Statistical Analysis of Car Data using Tidyverse in RStudio:

1. Introduction:

In this report we analyze the data of 405 cars on nine characteristics (variables). The data has been collected for cars that are mainly manufactured in US, Europe and Japan from year 1970 to 1982. In the report we will discuss the entire process of data analysis from loading data in RStudio to analyzing it using Tidyverse. We will also use ggplot2 and dplyr in analyzing this dataset.

2. Loading dataset and making it tidy:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.
1 --

## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.1     v dplyr   0.7.4
## v tidyr   0.7.2     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.2.0

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

Now we will load the dataset.

```
cardata<-read_csv("D:/Box Sync/MS IS/R Projects/Data Wrangling in R/cars.c
sv")
```

```
## Parsed with column specification:
## cols(
##   `Car;MPG;Cylinders;Displacement;Horsepower;Weight;Acceleration;Model;
Origin` = col_character()
## )
```

```
glimpse(cardata) #for overview of the dataset
```

```
## Observations: 407
## Variables: 1
## $ `Car;MPG;Cylinders;Displacement;Horsepower;Weight;Acceleration;Model;
Origin` <chr> ...
```

We can see that the entire dataset is in just one column and we need to separate the data. To do that we will use “read_delim” function on our dataset.

```
caradata<-read_delim("D:/Box Sync/MS IS/R Projects/Data Wrangling in R/car
s.csv",";")
```

```
## Parsed with column specification:
## cols(
##   Car = col_character(),
##   MPG = col_character(),
##   Cylinders = col_character(),
##   Displacement = col_character(),
##   Horsepower = col_character(),
##   Weight = col_character(),
##   Acceleration = col_character(),
##   Model = col_character(),
##   Origin = col_character()
## )
```

```
glimpse(caradata)
```

```
## Observations: 407
## Variables: 9
## $ Car          <chr> "STRING", "Chevrolet Chevelle Malibu", "Buick Sky.
..
## $ MPG          <chr> "DOUBLE", "18.0", "15.0", "18.0", "16.0", "17.0",.
..
## $ Cylinders    <chr> "INT", "8", "8", "8", "8", "8", "8", "8", "8", "8.
..
## $ Displacement <chr> "DOUBLE", "307.0", "350.0", "318.0", "304.0", "30.
..
## $ Horsepower   <chr> "DOUBLE", "130.0", "165.0", "150.0", "150.0", "14.
..
## $ Weight       <chr> "DOUBLE", "3504.", "3693.", "3436.", "3433.", "34.
..
## $ Acceleration <chr> "DOUBLE", "12.0", "11.5", "11.0", "12.0", "10.5",.
..
## $ Model        <chr> "INT", "70", "70", "70", "70", "70", "70", "70", .
..
## $ Origin       <chr> "CAT", "US", "US", "US", "US", "US", "US", "US", .
..
```

As we can see that the first row in our dataset is datatypes instead of values. We would like to change that and skip this line while loading the data. We will also want to change to names of our variables and keep them in lower case for ease of usage. We can do all of this by reloading data with some additional parameters as shown below.

```
#created a vector to change the names of the variables in our dataset.
```

```
varnames<-
c("car","mpg","cylinders","displacement","horsepower","weight","accelerati
on","model","origin")
```

```
cardata<-read_delim("D:/Box Sync/MS IS/R Projects/Data Wrangling in R/cars
.csv",";", skip=2, col_names = varnames)
```

```
## Parsed with column specification:
## cols(
##   car = col_character(),
##   mpg = col_double(),
##   cylinders = col_integer(),
##   displacement = col_double(),
##   horsepower = col_double(),
##   weight = col_double(),
##   acceleration = col_double(),
##   model = col_integer(),
##   origin = col_character()
## )
```

```
#having a look at the first 10 rows of the data.
```

```
head(cardata, 10)
```

```
## # A tibble: 10 x 9
##    car              mpg cylind~ displac~ horsep~ weight accele~ model or
ig~
##    <chr>          <dbl>   <int>    <dbl>   <dbl>  <dbl>   <dbl> <int> <c
hr>
##  1 Chevrolet Ch~  18.0       8      307     130   3504   12.0     70 US
##  2 Buick Skylar~  15.0       8      350     165   3693   11.5     70 US
##  3 Plymouth Sat~  18.0       8      318     150   3436   11.0     70 US
##  4 AMC Rebel SST  16.0       8      304     150   3433   12.0     70 US
##  5 Ford Torino    17.0       8      302     140   3449   10.5     70 US
##  6 Ford Galaxie~  15.0       8      429     198   4341   10.0     70 US
##  7 Chevrolet Im~  14.0       8      454     220   4354    9.00    70 US
##  8 Plymouth Fur~  14.0       8      440     215   4312    8.50    70 US
##  9 Pontiac Cata~  14.0       8      455     225   4425   10.0     70 US
## 10 AMC Ambassad~  15.0       8      390     190   3850    8.50    70 US
```

