Report on Used Car DataSet

Adit Doshi

2022-08-08

Introduction

One of the key elements in boosting a nation's economy is its automobile sector. Prediction of used automobile sales has grown significantly in relevance in recent years. Over time, the amount of vehicles produced has steadily increased. People buy new or used cars based on their budgets as a result of the rising standard of living experienced by people worldwide. People who live in developing economies typically lease cars through contracts or loans. The car is delivered back to the dealer at the conclusion of the agreement. Thus, it is quite usual to see the market for used car sales outpace that of new car sales. Currently, as the car is sold on the secondary market, its price changes. The price of a new car is set by the manufacturer and is constant throughout the auto sales industry.

However, the price of a car in the used automobile market is relative, fluctuating, and dependent on a number of criteria, including the vehicle's brand or manufacturer, make, age, cylinder capacity, and fuel type. As a result,there is no established criteria for determining the second-hand car's selling price. The prices of autos are occasionally inflated by auto dealers who prey on the customer's ignorance. It's a fascinating problem to figure out how to maintain a regular price range for various cars on the used market while also knowing the needs of the market. In other cases, the client is unwilling to spend a lot of money on the vehicle, or frequently, the customer's decision to buy a vehicle is impacted by the vehicle's features and how comfortable it is.

When someone is considering to buy a used car, it is important to get the best possible deal for the car. The same can be achieved by keeping into account the factors that affect the pricing of the car. Factors such as Manufacturer's brand, condition of the car, odometer, title status, which year is the car model and state. This factors play an important role in deciding the price of the car. In order to get the best deal for a used car, it is important to understand what the average price of the car is in the used market, Once someone get a Idea of the average price, they can compare the listing price with the average price and decide if it is a good deal based on the specifications of the car.

This visualization helps identify what the average price of a car is in a state and how the features are correlated to each other to help a user identify the needs and price of the car.

Context

A data Visualization can be used by people to understand what car brand and model are popular and how are they prices based on different specifications of the car. It also helps a customer understand how the used car market work and make a correlation between the features of the car and what features are related. It helps visualize how the price of different cars and model are different in all the states and which car is more used in a state

Purpose of the Report

1. Which state has highest amount of car listing?
2. Which brand is highest for sale in the top 4 State's with highest number of car listing?
3. What is the mean & the price range of cars with different fuel type?
4. What is the price range of the different type of cars based on the odometer status?
5. Correlation of different types of features of the car
6. What is the average price per manufacturer and State?

Data-set

The vehicle data set consists of 426880 rows and 26 columns. An entry for a secondhand car can be found in each row.

Cleaning the Data-Set

Although not completely, I have cleaned the data set in accordance with the query. The data set was not completely cleaned because doing so would have resulted in all of the rows having some sort of missing value. I tried using the omit.na() command to solve the missing value problem in the entire data set at once. As a consequence, every row and column in the data set were entirely eliminated. To prevent this from occurring, I refrained from using this command and switched to cleaning each row/column separately as the necessity for the query arose. Here are a few instances of data cleaning:

1. fl_manu_count <- fl_manu_count[!(fl_manu_count$manufacturer == “ “),]
2. ca_manu_count <- ca_manu_count[!(ca_manu_count$manufacturer == “ “),]
3. manu_fuel_count <- manu_fuel_count[!(manu_fuel_count$price == 0),]
4. manu_fuel_count <- manu_fuel_count[!(manu_fuel_count$manufacturer == “NA”),]
5. manu_fuel_count <- manu_fuel_count[!(manu_fuel_count$fuel == “ “),]

Loading The Packages

```
packages <-
  c(
    "tidyverse",
    "janitor",
    "data.table",
    "devtools",
    "ggmap",
    "viridis",
    "lubridate",
    "reshape",
    "ggplot2",
    "wordcloud",
    "knitr",
    "rworldmap",
    "rworldxtra",
    "dplyr",
    "tidyr",
    "kableExtra"
  )
for (package in packages) {
  library(package, character.only = TRUE)
}
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Warning: package 'janitor' was built under R version 4.2.1
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
## Loading required package: usethis
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
## Loading required package: viridisLite
```

```
##
## Attaching package: 'lubridate'
## The following objects are masked from ’package:data.table’:
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union

## Warning: package ’reshape’ was built under R version 4.2.1

##
## Attaching package: ’reshape’

## The following object is masked from ’package:lubridate’:
##
##     stamp

## The following object is masked from ’package:data.table’:
##
##     melt

## The following object is masked from ’package:dplyr’:
##
##     rename

## The following objects are masked from ’package:tidyr’:
##
##     expand, smiths

## Warning: package ’wordcloud’ was built under R version 4.2.1

## Loading required package: RColorBrewer

## Warning: package ’rworldmap’ was built under R version 4.2.1

## Loading required package: sp

## ### Welcome to rworldmap ###

## For a short introduction type :   vignette(’rworldmap’)

## Warning: package ’rworldxtra’ was built under R version 4.2.1

## Warning: package ’kableExtra’ was built under R version 4.2.1

##
## Attaching package: ’kableExtra’

## The following object is masked from ’package:dplyr’:
##
##     group_rows
```

