AS474_Automobile_Analysis

Group06

2023-07-21

```r
library(tidyverse)
```

Import the libraries

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dplyr)
library(moments)
library(repr)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(dplyr)
library(purrr)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
colNames <- c("symboling",
               "normalized_losses",
               "make",
               "fuel_type",
               "aspiration",
               "num_of_doors",
              "body_style",
               "drive_wheels",
               "engine_location",
               "wheel_base",
               "length",
               "width",
               "height",
               "curb_weight",
               "engine_type",
               "num_of_cylinders",
               "engine_size",
               "fuel_system",
              "bore",
               "stroke",
               "compression_ratio",
               "horsepower",
               "peak_rpm",
               "city_mpg",
               "highway_mpg",
               "price")



autoMobile <- read.csv("../Data/Automobile_data.csv", header = TRUE, col.names = colNames)

attach(autoMobile)
```

Loading the data set

```r
head(autoMobile)
```

Displaying the first 6 rows of data

```
##   symboling normalized_losses        make fuel_type aspiration num_of_doors
## 1         3                 ? alfa-romero       gas        std          two
## 2         3                 ? alfa-romero       gas        std          two
## 3         1                 ? alfa-romero       gas        std          two
## 4         2               164        audi       gas        std         four
## 5         2               164        audi       gas        std         four
## 6         2                 ?        audi       gas        std          two
##    body_style drive_wheels engine_location wheel_base length width height
## 1 convertible          rwd           front       88.6  168.8  64.1   48.8
## 2 convertible          rwd           front       88.6  168.8  64.1   48.8
## 3   hatchback          rwd           front       94.5  171.2  65.5   52.4
## 4       sedan          fwd           front       99.8  176.6  66.2   54.3
## 5       sedan          4wd           front       99.4  176.6  66.4   54.3
## 6       sedan          fwd           front       99.8  177.3  66.3   53.1
##   curb_weight engine_type num_of_cylinders engine_size fuel_system bore stroke
## 1        2548        dohc             four         130        mpfi 3.47   2.68
## 2        2548        dohc             four         130        mpfi 3.47   2.68
## 3        2823        ohcv              six         152        mpfi 2.68   3.47
## 4        2337         ohc             four         109        mpfi 3.19    3.4
## 5        2824         ohc             five         136        mpfi 3.19    3.4
## 6        2507         ohc             five         136        mpfi 3.19    3.4
##   compression_ratio horsepower peak_rpm city_mpg highway_mpg price
## 1               9.0        111     5000       21          27 13495
## 2               9.0        111     5000       21          27 16500
## 3               9.0        154     5000       19          26 16500
## 4              10.0        102     5500       24          30 13950
## 5               8.0        115     5500       18          22 17450
## 6               8.5        110     5500       19          25 15250
```

Steps for working with missing data:

1. Identify missing data
2. Deal with missing data
3. Correct data format

Identify Missing Value

- Convert “?” to NA In the data set missing data comes with the question mark “?”. We replace it with NA.

```
autoMobile[autoMobile == '?'] <- NA

head(autoMobile)
```

```
##   symboling normalized_losses        make fuel_type aspiration num_of_doors
## 1         3              <NA> alfa-romero       gas        std          two
## 2         3              <NA> alfa-romero       gas        std          two
## 3         1              <NA> alfa-romero       gas        std          two
## 4         2               164        audi       gas        std         four
## 5         2               164        audi       gas        std         four
## 6         2              <NA>        audi       gas        std          two
##    body_style drive_wheels engine_location wheel_base length width height
## 1 convertible          rwd           front       88.6  168.8  64.1   48.8
## 2 convertible          rwd           front       88.6  168.8  64.1   48.8
## 3   hatchback          rwd           front       94.5  171.2  65.5   52.4
## 4       sedan          fwd           front       99.8  176.6  66.2   54.3
## 5       sedan          4wd           front       99.4  176.6  66.4   54.3
## 6       sedan          fwd           front       99.8  177.3  66.3   53.1
##   curb_weight engine_type num_of_cylinders engine_size fuel_system bore stroke
## 1        2548        dohc             four         130        mpfi 3.47   2.68
## 2        2548        dohc             four         130        mpfi 3.47   2.68
## 3        2823        ohcv              six         152        mpfi 2.68   3.47
## 4        2337         ohc             four         109        mpfi 3.19    3.4
## 5        2824         ohc             five         136        mpfi 3.19    3.4
## 6        2507         ohc             five         136        mpfi 3.19    3.4
##   compression_ratio horsepower peak_rpm city_mpg highway_mpg price
## 1               9.0        111     5000       21          27 13495
## 2               9.0        111     5000       21          27 16500
## 3               9.0        154     5000       19          26 16500
## 4              10.0        102     5500       24          30 13950
## 5               8.0        115     5500       18          22 17450
## 6               8.5        110     5500       19          25 15250
```

```r
glimpse(autoMobile)
```

Getting a description about the dataset

```
## Rows: 205
## Columns: 26
## $ symboling         <int> 3, 3, 1, 2, 2, 2, 1, 1, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0~
## $ normalized_losses <chr> NA, NA, NA, "164", "164", NA, "158", NA, "158", NA, ~
## $ make              <chr> "alfa-romero", "alfa-romero", "alfa-romero", "audi",~
## $ fuel_type         <chr> "gas", "gas", "gas", "gas", "gas", "gas", "gas", "ga~
## $ aspiration        <chr> "std", "std", "std", "std", "std", "std", "std", "st~
## $ num_of_doors      <chr> "two", "two", "two", "four", "four", "two", "four", ~
## $ body_style        <chr> "convertible", "convertible", "hatchback", "sedan", ~
## $ drive_wheels      <chr> "rwd", "rwd", "rwd", "fwd", "4wd", "fwd", "fwd", "fw~
## $ engine_location   <chr> "front", "front", "front", "front", "front", "front"~
## $ wheel_base        <dbl> 88.6, 88.6, 94.5, 99.8, 99.4, 99.8, 105.8, 105.8, 10~
## $ length            <dbl> 168.8, 168.8, 171.2, 176.6, 176.6, 177.3, 192.7, 192~
## $ width             <dbl> 64.1, 64.1, 65.5, 66.2, 66.4, 66.3, 71.4, 71.4, 71.4~
## $ height            <dbl> 48.8, 48.8, 52.4, 54.3, 54.3, 53.1, 55.7, 55.7, 55.9~
## $ curb_weight       <int> 2548, 2548, 2823, 2337, 2824, 2507, 2844, 2954, 3086~
## $ engine_type       <chr> "dohc", "dohc", "ohcv", "ohc", "ohc", "ohc", "ohc", ~
## $ num_of_cylinders  <chr> "four", "four", "six", "four", "five", "five", "five~
## $ engine_size       <int> 130, 130, 152, 109, 136, 136, 136, 136, 131, 131, 10~
## $ fuel_system       <chr> "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpf~
## $ bore              <chr> "3.47", "3.47", "2.68", "3.19", "3.19", "3.19", "3.1~
## $ stroke            <chr> "2.68", "2.68", "3.47", "3.4", "3.4", "3.4", "3.4", ~
## $ compression_ratio <dbl> 9.00, 9.00, 9.00, 10.00, 8.00, 8.50, 8.50, 8.50, 8.3~
## $ horsepower        <chr> "111", "111", "154", "102", "115", "110", "110", "11~
## $ peak_rpm          <chr> "5000", "5000", "5000", "5500", "5500", "5500", "550~
## $ city_mpg          <int> 21, 21, 19, 24, 18, 19, 19, 19, 17, 16, 23, 23, 21, ~
## $ highway_mpg       <int> 27, 27, 26, 30, 22, 25, 25, 25, 20, 22, 29, 29, 28, ~
## $ price             <chr> "13495", "16500", "16500", "13950", "17450", "15250"~
```

```r
sum(is.na(autoMobile))
```

```
## [1] 59
```

```r
NAsByFeature <- apply(autoMobile, 2,
                      function(x){
                        length(which(is.na(x)))
                        }
                      )

NAsByFeature
```

Check the missing values in each column

```
##         symboling normalized_losses              make         fuel_type
##                 0                41                 0                 0
##        aspiration      num_of_doors        body_style      drive_wheels
##                 0                 2                 0                 0
##   engine_location        wheel_base            length             width
##                 0                 0                 0                 0
##            height       curb_weight       engine_type  num_of_cylinders
##                 0                 0                 0                 0
##       engine_size       fuel_system              bore            stroke
##                 0                 0                 4                 4
## compression_ratio        horsepower          peak_rpm          city_mpg
##                 0                 2                 2                 0
##       highway_mpg             price
##                 0                 4
```

Each column has 205 rows of data and 7 columns containing missing data:

1. normalized_losses: 41 NA
2. num_of_doors: 2 NA
3. bore: 4 NA
4. stroke: 4 NA
5. horsepower: 2 NA
6. peak_rpm: 2 NA
7. price: 4 NA

Deal with missing data

1. **Drop data**

   a.drop the whole row

   b.drop the whole column

2. **Replace data**

   a.replace it by mean

   b.replace it by frequency

   c.replace it based on other functions

- Whole columns should be dropped only if most entries in the column are empty. In our dataset, none of the columns are empty enough to drop entirely.
- “normalized-losses”: 41 missing data, replace them with mean.
- For other missing values we remove the rows which contain missing values

View the structure of the data set

```
# Calculate the average of normalized-losses
avg_norm_loss <- mean(as.numeric(autoMobile[["normalized_losses"]]), na.rm = TRUE)

# Print the average of normalized-losses
cat("Average of normalized-losses:", avg_norm_loss, "\n")
```

```
## Average of normalized-losses: 122
```

```
# Replace missing values in normalized-losses with the average
autoMobile[["normalized_losses"]][is.na(autoMobile[["normalized_losses"]])] <- avg_norm_loss

head(autoMobile,10)
```

```
##    symboling normalized_losses        make fuel_type aspiration num_of_doors
## 1          3               122 alfa-romero       gas        std          two
## 2          3               122 alfa-romero       gas        std          two
## 3          1               122 alfa-romero       gas        std          two
## 4          2               164        audi       gas        std         four
## 5          2               164        audi       gas        std         four
## 6          2               122        audi       gas        std          two
## 7          1               158        audi       gas        std         four
## 8          1               122        audi       gas        std         four
## 9          1               158        audi       gas      turbo         four
## 10         0               122        audi       gas      turbo          two
##     body_style drive_wheels engine_location wheel_base length width height
## 1  convertible          rwd           front       88.6  168.8  64.1   48.8
## 2  convertible          rwd           front       88.6  168.8  64.1   48.8
## 3    hatchback          rwd           front       94.5  171.2  65.5   52.4
## 4        sedan          fwd           front       99.8  176.6  66.2   54.3
## 5        sedan          4wd           front       99.4  176.6  66.4   54.3
## 6        sedan          fwd           front       99.8  177.3  66.3   53.1
## 7        sedan          fwd           front      105.8  192.7  71.4   55.7
## 8        wagon          fwd           front      105.8  192.7  71.4   55.7
## 9        sedan          fwd           front      105.8  192.7  71.4   55.9
## 10   hatchback          4wd           front       99.5  178.2  67.9   52.0
##    curb_weight engine_type num_of_cylinders engine_size fuel_system bore stroke
## 1         2548        dohc             four         130        mpfi 3.47   2.68
## 2         2548        dohc             four         130        mpfi 3.47   2.68
## 3         2823        ohcv              six         152        mpfi 2.68   3.47
## 4         2337         ohc             four         109        mpfi 3.19    3.4
## 5         2824         ohc             five         136        mpfi 3.19    3.4
## 6         2507         ohc             five         136        mpfi 3.19    3.4
## 7         2844         ohc             five         136        mpfi 3.19    3.4
## 8         2954         ohc             five         136        mpfi 3.19    3.4
## 9         3086         ohc             five         131         mpfi 3.13    3.4
## 10        3053         ohc             five         131         mpfi 3.13    3.4
##    compression_ratio horsepower peak_rpm city_mpg highway_mpg price
## 1                9.0        111     5000       21          27 13495
## 2                9.0        111     5000       21          27 16500
## 3                9.0        154     5000       19          26 16500
## 4               10.0        102     5500       24          30 13950
## 5                8.0        115     5500       18          22 17450
## 6                8.5        110     5500       19          25 15250
## 7                8.5        110     5500       19          25 17710
## 8                8.5        110     5500       19          25 18920
## 9                8.3        140     5500       17          20 23875
## 10               7.0        160     5500       16          22  <NA>
```

```
sum(duplicated(autoMobile))
```

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

