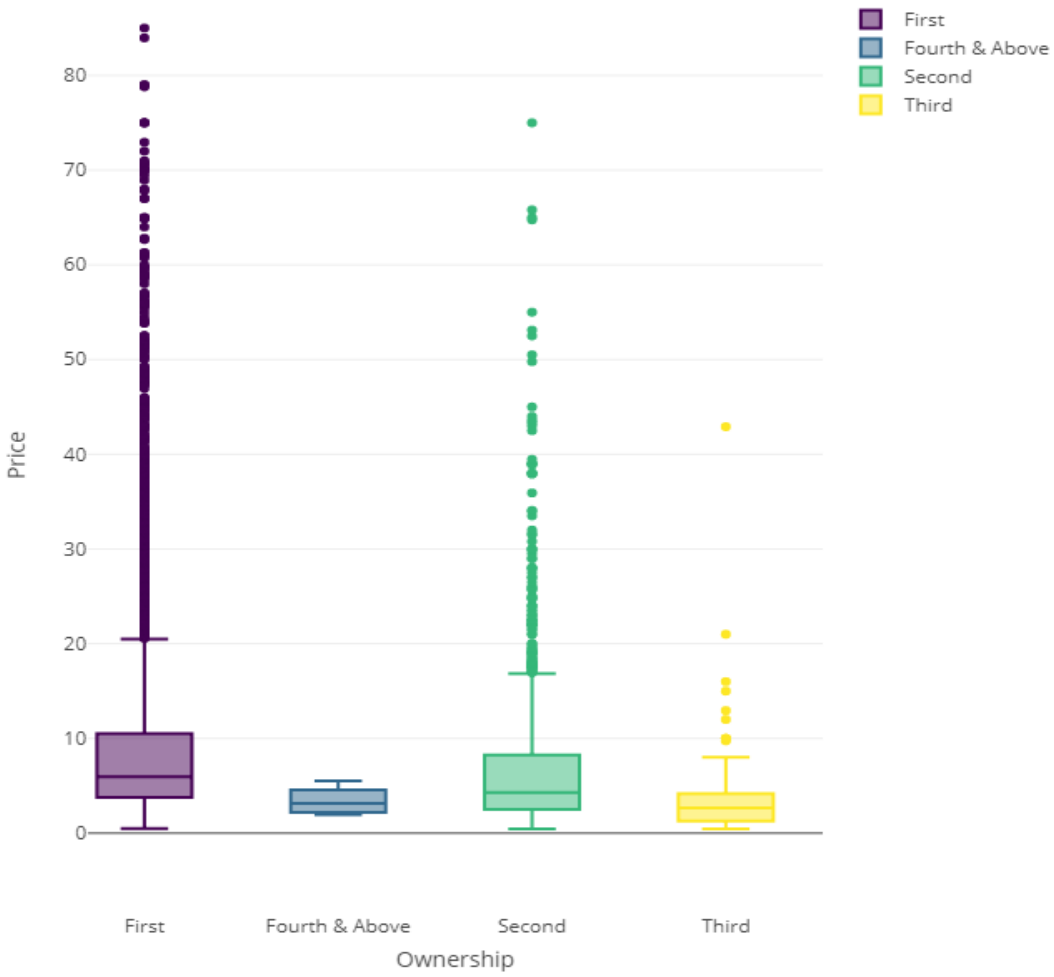


Fig 2.12

VIOLINPLOTS

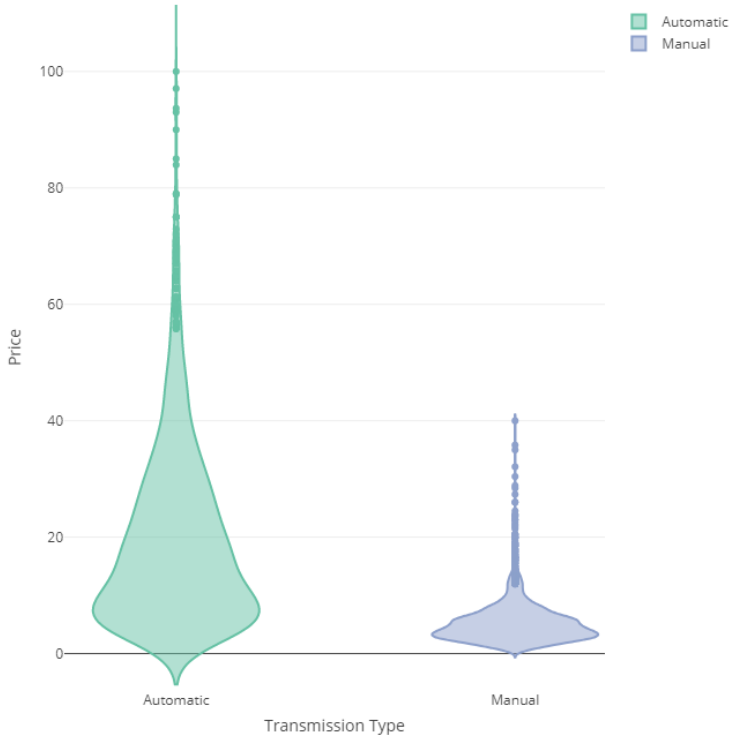


Fig 2.13

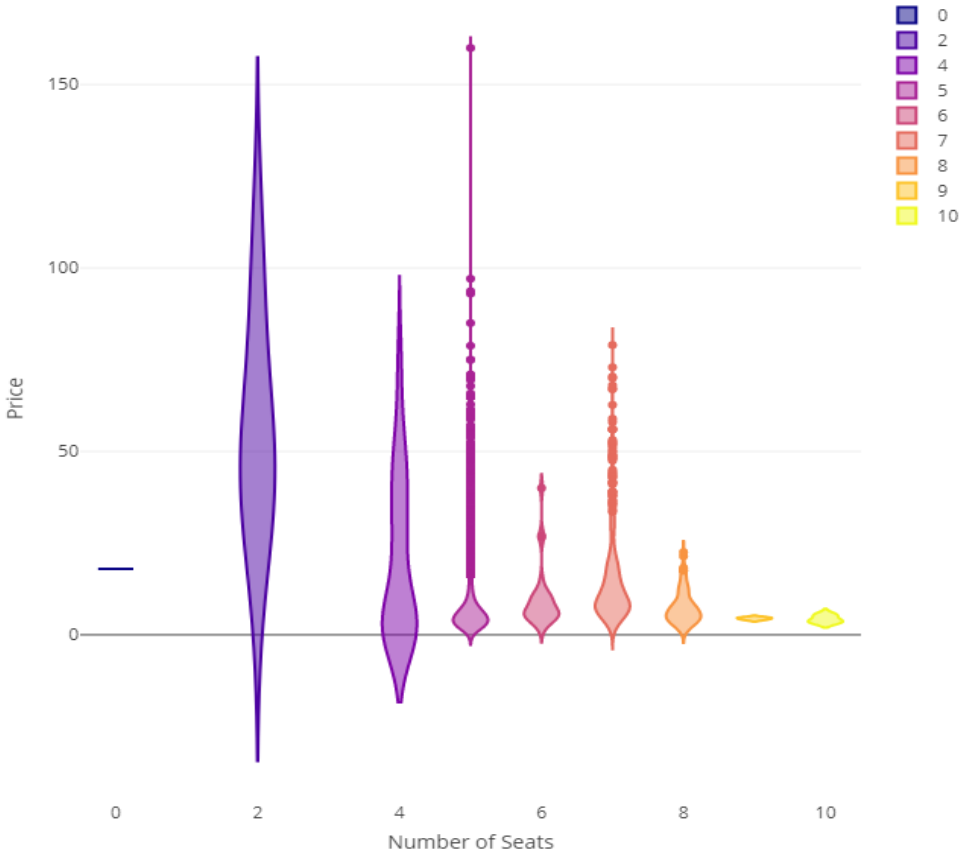


Fig 2.14

MULTIVARIATE ANALYSIS

BARPLOTS

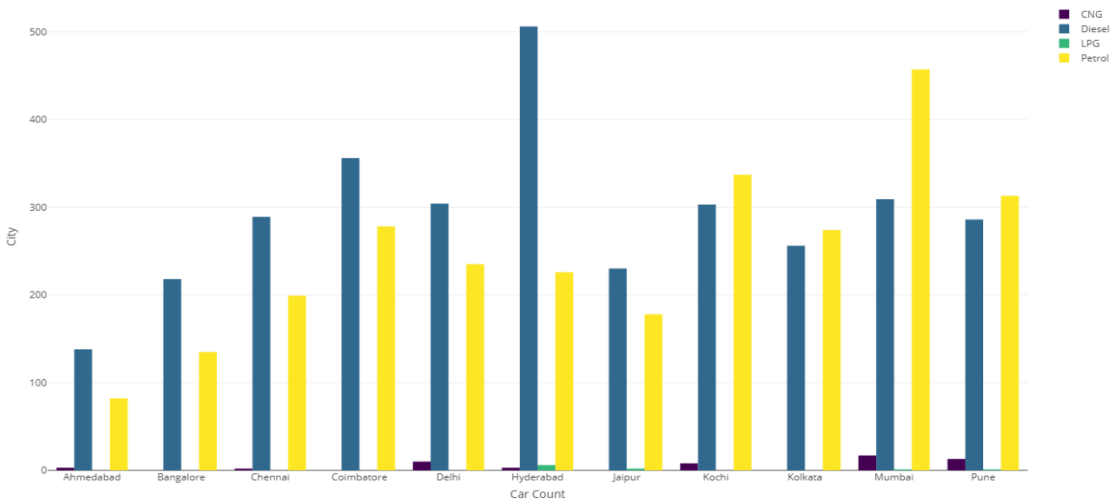


Fig 3.1

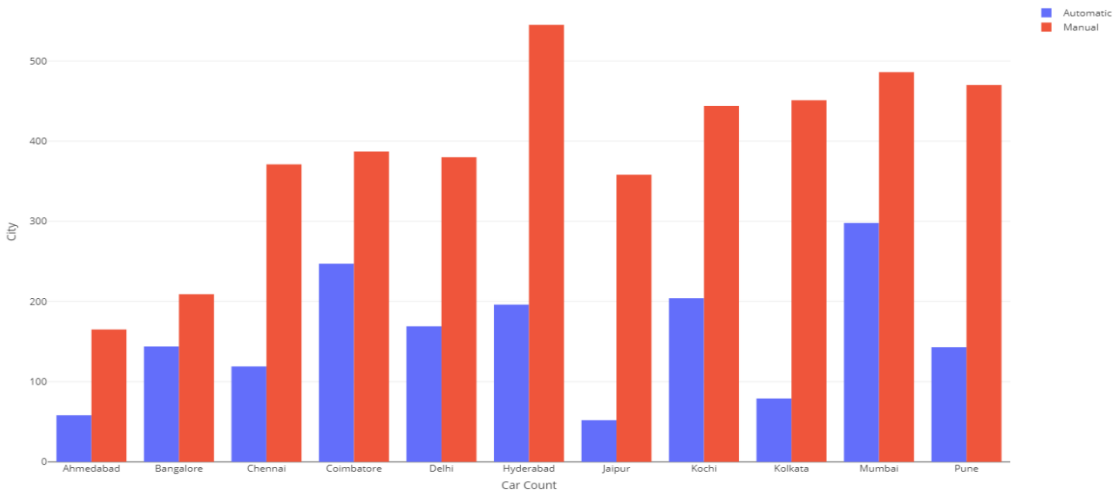


Fig 3.2

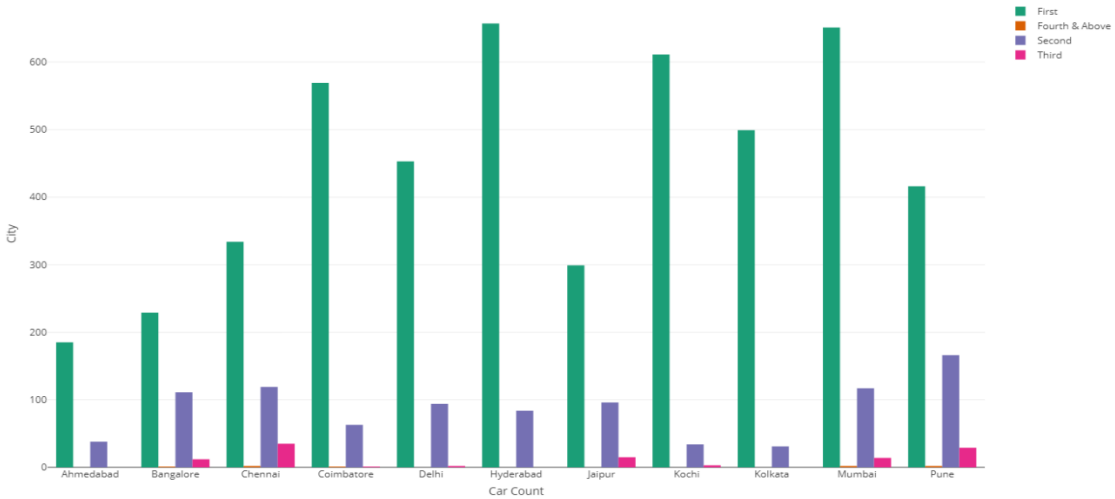


Fig 3.3

SCATTERPLOTS

Fig 3.4

Fig 3.5

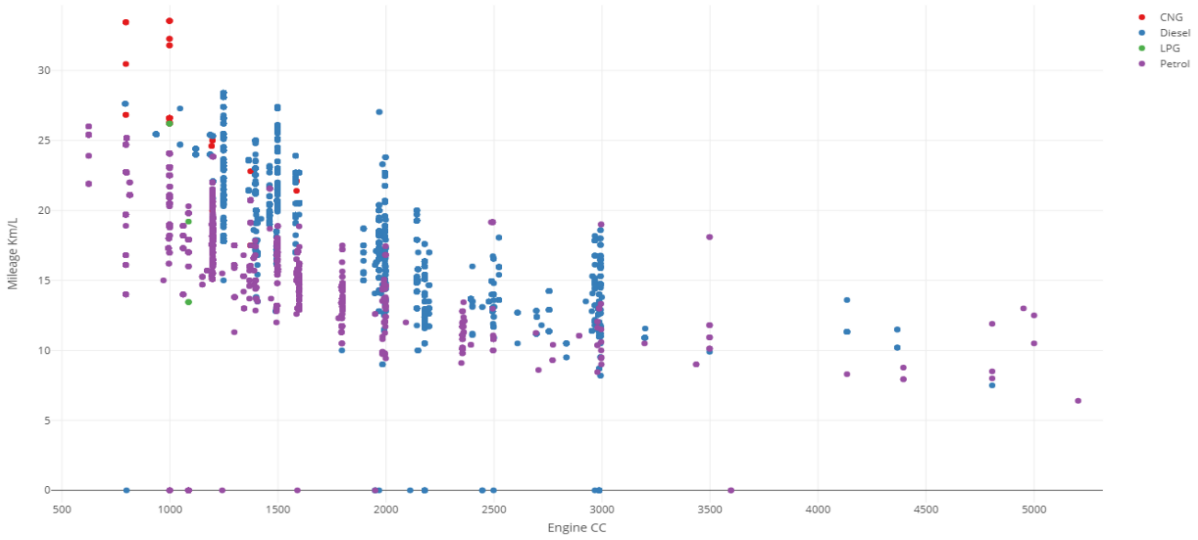


Fig 3.6

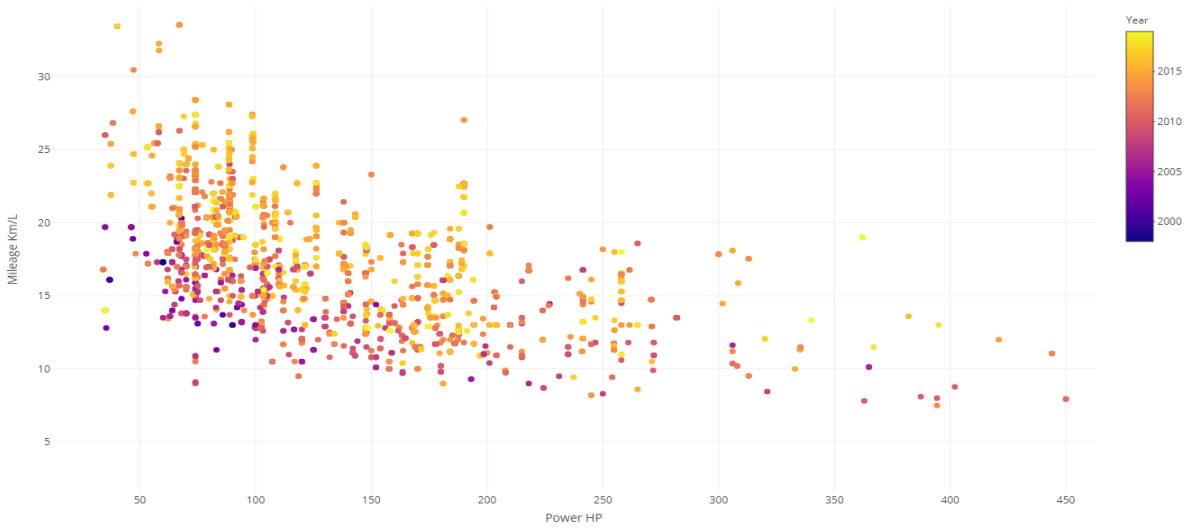


Fig 3.7

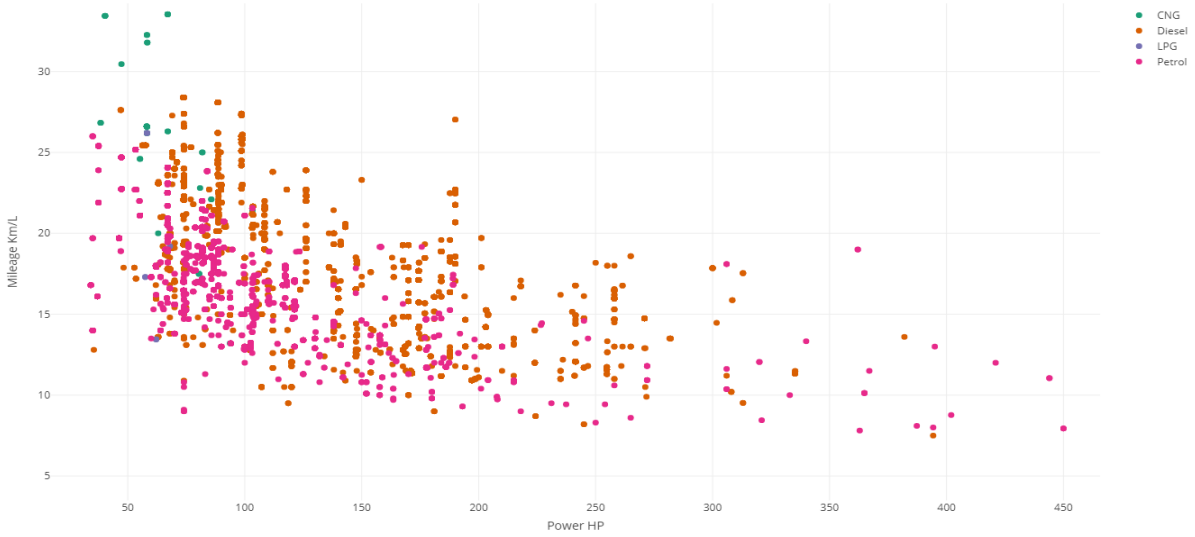


Fig 3.8

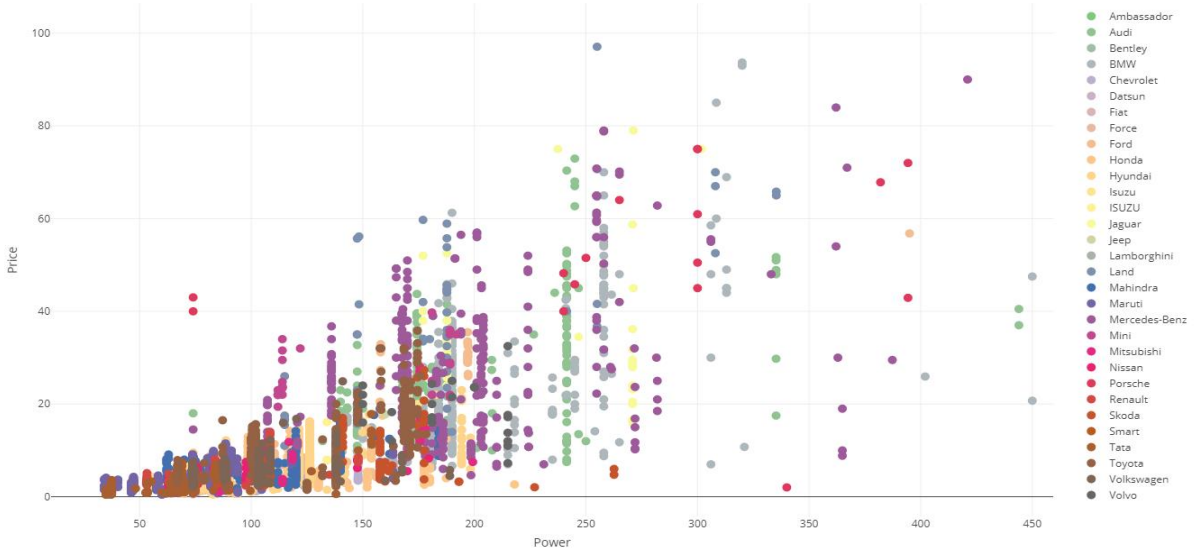


Fig 3.9

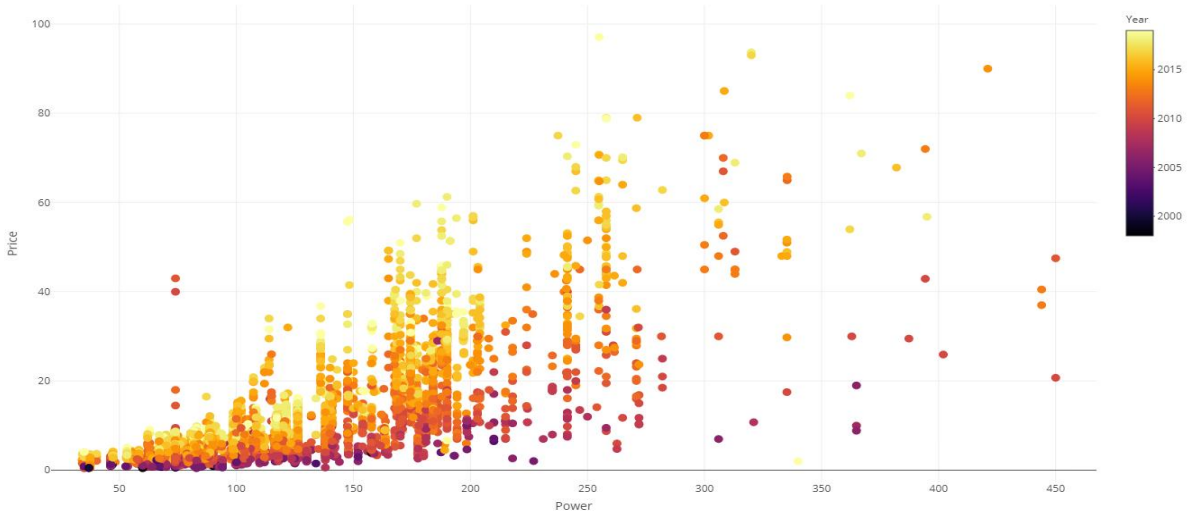


Fig 3.10