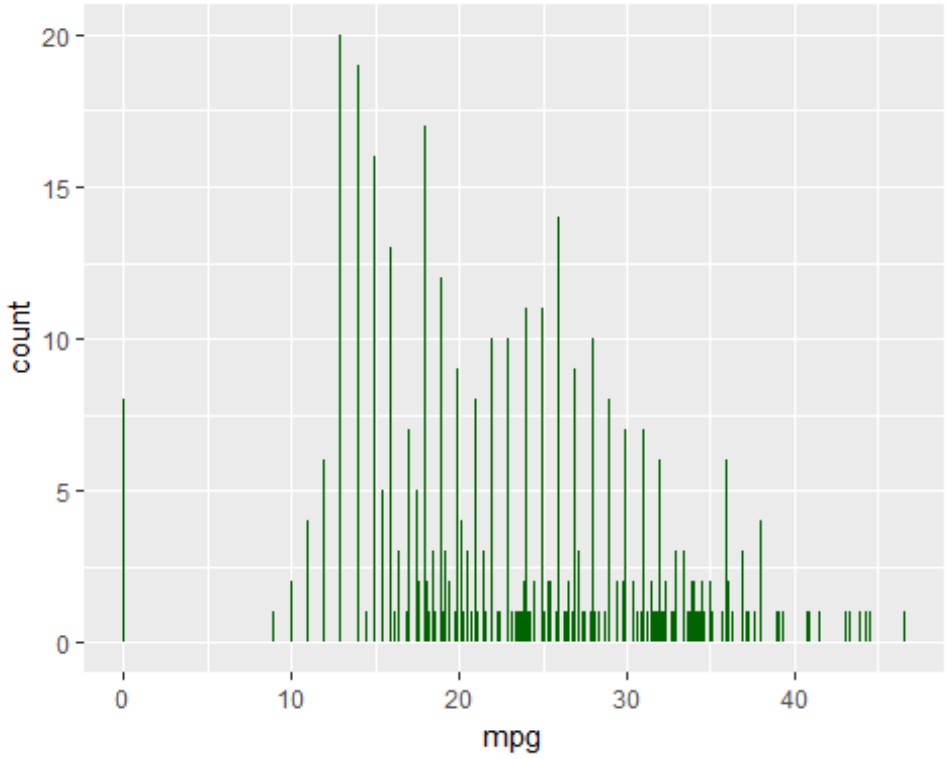
3. Detecting outliers and missing values:

```
which(is.na(cardata)) # Selecting which values in our dataset are NA
```

```
## integer(0)
```

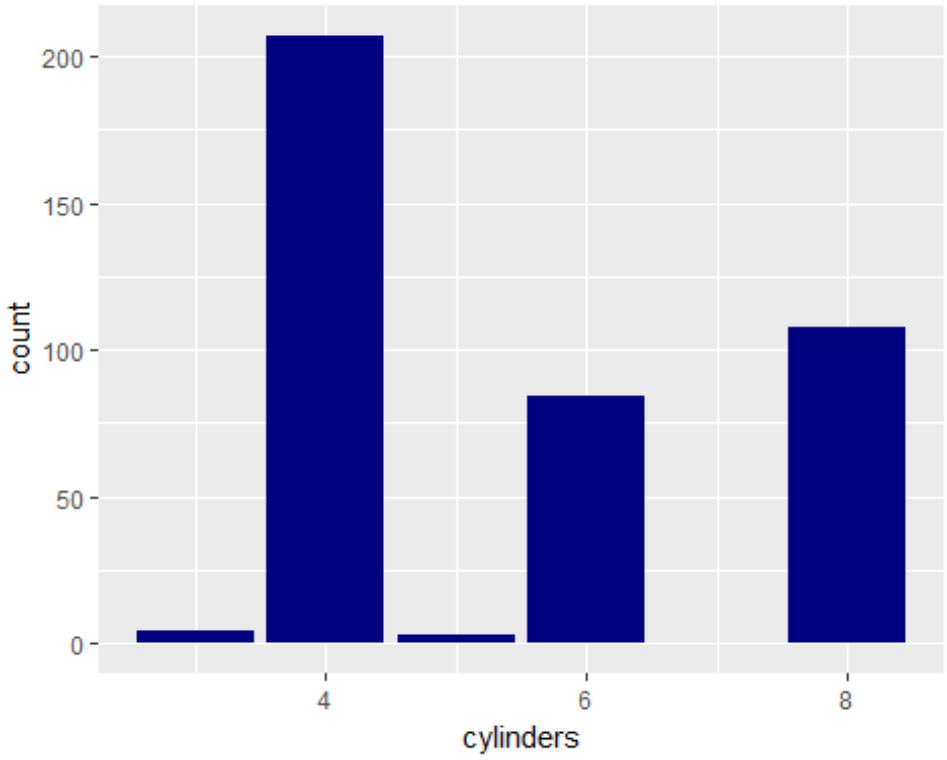
We can see that there no NA values in our data set. Now we will try to find out if there are any outliers in our dataset that are there due to some error. For that, we will make quick bar charts without much formatting of all our variables to see the range of values.

```
# Making quick bargraphs to analyse outliers in the dataset
ggplot(cardata, aes(mpg))+geom_bar(fill="dark green")
```



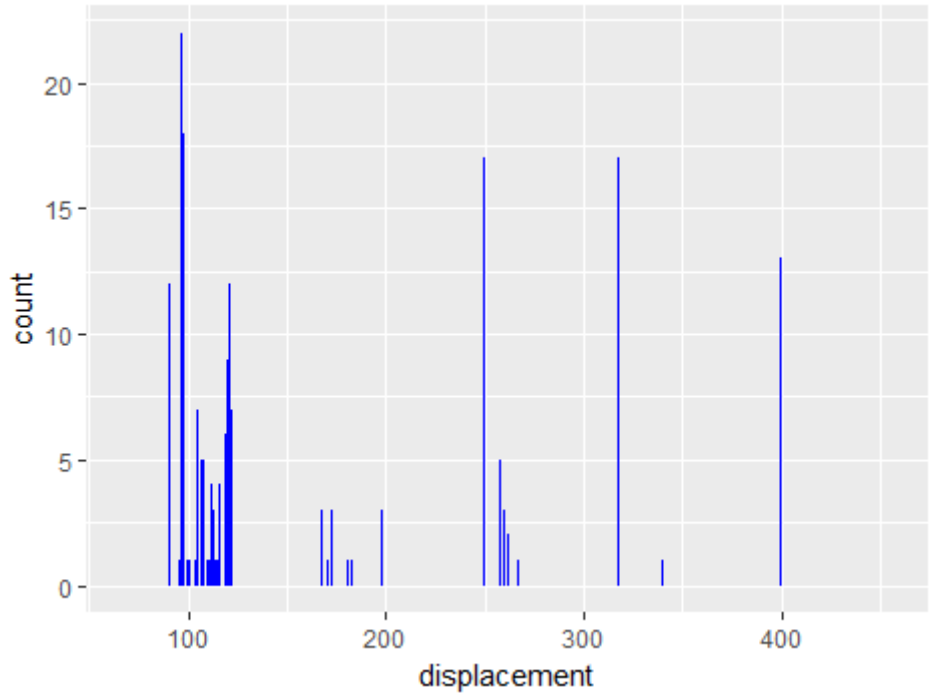
Here the graph shows that some of the cars have an mpg of 0 which is not possible in real world. We can safely say that this data was entered by mistake.

```
ggplot(cardata, aes(cylinders))+geom_bar(fill="navy")
```