```
glimpse(autoMobile)
```

```
## Rows: 205
## Columns: 26
## $ symboling         <int> 3, 3, 1, 2, 2, 2, 1, 1, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0~
## $ normalized_losses <chr> "122", "122", "122", "164", "164", "122", "158", "12~
## $ make              <chr> "alfa-romero", "alfa-romero", "alfa-romero", "audi",~
## $ fuel_type         <chr> "gas", "gas", "gas", "gas", "gas", "gas", "gas", "ga~
## $ aspiration        <chr> "std", "std", "std", "std", "std", "std", "std", "st~
## $ num_of_doors      <chr> "two", "two", "two", "four", "four", "two", "four", ~
## $ body_style        <chr> "convertible", "convertible", "hatchback", "sedan", ~
## $ drive_wheels      <chr> "rwd", "rwd", "rwd", "fwd", "4wd", "fwd", "fwd", "fw~
## $ engine_location   <chr> "front", "front", "front", "front", "front", "front"~
## $ wheel_base        <dbl> 88.6, 88.6, 94.5, 99.8, 99.4, 99.8, 105.8, 105.8, 10~
## $ length            <dbl> 168.8, 168.8, 171.2, 176.6, 176.6, 177.3, 192.7, 192~
## $ width             <dbl> 64.1, 64.1, 65.5, 66.2, 66.4, 66.3, 71.4, 71.4, 71.4~
## $ height            <dbl> 48.8, 48.8, 52.4, 54.3, 54.3, 53.1, 55.7, 55.7, 55.9~
## $ curb_weight       <int> 2548, 2548, 2823, 2337, 2824, 2507, 2844, 2954, 3086~
## $ engine_type       <chr> "dohc", "dohc", "ohcv", "ohc", "ohc", "ohc", "ohc", ~
## $ num_of_cylinders  <chr> "four", "four", "six", "four", "five", "five", "five~
## $ engine_size       <int> 130, 130, 152, 109, 136, 136, 136, 136, 131, 131, 10~
## $ fuel_system       <chr> "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpf~
## $ bore              <chr> "3.47", "3.47", "2.68", "3.19", "3.19", "3.19", "3.1~
## $ stroke            <chr> "2.68", "2.68", "3.47", "3.4", "3.4", "3.4", "3.4", ~
## $ compression_ratio <dbl> 9.00, 9.00, 9.00, 10.00, 8.00, 8.50, 8.50, 8.50, 8.3~
## $ horsepower        <chr> "111", "111", "154", "102", "115", "110", "110", "11~
## $ peak_rpm          <chr> "5000", "5000", "5000", "5500", "5500", "5500", "550~
## $ city_mpg          <int> 21, 21, 19, 24, 18, 19, 19, 19, 17, 16, 23, 23, 21, ~
## $ highway_mpg       <int> 27, 27, 26, 30, 22, 25, 25, 25, 20, 22, 29, 29, 28, ~
## $ price             <chr> "13495", "16500", "16500", "13950", "17450", "15250"~
```

Remove the na vaulues

```r
autoMobile <- autoMobile %>%
  na.omit()

NAsByFeature <- sapply(autoMobile, function(x) sum(is.na(x)))

NAsByFeature
```

```
##         symboling normalized_losses              make         fuel_type
##                 0                 0                 0                 0
##        aspiration      num_of_doors        body_style      drive_wheels
##                 0                 0                 0                 0
##   engine_location        wheel_base            length             width
##                 0                 0                 0                 0
##            height       curb_weight       engine_type  num_of_cylinders
##                 0                 0                 0                 0
##       engine_size       fuel_system              bore            stroke
##                 0                 0                 0                 0
## compression_ratio        horsepower          peak_rpm          city_mpg
##                 0                 0                 0                 0
##       highway_mpg             price
##                 0                 0
```

- Now we can see that data set is cleaned from missing values.
- Now we should check the data types for each column.

```r
glimpse(autoMobile)
```

```
## Rows: 193
## Columns: 26
## $ symboling         <int> 3, 3, 1, 2, 2, 2, 1, 1, 1, 2, 0, 0, 0, 1, 0, 0, 0, 2~
## $ normalized_losses <chr> "122", "122", "122", "164", "164", "122", "158", "12~
## $ make              <chr> "alfa-romero", "alfa-romero", "alfa-romero", "audi",~
## $ fuel_type         <chr> "gas", "gas", "gas", "gas", "gas", "gas", "gas", "ga~
## $ aspiration        <chr> "std", "std", "std", "std", "std", "std", "std", "st~
## $ num_of_doors      <chr> "two", "two", "two", "four", "four", "two", "four", ~
## $ body_style        <chr> "convertible", "convertible", "hatchback", "sedan", ~
## $ drive_wheels      <chr> "rwd", "rwd", "rwd", "fwd", "4wd", "fwd", "fwd", "fw~
## $ engine_location   <chr> "front", "front", "front", "front", "front", "front"~
## $ wheel_base        <dbl> 88.6, 88.6, 94.5, 99.8, 99.4, 99.8, 105.8, 105.8, 10~
## $ length            <dbl> 168.8, 168.8, 171.2, 176.6, 176.6, 177.3, 192.7, 192~
## $ width             <dbl> 64.1, 64.1, 65.5, 66.2, 66.4, 66.3, 71.4, 71.4, 71.4~
## $ height            <dbl> 48.8, 48.8, 52.4, 54.3, 54.3, 53.1, 55.7, 55.7, 55.9~
## $ curb_weight       <int> 2548, 2548, 2823, 2337, 2824, 2507, 2844, 2954, 3086~
## $ engine_type       <chr> "dohc", "dohc", "ohcv", "ohc", "ohc", "ohc", "ohc", ~
## $ num_of_cylinders  <chr> "four", "four", "six", "four", "five", "five", "five~
## $ engine_size       <int> 130, 130, 152, 109, 136, 136, 136, 136, 131, 108, 10~
## $ fuel_system       <chr> "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpf~
## $ bore              <chr> "3.47", "3.47", "2.68", "3.19", "3.19", "3.19", "3.1~
## $ stroke            <chr> "2.68", "2.68", "3.47", "3.4", "3.4", "3.4", "3.4", ~
## $ compression_ratio <dbl> 9.00, 9.00, 9.00, 10.00, 8.00, 8.50, 8.50, 8.50, 8.3~
## $ horsepower        <chr> "111", "111", "154", "102", "115", "110", "110", "11~
## $ peak_rpm          <chr> "5000", "5000", "5000", "5500", "5500", "5500", "550~
## $ city_mpg          <int> 21, 21, 19, 24, 18, 19, 19, 19, 17, 23, 23, 21, 21, ~
## $ highway_mpg       <int> 27, 27, 26, 30, 22, 25, 25, 25, 20, 29, 29, 28, 28, ~
## $ price             <chr> "13495", "16500", "16500", "13950", "17450", "15250"~
```

- Change the variable types for specific columns Some columns are not of the correct data type. We have to convert data types into a proper format for each column.

```
factorCols = c('make',
               'fuel_type',
               'aspiration',
               'num_of_doors',
               'body_style',
               'drive_wheels',
               'engine_location',
               'engine_type',
               'num_of_cylinders',
               'fuel_system'
               )

intCols =c('horsepower',
           'symboling',
           'normalized_losses',
           'curb_weight',
           'engine_size',
           'city_mpg',
           'highway_mpg'
           )

numCols = c('bore',
            'stroke',
            'compression_ratio',
            'peak_rpm',
            'price',
            'wheel_base',
            'length',
            'width',
            'height')


autoMobile <- autoMobile %>%
  mutate_at(factorCols, as.factor) %>%
  mutate_at(intCols, as.integer) %>%
  mutate_at(numCols, as.numeric)
```

```
str(autoMobile)
```

```
## 'data.frame':    193 obs. of  26 variables:
##  $ symboling        : int  3 3 1 2 2 2 1 1 1 2 ...
##  $ normalized_losses: int  122 122 122 164 164 122 158 122 158 192 ...
##  $ make             : Factor w/ 21 levels "alfa-romero",..: 1 1 1 2 2 2 2 2 2 3 ...
##  $ fuel_type        : Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...
##  $ aspiration       : Factor w/ 2 levels "std","turbo": 1 1 1 1 1 1 1 1 2 1 ...
##  $ num_of_doors     : Factor w/ 2 levels "four","two": 2 2 2 1 1 2 1 1 1 2 ...
##  $ body_style       : Factor w/ 5 levels "convertible",..: 1 1 3 4 4 4 4 5 4 4 ...
##  $ drive_wheels     : Factor w/ 3 levels "4wd","fwd","rwd": 3 3 3 2 1 2 2 2 2 3 ...
##  $ engine_location  : Factor w/ 2 levels "front","rear": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wheel_base       : num  88.6 88.6 94.5 99.8 99.4 ...
##  $ length           : num  169 169 171 177 177 ...
##  $ width            : num  64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 64.8 ...
##  $ height           : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 54.3 ...
##  $ curb_weight      : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 2395 ...
##  $ engine_type      : Factor w/ 5 levels "dohc","l","ohc",..: 1 1 5 3 3 3 3 3 3 3 ...
##  $ num_of_cylinders : Factor w/ 6 levels "eight","five",..: 3 3 4 3 2 2 2 2 2 3 ...
##  $ engine_size      : int  130 130 152 109 136 136 136 136 131 108 ...
##  $ fuel_system      : Factor w/ 7 levels "1bbl","2bbl",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ bore             : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.5 ...
##  $ stroke           : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 2.8 ...
##  $ compression_ratio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 8.8 ...
##  $ horsepower       : int  111 111 154 102 115 110 110 110 140 101 ...
##  $ peak_rpm         : num  5000 5000 5000 5500 5500 5500 5500 5500 5500 5800 ...
##  $ city_mpg         : int  21 21 19 24 18 19 19 19 17 23 ...
##  $ highway_mpg      : int  27 27 26 30 22 25 25 25 20 29 ...
##  $ price            : num  13495 16500 16500 13950 17450 ...
##  - attr(*, "na.action")= 'omit' Named int [1:12] 10 28 45 46 56 57 58 59 64 130 ...
##   ..- attr(*, "names")= chr [1:12] "10" "28" "45" "46" ...
```

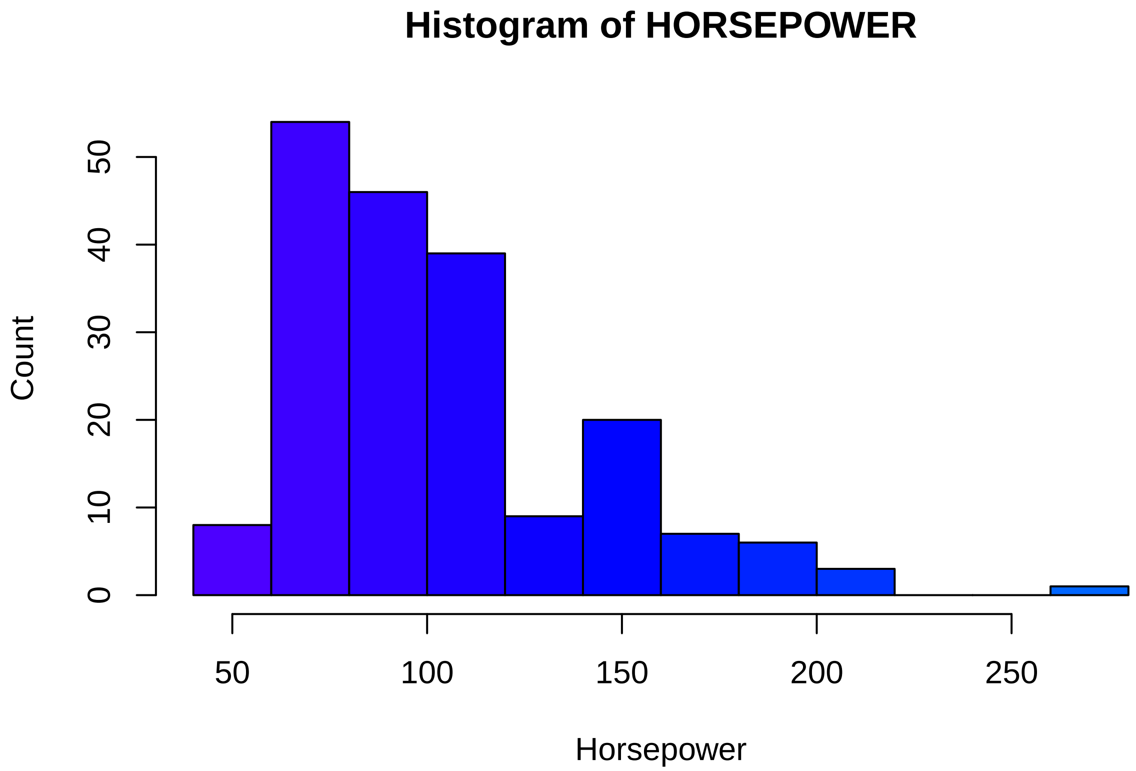
- Checking whether there is any duplicate value.