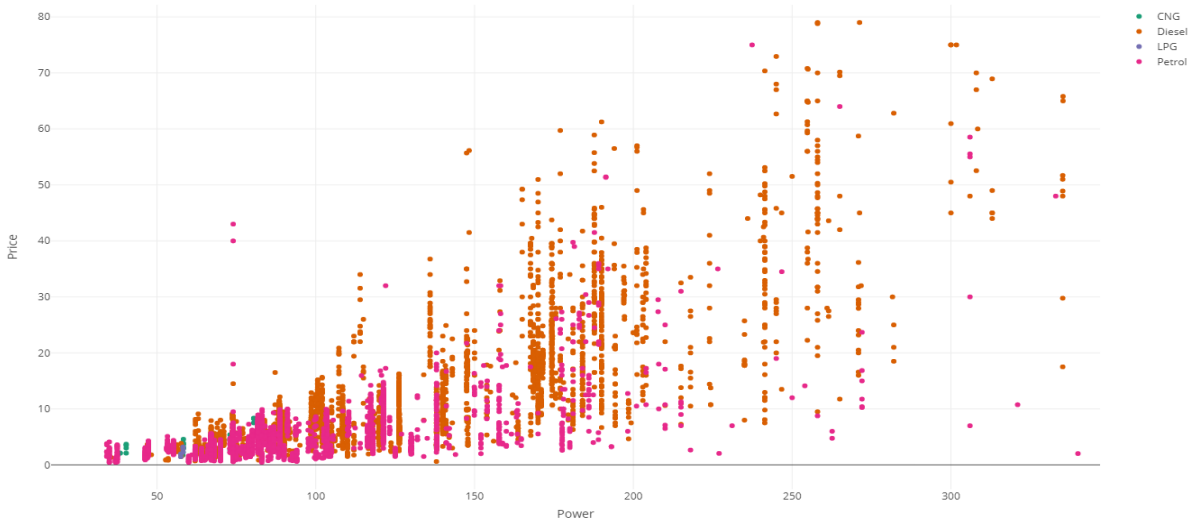


Fig 3.11

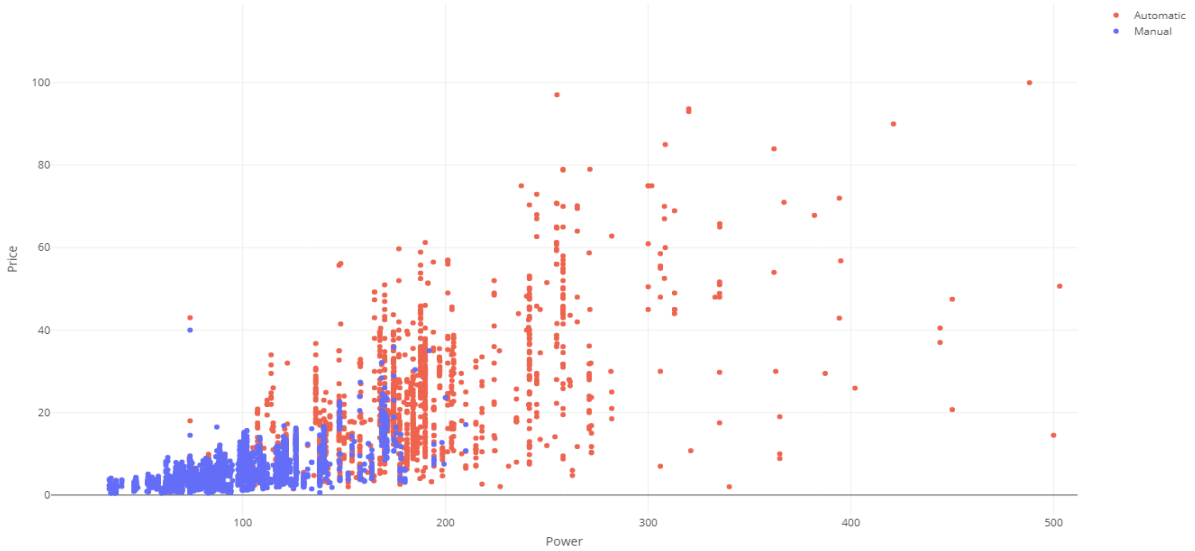


Fig 3.12

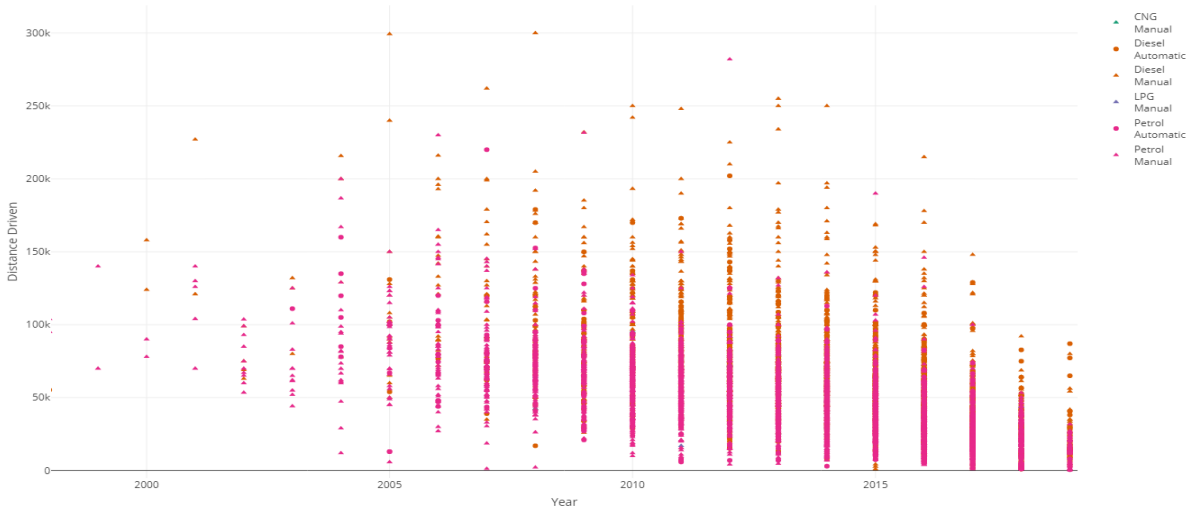


Fig 3.13

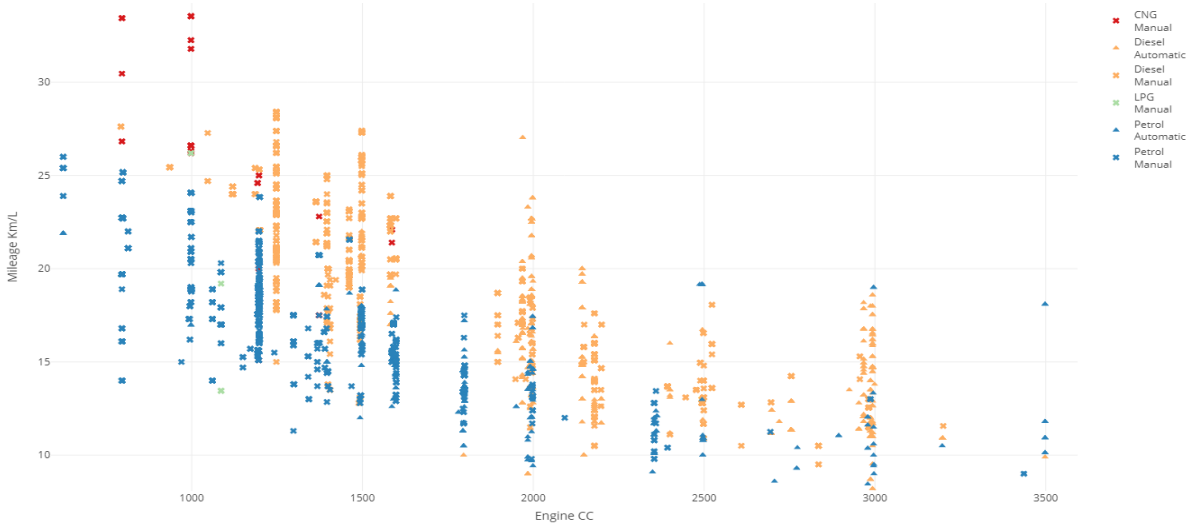


Fig 3.14

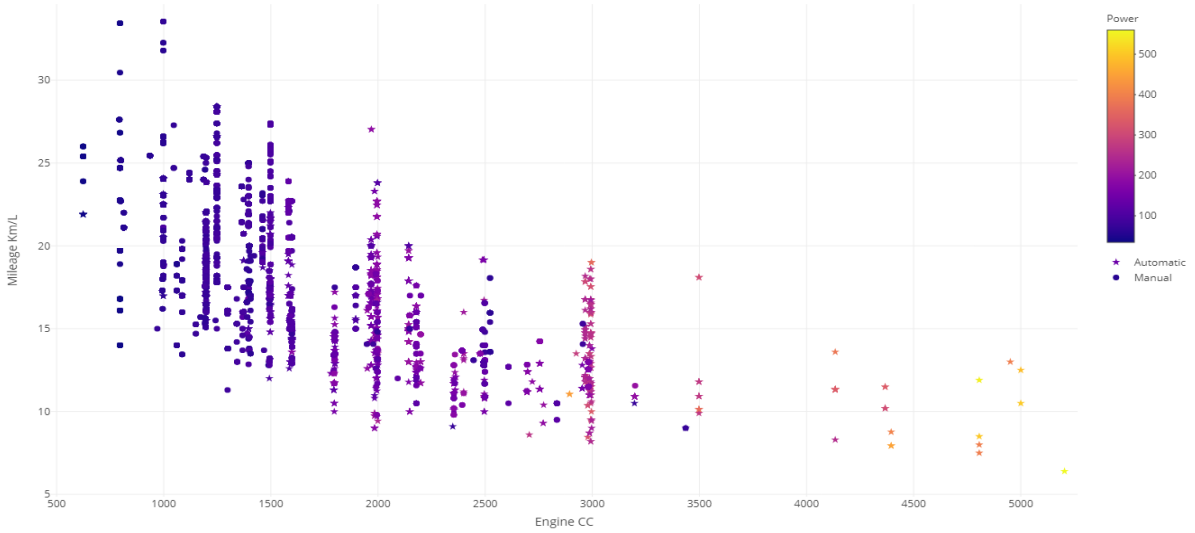


Fig 3.15

BOXPLOTS

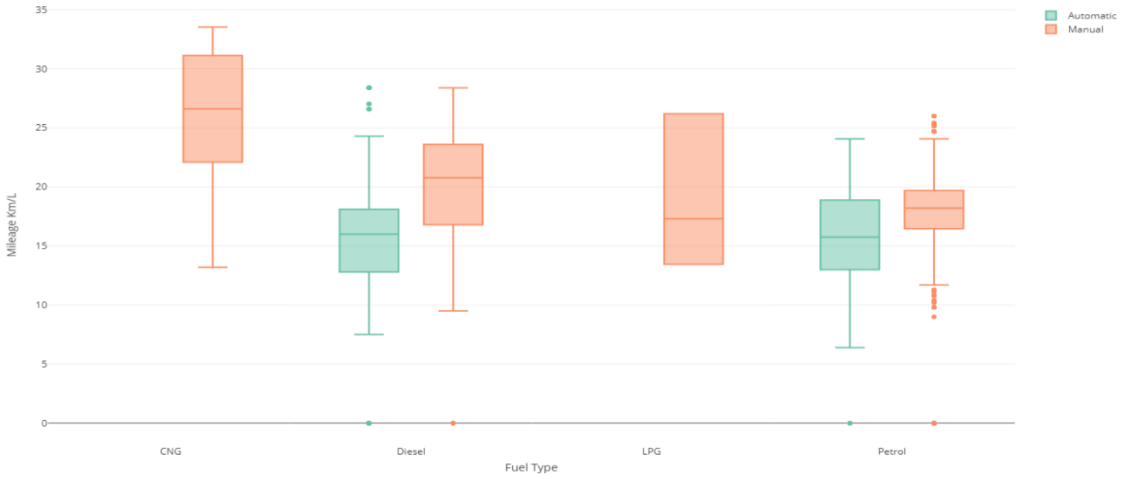


Fig 3.16

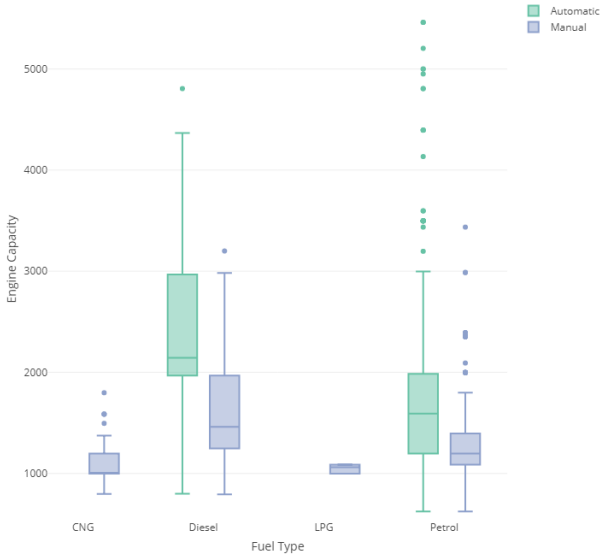


Fig 3.17

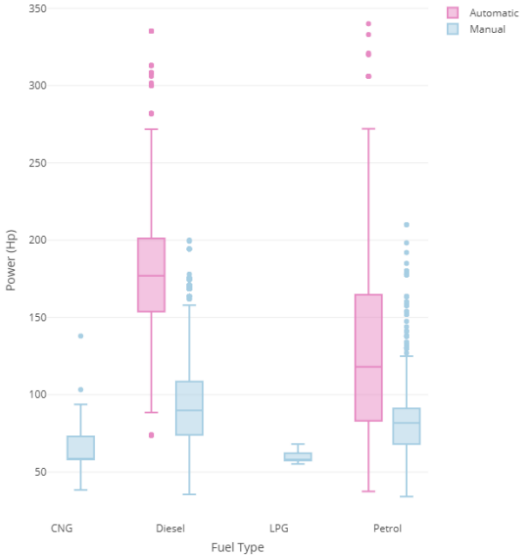


Fig 3.18

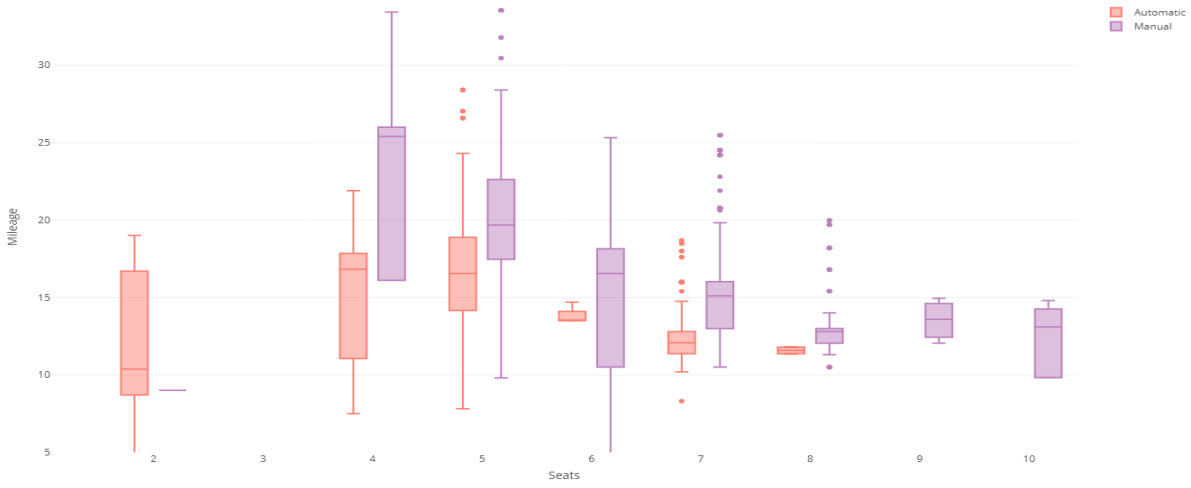


Fig 3.19

VIOLINPLOTS

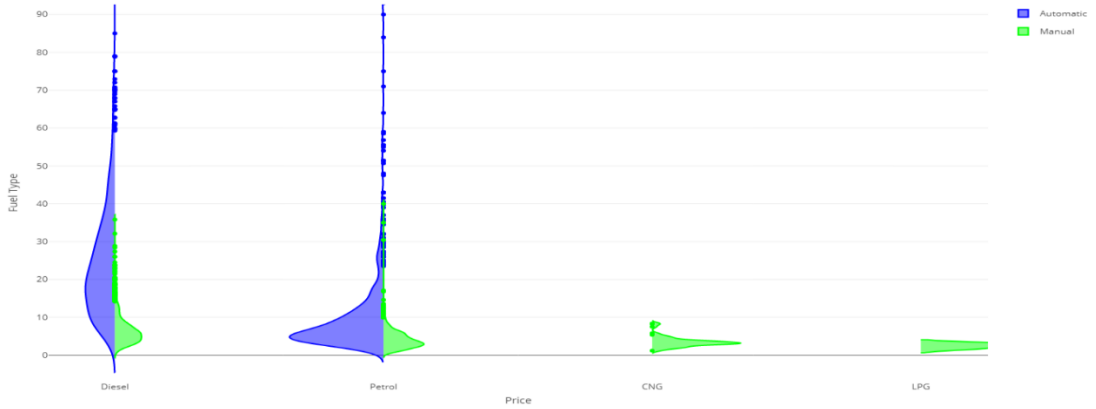


Fig 3.20

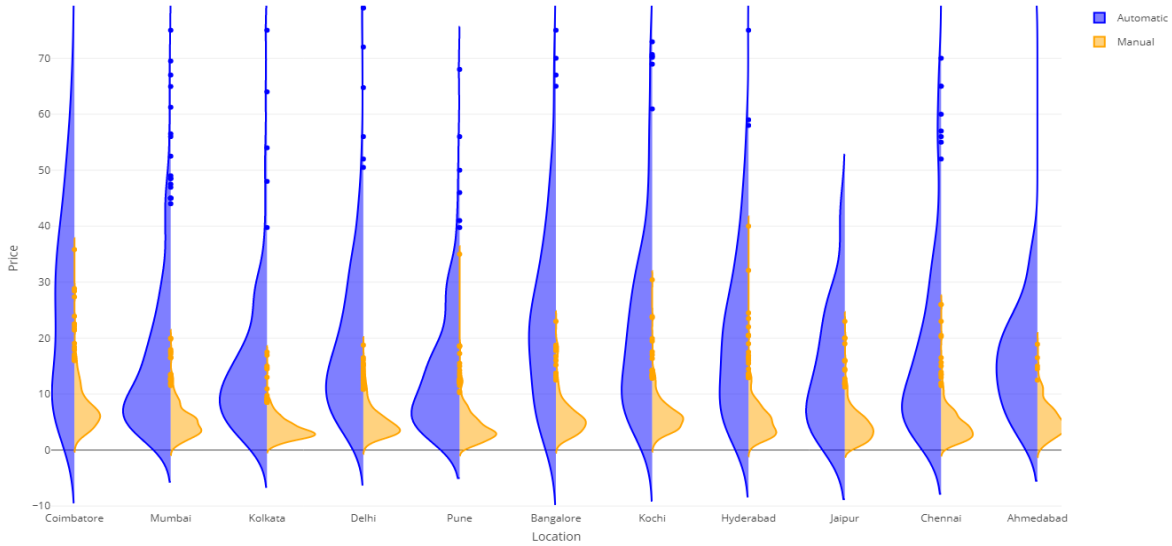


Fig 3.21

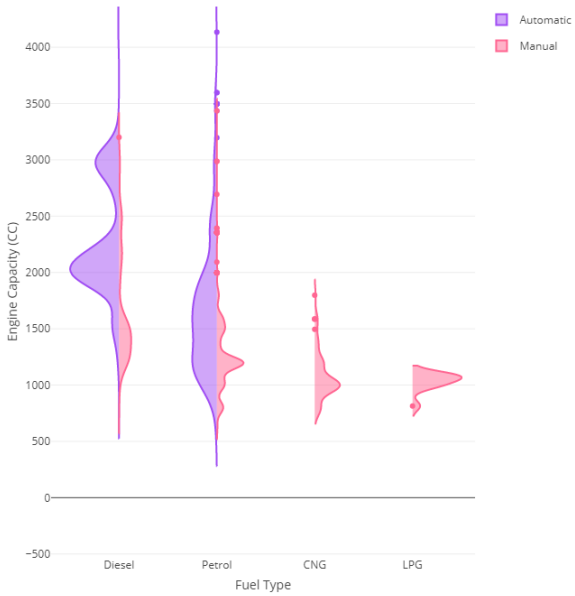


Fig 3.22