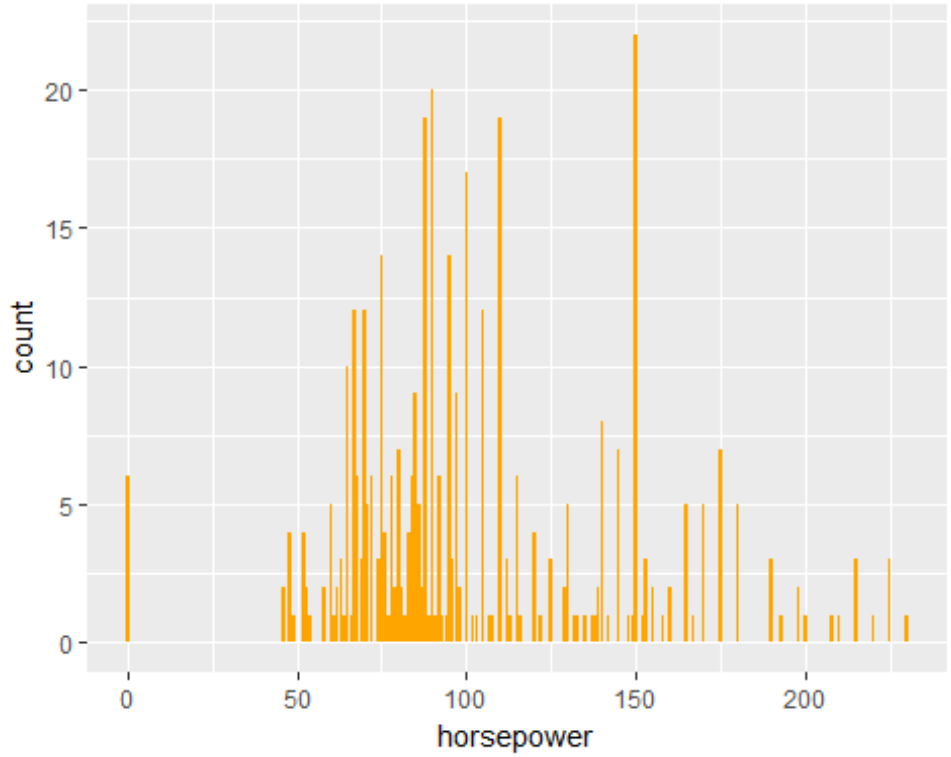


This looks fine. The range of cylinders in cars in our dataset is from 3 to 8.

```
ggplot(cardata, aes(displacement))+geom_bar(fill="blue")
```

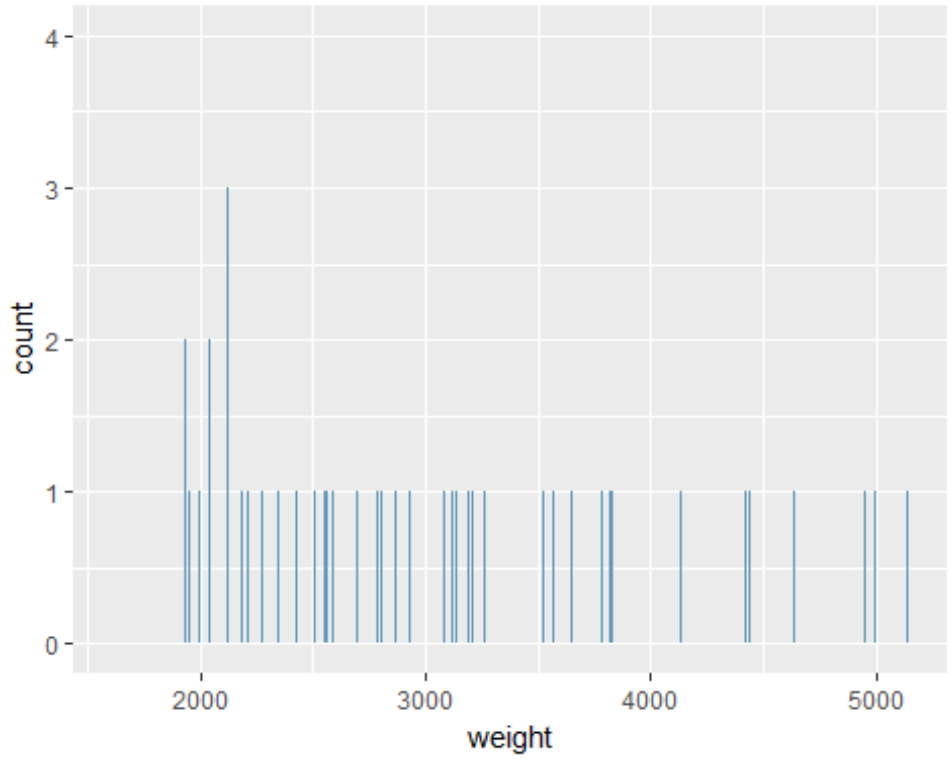



```
ggplot(cardata, aes(horsepower))+geom_bar(fill="orange")
```