```
#dealing with the duplicate data
sum(duplicated(autoMobile))
```

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

- Finally the cleaned data set is obtained with no missing values and all data in its proper format.

```
horsepowerTable <- table(autoMobile[['horsepower']])


hist(autoMobile$horsepower, col = topo.colors(length(horsepowerTable)), border = "black",
     xlab = "Horsepower", ylab = "Count", main = "Histogram of HORSEPOWER")
```



Explantory Data Analysis

- When visualizing individual variables, it is important to first understand what type of variable you are dealing with.This helps to find the right visualization method for that variable.

```
str(autoMobile)
```

```
## 'data.frame':    193 obs. of  26 variables:
##  $ symboling        : int  3 3 1 2 2 2 1 1 1 2 ...
##  $ normalized_losses: int  122 122 122 164 164 122 158 122 158 192 ...
##  $ make             : Factor w/ 21 levels "alfa-romero",..: 1 1 1 2 2 2 2 2 2 3 ...
##  $ fuel_type        : Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...
##  $ aspiration       : Factor w/ 2 levels "std","turbo": 1 1 1 1 1 1 1 1 2 1 ...
##  $ num_of_doors     : Factor w/ 2 levels "four","two": 2 2 2 1 1 2 1 1 1 2 ...
##  $ body_style       : Factor w/ 5 levels "convertible",..: 1 1 3 4 4 4 4 5 4 4 ...
##  $ drive_wheels     : Factor w/ 3 levels "4wd","fwd","rwd": 3 3 3 2 1 2 2 2 2 3 ...
##  $ engine_location  : Factor w/ 2 levels "front","rear": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wheel_base       : num  88.6 88.6 94.5 99.8 99.4 ...
##  $ length           : num  169 169 171 177 177 ...
##  $ width            : num  64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 64.8 ...
##  $ height           : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 54.3 ...
##  $ curb_weight      : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 2395 ...
##  $ engine_type      : Factor w/ 5 levels "dohc","l","ohc",..: 1 1 5 3 3 3 3 3 3 3 ...
##  $ num_of_cylinders : Factor w/ 6 levels "eight","five",..: 3 3 4 3 2 2 2 2 2 3 ...
##  $ engine_size      : int  130 130 152 109 136 136 136 136 131 108 ...
##  $ fuel_system      : Factor w/ 7 levels "1bbl","2bbl",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ bore             : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.5 ...
##  $ stroke           : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 2.8 ...
##  $ compression_ratio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 8.8 ...
##  $ horsepower       : int  111 111 154 102 115 110 110 110 140 101 ...
##  $ peak_rpm         : num  5000 5000 5000 5500 5500 5500 5500 5500 5500 5800 ...
##  $ city_mpg         : int  21 21 19 24 18 19 19 19 17 23 ...
##  $ highway_mpg      : int  27 27 26 30 22 25 25 25 20 29 ...
##  $ price            : num  13495 16500 16500 13950 17450 ...
##  - attr(*, "na.action")= ’omit’ Named int [1:12] 10 28 45 46 56 57 58 59 64 130 ...
##   ..- attr(*, "names")= chr [1:12] "10" "28" "45" "46" ...
```

- summary of the dataset

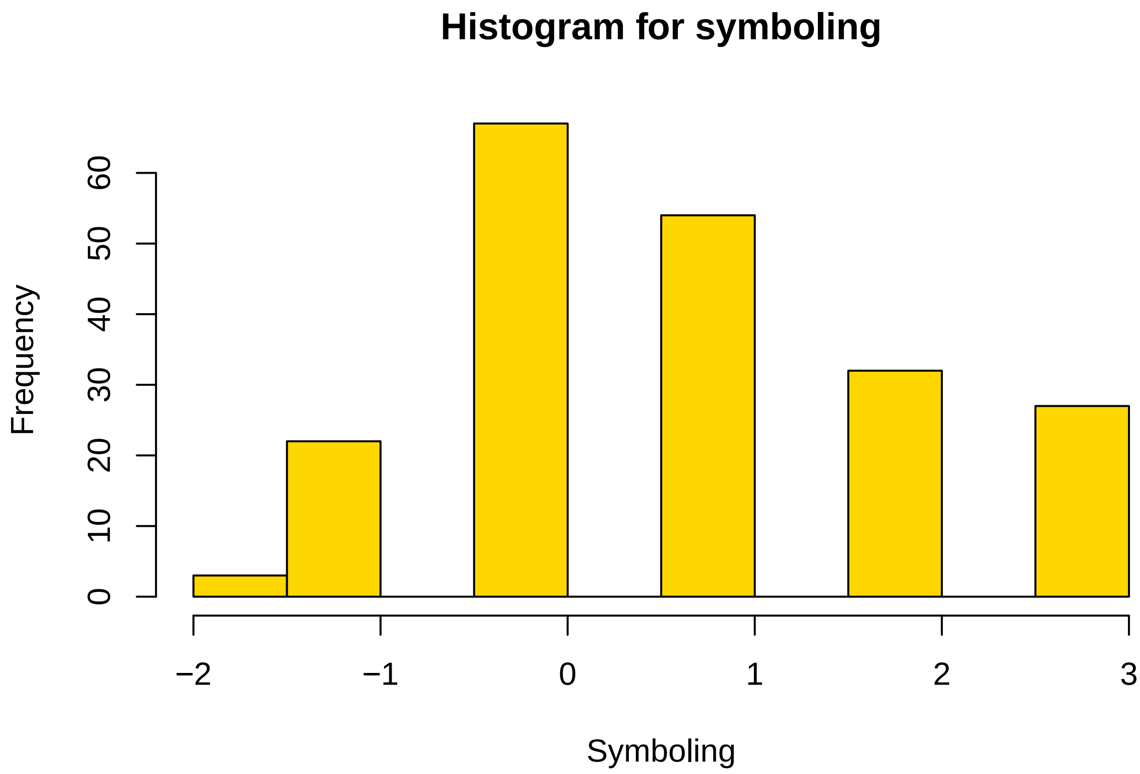
```
summary(autoMobile)
```

```
##    symboling       normalized_losses         make     fuel_type   aspiration
##  Min.   :-2.0000   Min.   : 65.0     toyota    :32   diesel: 19   std  :158
##  1st Qu.: 0.0000   1st Qu.: 95.0     nissan    :18   gas   :174   turbo: 35
##  Median : 1.0000   Median :122.0     honda     :13
##  Mean   : 0.7979   Mean   :121.3     mitsubishi:13
##  3rd Qu.: 2.0000   3rd Qu.:134.0     mazda     :12
##  Max.   : 3.0000   Max.   :256.0     subaru    :12
##                                      (Other)   :93
##  num_of_doors       body_style drive_wheels engine_location   wheel_base
##  four:112     convertible: 6   4wd:  8      front:190       Min.   : 86.60
##  two : 81     hardtop    : 8   fwd:114      rear :  3       1st Qu.: 94.50
##               hatchback  :63   rwd: 71                      Median : 97.00
##               sedan      :92                                Mean   : 98.92
##               wagon      :24                                3rd Qu.:102.40
##                                                             Max.   :120.90
##
##      length          width           height       curb_weight   engine_type
##  Min.   :141.1   Min.   :60.30   Min.   :47.80   Min.   :1488   dohc: 12
##  1st Qu.:166.3   1st Qu.:64.10   1st Qu.:52.00   1st Qu.:2145   l   : 12
##  Median :173.2   Median :65.40   Median :54.10   Median :2414   ohc :141
##  Mean   :174.3   Mean   :65.89   Mean   :53.87   Mean   :2562   ohcf: 15
##  3rd Qu.:184.6   3rd Qu.:66.90   3rd Qu.:55.70   3rd Qu.:2952   ohcv: 13
##  Max.   :208.1   Max.   :72.00   Max.   :59.80   Max.   :4066
##
##  num_of_cylinders  engine_size    fuel_system      bore           stroke
##  eight :  4       Min.   : 61.0   1bbl:11     Min.   :2.540   Min.   :2.070
##  five  : 10       1st Qu.: 98.0   2bbl:64     1st Qu.:3.150   1st Qu.:3.110
##  four  :153       Median :120.0   idi :19     Median :3.310   Median :3.290
##  six   : 24       Mean   :128.1   mfi : 1     Mean   :3.331   Mean   :3.249
##  three :  1       3rd Qu.:146.0   mpfi:88     3rd Qu.:3.590   3rd Qu.:3.410
##  twelve:  1       Max.   :326.0   spdi: 9     Max.   :3.940   Max.   :4.170
##                                   spfi: 1
##  compression_ratio   horsepower       peak_rpm       city_mpg
##  Min.   : 7.00     Min.   : 48.0   Min.   :4150   Min.   :13.00
##  1st Qu.: 8.50     1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00     Median : 95.0   Median :5100   Median :25.00
##  Mean   :10.14     Mean   :103.5   Mean   :5100   Mean   :25.33
##  3rd Qu.: 9.40     3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00     Max.   :262.0   Max.   :6600   Max.   :49.00
##
##   highway_mpg       price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7738
##  Median :30.00   Median :10245
##  Mean   :30.79   Mean   :13285
##  3rd Qu.:34.00   3rd Qu.:16515
##  Max.   :54.00   Max.   :45400
##
```

Continuous Numerical Variables

- Continuous numerical variables are variables that may contain any value within some range. Continuous numerical variables can have the type "num".
- In order to start understanding the (linear) relationship between an individual variable and the price. This can be done by using the scatterplot plus the fitted regression line for the data.

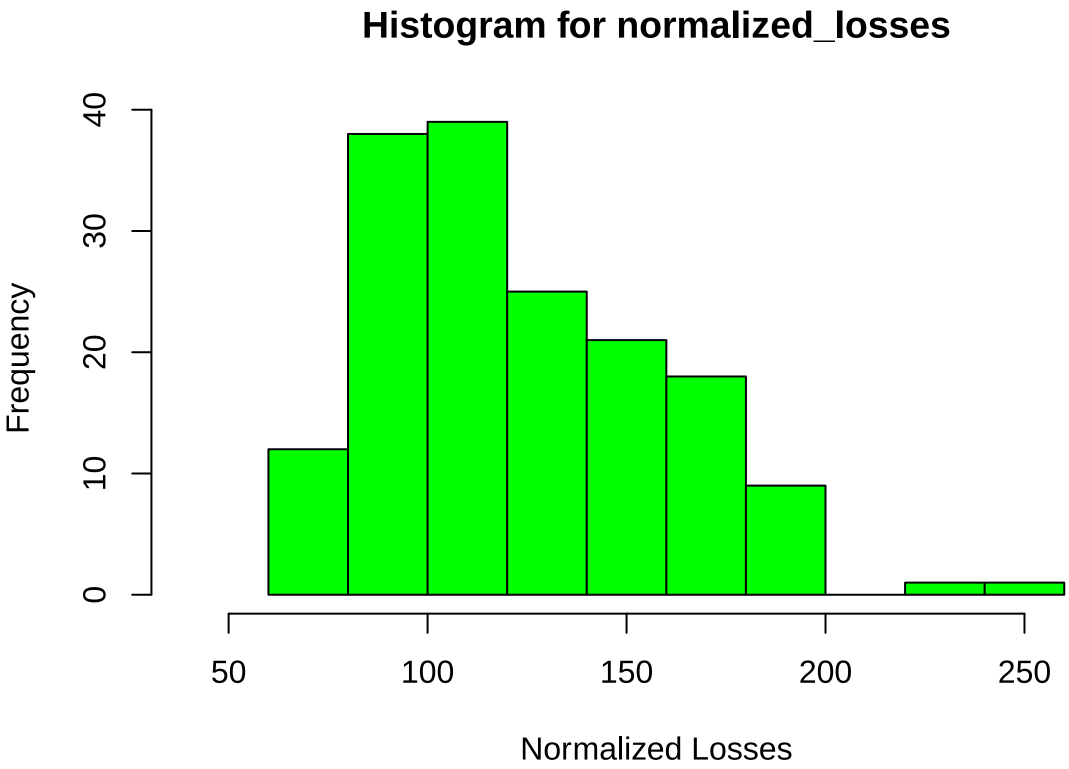
```r
hist(symboling,main= "Histogram for symboling",xlab= "Symboling",ylab= "Frequency", col= "gold",border=
```



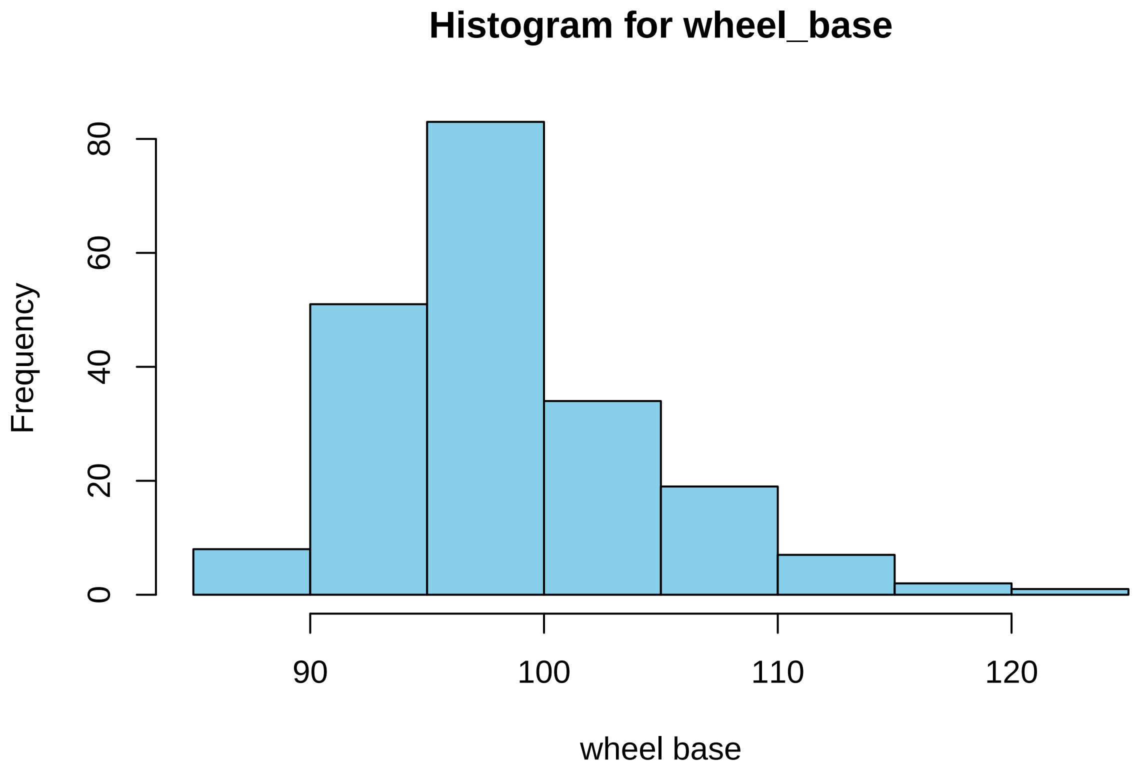
```
norm_loss <- as.numeric(normalized_losses)
```