Loading the Data-set

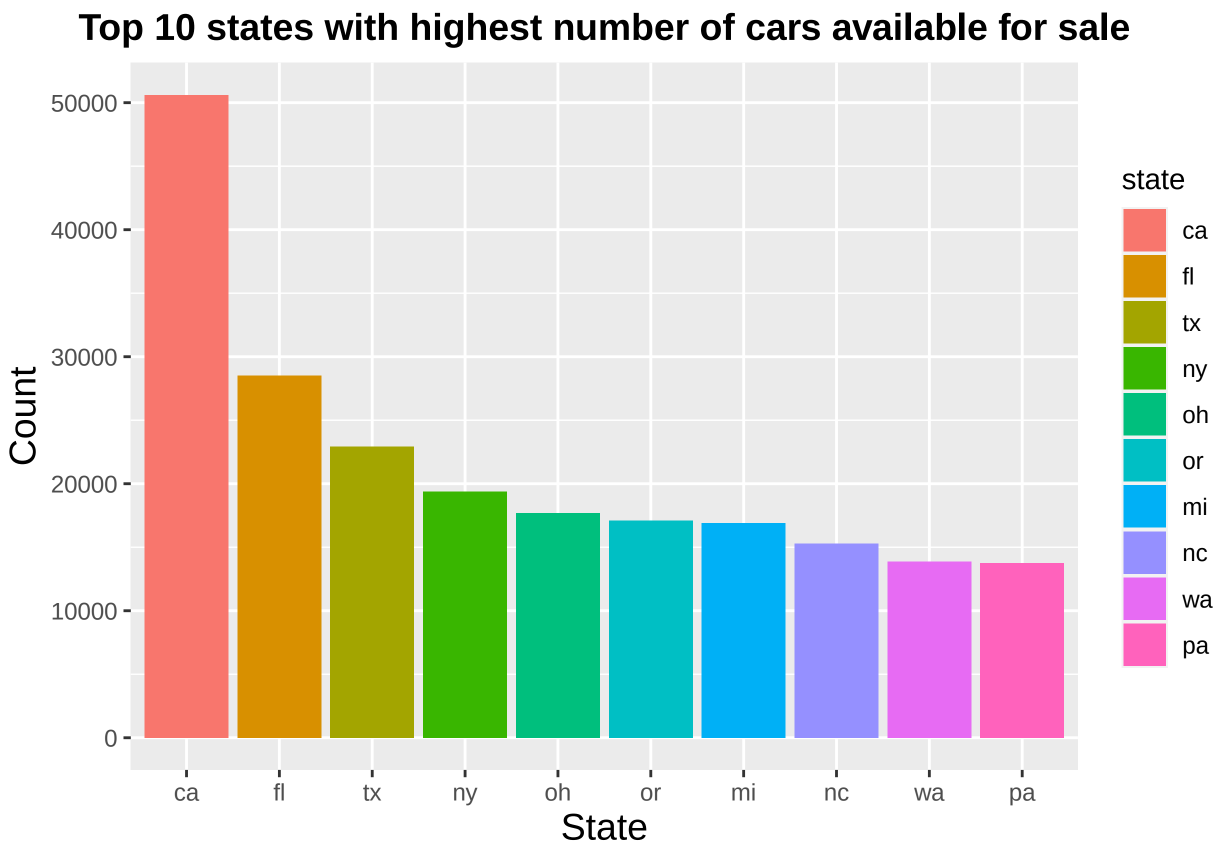
```r
vehicle<-fread("vehicles.csv")
```

Findings

```r
State_count <- vehicle %>% group_by(state) %>%
  count(sort = TRUE)
State_count1 <- State_count[1:10,]
State_count1
```

```
## # A tibble: 10 x 2
## # Groups:   state [10]
##    state     n
##    <chr> <int>
##  1 ca    50614
##  2 fl    28511
##  3 tx    22945
##  4 ny    19386
##  5 oh    17696
##  6 or    17104
##  7 mi    16900
##  8 nc    15277
##  9 wa    13861
## 10 pa    13753
```

```r
State_count1$state <- factor(State_count1$state,
                levels = State_count1$state[order(State_count1$n,
                decreasing = TRUE)])
state <- ggplot(State_count1, aes(x=state ,y=n, fill = state))+
  geom_bar(stat = "identity") + labs(x="State",y="Count") +
  ggtitle("Top 10 states with highest number of cars available for sale")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5))
state
```



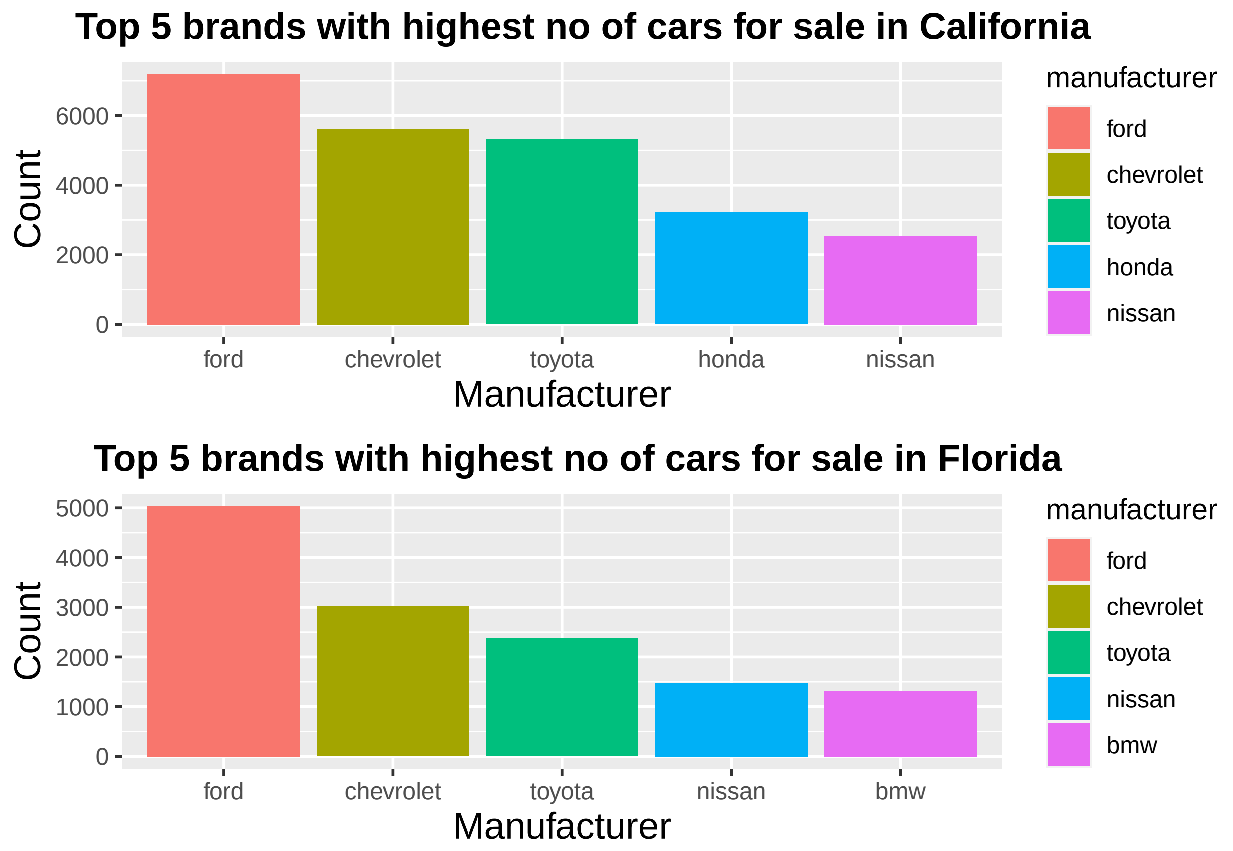
From the above bar plot, we can see that California has the highest number of listings in the Craigslist website. It is followed by the Florida state and Texas. California has more than 50k listing. From the plot we can see that Florida has approximately 50% less listing than California

```
df <- vehicle %>% filter(state == "ca")
df1 <- vehicle %>% filter(state == "fl")
df2 <- vehicle %>% filter(state == "tx")
df3 <- vehicle %>% filter(state == "ny")

ca_manu_count <- df %>% group_by(manufacturer) %>%
  count(sort = TRUE)
ca_manu_count <- ca_manu_count[!(ca_manu_count$manufacturer == ""), ]

ca_manu_count1 <- ca_manu_count[1:5,]
ca_manu_count1$manufacturer <- factor(ca_manu_count1$manufacturer,
                                levels = ca_manu_count1$manufacturer
                                [order(ca_manu_count1$n, decreasing = TRUE)])
fl_manu_count <- df1 %>% group_by(manufacturer) %>%
  count(sort = TRUE)
fl_manu_count <- fl_manu_count[!(fl_manu_count$manufacturer == ""), ]

fl_manu_count1 <- fl_manu_count[1:5,]
fl_manu_count1$manufacturer <- factor(fl_manu_count1$manufacturer,
                                levels = fl_manu_count1$manufacturer
                                [order(fl_manu_count1$n, decreasing = TRUE)])
```