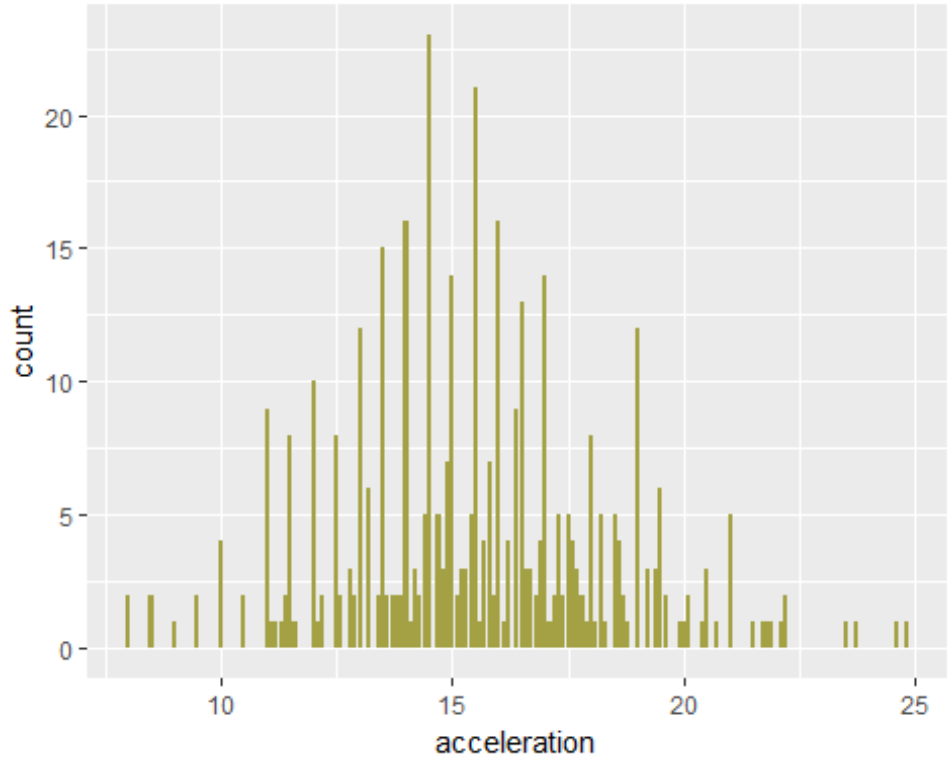


There are some cars which have zero horsepower as per the graph. We know that is not possible, therefore we will remove these values.

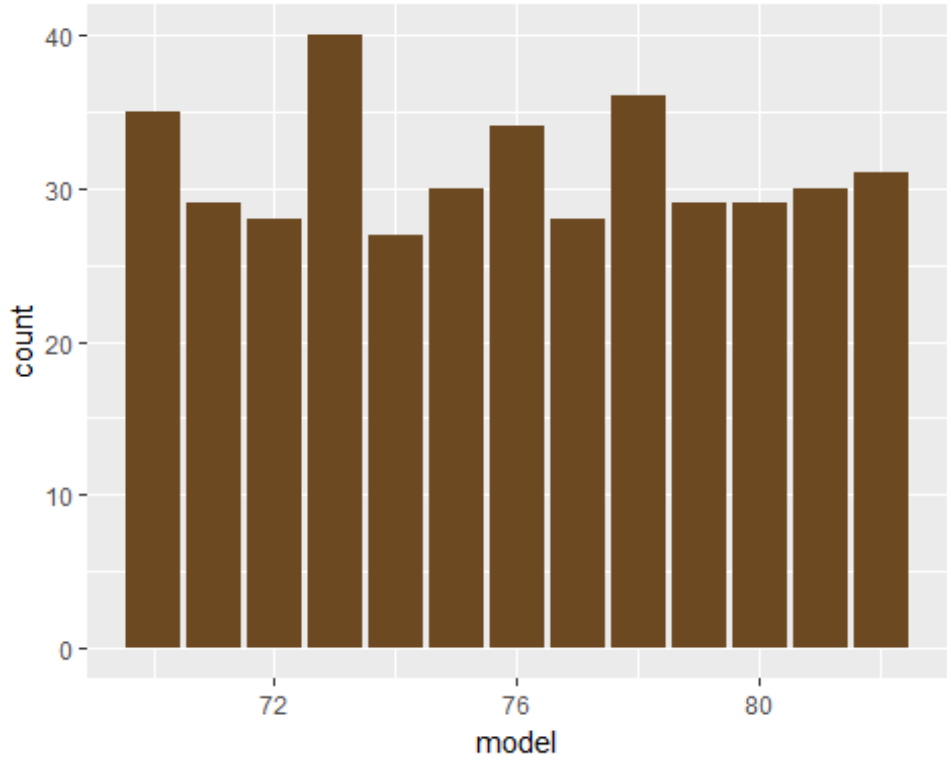
```
ggplot(cardata, aes(weight))+geom_bar(fill="#6299BA")
```