```
## Warning: NAs introduced by coercion
```

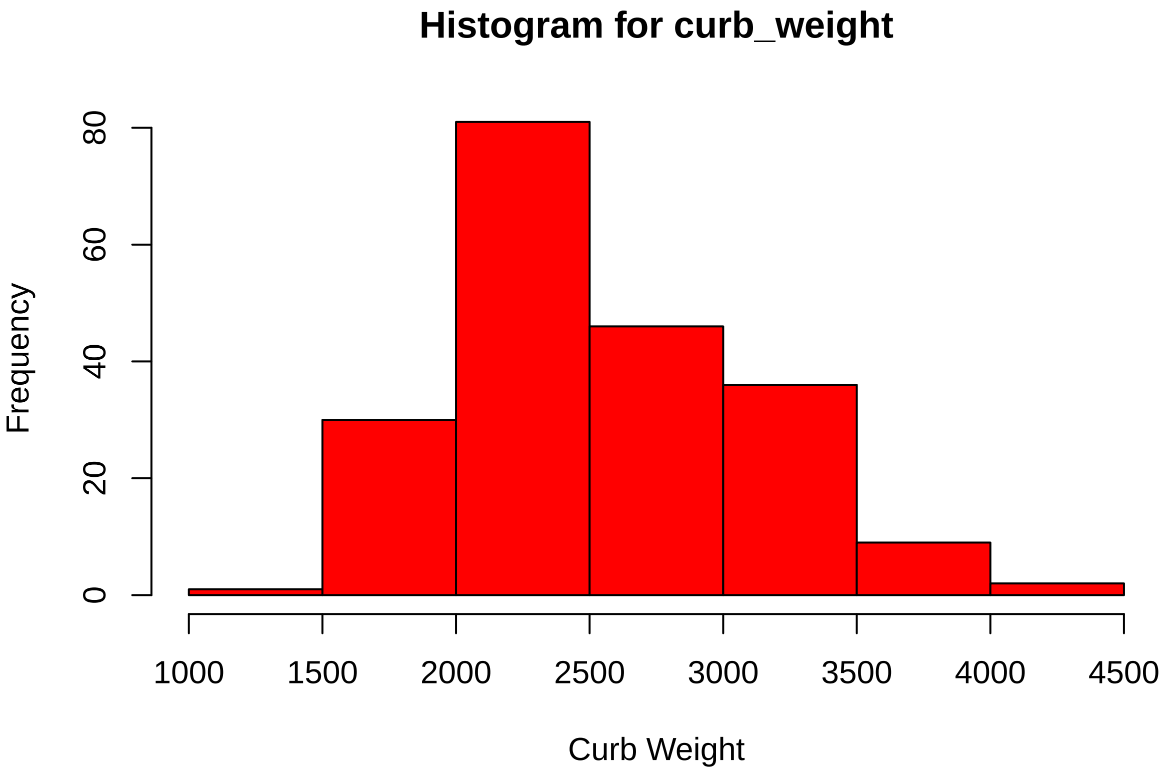
```
hist(norm_loss, main= "Histogram for normalized_losses",xlab= "Normalized Losses",xlim = c(40,275), ylab
```



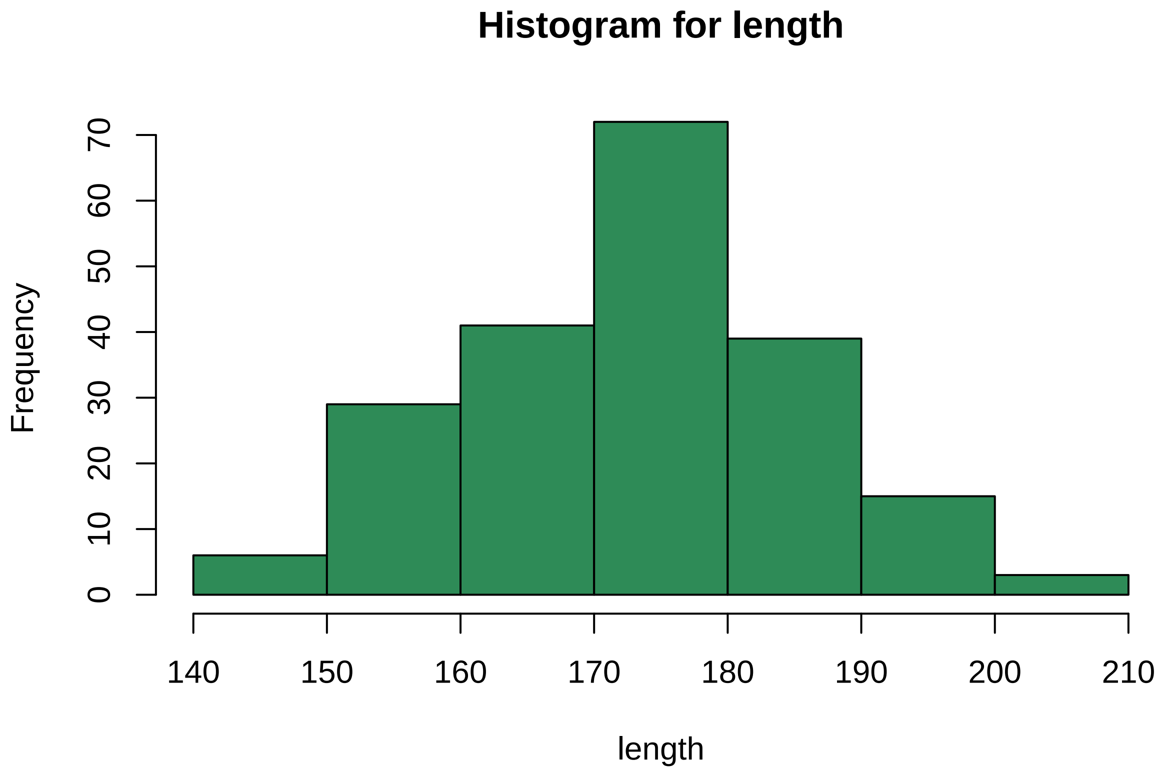
```
hist(wheel_base, main= "Histogram for wheel_base",xlab= "wheel base", ylab= "Frequency", col= "skyblue"
```



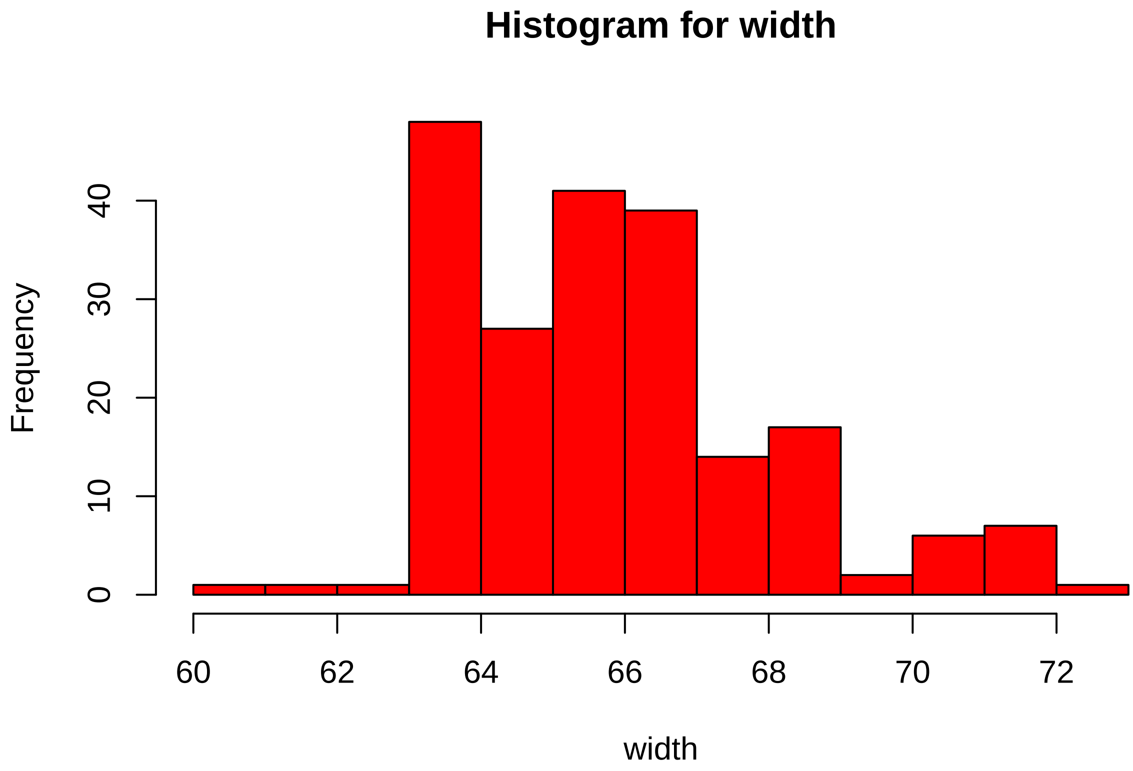
```r
hist(curb_weight,main= "Histogram for curb_weight",xlab= "Curb Weight", ylab= "Frequency", col= rainbow
```



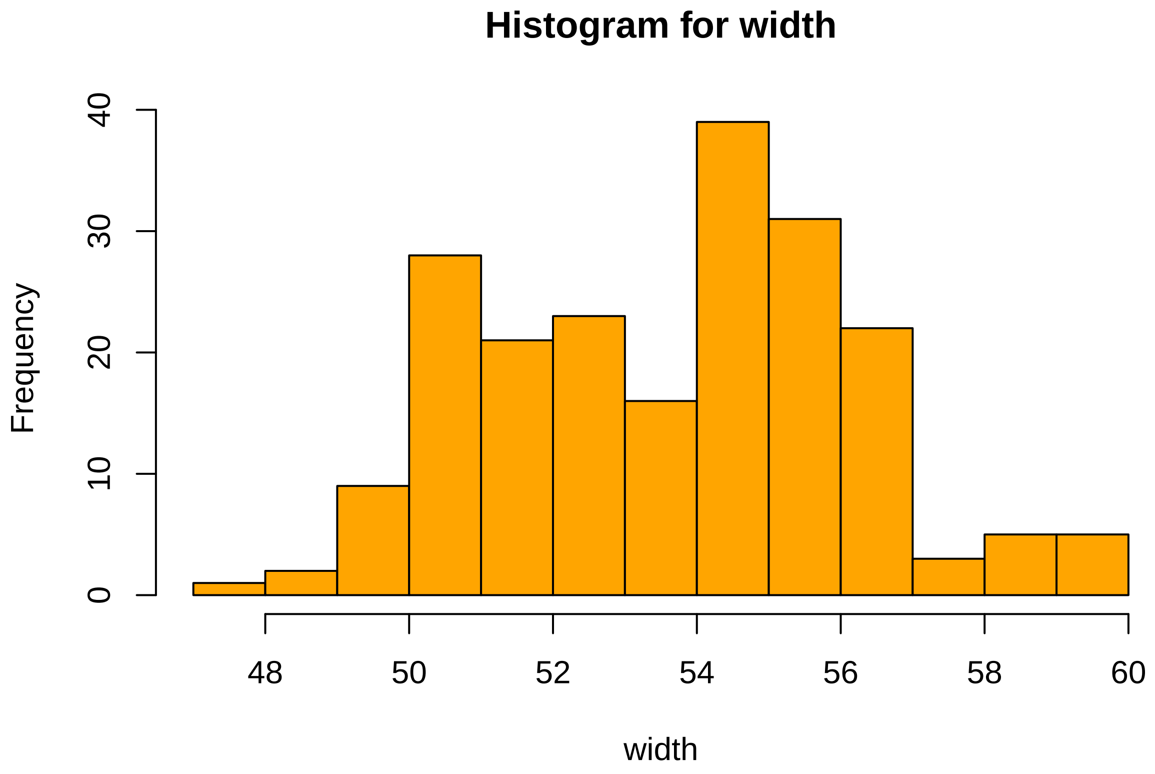
```r
hist(length, main= "Histogram for length",xlab= "length", ylab= "Frequency", col= "seagreen",border= "bl
```



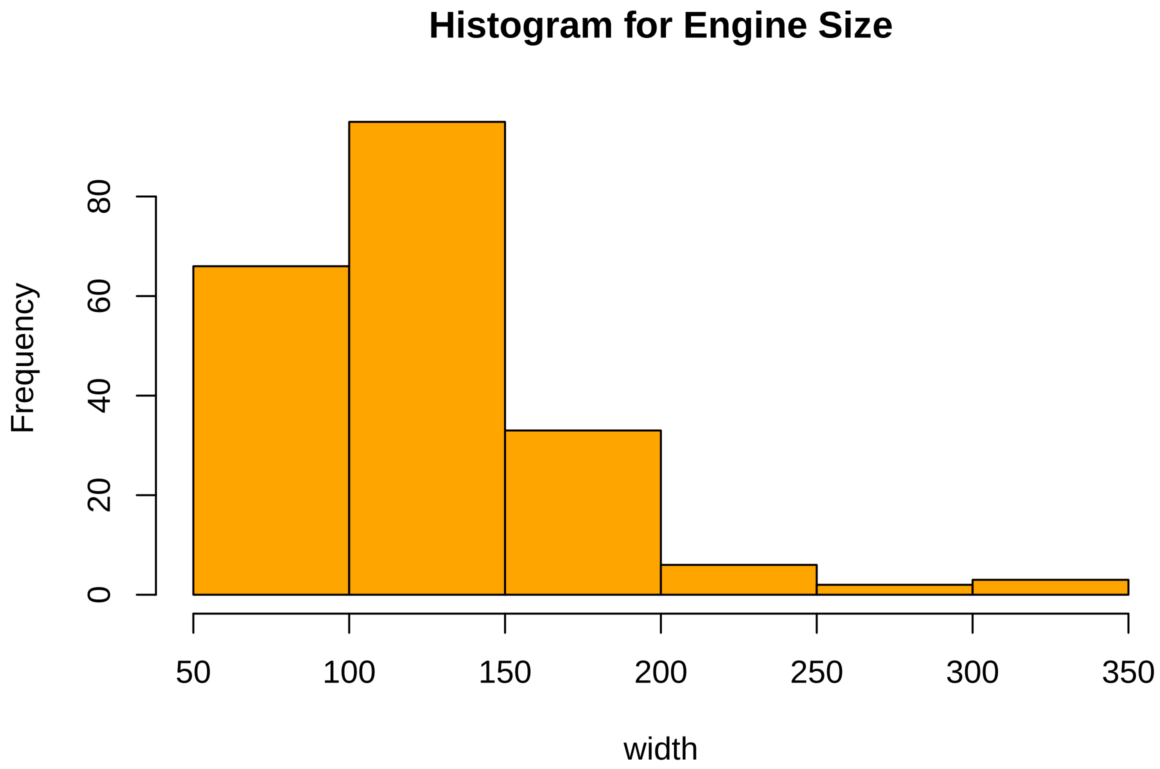
```r
hist(width, main= "Histogram for width",xlab= "width", ylab= "Frequency", col= "red",border= "black")
```



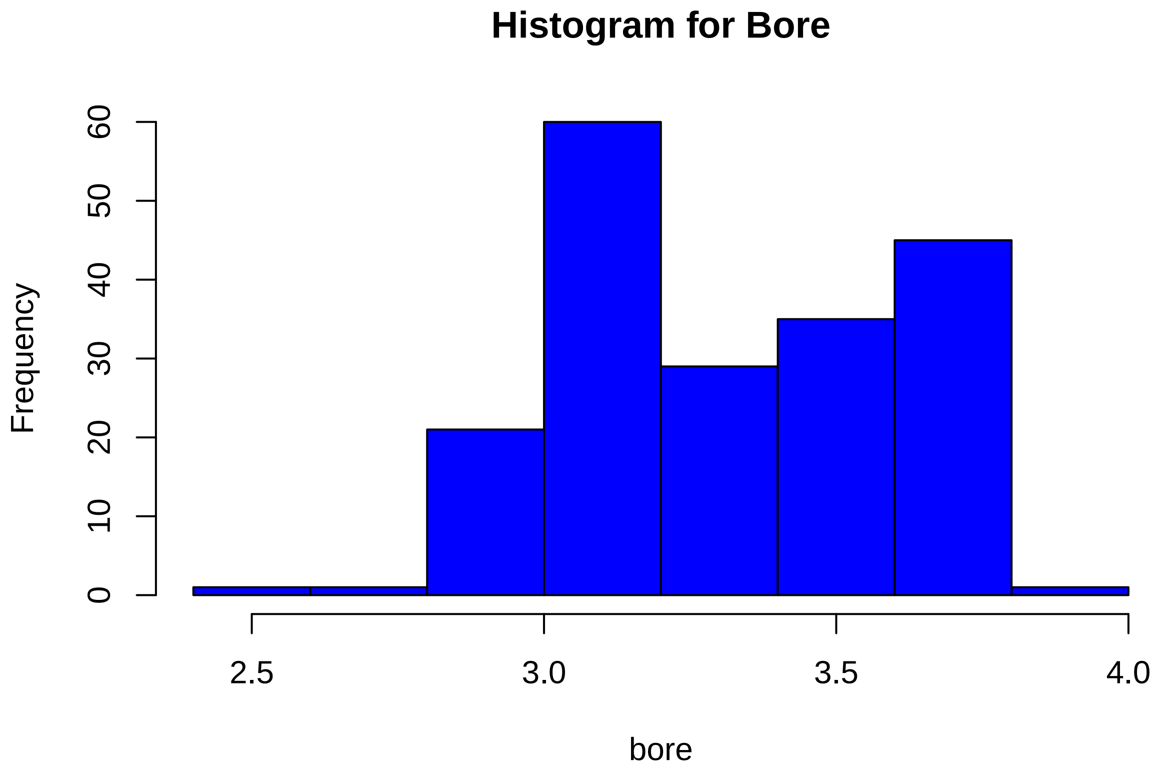
```
hist(height, main= "Histogram for width",xlab= "width", ylab= "Frequency", col= "orange",border= "black
```



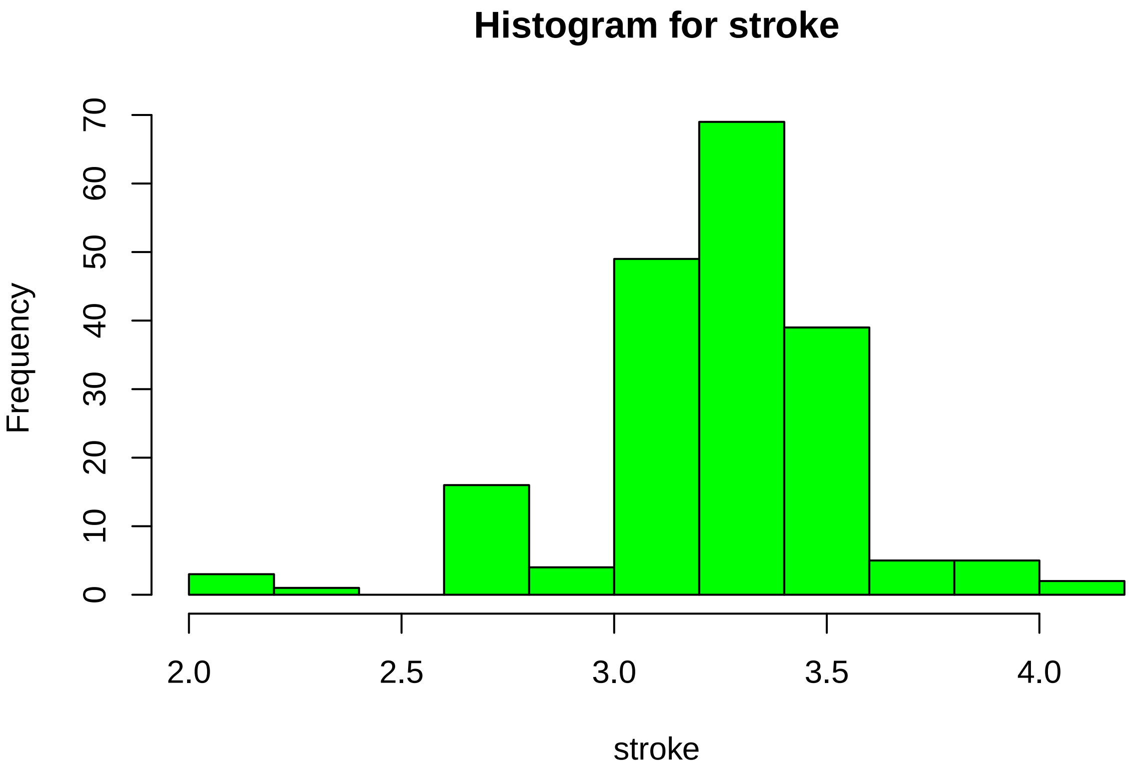
```
hist(engine_size, main= "Histogram for Engine Size",xlab= "width", ylab= "Frequency", col= "orange",bord