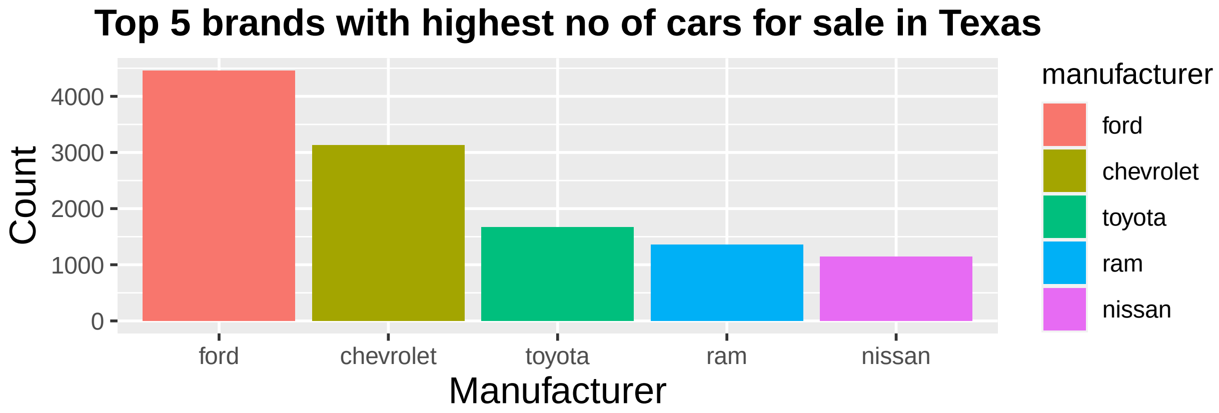
```r
tx_manu_count <- df2 %>% group_by(manufacturer) %>%
  count(sort = TRUE)
tx_manu_count <- tx_manu_count[!(tx_manu_count$manufacturer == ""), ]

tx_manu_count1 <- tx_manu_count[1:5,]
tx_manu_count1$manufacturer <- factor(tx_manu_count1$manufacturer,
                                    levels = tx_manu_count1$manufacturer
                                [order(tx_manu_count1$n, decreasing = TRUE)])

ny_manu_count <- df3 %>% group_by(manufacturer) %>%
  count(sort = TRUE)
ny_manu_count <- ny_manu_count[!(ny_manu_count$manufacturer == ""), ]


ny_manu_count1 <- ny_manu_count[1:5,]
ny_manu_count1$manufacturer <- factor(ny_manu_count1$manufacturer,
                                  levels = ny_manu_count1$manufacturer
                                  [order(ny_manu_count1$n, decreasing = TRUE)])

ca_manu <- ggplot(ca_manu_count1, aes(x=manufacturer ,y=n, fill = manufacturer))+
  geom_bar(stat = "identity") + labs(x="Manufacturer",y="Count") +
  ggtitle("    Top 5 brands with highest no of cars for sale in California")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5))

fl_manu <- ggplot(fl_manu_count1, aes(x=manufacturer ,y=n, fill = manufacturer))+
  geom_bar(stat = "identity") + labs(x="Manufacturer",y="Count") +
  ggtitle("   Top 5 brands with highest no of cars for sale in Florida")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5))

tx_manu <- ggplot(tx_manu_count1, aes(x=manufacturer ,y=n, fill = manufacturer))+
  geom_bar(stat = "identity") + labs(x="Manufacturer",y="Count") +
  ggtitle("  Top 5 brands with highest no of cars for sale in Texas")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5))

ny_manu <- ggplot(ny_manu_count1, aes(x=manufacturer ,y=n, fill = manufacturer))+
  geom_bar(stat = "identity") + labs(x="Manufacturer",y="Count") +
  ggtitle("  Top 5 brands with highest no of cars for sale in New York")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5))

library(ggpubr)
```

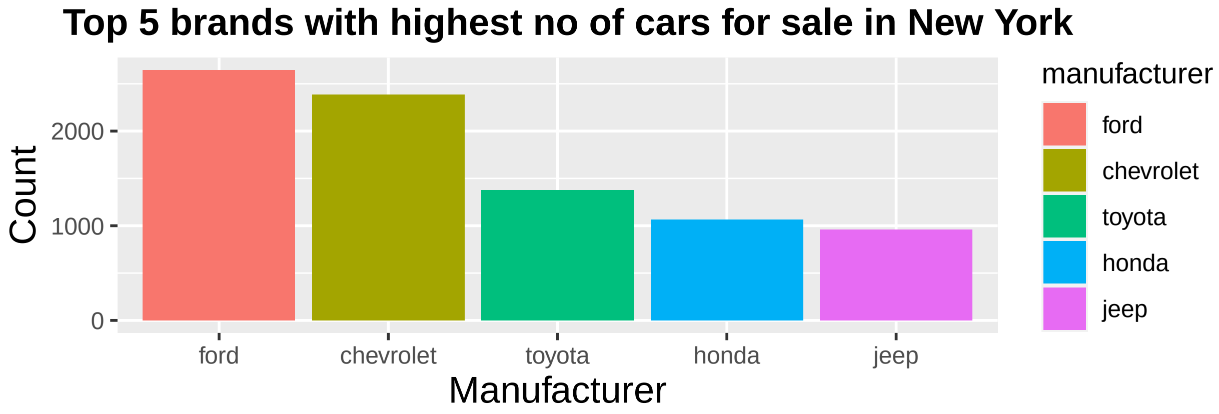
```
## Warning: package 'ggpubr' was built under R version 4.2.1
```

```
ggarrange(ca_manu, fl_manu, labels = c(), nrow= 2, ncol=1)
```



```
ggarrange(tx_manu, ny_manu, labels = c(), nrow= 2, ncol=1)
```





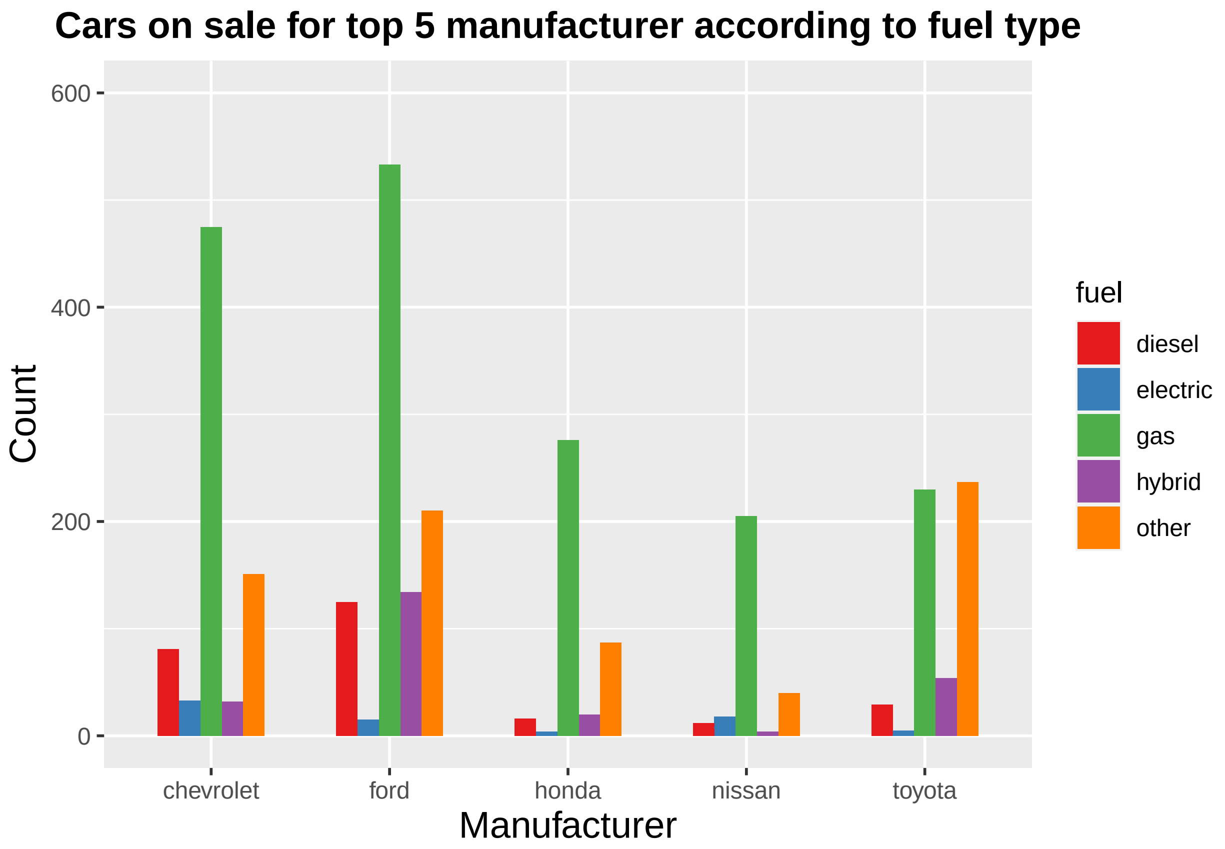
Here, I am filtering out Top 4 states which are - California, Florida, Texas & New York. From these graph, we can confirm that Ford has the highest number of listing in all the above mentioned states. This is because Ford stands out as the most popular automaker in the United States. This is hardly surprising considering that the Ford F-Series pickup line has been the best-selling truck in America for 43 years running.