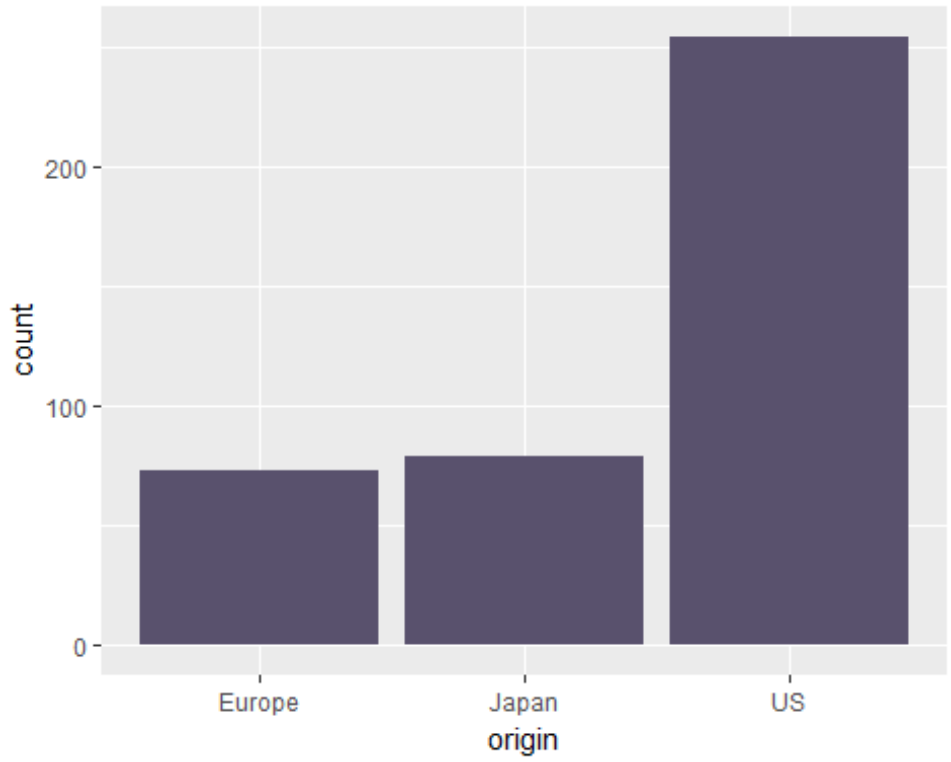

```
ggplot(cardata, aes(acceleration))+geom_bar(fill="#A3A144")
```




```
ggplot(cardata, aes(model))+geom_bar(fill="#6C4921")
```

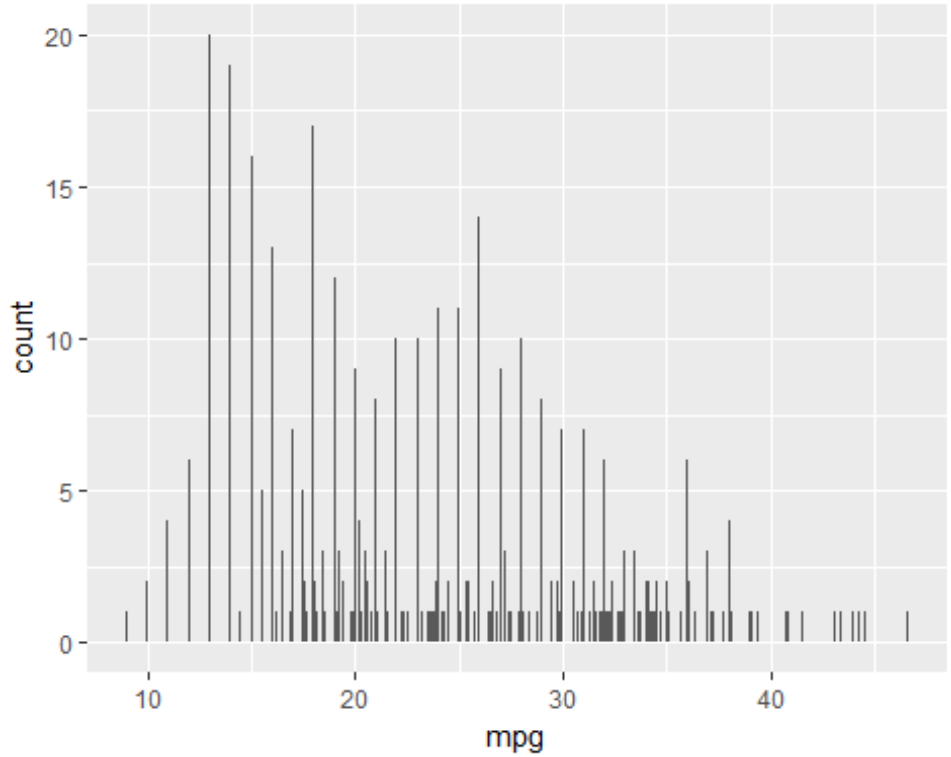



```
ggplot(cardata, aes(origin))+geom_bar(fill="#59516D")
```