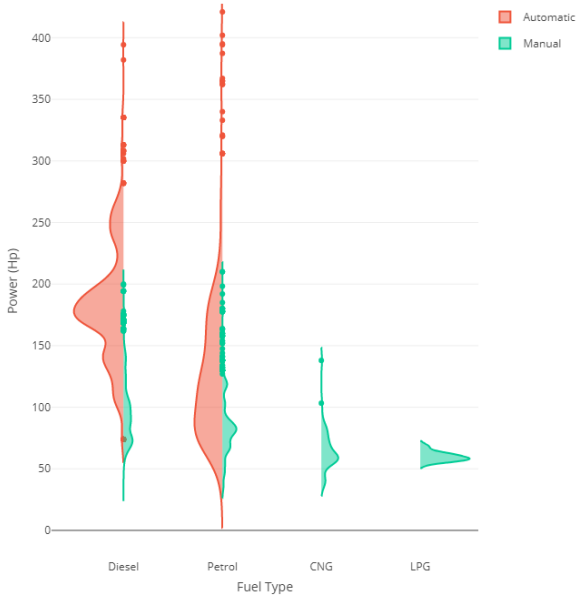


Fig 3.23

CORRELATION HEATMAP

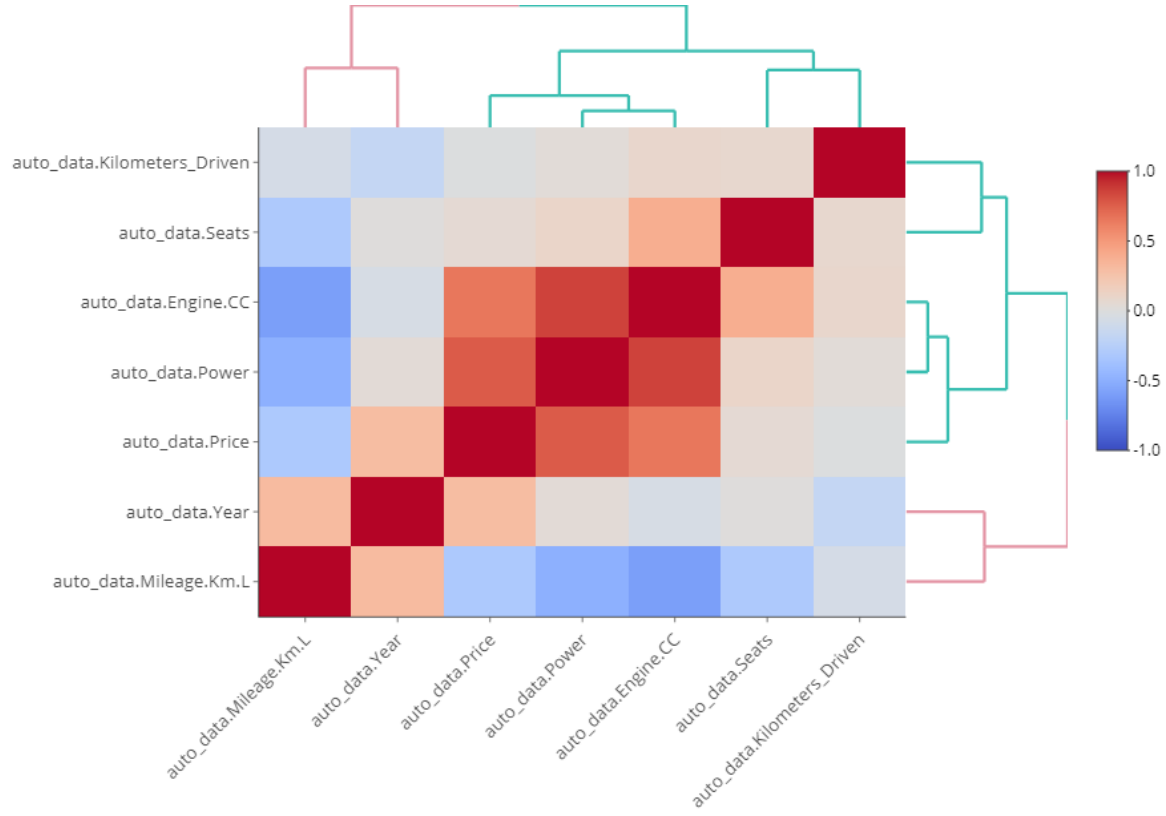


Fig 4.1

REGRESSION PLOTS

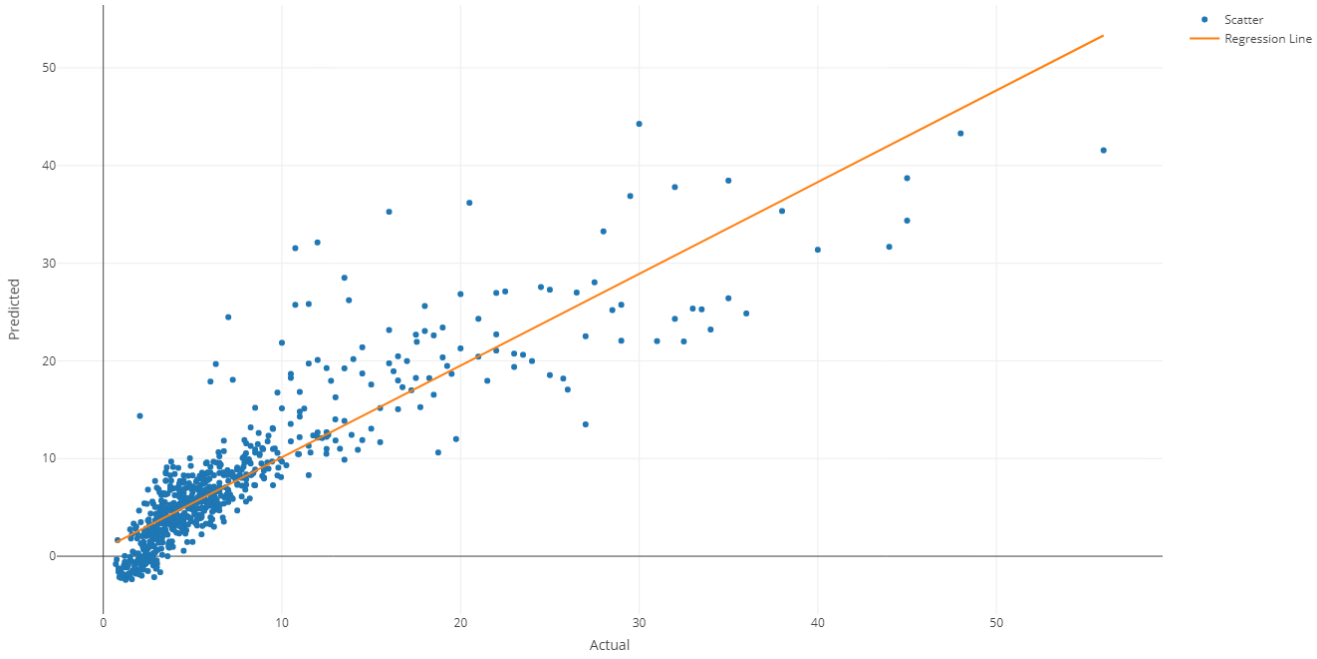


Fig 4.2

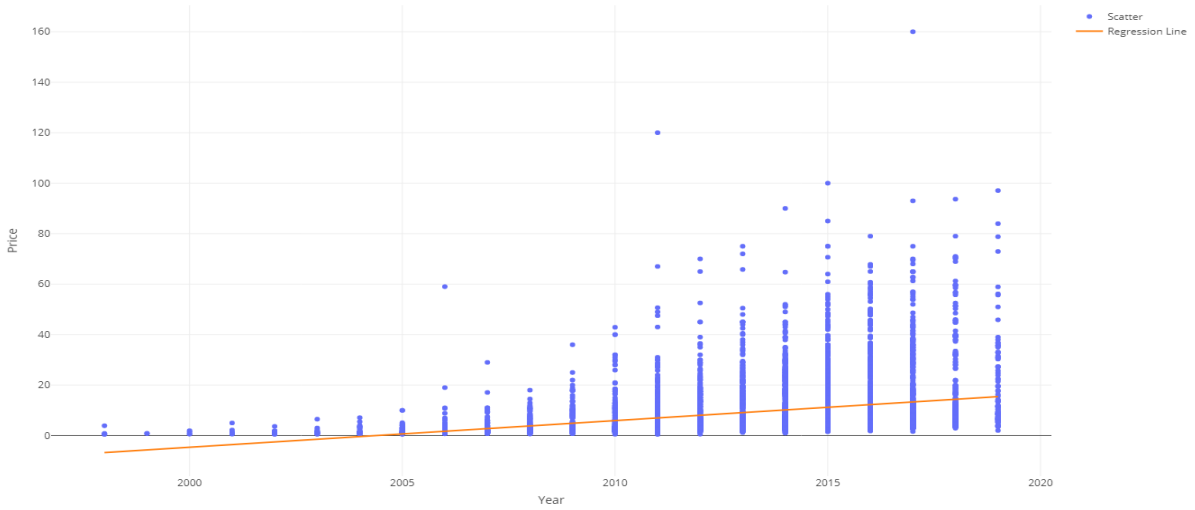


Fig 4.3

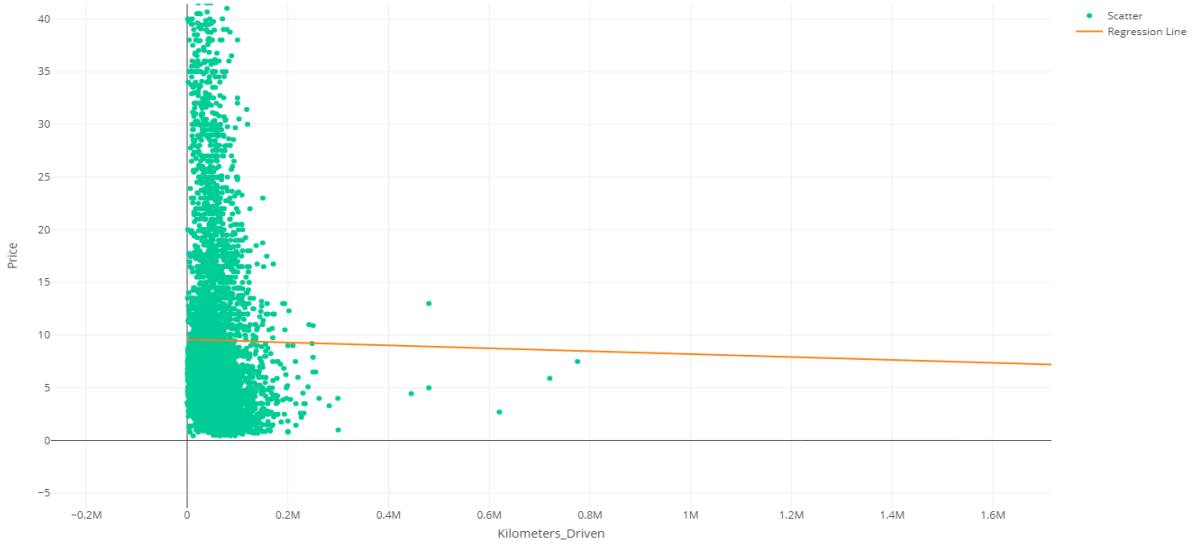


Fig 4.4

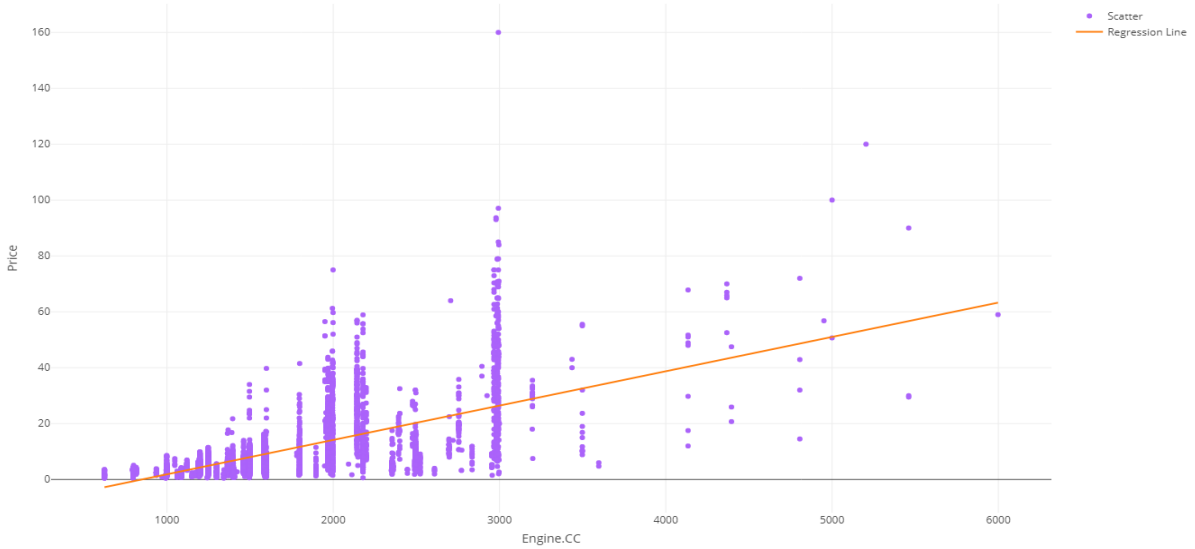


Fig 4.5

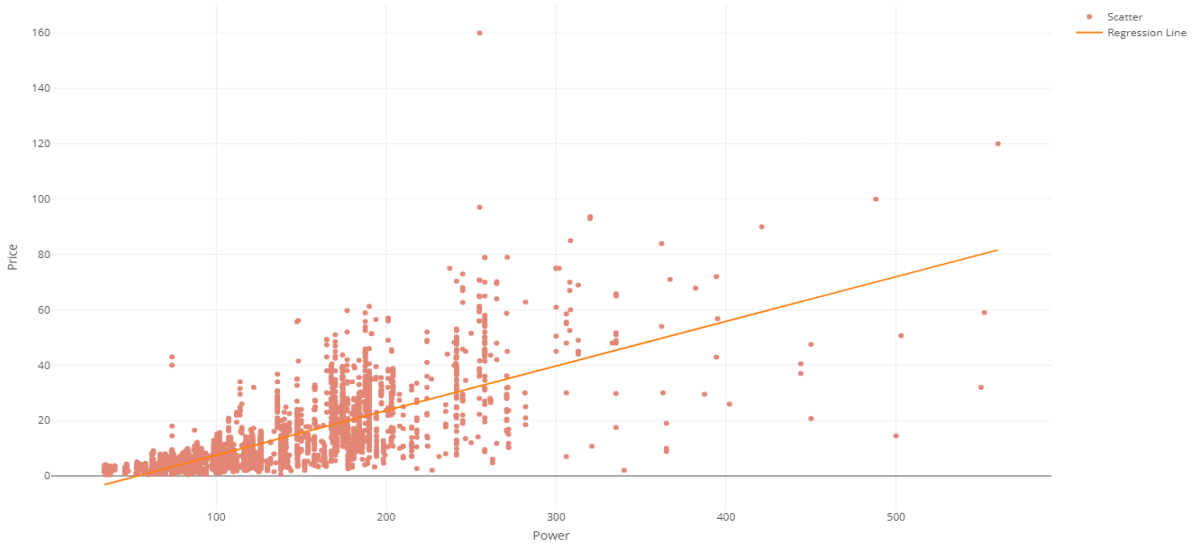


Fig 4.6

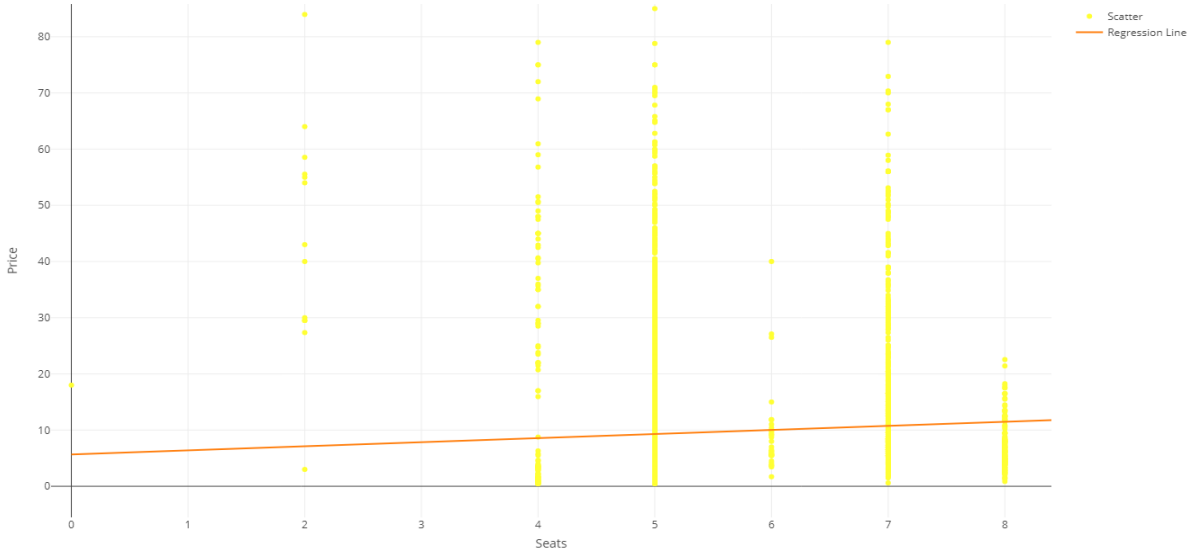


Fig 4.7

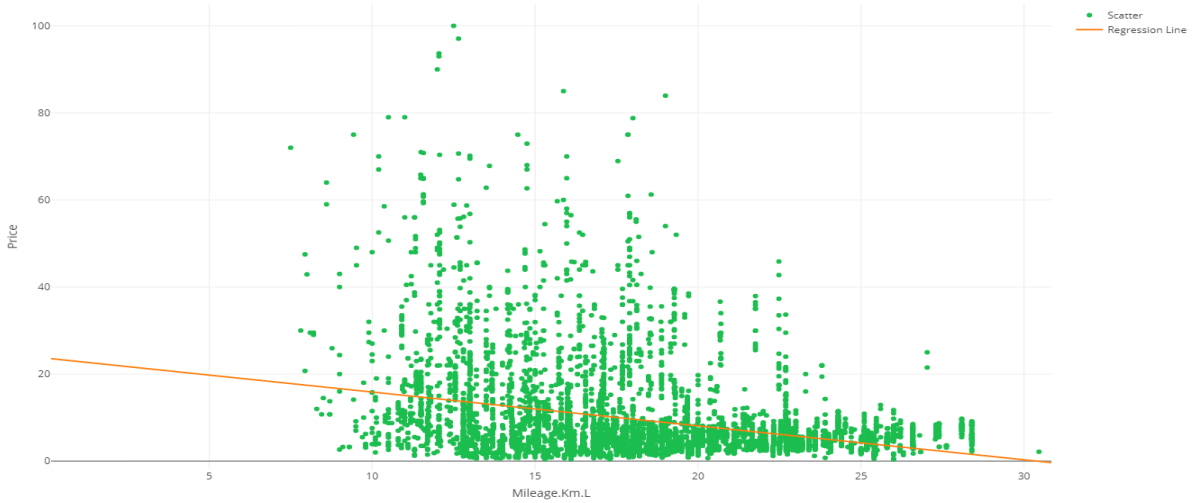


Fig 4.8

CONCLUSION AND FUTURE WORK

CONCLUSION

Thus, we have visualized and derived various insights from the considered Indian automobile dataset by performing