The listing of Chevrolet and Toyota is approximately similar in the state of California and Florida. The listing of Chevrolet is approximately double than that of Toyota in the state of Texas and New York

```
veh1<- vehicle %>% filter(manufacturer %in% c("ford", "chevrolet", "toyota", "nissan", "honda"))

manu_fuel_count <- veh1 %>% group_by(manufacturer, price, fuel) %>%
  count(sort = TRUE)
manu_fuel_count <- manu_fuel_count[!(manu_fuel_count$price == 0), ]
manu_fuel_count <- manu_fuel_count[!(manu_fuel_count$manufacturer == "NA"), ]
manu_fuel_count <- manu_fuel_count[!(manu_fuel_count$fuel == ""), ]
manu_fuel_count$price  <- as.integer(manu_fuel_count$price)
```

```
## Warning in as.integer.integer64(manu_fuel_count$price): NAs produced by integer
## overflow
```

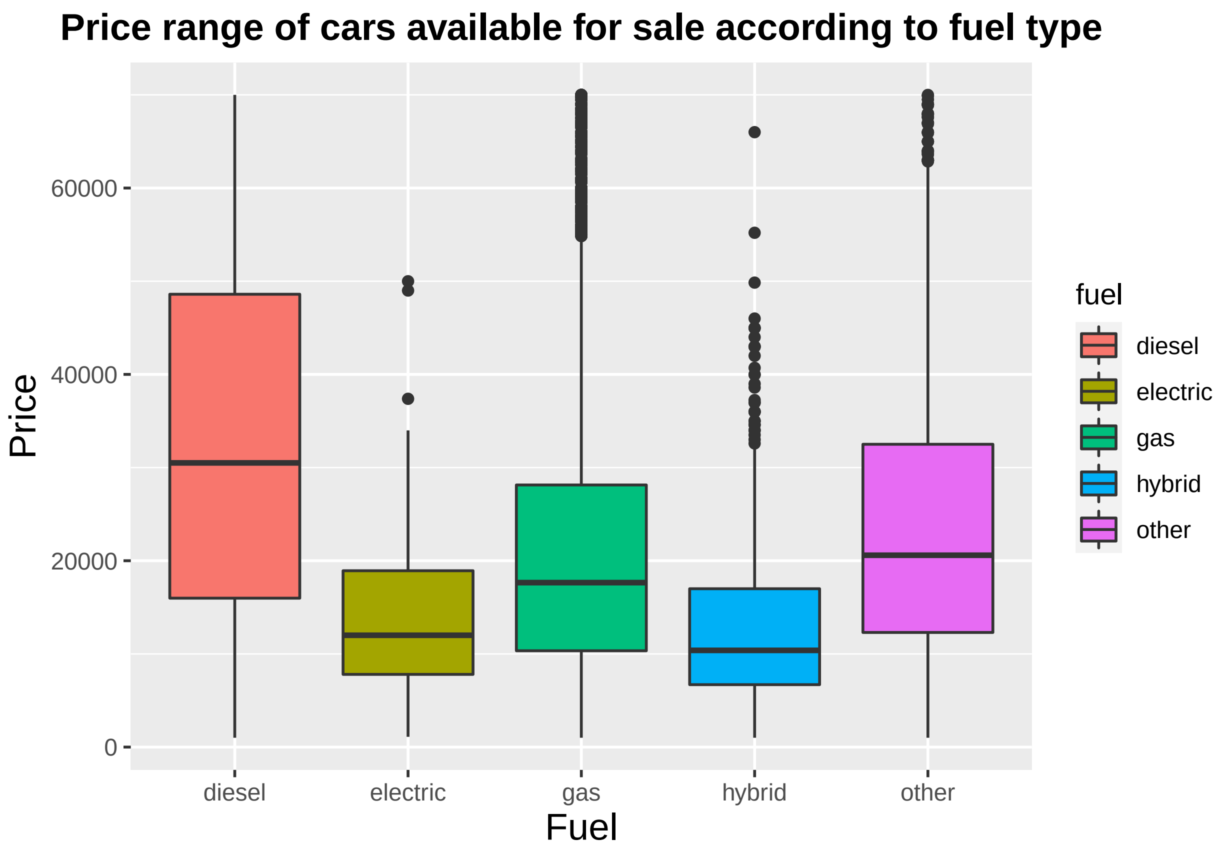
```
manu_fuel <- ggplot(manu_fuel_count, aes(x=manufacturer ,y=n, fill = fuel))+
  geom_col(position = "dodge" , width = 0.6)+
  ggtitle("Cars on sale for top 5 manufacturer according to fuel type")+
  labs(x="Manufacturer",y="Count")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5)) + ylim(0,600)
manu_fuel + scale_fill_brewer(palette = "Set1")
```



Irrespective of the car brand, Gas fuel type is the most listed On the website. This is because Diesel engines are heavier and perform less well than gasoline engines. Diesel is highest preferred after "gas" & "other" fuel type which is unknown. This can be because of Diesel fuel type which is thicker and provides more power and mileage than other fuel types

```
price_fuel <- ggplot(manu_fuel_count, aes(x=fuel ,y=price, fill = fuel))+
  geom_boxplot()+ labs(x="Fuel",y="Price") +
  ggtitle("Price range of cars available for sale according to fuel type")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5)) + ylim(1000, 70000)
price_fuel
```

```
## Warning: Removed 1951 rows containing non-finite values (stat_boxplot).
```



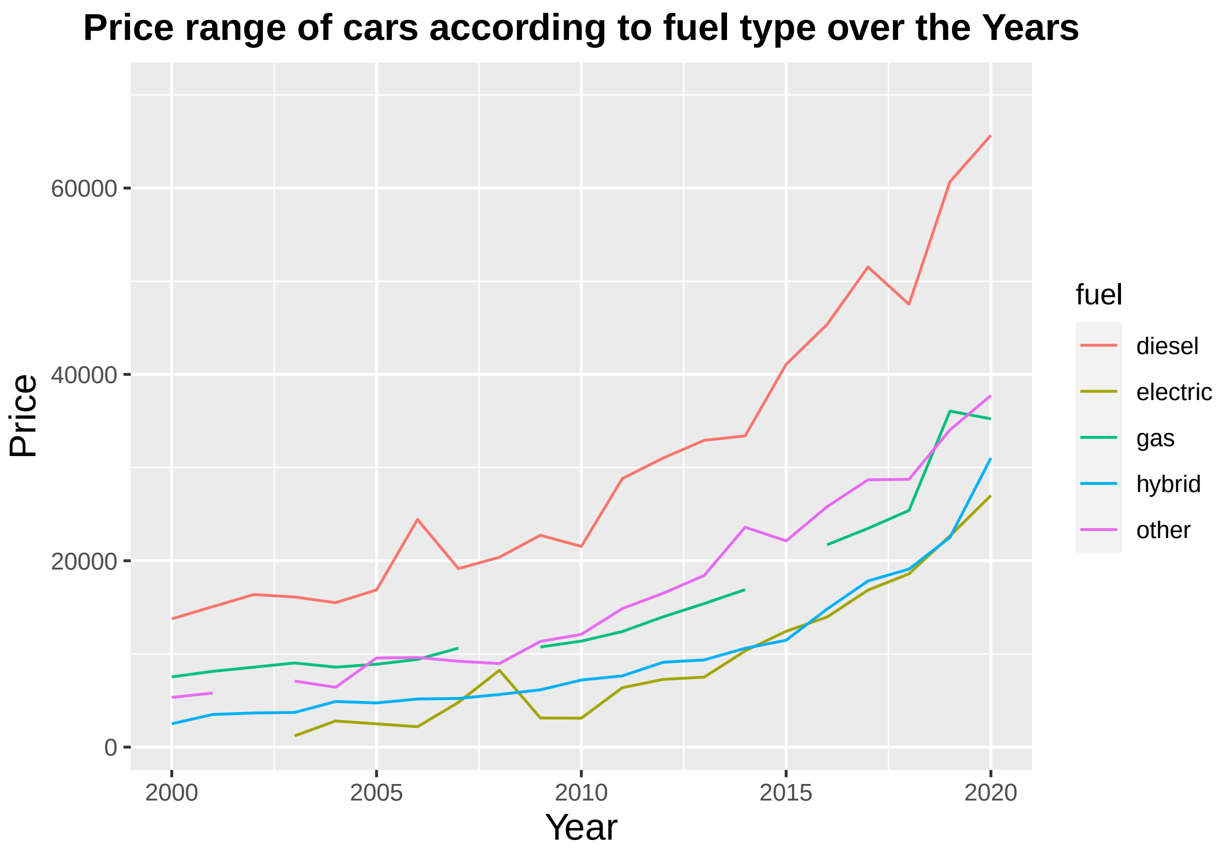
Here is a Box-plot which represents that the median of Diesel Fuel Cars is more than all the other fuel type cars even though as seen in the previous graph, Gas cars are the highest and more preferred on craigslist website.

```
year_fuel_count <- veh1 %>% group_by(year, price, fuel) %>%
  count(sort = TRUE)

year_fuel_count <- year_fuel_count[!(year_fuel_count$price == 0), ]
year_fuel_count <- year_fuel_count[!(year_fuel_count$fuel == ""), ]
year_fuel_count$price  <- as.integer(year_fuel_count$price)
```

```
## Warning in as.integer.integer64(year_fuel_count$price): NAs produced by integer
## overflow
```

```
year_fuel_count1 <-aggregate( price ~ year + fuel, year_fuel_count, mean )
year_count <- ggplot(year_fuel_count1, aes(x=year, y=price, colour = fuel)) +
  geom_line() + labs(x="Year",y="Price") +
  ggtitle("Price range of cars according to fuel type over the Years")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5)) + ylim(1000, 70000) + xlim(2000, 2020)

year_count
```

```
## Warning: Removed 173 row(s) containing missing values (geom_path).
```