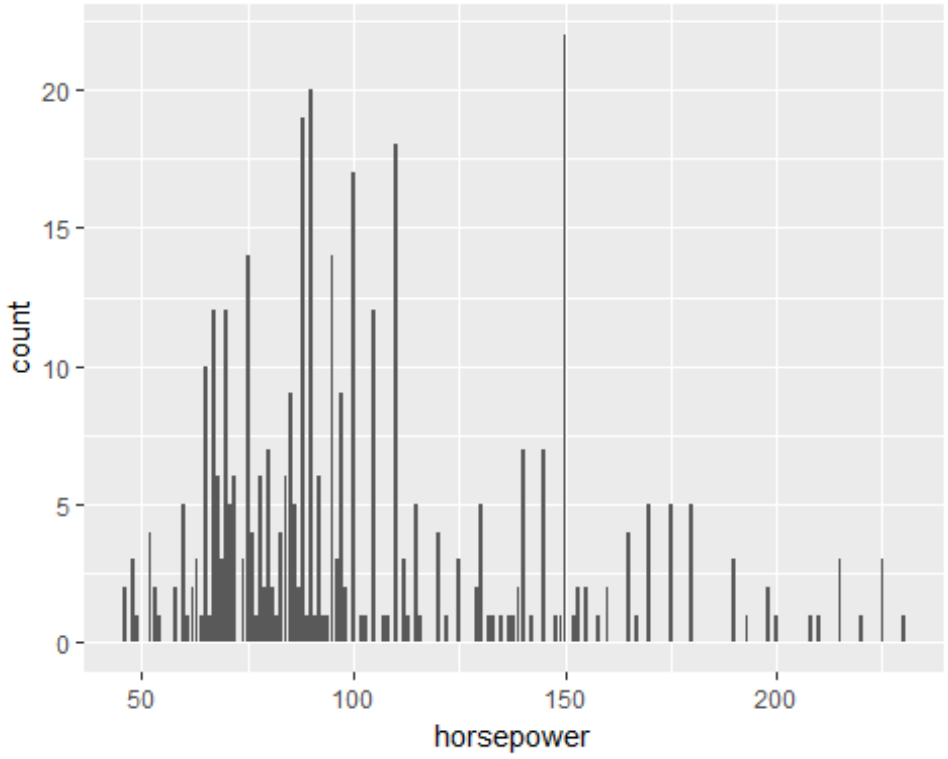


Now we will remove all unwanted values from our dataset using the following code:

```
remove<-which(cardata$mpg==0) # storing tibble where mpg=0.
cardata<-cardata[-remove,] # Removing rows from dataset

ggplot(cardata, aes(mpg))+geom_bar() # Creating barchart to confirm that t
here are no more outliers
```




```
remove2<-which(cardata$horsepower==0) # Storing tibble where horsepower=0.

cardata<-cardata[-remove2,]

ggplot(cardata, aes(horsepower))+geom_bar()
```



4. Analyzing Dataset:

```
summary(cardata) #Overview of cleaned dataset
```

```
##      car                 mpg          cylinders      displacement
##  Length:392         Min.   : 9.00   Min.   :3.000   Min.   : 68.0
##  Class :character   1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0
##  Mode  :character   Median :22.75   Median :4.000   Median :151.0
##                     Mean   :23.45   Mean   :5.472   Mean   :194.4
##                     3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8
##                     Max.   :46.60   Max.   :8.000   Max.   :455.0
##    horsepower        weight      acceleration       model
##  Min.   : 46.0   Min.   :1613   Min.   : 8.00   Min.   :70.00
##  1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00
##  Median : 93.5   Median :2804   Median :15.50   Median :76.00
##  Mean   :104.5   Mean   :2978   Mean   :15.54   Mean   :75.98
##  3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00
##  Max.   :230.0   Max.   :5140   Max.   :24.80   Max.   :82.00
##     origin
##  Length:392
##  Class :character
##  Mode  :character
##
##
##
```

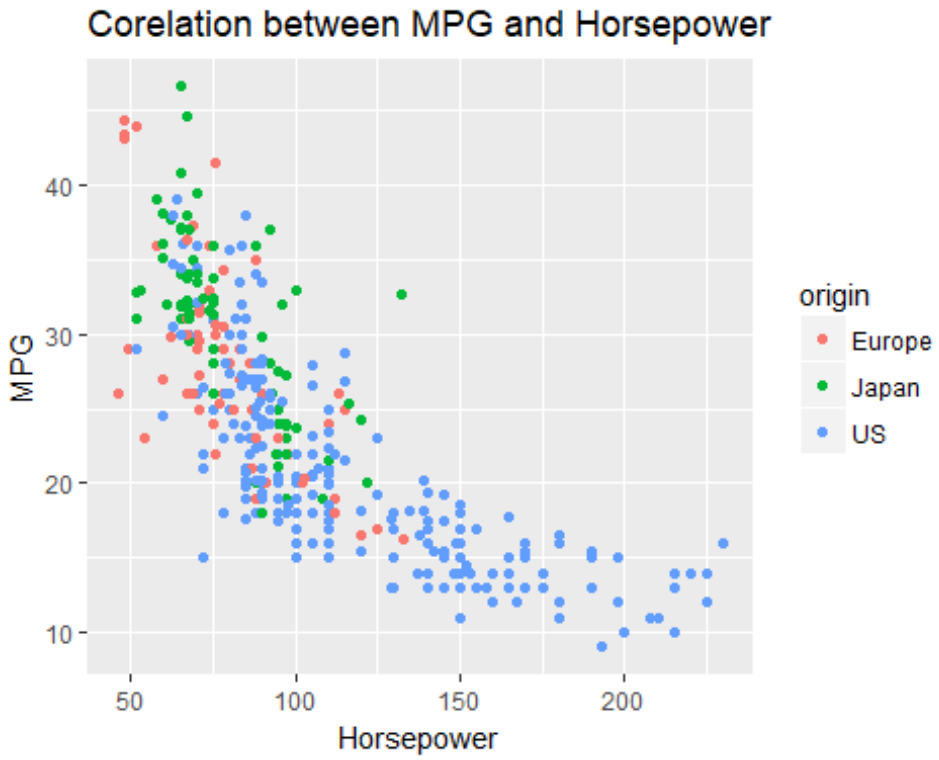
Using the above descriptive statistics, we can extract vital information from our dataset such as:

- The mean car mileage is 23.48 miles per gallon. The most fuel-efficient car has a mileage of 46.6 mpg while the least fuel-efficient car has mileage of 9 mpg.
- The minimum cylinders a car have is 3 while the maximum cylinders is 8.
- The mean engine displacement is 194.4 cubic inches. The smallest engine in our dataset is 68 cubic inches while the biggest is 455.
- The heaviest car is 5,140 lbs. while the lightest car is 1,613 lbs., and the mean value of weight is 2,978 lbs.
- The most powerful car in our dataset has 230 hp while the least powerful car has 46 hp. Average value of horsepower is 104.5 hp.
- The fastest car has an acceleration of 8 seconds while the slowest car has acceleration of 24.8 seconds for going from the speed of zero to 60. The average value is 15.50 seconds.
- The oldest car is from 1970 while the latest car is from 1980.

```
cor(cardata$horsepower, cardata$mpg) # Finding relation between horsepower
and mpg variable.
```

```
## [1] -0.7784268
```

```
ggplot(cardata, aes(horsepower, mpg ))+geom_point(aes(color=origin))+
  labs(x="Horsepower", y="MPG", title="Corelation between MPG and Horsepow
er") # Creating a scatterplot to visulize the correlation.
```