data analysis that utilizing machine learning algorithms in R programming language. We have performed univariate analysis, analyzing the data in perspective of a single attribute then with bivariate analysis, analysis using two attributes and then with multivariate which deals with more than two attributes at the same time presenting various levels of visualizations using barplots, histograms, scatter plots, boxplots, violinplots. The result of finding this relationship between various attributes of a vehicle will provided useful insights in building in a prediction model capable of predicting the price of a vehicle based on the other attributes. We have derived one polynomial regression model and studied the results, outcomes, and interpretations in addition to the methodologies to evaluate these models. From the data analysis we have summarized that the attributes Engine, Power, Mileage are the major factor which effected the price of the car largely and the rest of the attributes have some impact but not a huge one. Thus, we could conclude that price is heavily correlated with car engine, power and mileage attributes of the dataset.

FUTURE WORK

In the future extension of this project, more data can be collected that are related this dataset so that this could add more features for the predicting and finding the correlation between the different variables which effect the price of the vehicle. Also, more advanced machine learning models can be used to reduce the amount of error the current had produced. Also, the various hyper parameters can be tuned during the training of the model in order to decrease the RSME value making the prediction model closer to the actual values, increasing the precision of the model. Also, if more and more attributes are added to the dataset them, a deep learning neural network approach can be taken to train the ANN model which has higher chances of predicting the price of automobile with more accuracy.