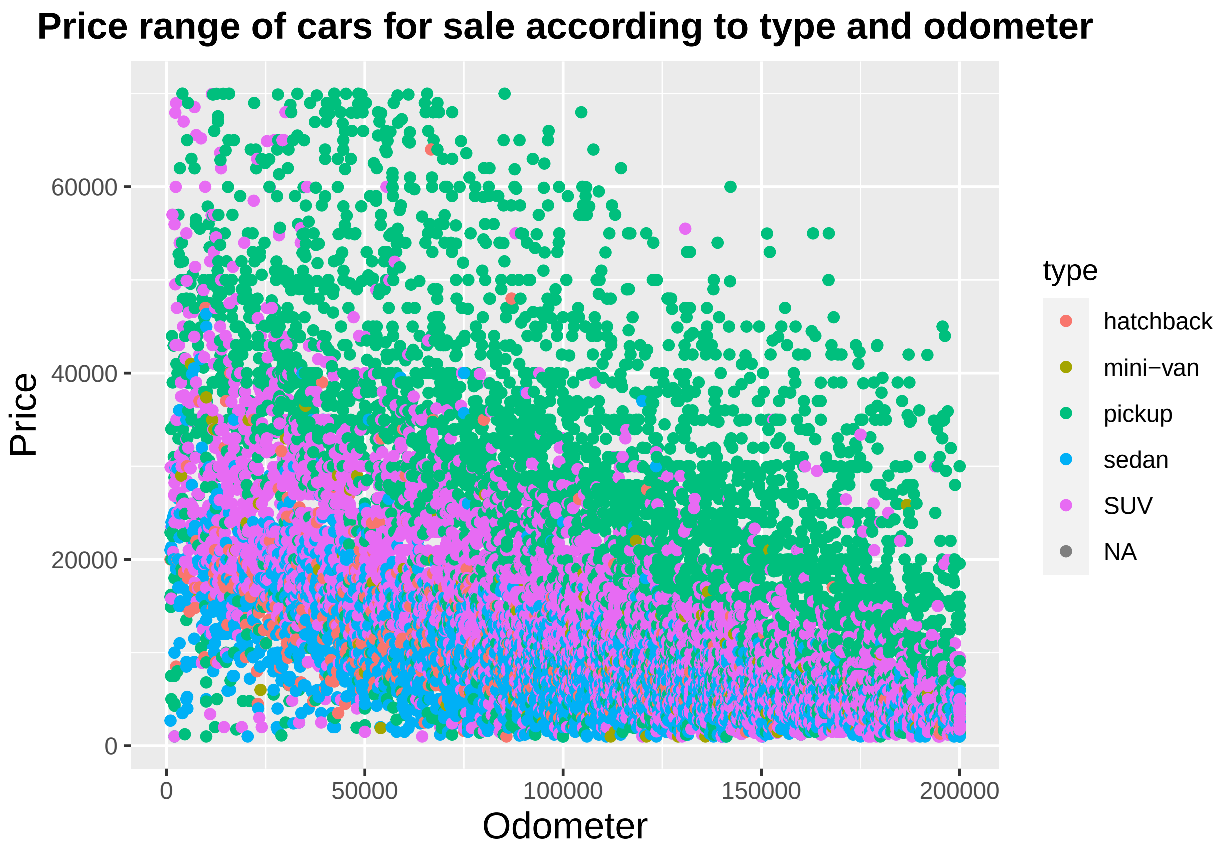
From the above line graph, we can interpret that the diesel cars prices have increase significantly over the years. We can even observe a small downfall during 2007-2008 that is due to the recession due to the economic crisis. We can observe that break in the line graph which is due to the missing values in the data set.

```
veh2 <- vehicle %>% filter(manufacturer == "ford") %>% filter(type %in%
                                      c("hatchback", "mini-van", "pickup", "sedan", "SUV"))
type_odo_count <- veh2%>% group_by(odometer, type, price) %>%
  count(sort = TRUE)
type_odo_count <- type_odo_count[!(type_odo_count$price == 0), ]
type_odo_count <- type_odo_count[!(type_odo_count$odometer == 0), ]
type_odo_count <- type_odo_count[!(type_odo_count$type == ""), ]
type_odo_count
```

```
## # A tibble: 15,646 x 4
## # Groups:   odometer, type, price [15,557]
##    odometer type     price     n
##       <int> <chr>  <int64> <int>
##  1    70760 SUV      29590   261
##  2    10688 pickup   27990   231
##  3    14230 pickup   26990   195
##  4    14169 pickup   22590   132
##  5    10740 pickup   30590   121
##  6     5468 pickup   30990   109
##  7     1834 pickup   30990    87
##  8    32652 pickup   21990    84
## 9      9313 pickup   40590    78
## 10    67344 pickup   30990    72
## # ... with 15,636 more rows
```

```
type_odo_count$price  <- as.integer(type_odo_count$price)
type_count <- ggplot(type_odo_count, aes(x=odometer, y=price, colour = type)) +
  geom_point() + labs(x="Odometer",y="Price") +
  ggtitle("Price range of cars for sale according to type and odometer")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5)) + ylim(1000, 70000) + xlim(1000, 200000)
type_count
```

```
## Warning: Removed 1885 rows containing missing values (geom_point).
```



Here, We are trying to visualize that the price of a car decreases if the car has high amount of miles on it and the price is high if the car has low amount of miles on it. We can also observe that the cost of the pickup trucks/cars is significantly high even if it has more number of miles on it.

```
veh3 <- vehicle %>% filter(manufacturer == "ford")

ford_model_count <- veh3 %>% group_by(model) %>%
  count(sort = TRUE)
ford_model_count
```