```



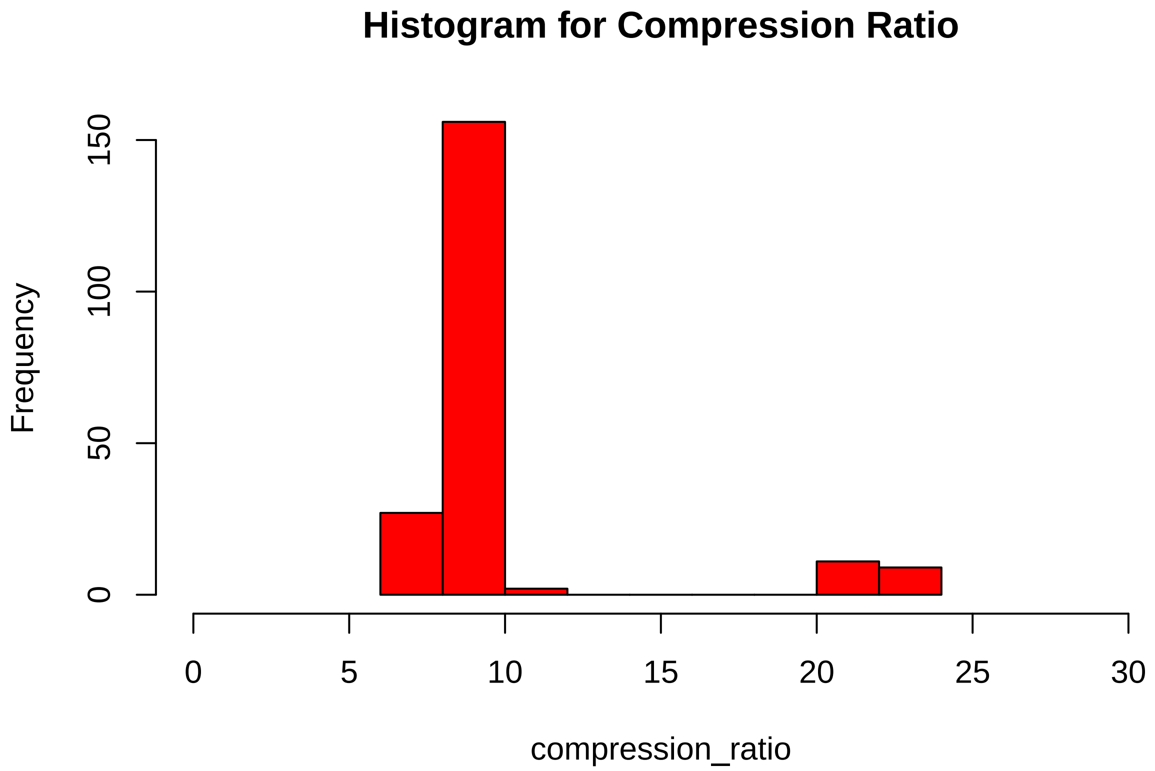
```
bore_num <- as.numeric(autoMobile$bore)
hist(bore_num, main= "Histogram for Bore",xlab= "bore", ylab= "Frequency", col= "blue",border= "black")
```



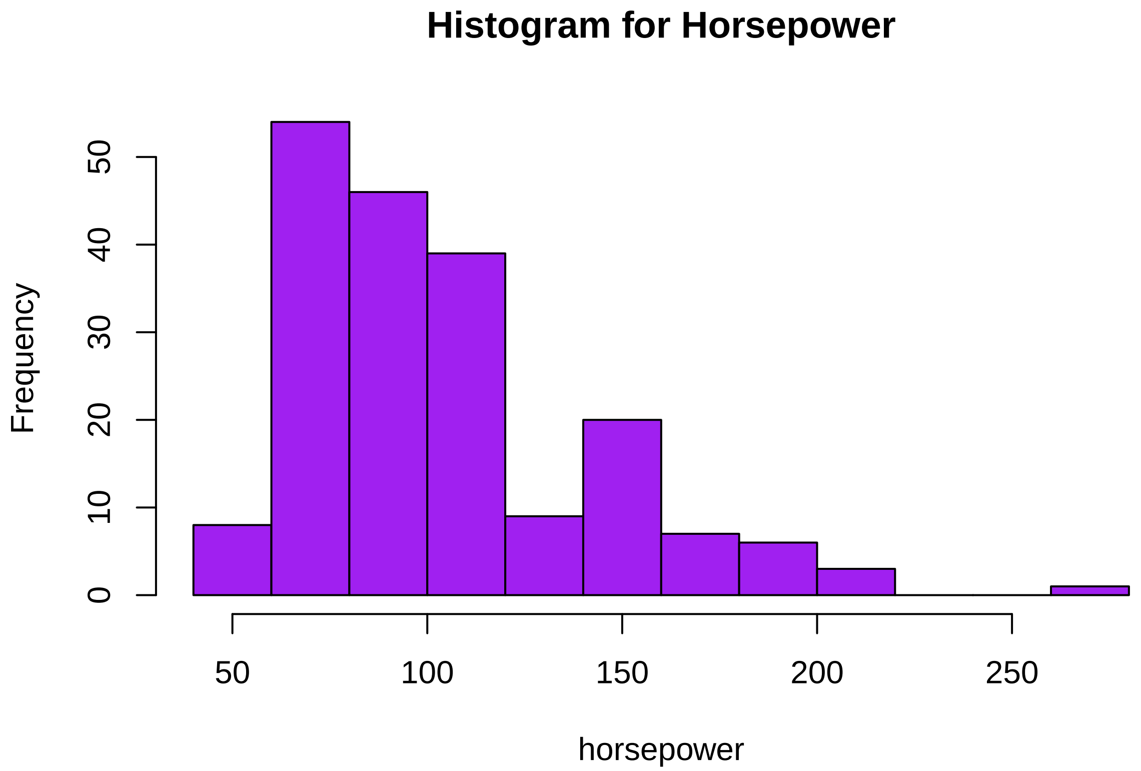
```
stroke_num <- as.numeric(autoMobile$stroke)
hist(stroke_num, main= "Histogram for stroke",xlab= "stroke", ylab= "Frequency", col= "green",border= "
```



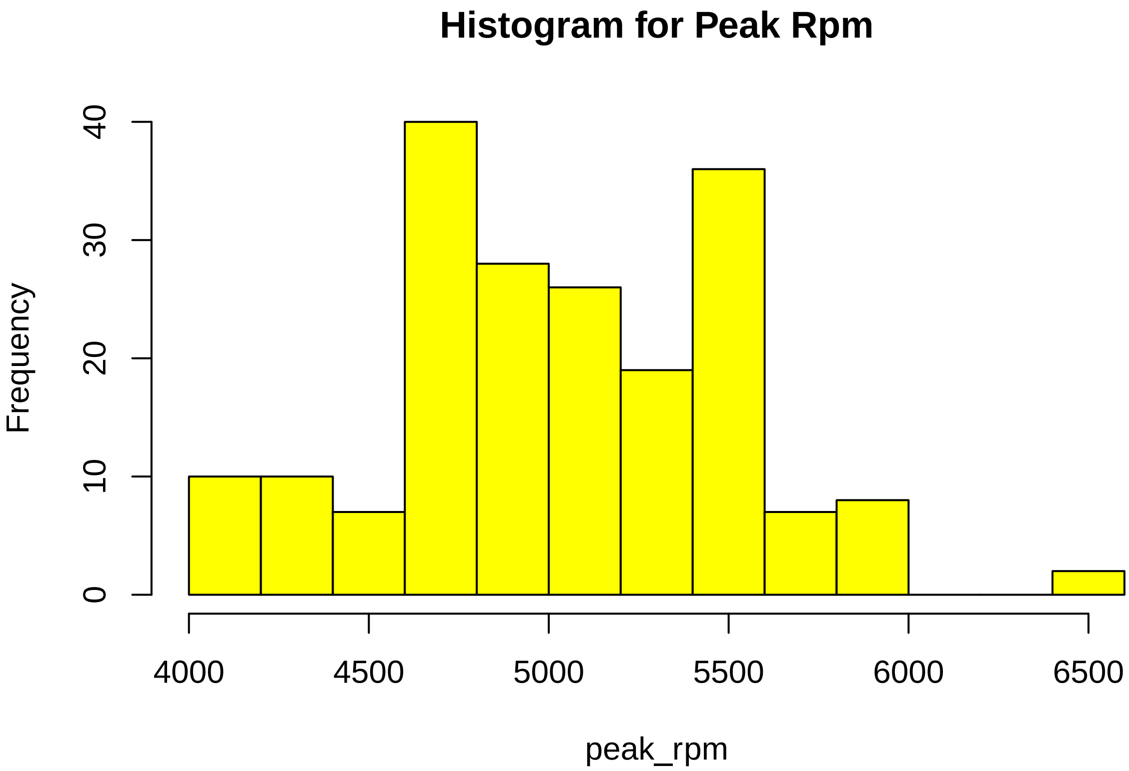
```
hist(compression_ratio, main= "Histogram for Compression Ratio",xlab= "compression_ratio", ylab= "Freque
```



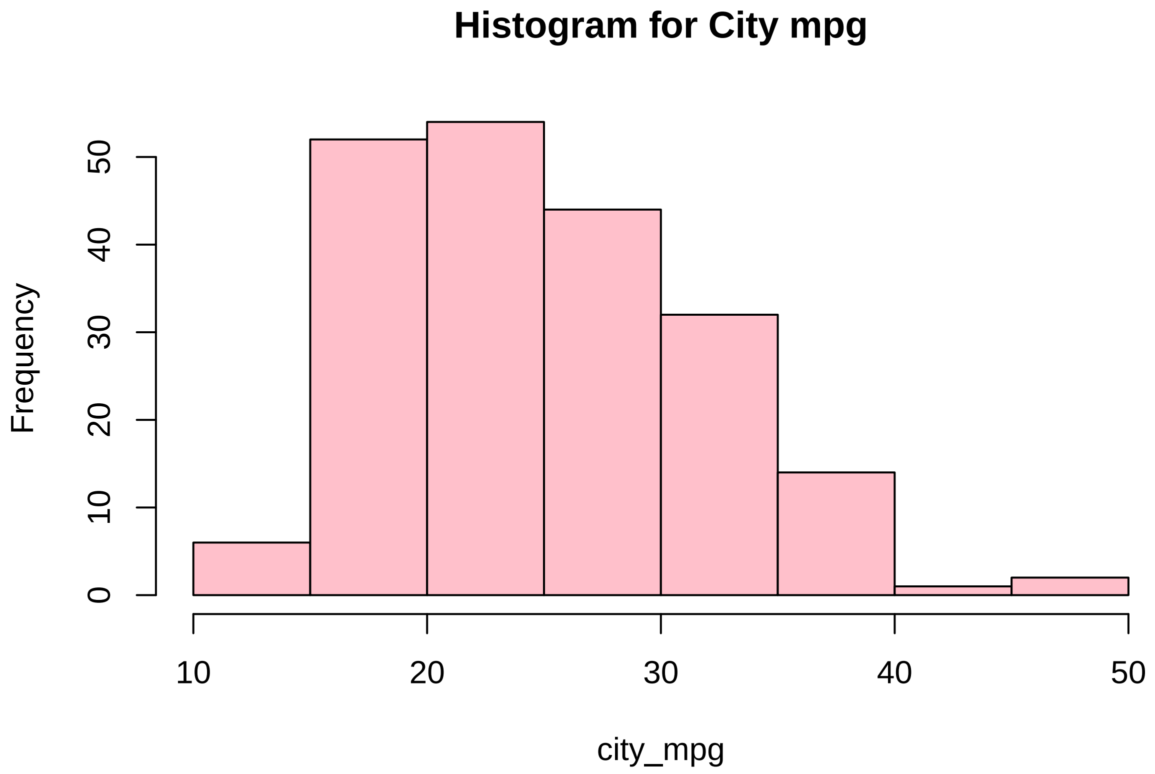
```
horsepower_num <- as.numeric(autoMobile$horsepower)
hist(horsepower_num, main= "Histogram for Horsepower",xlab= "horsepower", ylab= "Frequency", col= "purpl
```



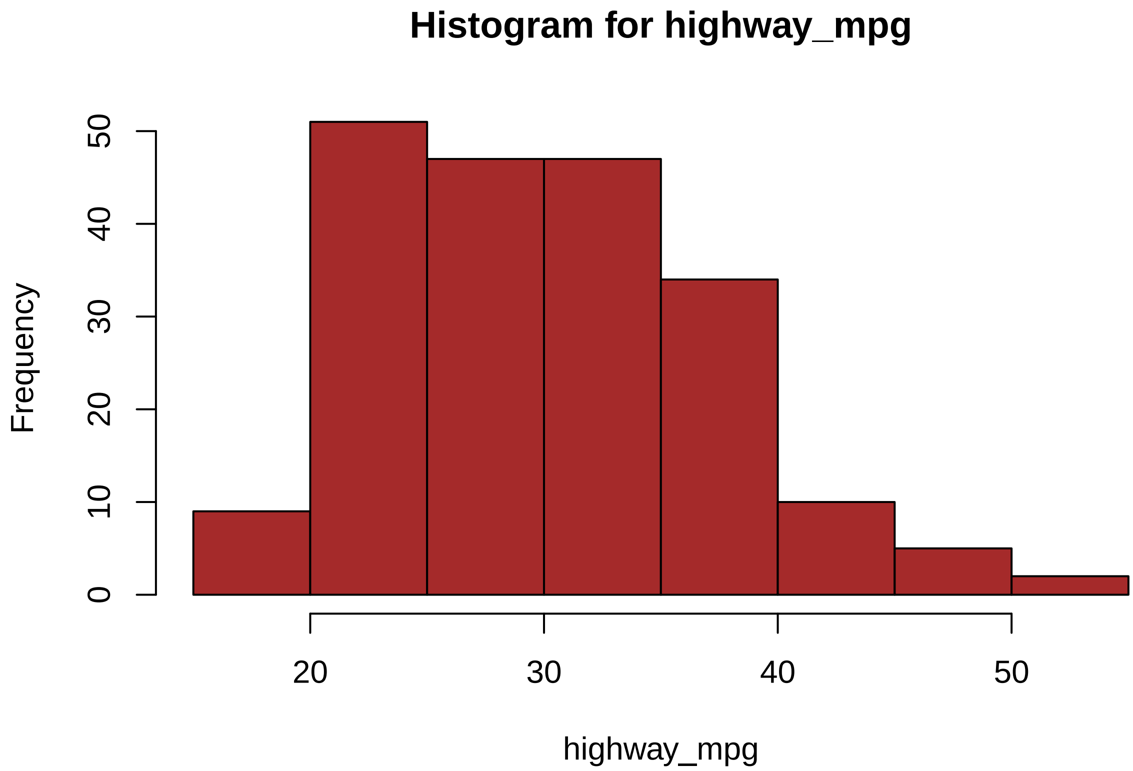
```
peak_rpm_num <- as.numeric(autoMobile$peak_rpm)
hist(peak_rpm_num, main= "Histogram for Peak Rpm",xlab= "peak_rpm", ylab= "Frequency", col= "yellow",bo
```



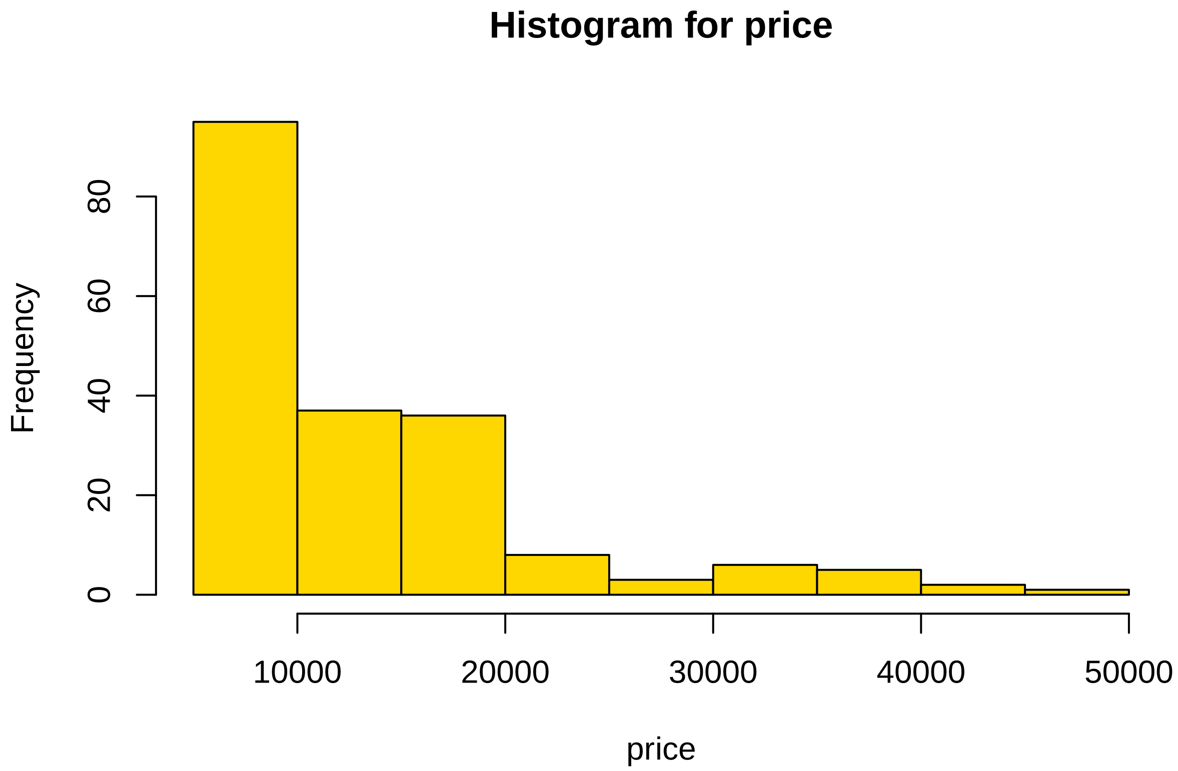
```r
hist(city_mpg, main= "Histogram for City mpg",xlab= "city_mpg", ylab= "Frequency", col= "pink",border=
```