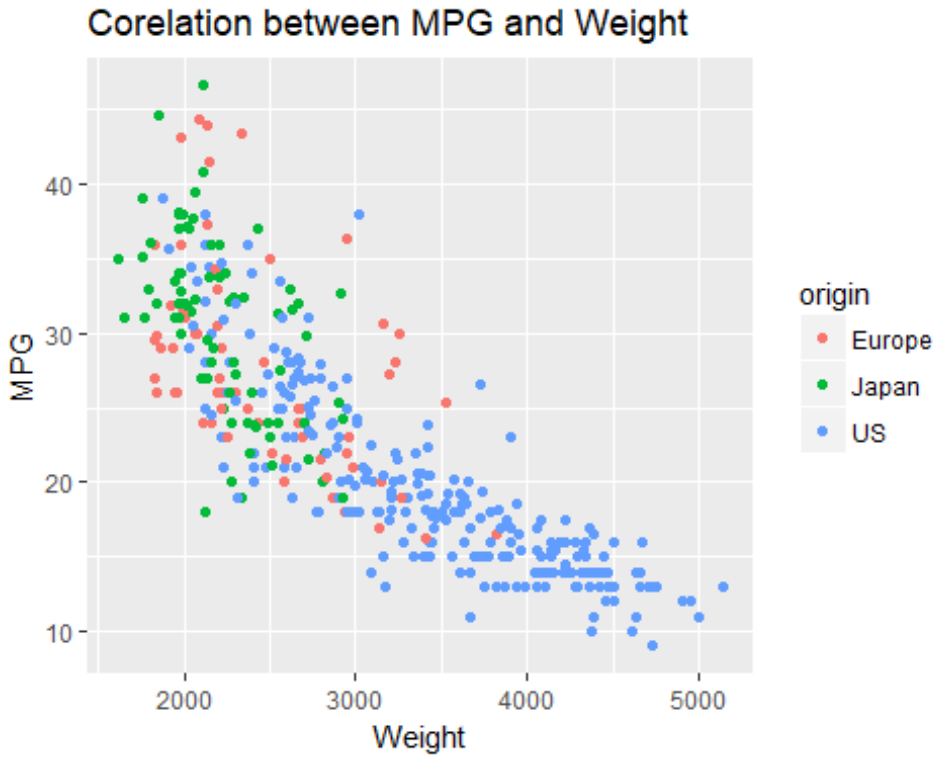


The correlation between mileage and horsepower is negative. This means that cars that have higher weight tend to have lower mpg. From the graph, we can see that correlation is very high.

```
cor(cardata$weight, cardata$mpg)
```

```
## [1] -0.8322442
```

```
ggplot(cardata, aes(weight, mpg ))+geom_point(aes(color=origin))+
  labs(x="Weight", y="MPG", title="Corelation between MPG and Weight")
```

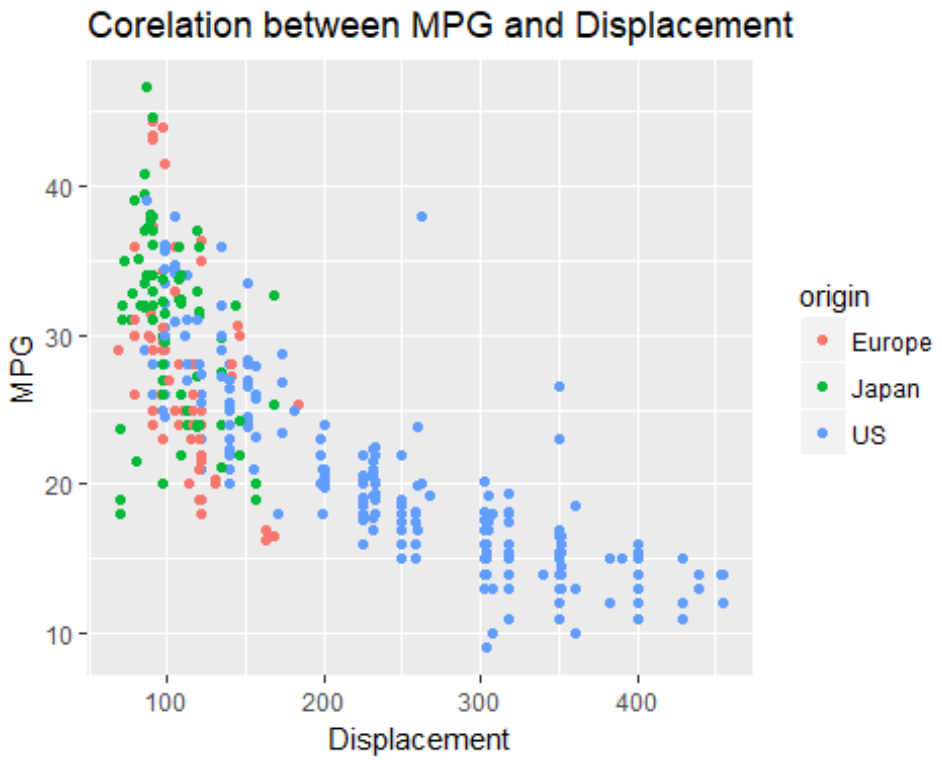


The correlation between mileage and weight is negative. This means that as weight of the car increases, its mileage decreases. From the graph, we can see that correlation is very high.

```
cor(cardata$displacement, cardata$mpg)
```

```
## [1] -0.8051269
```

```
ggplot(cardata, aes(displacement, mpg ))+geom_point(aes(color=origin))+
  labs(x="Displacement", y="MPG", title="Corelation between MPG and Displa
cement")
```