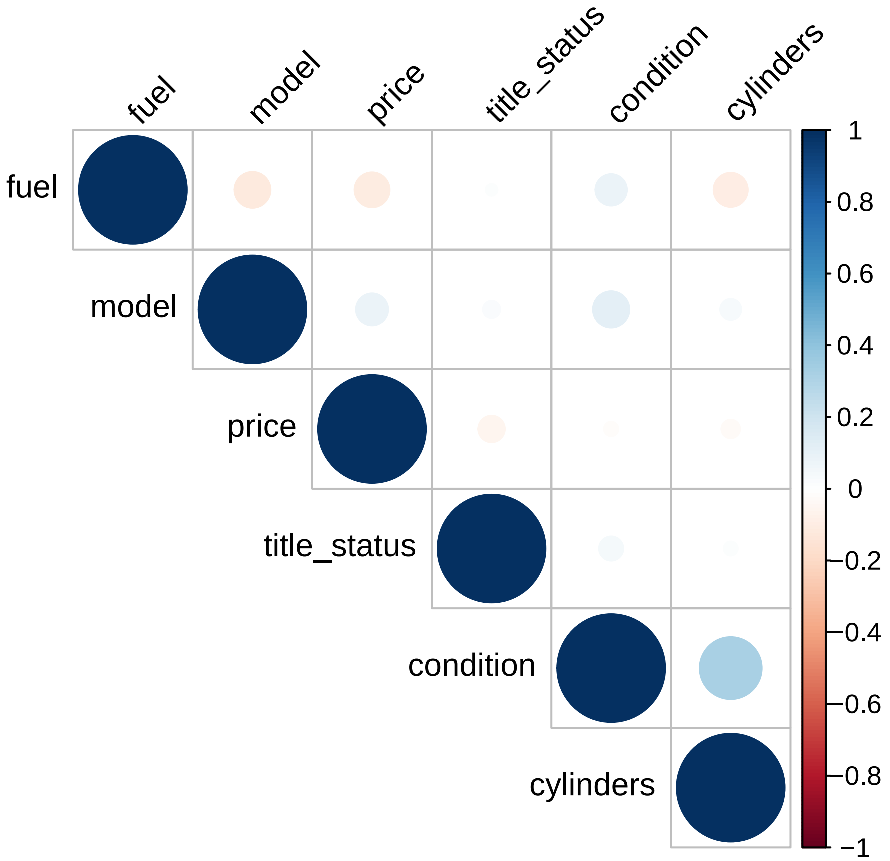
```
## # A tibble: 3,875 x 2
## # Groups:   model [3,875]
##    model        n
##    <chr>    <int>
##  1 f-150     8009
##  2 escape    2745
##  3 explorer  2499
##  4 mustang   2220
##  5 fusion    1979
##  6 focus     1828
##  7 f-250     1529
##  8 edge      1471
##  9 f-350     1116
## 10 f150      1085
## # ... with 3,865 more rows
```

```
ford_char <- veh3 %>% select(model, condition, cylinders, fuel, title_status, price)
ford_char$model <- as.numeric(as.factor(ford_char$model))
ford_char$condition <- as.numeric(as.factor(ford_char$condition))
ford_char$cylinders <- as.numeric(as.factor(ford_char$cylinders))
ford_char$fuel <- as.numeric(as.factor(ford_char$fuel))
ford_char$title_status <- as.numeric(as.factor(ford_char$title_status))
ford_char$price <- as.numeric(as.factor(ford_char$price))
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.1
```

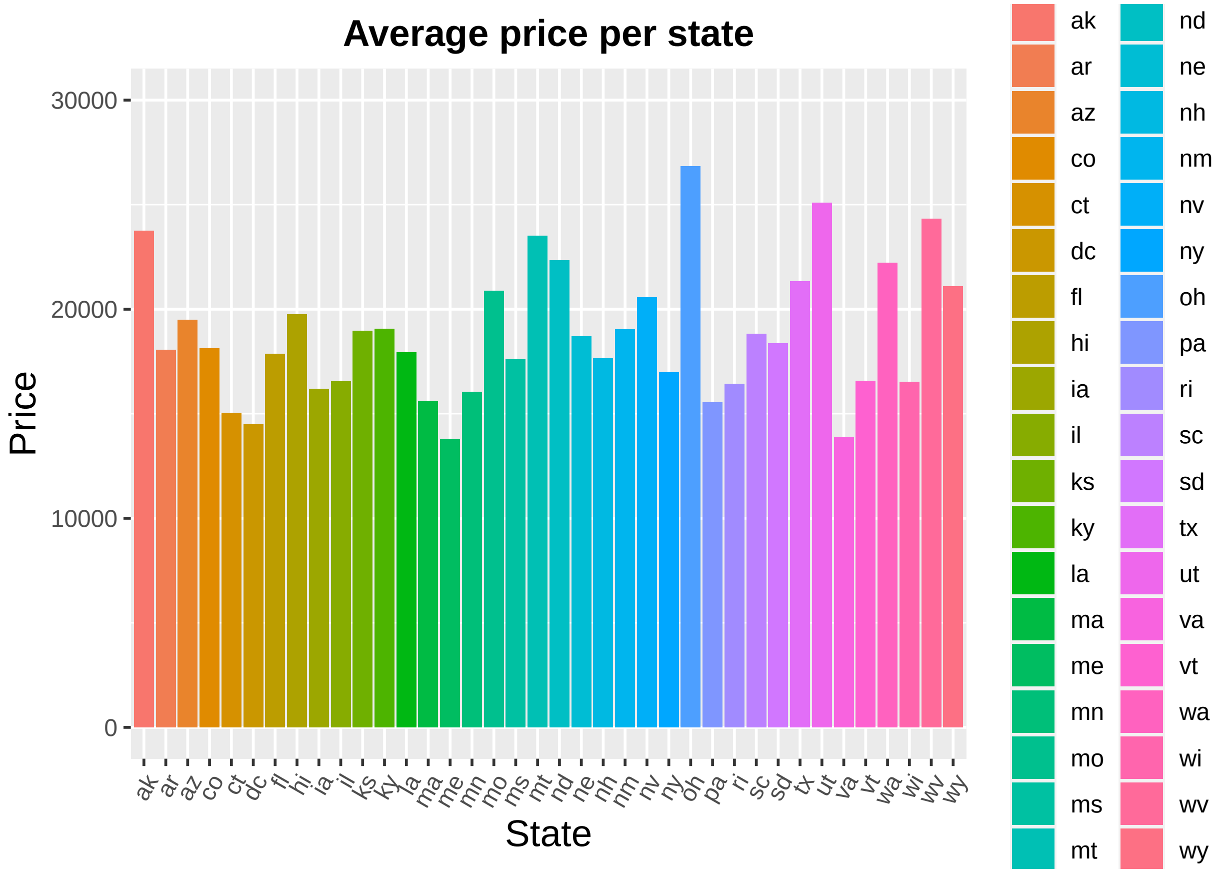
```
## corrplot 0.92 loaded
```

```
corrplot(cor(ford_char), type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```



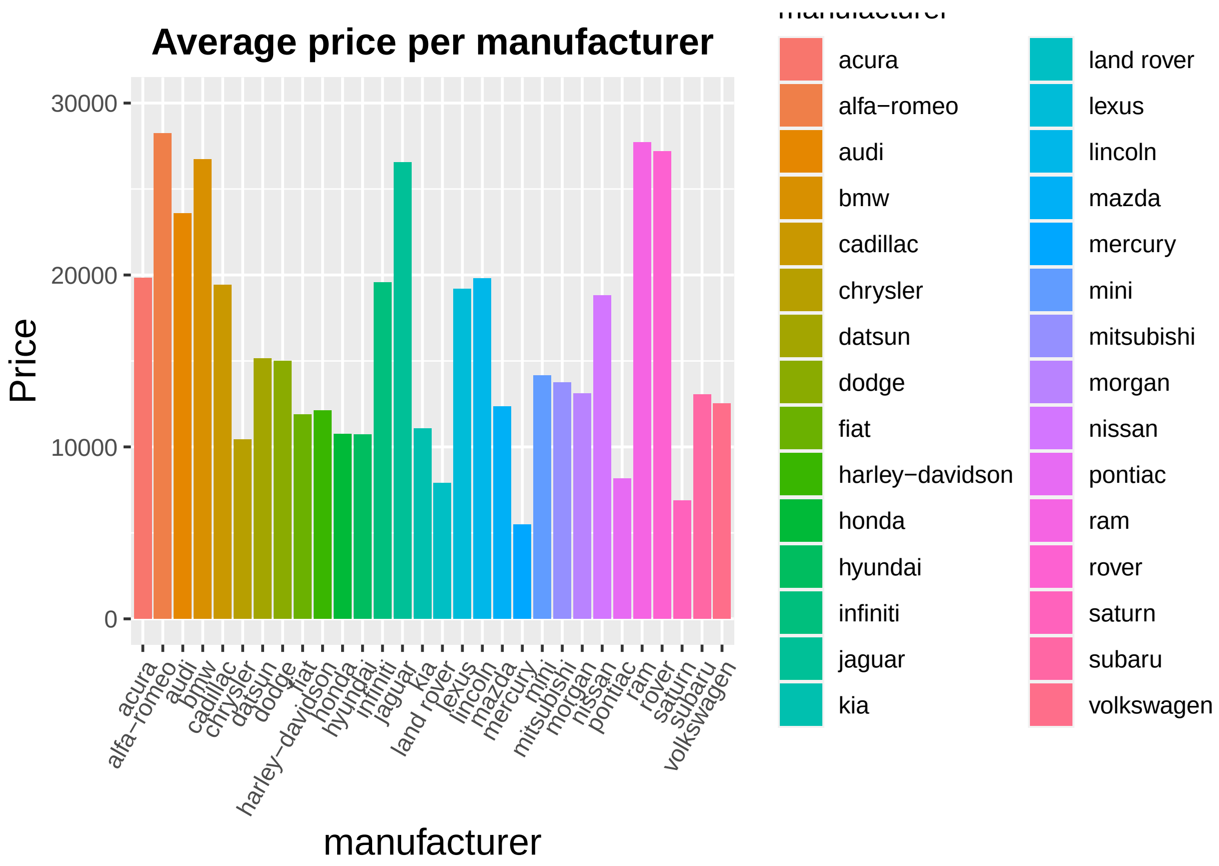
This is a correlation graph. Here, 0 to 1 represents a positive correlation(in Blue), while 0 to -1 represents a negative correlation(In Red) Here we can observe that, the Condition is positively correlated with the number of cylinders in the car. cars with 6 cylinders is highest in the data listing. Similarly we can also see a correlation between fuel type, model, price, number of cylinder and conditions.