APPENDIX

VISUALIZATION CODE:

UNIVARIATE ANALYSIS

```
library(plotly)

auto_data = read.csv("C://Users/subha/Desktop/Visualization & Analysis on Automobile Dataset using Machine Learning in R/indian-auto-mpg.csv")

summary(auto_data)

fig <- plot_ly(x = auto_data$Manufacturer, type = "histogram",
               marker = list(color = "rgba(255, 0, 0, 0.7)")) %>%
      layout(xaxis = list(title = "Manufacturer",categoryorder="total descending"),
             yaxis = list(title = "Car Count"))

fig <- plot_ly(x = auto_data$Location, type = "histogram")%>%
      layout(xaxis = list(title = "Location"),
             yaxis = list(title = "Car Count"))


fig <- plot_ly(x = as.factor(auto_data$Year), histfunc='sum',type = "histogram",
               marker = list(color = "rgba(0,205,149, 1)"))%>%
      layout(xaxis = list(title = "Year"),
             yaxis = list(title = "Car Count"))

fig <- plot_ly(x = auto_data$Transmission, type = "histogram",
               marker = list(color = "#e0812c"))%>%
      layout(xaxis = list(title = "Transmission"),
             yaxis = list(title = "Car Count"))

fig <- plot_ly(x = auto_data$Fuel_Type, type = "histogram",
               marker = list(color = "#d89f38"))%>%
      layout(xaxis = list(title = "Fuel Type"),
             yaxis = list(title = "Car Count"))

fig <- plot_ly(x = as.factor(auto_data$Seats), type = "histogram",
               marker = list(color = "#707bfa")) %>%
       layout(xaxis = list(title = "Number of Seats"),
              yaxis = list(title = "Car Count"))


fig <- plot_ly(y = auto_data$Engine.CC, type = "box")%>%
      layout(xaxis = list(title = "Boxplot"),
             yaxis = list(title = "Engine CC"))
```

```
fig <- plot_ly(y = auto_data$Power, type = "box") %>%
      layout(xaxis = list(title = "Horse Power"),
             yaxis = list(title = "Engine Power"))

fig <- plot_ly(y = auto_data$Mileage.Km.L, type = "box")%>%
      layout(xaxis = list(title = "Mileage"),
             yaxis = list(title = "Km per Litre"))

fig <- plot_ly(y = auto_data$Price, type = "box", color=".")%>%
     layout(xaxis = list(title = "Price"),
            yaxis = list(title = "Lakhs"))
```

BIVARIATE ANALYSIS

```
library(plotly)

auto_data = read.csv("C://Users/subha/Desktop/Visualization & Analysis on Automobile Dataset using Machine Learning in R/indian-auto-mpg.csv")


fig <- plot_ly(y = auto_data$Mileage.Km.L, x=auto_data$Engine.CC,
               marker = list(line = list(width = 1))) %>%
       layout(xaxis = list(title = "Engine CC"),
              yaxis = list(title = "Mileage Km/L"))


fig <- plot_ly(y = auto_data$Mileage.Km.L, x=auto_data$Power,
               marker = list(color="#e2594e")) %>%
      layout(xaxis = list(title = "Power Hp"),
             yaxis = list(title = "Mileage Km/L"))


fig <- plot_ly(y = auto_data$Price, x=auto_data$Year,
               marker = list(color="rgba(0,205,149, 1)")) %>%
     layout(xaxis = list(title = "Year"),
            yaxis = list(title = "Price"))

fig <- plot_ly(y = auto_data$Kilometers_Driven, x=auto_data$Year,ttype="scatter",
               marker = list(color="#8b0d86")) %>%
      layout(xaxis = list(title = "Year"),
             yaxis = list(title = "Distance Driven"))


fig <- plot_ly(y = auto_data$Price, x=auto_data$Engine.CC, type="scatter",
               marker = list(color="#e0812c")) %>%
      layout(xaxis = list(title = "Engine CC"),
             yaxis = list(title = "Price"))
```

```
fig <- plot_ly(y = auto_data$Price, x=auto_data$Power, type="scatter",
               marker = list(color="#f3da2d")) %>%
      layout(xaxis = list(title = "Power Hp"),
             yaxis = list(title = "Price"))

fig <- plot_ly(y = auto_data$Power, x=auto_data$Engine.CC, type="scatter",
               marker = list(color="#9932CC")) %>%
       layout(xaxis = list(title = "Engine"),
              yaxis = list(title = "Power"))


fig <- plot_ly(y = auto_data$Mileage.Km.L, color=auto_data$Fuel_Type, type="box") %>%
      layout(xaxis = list(title = "Fuel Type"),
             yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(y = auto_data$Mileage.Km.L, color=auto_data$Transmission, type="box")
%>%
      layout(xaxis = list(title = "Transmission Type"),
             yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(y = auto_data$Mileage.Km.L, color=auto_data$Owner_Type,
               type="box",colors = "viridis") %>%
      layout(xaxis = list(title = "Ownership"),
             yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(y = auto_data$Mileage.Km.L, color=as.factor(auto_data$Seats),
               type="violin", colors = "plasma") %>%
      layout(xaxis = list(title = "Number of Seats"),
             yaxis = list(title = "Mileage"))


fig <- plot_ly(y = auto_data$Price, color=auto_data$Fuel_Type, type="box") %>%
      layout(xaxis = list(title = "Fuel Type"),
             yaxis = list(title = "Price"))


fig <- plot_ly(y = auto_data$Price, color=auto_data$Transmission, type="violin") %>%
      layout(xaxis = list(title = "Transmission Type"),
             yaxis = list(title = "Price"))


fig <- plot_ly(y = auto_data$Price, color=auto_data$Owner_Type,
               type="box",colors = "viridis") %>%
      layout(xaxis = list(title = "Ownership"),
             yaxis = list(title = "Price"))

fig <- plot_ly(y = auto_data$Price, color=as.factor(auto_data$Seats),
               type="violin", colors = "plasma") %>%
      layout(xaxis = list(title = "Number of Seats"),
             yaxis = list(title = "Price"))
```