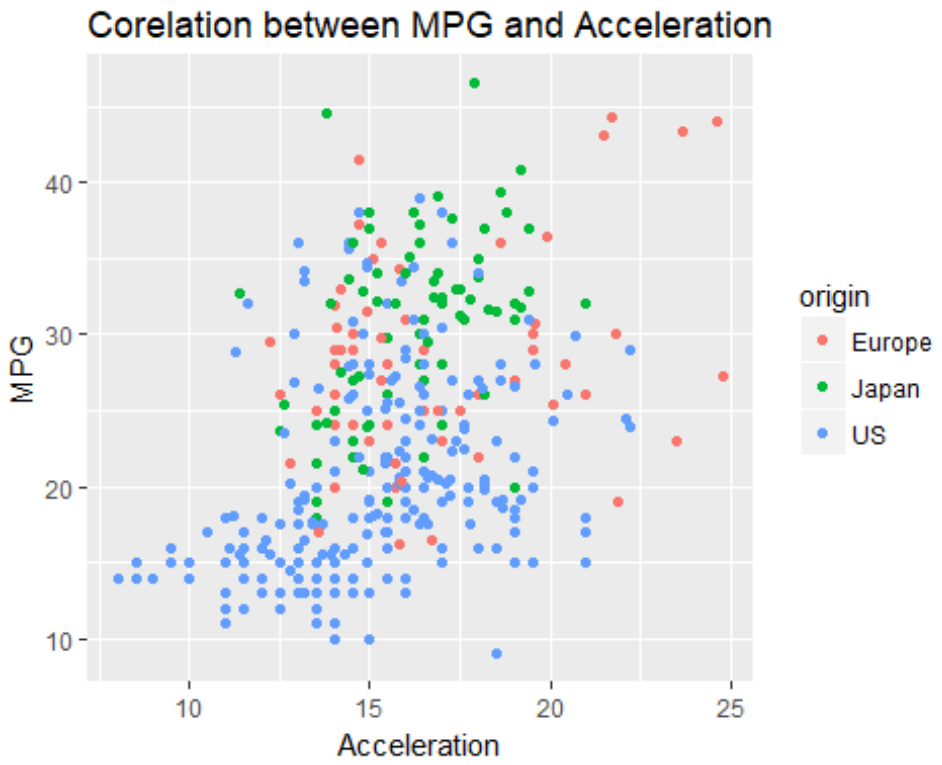


The correlation is highly negative which means that as displacement i.e. engine size increases, the mileage of the car decreases.

```
cor(cardata$acceleration, cardata$mpg)
```

```
## [1] 0.4233285
```

```
ggplot(cardata, aes(acceleration, mpg ))+geom_point(aes(color=origin))+
  labs(x="Acceleration", y="MPG", title="Corelation between MPG and Accele
ration")
```



Acceleration and mileage are not highly correlated. The data shows that higher the acceleration time (means smaller engine size), the higher the mileage. The overall correlation, however, is positive.

5. Analysis cars on various other parameters:

Now we will have a look at our dataset and try to answer some of questions that a business or buyer might be interested in. We will try to find out which cars are fastest, most fuel efficient, or most powerful in our dataset.

```
# Analysing cars on various parameters.

arrange(cardata, desc(cardata$mpg))%>%
  select(car, mpg)%>%
  head(5) # most fuel efficient cars in our dataset
```

```
## # A tibble: 5 x 2
##   car                           mpg
##   <chr>                       <dbl>
## 1 Mazda GLC                    46.6
## 2 Honda Civic 1500 gl          44.6
## 3 Volkswagen Rabbit C (Diesel)  44.3
## 4 Volkswagen Pickup            44.0
## 5 Volkswagen Dasher (diesel)   43.4
```

```
arrange(cardata, desc(cardata$mpg))%>%
    select(car, mpg)%>%
      tail(5) # least fuel efficient cars in our dataset
```

```
## # A tibble: 5 x 2
##   car                mpg
##   <chr>            <dbl>
```

```
## 1 Chevrolet Impala 11.0
## 2 Oldsmobile Omega 11.0
## 3 Ford F250        10.0
## 4 Chevy C20        10.0
## 5 Hi 1200D          9.00
```

```
arrange(cardata, desc(cardata$horsepower))%>%
  select(car, horsepower)%>%
  head(5) # most powerful in our dataset
```

```
## # A tibble: 5 x 2
##   car                      horsepower
##   <chr>                         <dbl>
## 1 Pontiac Grand Prix              230
## 2 Pontiac Catalina                225
## 3 Buick Estate Wagon (sw)         225
## 4 Buick Electra 225 Custom        225
## 5 Chevrolet Impala                220
```

```
arrange(cardata, desc(cardata$horsepower))%>%
  select(car, horsepower)%>%
  tail(5) # least powerful cars in our dataset
```

```
## # A tibble: 5 x 2
##   car                              horsepower
##   <chr>                                 <dbl>
## 1 Volkswagen Rabbit Custom Diesel        48.0
## 2 Volkswagen Rabbit C (Diesel)           48.0
## 3 Volkswagen Dasher (diesel)             48.0
## 4 Volkswagen 1131 Deluxe Sedan           46.0
## 5 Volkswagen Super Beetle                46.0
```

```
arrange(cardata, desc(cardata$mpg), desc(horsepower))%>%
  select(car, horsepower, mpg)%>%
  head(5) # selecting most powerful car that gives best mpg
```

```
## # A tibble: 5 x 3
##   car                           horsepower   mpg
##   <chr>                              <dbl> <dbl>
## 1 Mazda GLC                           65.0  46.6
## 2 Honda Civic 1500 gl                 67.0  44.6
## 3 Volkswagen Rabbit C (Diesel)        48.0  44.3
## 4 Volkswagen Pickup                   52.0  44.0
## 5 Volkswagen Dasher (diesel)          48.0  43.4
```

6. Source of Dataset:

The Cars dataset has been obtained from the following url:
https://perso.telecom-paristech.fr/eagan/class/igr204/datasets