```
hist(highway_mpg, main= "Histogram for highway_mpg",xlab= "highway_mpg", ylab= "Frequency", col= "brown
```



```
price_num <-as.numeric(autoMobile$price)
hist(price_num, main= "Histogram for price",xlab= "price", ylab= "Frequency", col= "gold",border= "blac
```

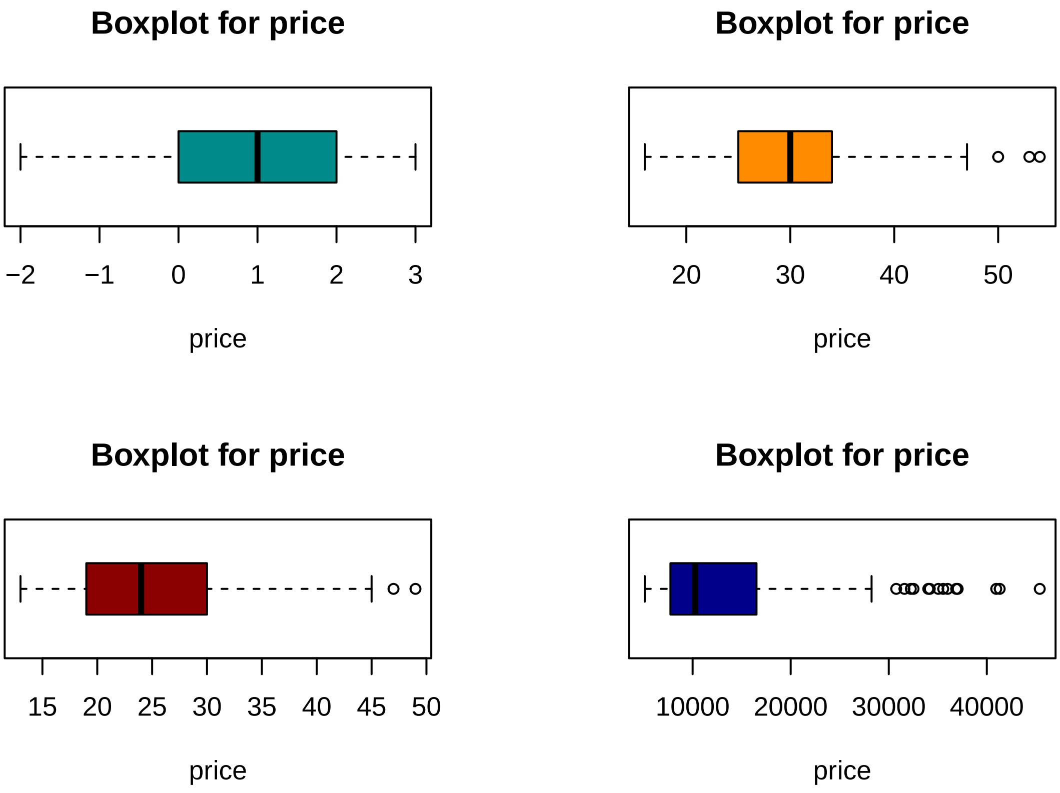