MULTIVARIATE ANALYSIS

```
library(plotly)


auto_data = read.csv("C://Users/subha/Desktop/Visualization & Analysis on Automobile Dataset using Machine Learning in R/indian-auto-mpg.csv")


fig <- plot_ly(auto_data, x = ~Location, color = ~Fuel_Type, colors = "viridis") %>% add_histogram()%>%
      layout(xaxis = list(title = "Car Count"),
             yaxis = list(title = "City"))

fig <- plot_ly(auto_data, x = ~Location, color = ~Transmission, colors=c("#636efa","#ef553b")) %>% add_histogram()%>%
     layout(xaxis = list(title = "Car Count"),
            yaxis = list(title = "City"))

fig <- plot_ly(auto_data, x = ~Location, color = ~Owner_Type, colors="Dark2") %>% add_histogram()%>%
     layout(xaxis = list(title = "Car Count"),
            yaxis = list(title = "City"))


fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Manufacturer,colors="Set2",
               mode = "markers",marker = list(size = 10)) %>%
      layout(xaxis = list(title = "Engine CC"),
             yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Year,colors="viridis",
               mode = "markers",marker = list(size = 10)) %>%
      layout(xaxis = list(title = "Engine CC"),
             yaxis = list(title = "Mileage Km/L"))


fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Fuel_Type, colors = "Set1",
               mode = "markers",marker = list(size = 7)) %>%
     layout(xaxis = list(title = "Engine CC"),
            yaxis = list(title = "Mileage Km/L"))


fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Power,
               type = "scatter",color = ~Year, colors="plasma",
               mode = "markers",marker = list(size = 7)) %>%
     layout(xaxis = list(title = "Power HP"),
            yaxis = list(title = "Mileage Km/L"))
```

```
fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Power,
               type = "scatter",color = ~Fuel_Type,colors="Dark2",
               mode = "markers",marker = list(size = 7)) %>%
      layout(xaxis = list(title = "Power HP"),
             yaxis = list(title = "Mileage Km/L"))


fig <- plot_ly(data = auto_data, y = ~Price, x=~Power,
               type = "scatter",color = ~Manufacturer,
               mode = "markers",marker = list(size = 10)) %>%
     layout(xaxis = list(title = "Power"),
            yaxis = list(title = "Price"))


fig <- plot_ly(data = auto_data, y = ~Price, x=~Power,
               type = "scatter",color = ~Year,colors="inferno",
               mode = "markers",marker = list(size = 10)) %>%
      layout(xaxis = list(title = "Power"),
             yaxis = list(title = "Price"))


fig <- plot_ly(data = auto_data, y = ~Price, x=~Power,
               type = "scatter",color = ~Fuel_Type ,colors="Dark2",
               mode = "markers") %>%
      layout(xaxis = list(title = "Power"),
             yaxis = list(title = "Price"))


fig <- plot_ly(data = auto_data, y = ~Price, x=~Power,colors=c("#ee644e","#646ef8"),
               type = "scatter",color = ~Transmission,
               mode = "markers") %>%
  layout(xaxis = list(title = "Power"),
         yaxis = list(title = "Price"))



fig <- plot_ly(data = auto_data,  y = ~Kilometers_Driven, x=~Year,
               type = "scatter",color = ~Fuel_Type ,colors="Dark2",
               mode = "markers", symbol = ~Transmission) %>%
      layout(xaxis = list(title = "Year"),
             yaxis = list(title = "Distance Driven"))


fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Fuel_Type,colors="Spectral",
               mode = "markers",symbol = ~Transmission,
               symbols = c('triangle-up','x'),marker = list(size = 7)) %>%
      layout(xaxis = list(title = "Engine CC"),
             yaxis = list(title = "Mileage Km/L"))
```

```
fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Fuel_Type,colors="Spectral",
               mode = "markers",symbol = ~Transmission,
               symbols = c('triangle-up','x'),marker = list(size = 7)) %>%
  layout(xaxis = list(title = "Engine CC"),
         yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Power,colors="plasma",
               symbol = ~Transmission,symbols = c('star','circle'),
               mode = "markers",marker = list(size = 7)) %>%
      layout(xaxis = list(title = "Engine CC"),
             yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(data = auto_data, y = ~Mileage.Km.L, x=~Engine.CC,
               type = "scatter",color = ~Seats,colors="viridis",
               symbol = ~Transmission,symbols = c('star','circle'),
               mode = "markers",marker = list(size = 7)) %>%
      layout(xaxis = list(title = "Engine CC"),
             yaxis = list(title = "Mileage Km/L"))



fig <- plot_ly(data = auto_data,y = ~Mileage.Km.L, x=~Fuel_Type,
               color=~Transmission,colors = "Set2" ,type="box") %>%
      layout(boxmode = "group",
             xaxis = list(title = "Fuel Type"),
             yaxis = list(title = "Mileage Km/L"))

fig <- plot_ly(data = auto_data,y = ~Engine.CC, x=~Fuel_Type,
               color=~Transmission,type="box") %>%
      layout(boxmode = "group",
             xaxis = list(title = "Fuel Type"),
             yaxis = list(title = "Engine Capacity"))

fig <- plot_ly(data = auto_data,y = ~Power, x=~Fuel_Type,
               color=~Transmission,colors = c("#e78ac3","#a6cee3"), type="box") %>%
      layout(boxmode = "group",
             xaxis = list(title = "Fuel Type"),
             yaxis = list(title = "Power (Hp)"))



fig <- plot_ly(data = auto_data,y = ~Mileage.Km.L, x=~Seats,
               color=~Transmission,colors = c("#fb8072","#bc80bd"), type="box") %>%
      layout(boxmode = "group",
             xaxis = list(title = "Seats"),
             yaxis = list(title = "Mileage"))

fig <- plot_ly(data = auto_data,y = ~Power, x=~Seats,
```

```
          color=~Transmission, colors=c("#6684ca","#ffd92f") ,type="box") %>%
      layout(boxmode = "group",
            xaxis = list(title = "Seats"),
            yaxis = list(title = "Power"))


fig <- plot_ly(data = auto_data,y = ~Price, x=~Fuel_Type,
           color=I(~Transmission), colors=c("#ef553b","#00cc96") ,type="violin") %>%
      layout(violinmode = 'group',
            xaxis = list(title = "Fuel Type"),
            yaxis = list(title = "Price"))
fig <- plot_ly(data = auto_data,y = ~Price, x=~Owner_Type,
           color=I(~Transmission), colors="Dark2" ,type="violin") %>%
      layout(violinmode = 'group',
            xaxis = list(title = "Owner Type"),
            yaxis = list(title = "Price"))



fig <- auto_data %>% plot_ly(type = 'violin')
fig <- fig %>% add_trace(x = ~Fuel_Type[auto_data$Transmission == 'Automatic'],
                  y = ~Price[auto_data$Transmission == 'Automatic'],
                  legendgroup = 'Automatic', scalegroup = 'Automatic',
                  name = 'Automatic',side = 'negative',color = I("blue"))
fig <- fig %>% add_trace(x = ~Fuel_Type[auto_data$Transmission == 'Manual'],
                  y = ~Price[auto_data$Transmission == 'Manual'],
                  legendgroup = 'Manual', scalegroup = 'Manual',
                  name = 'Manual',side = 'positive',color = I("green"))%>%
      layout(violingroupgap = 0,
            xaxis = list(title = "Price"),
            yaxis = list(title = "Fuel Type"))


fig <- auto_data %>% plot_ly(type = 'violin')
fig <- fig %>% add_trace(x = ~Location[auto_data$Transmission == 'Automatic'],
                  y = ~Price[auto_data$Transmission == 'Automatic'],
                  legendgroup = 'Automatic', scalegroup = 'Automatic',
                  name = 'Automatic',side = 'negative',color = I("blue"))

fig <- fig %>% add_trace(x = ~Location[auto_data$Transmission == 'Manual'],
                  y = ~Price[auto_data$Transmission == 'Manual'],
                  legendgroup = 'Manual', scalegroup = 'Manual',
                  name = 'Manual',side = 'positive',color = I("orange"))%>%
           layout(violingap = 0,
                 violingroupgap = 0,
                 violinmode = 'overlay',
                 xaxis = list(title = "Location"),
                 yaxis = list(title = "Price"))
```

```
fig <- plot_ly(data = auto_data,y = ~Price, x=~Seats,type="violin") %>%
      layout(violingap = 0,violingroupgap = 0,
            violinmode = "group",
            xaxis = list(title = "Seats"),
            yaxis = list(title = "Price"))


fig <- auto_data %>% plot_ly(type = 'violin')
fig <- fig %>% add_trace(x = ~Fuel_Type[auto_data$Transmission == 'Automatic'],
                     y = ~Engine.CC[auto_data$Transmission == 'Automatic'],
                     legendgroup = 'Automatic', scalegroup = 'Automatic',
                     name = 'Automatic',side = 'negative',color = I("#a14ef4"))

fig <- fig %>% add_trace(x = ~Fuel_Type[auto_data$Transmission == 'Manual'],
                     y = ~Engine.CC[auto_data$Transmission == 'Manual'],
                     legendgroup = 'Manual', scalegroup = 'Manual',
                     name = 'Manual',side = 'positive',color = I("#ff6692"))%>%
             layout(xaxis = list(title = "Fuel Type"),
                   yaxis = list(title = "Engine Capacity (CC)"),
                   violingap = 0,violingroupgap = 0,violinmode = 'overlay')



fig <- auto_data %>% plot_ly(type = 'violin')
fig <- fig %>% add_trace(x = ~Fuel_Type[auto_data$Transmission == 'Automatic'],
                     y = ~Power[auto_data$Transmission == 'Automatic'],
                     legendgroup = 'Automatic', scalegroup = 'Automatic',
                     name = 'Automatic',side = 'negative',color = I("#ef553b"))

fig <- fig %>% add_trace(x = ~Fuel_Type[auto_data$Transmission == 'Manual'],
                     y = ~Power[auto_data$Transmission == 'Manual'],
                     legendgroup = 'Manual', scalegroup = 'Manual',
                     name = 'Manual',side = 'positive',color = I("#00cc96"))%>%
             layout(xaxis = list(title = "Fuel Type"),
                   yaxis = list(title = "Power (Hp)"),
                   violingap = 0,violingroupgap = 0,violinmode = 'overlay')
```

REGRESSION MODEL TRAINING AND TESTING

```
library(caTools)
library(caret)
library(fastDummies)
library(plotly)


auto_data = read.csv("C://Users/subha/Desktop/Visualization & Analysis on Automobile
Dataset using Machine Learning in R/indian-auto-mpg.csv")



# Cleaning Data
new_df = auto_data
new_df %>% filter(Seats>0) -> new_df
new_df %>% filter(Mileage.Km.L>0) -> new_df
new_df %>% filter(Price<70) -> new_df
new_df = new_df[3:14]
new_df = new_df[,-2]

new_df$Price = new_df$Price*100000

# Checking
sum(is.na(auto_data))==0


# Creating dummy variables for categorical variables
new_df <- dummy_cols(new_df,
                     select_columns = c('Manufacturer','Fuel_Type',
                                        'Transmission','Owner_Type'),
                     remove_selected_columns = TRUE)

auto_data = new_df



# Train Test Split
sample.split(auto_data$Price, SplitRatio = 0.85) -> split_tag
subset(auto_data, split_tag==T) -> train
subset(auto_data, split_tag==F) -> test


# Building Linear Regression Model
ML_Model = train(Price ~ .+poly(Power,5)+poly(Engine.CC,5)+poly(Year,5)+poly(Seats,5),
                 data = auto_data, method = "lm",na.action = na.omit,
                 preProcess=c("scale","center"),
                 trControl= trainControl(method="none"))
```

```
# Prediction
test_pred_data = predict(ML_Model, newdata = test)
pred_data = cbind(Actual=test$Price/100000, Predicted=test_pred_data/100000)
pred_df = as.data.frame(pred_data)
error = (pred_df$Actual-pred_df$Predicted)
pred_df = cbind(pred_df,error)
rmse = sqrt(mean(error^2))


# Plotting Prediction vs Actual values
fit <- lm(Predicted ~ Actual, data = pred_df)

pred_df %>% plot_ly(x = ~Actual) %>%
            add_markers(y = ~Predicted, name="Scatter") %>%
            add_lines(x = ~Actual, y = fitted(fit), name="Regression Line")%>%
            layout(xaxis = list(title = "Actual"),
                  yaxis = list(title = "Predicted"))


validate_df = read.csv("C://Users/subha/Desktop/Visualization & Analysis on Automobile
Dataset using Machine Learning in R/validate.csv")
val_pred_df = predict(ML_Model, newdata = validate_df)

predict_data = cbind(Actual=validate_df$Price/100000, Predicted=predict_df/100000)
val_pred_df = as.data.frame(predict_data)
error = (val_pred_df$Actual-val_pred_df$Predicted)
val_pred_df = cbind(val_pred_df,error)
rmse = sqrt(mean(error^2))
```

REGRESSION PLOT

```
library(caret)
library(plotly)
library(heatmaply)


auto_data = read.csv("C://Users/subha/Desktop/Visualization & Analysis on Automobile
Dataset using Machine Learning in R/indian-auto-mpg.csv")

corr_df = data.frame(auto_data$Year, auto_data$Kilometers_Driven, auto_data$Engine.CC,
                  auto_data$Power, auto_data$Seats, auto_data$Mileage.Km.L, auto_data$Price)
heatmaply_cor(cor(corr_df),k_col = 2, k_row = 2)
corr <- round(cor(corr_df), 1)
p.mat <- cor_pmat(corr_df)
corr.plot <- ggcorrplot(corr, hc.order = TRUE, type = "lower", outline.col = "white",p.mat =
p.mat)
corr.plot
```

```
fit <- lm(Price ~ Year, data = auto_data)
auto_data %>% plot_ly(x = ~Year) %>%
        add_markers(y = ~Price, name="Scatter",
                marker = list(color="#636efa")) %>%
        add_lines(x = ~Year, y = fitted(fit), name="Regression Line")%>%
        layout(showlegend = T)


fit <- lm(Price ~ Kilometers_Driven, data = auto_data)
auto_data %>% plot_ly(x = ~Kilometers_Driven) %>%
        add_markers(y = ~Price, name="Scatter",
                marker = list(color="#00cc96")) %>%
        add_lines(x = ~Kilometers_Driven, y = fitted(fit), name="Regression Line")%>%
        layout(showlegend = T)


fit <- lm(Price ~ Engine.CC, data = auto_data)
auto_data %>% plot_ly(x = ~Engine.CC) %>%
        add_markers(y = ~Price, name="Scatter",
                marker = list(color="#ab63fa")) %>%
        add_lines(x = ~Engine.CC, y = fitted(fit), name="Regression Line")%>%
        layout(showlegend = T)



fit <- lm(Price ~ Power, data = auto_data)
auto_data %>% plot_ly(x = ~Power) %>%
        add_markers(y = ~Price, name="Scatter",
                marker = list(color="#e28676")) %>%
        add_lines(x = ~Power, y = fitted(fit), name="Regression Line")%>%
        layout(showlegend = T)



fit <- lm(Price ~ Seats, data = auto_data)
auto_data %>% plot_ly(x = ~Seats) %>%
        add_markers(y = ~Price, name="Scatter",
                marker = list(color="#ffff33")) %>%
        add_lines(x = ~Seats, y = fitted(fit), name="Regression Line")%>%
        layout(showlegend = T)



fit <- lm(Price ~ Mileage.Km.L, data = auto_data)

auto_data %>% plot_ly(x = ~Mileage.Km.L) %>%
        add_markers(y = ~Price, name="Scatter",
                marker = list(color="#1cbe4f")) %>%
        add_lines(x = ~Mileage.Km.L, y = fitted(fit), name="Regression Line")%>%
  layout(showlegend = T)
```