```
avg_price_state <-aggregate( price ~ state, vehicle, mean)
avg_price_state <- filter(avg_price_state, !price > 30000)
avg_price_manufacturer <-aggregate( price ~ manufacturer, vehicle, mean)
avg_price_manufacturer <- filter(avg_price_manufacturer, !price > 30000)

avg_state <- ggplot(avg_price_state, aes(x=state ,y=price, fill = state))+
  geom_bar(stat = "identity") + labs(x="State",y="Price") +
  ggtitle("Average price per state")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5))+
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
 ylim(0, 30000)
avg_state
```



The above Bar-plot represent average price of cars per state. We can see that Ohio(OH) has the highest average of car price, this can be because the real-estate in Ohio is comparatively cheap than other states, this is why people might prefer buying luxury cars to commute. Similarly, Maine(ME) has the lowest average car price compared to other states.

```
avg_manufacturer <- ggplot(avg_price_manufacturer, aes(x=manufacturer ,y=price, fill = manufacturer))+
  geom_bar(stat = "identity") + labs(x="manufacturer",y="Price") +
  ggtitle("Average price per manufacturer")+
  theme(plot.title = element_text(color="black", size=14, face="bold"),
        axis.title.x = element_text(color="black", size=14),
        axis.title.y = element_text(color="black", size=14)) +
  theme(plot.title=element_text(hjust=0.5)) + ylim(0, 30000) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
avg_manufacturer
```

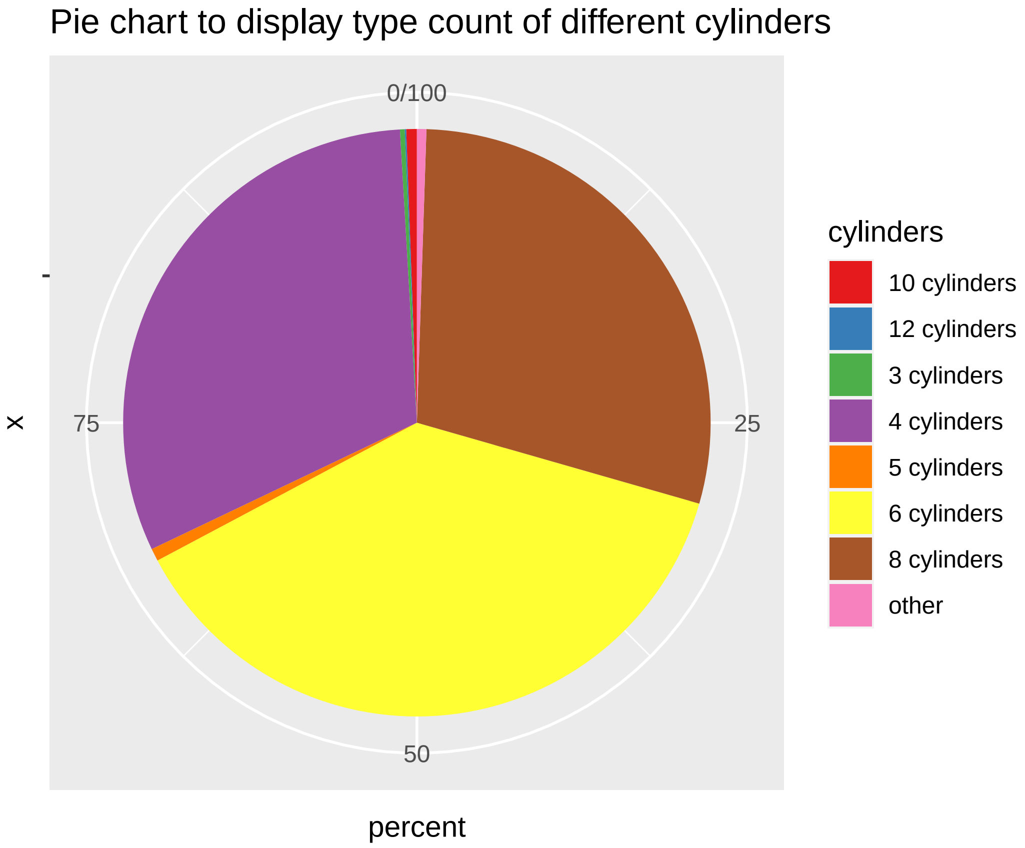


The above Bar-plot represents the average price of the cars based on the brands. We can see that the brand Alfa-Romeo has the highest average price of the cars. This is because Alfa-Romeo is an Italian luxury car manufacturer.