```
skewness_automobile = c(skewness(autoMobile$highway_mpg),
                skewness(autoMobile$city_mpg),
                skewness(autoMobile$price),
                skewness(autoMobile$peak_rpm),
                skewness(autoMobile$horsepower),
                skewness(autoMobile$compression_ratio),
                skewness(autoMobile$bore),
                skewness(autoMobile$stroke),
                skewness(autoMobile$engine_size),
                skewness(autoMobile$height),
                skewness(autoMobile$width),
                skewness(autoMobile$length),
                skewness(autoMobile$wheel_base),
                skewness(autoMobile$curb_weight)

              )

skew.Names <- c("highway_mpg", "city_mpg" , "price", "peak_rpm", "horsepower",
                "compression_ratio","bore","stroke","engine_size",
                "height","width","length","wheel_base","curb_weight")

skewnessDF <- data.frame(skew.Names, skewness_automobile)
skewnessDF
```

```
##             skew.Names skewness_automobile
## 1        highway_mpg          0.52925492
## 2           city_mpg          0.66937963
## 3              price          1.74745011
## 4           peak_rpm          0.09637108
## 5         horsepower          1.12744523
## 6  compression_ratio          2.58226566
## 7               bore         -0.02563592
## 8             stroke         -0.74139863
## 9        engine_size          1.99891286
## 10            height          0.03465855
## 11             width          0.85722465
## 12            length          0.13679913
## 13        wheel_base          0.96786876
## 14       curb_weight          0.66042442
```

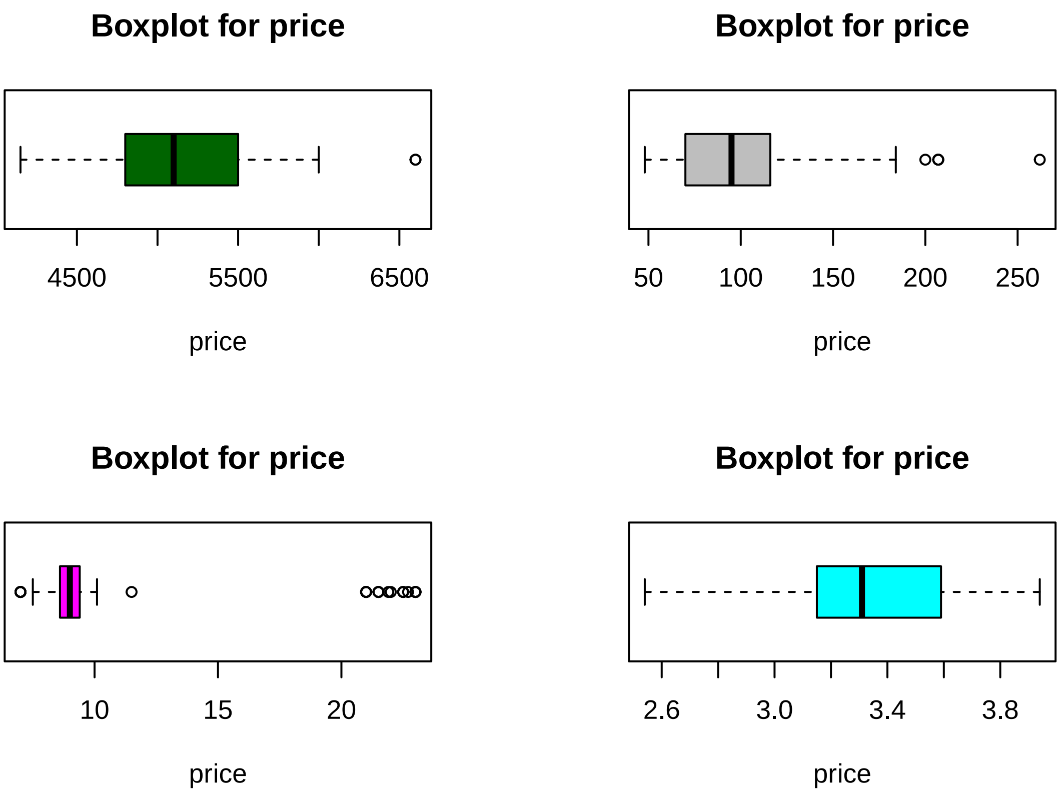
BOX PLOTS