```
title_status_count <- vehicle %>% group_by(cylinders) %>%
  count(sort = TRUE)
title_status_count <- title_status_count[!(title_status_count$cylinders == ""), ]
sum <- sum(title_status_count$n)
title_status_count$percent <- round(((title_status_count$n)/sum)*100, 2)
title_status_count
```

```
## # A tibble: 8 x 3
## # Groups:   cylinders [8]
##   cylinders        n percent
##   <chr>        <int>   <dbl>
## 1 6 cylinders  94169   37.8
## 2 4 cylinders  77642   31.2
## 3 8 cylinders  72062   28.9
## 4 5 cylinders   1712    0.69
## 5 10 cylinders  1455    0.58
## 6 other         1298    0.52
## 7 3 cylinders    655    0.26
## 8 12 cylinders   209    0.08
```

```r
pie <- ggplot(title_status_count, aes(x = "", y = percent, fill = cylinders)) +
  geom_col() +   ggtitle("Pie chart to display type count of different cylinders") +
  coord_polar(theta = "y")
pie + scale_fill_brewer(palette = "Set1")
```



The above pie-chart represents number of cars with different types of cylinders. We can interpret that cars with 6 cylinders have the highest amount of listing followed by 4 cylinders and 8 cylinder. This can be because people prefer 6 cylinders over 4 cylinders engine can produce low RPM torque and power better than a turbocharged four-cylinder

Conclusion

We can see that there are several categorical columns in the vehicle data set. The above visualizations helps us answer our questions. I have chosen 4 different states that is California, Florida, Texas and New york. These are the top 4 states with the highest number of listings. Among these 4 states, California has the highest number of listings. This can be due to the population of the California. California stands at the top in the United states for its population with the population of 39.35 Million people.

Later I am focusing on these 4 states. What we have observed from the above visualization is they had common Top Manufacturer which is ford. Out of the 4 states, 3 of the states have the same highest manufacturers. Ford was able to avoid bankruptcy in the 2008 financial collapse. Ford has the highest number of listings in all the states is because of the pickup trucks. Ford is among the companies to offer a range of transportation services rather than simply producing and selling cars.

Moving on, I am choosing Top 5 manufacturers and comparing their fuel type. We notice that most number of cars have Gas as their fuel type followed by Diesel. Gas is cost competitive, relatively abundant hence convenient and the cleanest burning fossil fuel. We even observe that electric cars are comparatively less in the listings. The main reason can be electric cars were introduced in the later years. The older electric cars were less convenient as there were less number of charging stations. Additionally, We notice that Electric cars are comparatively cheaper in price. This is because there is no need to change the oil regularly, No engine to manage and fewer parts to wear down. Electric cars are cost efficient and maintain than internal combustion engine vehicles. From the Box-plot we also notice that cars with gas fuel type have moderate prices. The graph also tells us about the steady increase in the price of diesel cars over the years. We have aggregated the price in the line graph according to the year and the fuel type. Even though all the fuel type cars prices have increase over the years, diesel has shown significant growth in the price.

I have then explored a scatter plot of odometer vs Price in-accordance to different types of car. We can see that Odometer and the price is inversely proportional. We can see that as the odometer increases, the price of the car decreases and vice-versa and it is quite visible in the scatter plot as we see a downward trend. We see that Pickup Cars(Green Dots) are the top. This means that price of Pickup trucks is quite higher than normal cars despite of the odometer. This is because Pickup Trucks are built to be more durable which gives them a high resell value. The prices are high because their demand out-strips the supply.

The correlation plot shows us the positive relation between Cylinder and Condition of the car. It also shows us a negative relation between fuel and cylinders because more the number of cylinders it will consume more amount of fuel. The plot explains us a positive relation between price and the model. Newer the model higher the price.

The next Graph represents Average price of cars in different states. We notice that Ohio(OH) has higher average price compared to other states. This can be because the standard of living and less population. We also observe that Maine(ME) has the lowest average car price. Maine is an affordable place to own a vehicle as its the cheapest state for car insurance.

Future Scope

1. Combine a new dataset to get a better analysis of the car listings. Also a new column to view if the car at the listed price was sold.
2. Compare a Weather dataset to find out which car is sold more in different weather conditions.

Data Dictionary

1. manufacturer: Brand of the vehicle
2. price: The vehicle's asking price as stated in the listing.
3. State: The state code of the listing's creation.
4. year: The year that the specified vehicle was first registered.
5. condition: List of the vehicle's conditions.
6. cylinders: The number of cylinders an engine has determines how big it is.
7. size: The vehicle's size classification.
8. lat: The listing's latitude point of origin.
9. long: The listing's longitude starting point.
10. region: The area from whence the listing was created.
11. model: The vehicle's model number.
12. odometer: The amount of miles displayed on the car's odometer.
13. type Distinguishes the cars according to their type
14. posting_date: Date of when the listing was made.
15. drive: Contains information about how the drive train delivers its power eg. AWD, FWD, RWD etc.
16. transmission: The type of transmission on the vehicle. eg Automatic, Manual

```
sessionInfo()
```

```
## R version 4.2.0 (2022-04-22 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_India.utf8  LC_CTYPE=English_India.utf8
## [3] LC_MONETARY=English_India.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_India.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] corrplot_0.92      ggpubr_0.4.0       kableExtra_1.3.4   rworldxtra_1.01
##  [5] rworldmap_1.3-6    sp_1.5-0           knitr_1.39         wordcloud_2.6
##  [9] RColorBrewer_1.1-3 reshape_0.8.9      lubridate_1.8.0    viridis_0.6.2
## [13] viridisLite_0.4.0  ggmap_3.0.0        devtools_2.4.3     usethis_2.1.6
## [17] data.table_1.14.2  janitor_2.1.0      forcats_0.5.1      stringr_1.4.0
## [21] dplyr_1.0.9        purrr_0.3.4        readr_2.1.2        tidyr_1.2.0
## [25] tibble_3.1.7       ggplot2_3.3.6      tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] colorspace_2.0-3    ggsignif_0.6.3      rjson_0.2.21
##  [4] ellipsis_0.3.2      rprojroot_2.0.3     snakecase_0.11.0
##  [7] fs_1.5.2            rstudioapi_0.13     farver_2.1.0
## [10] remotes_2.4.2       bit64_4.0.5         fansi_1.0.3
## [13] xml2_1.3.3          cachem_1.0.6        pkgload_1.2.4
## [16] spam_2.9-0          jsonlite_1.8.0      broom_0.8.0
## [19] dbplyr_2.2.0        png_0.1-7           compiler_4.2.0
## [22] httr_1.4.3          backports_1.4.1     assertthat_0.2.1
## [25] fastmap_1.1.0       cli_3.3.0           htmltools_0.5.2
## [28] prettyunits_1.1.1   tools_4.2.0         dotCall64_1.0-1
## [31] gtable_0.3.0        glue_1.6.2          maps_3.4.0
## [34] Rcpp_1.0.8.3        carData_3.0-5       cellranger_1.1.0
## [37] vctrs_0.4.1         svglite_2.1.0       xfun_0.31
## [40] ps_1.7.0            brio_1.1.3          testthat_3.1.4
## [43] rvest_1.0.2         lifecycle_1.0.1     rstatix_0.7.0
## [46] scales_1.2.0        hms_1.1.1           fields_14.0
## [49] yaml_2.3.5          memoise_2.0.1       gridExtra_2.3
## [52] stringi_1.7.6       highr_0.9           maptools_1.1-4
## [55] desc_1.4.1          pkgbuild_1.3.1      RgoogleMaps_1.4.5.3
## [58] rlang_1.0.2         pkgconfig_2.0.3     systemfonts_1.0.4
## [61] bitops_1.0-7        evaluate_0.15       lattice_0.20-45
## [64] labeling_0.4.2      cowplot_1.1.1       bit_4.0.4
## [67] processx_3.5.3      tidyselect_1.1.2    plyr_1.8.7
## [70] magrittr_2.0.3      R6_2.5.1            generics_0.1.2
## [73] DBI_1.1.2           pillar_1.7.0        haven_2.5.0
## [76] foreign_0.8-82      withr_2.5.0         abind_1.4-5
## [79] car_3.1-0           modelr_0.1.8        crayon_1.5.1
## [82] utf8_1.2.2          tzdb_0.3.0          rmarkdown_2.14
## [85] jpeg_0.1-9          grid_4.2.0          readxl_1.4.0
## [88] callr_3.7.0         reprex_2.0.1        digest_0.6.29
## [91] webshot_0.5.3       munsell_0.5.0       sessioninfo_1.2.2
```