```
par(mfrow = c(2,2))

boxplot(symboling, main= "Boxplot for price", xlab= "price", col= "darkcyan", horizontal = TRUE)
boxplot(highway_mpg, main= "Boxplot for price", xlab= "price", col= "darkorange", horizontal = TRUE)
boxplot(city_mpg, main= "Boxplot for price", xlab= "price", col= "darkred", horizontal = TRUE)
boxplot(price_num, main= "Boxplot for price", xlab= "price", col= "darkblue", horizontal = TRUE)
```

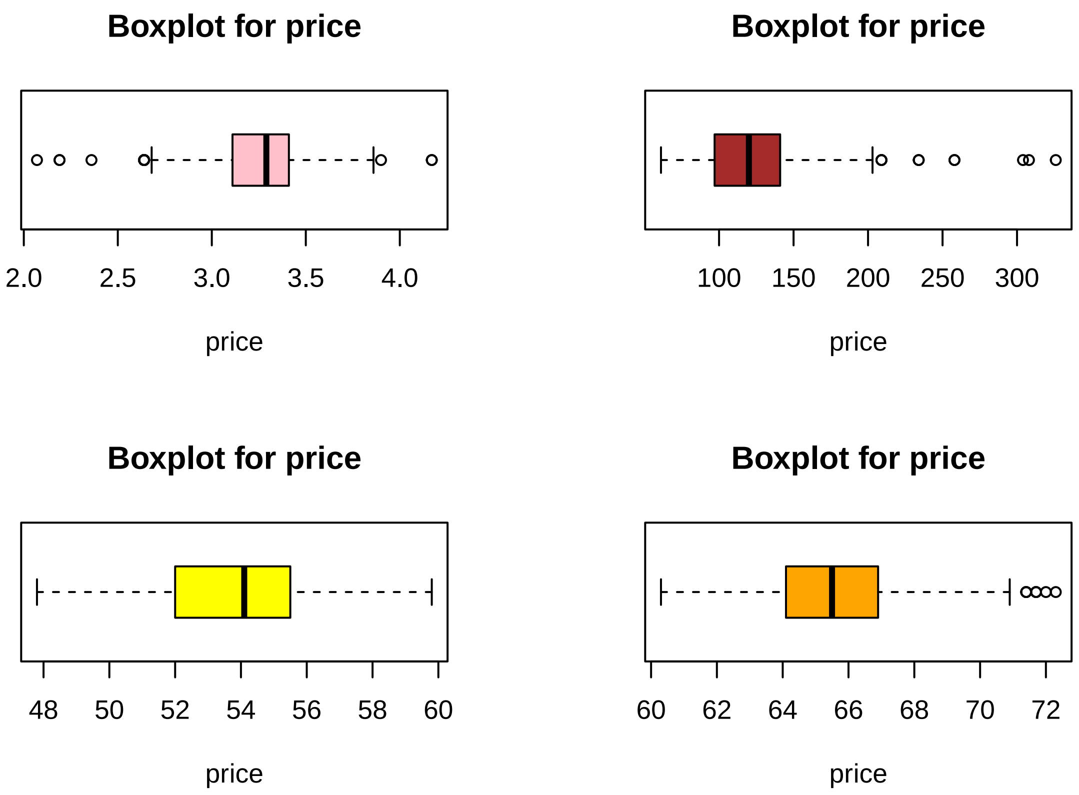


```
par(mfrow = c(2,2))
boxplot(peak_rpm_num, main= "Boxplot for price", xlab= "price", col= "darkgreen", horizontal = TRUE)
```

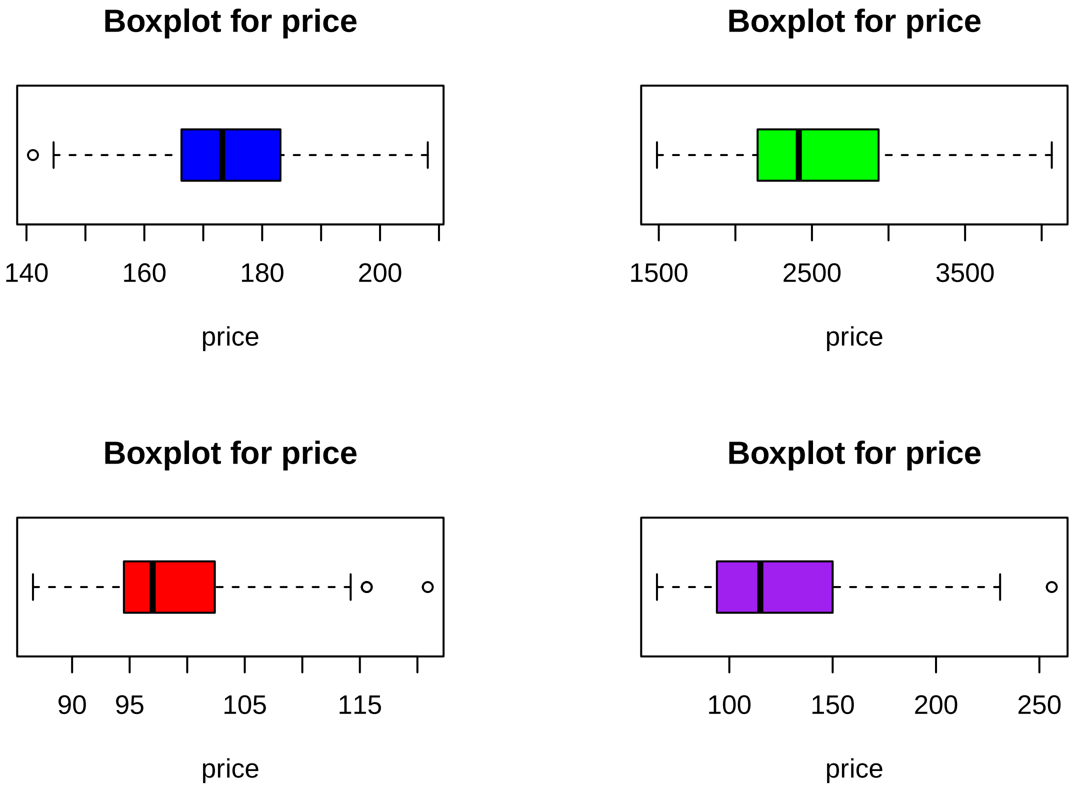
```r
boxplot(horsepower_num, main= "Boxplot for price", xlab= "price", col= "gray", horizontal = TRUE)
boxplot(compression_ratio, main= "Boxplot for price", xlab= "price", col= "magenta", horizontal = TRUE)
boxplot(bore_num, main= "Boxplot for price", xlab= "price", col= "cyan", horizontal = TRUE)
```



```r
par(mfrow = c(2,2))
boxplot(stroke_num, main= "Boxplot for price", xlab= "price", col= "pink", horizontal = TRUE)
boxplot(engine_size, main= "Boxplot for price", xlab= "price", col= "brown", horizontal = TRUE)
boxplot(height, main= "Boxplot for price", xlab= "price", col= "yellow", horizontal = TRUE)
boxplot(width, main= "Boxplot for price", xlab= "price", col= "orange", horizontal = TRUE)
```

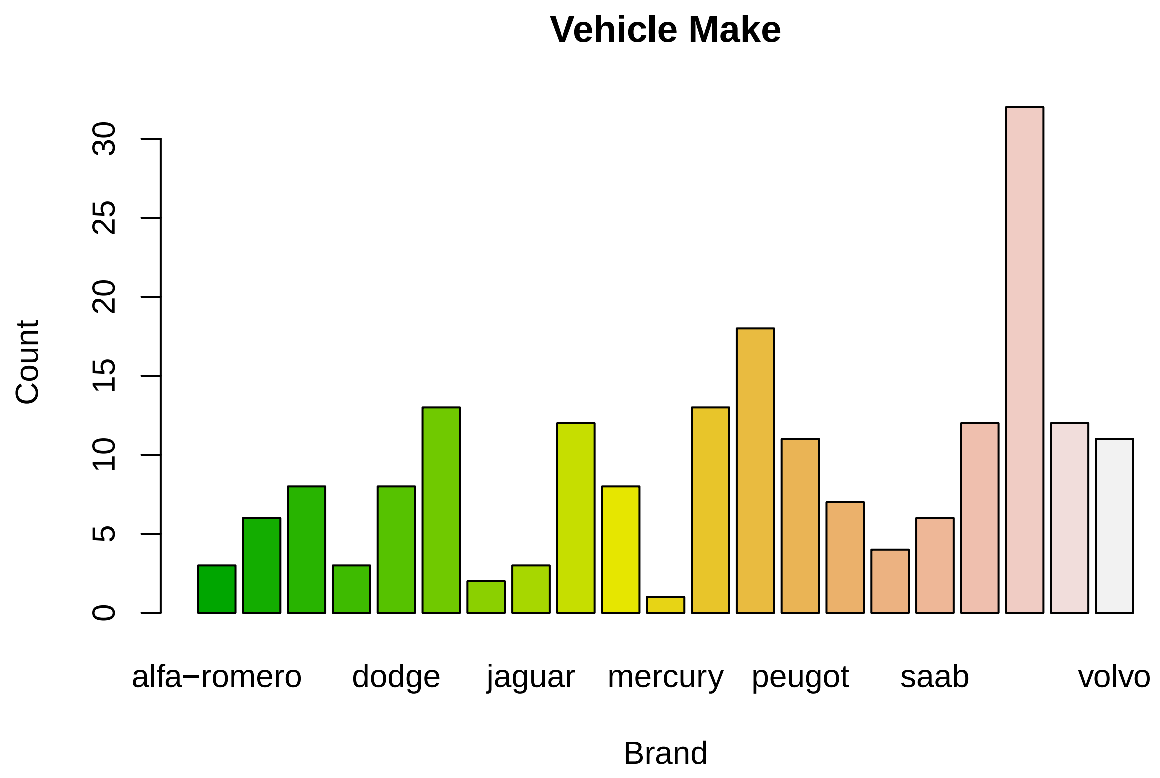


```
par(mfrow = c(2,2))
boxplot(length, main= "Boxplot for price", xlab= "price", col= "blue", horizontal = TRUE)
boxplot(curb_weight, main= "Boxplot for price", xlab= "price", col= "green", horizontal = TRUE)
boxplot(wheel_base, main= "Boxplot for price", xlab= "price", col= "red", horizontal = TRUE)
boxplot(norm_loss, main= "Boxplot for price", xlab= "price", col= "purple", horizontal = TRUE)
```



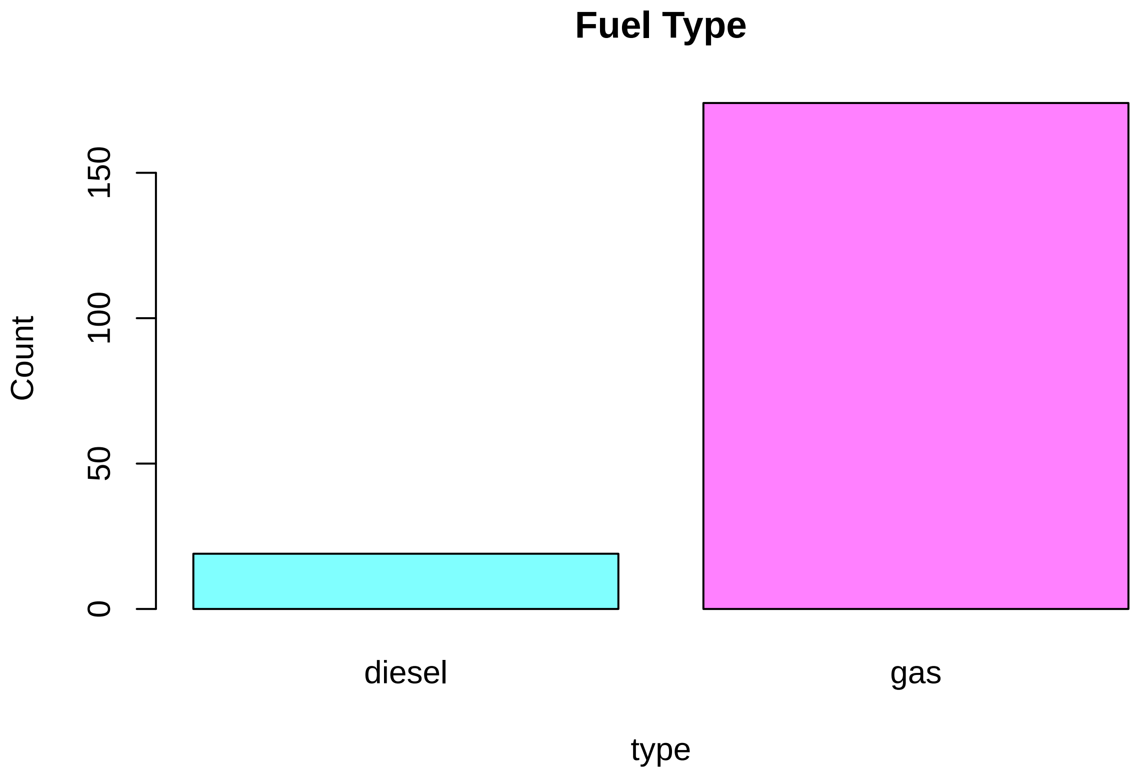
BAR PLOT

```
Make_Tbl <- table(autoMobile$make)

barplot(Make_Tbl, main = "Vehicle Make", xlab = "Brand", ylab = "Count", col = terrain.colors(21), bord
```



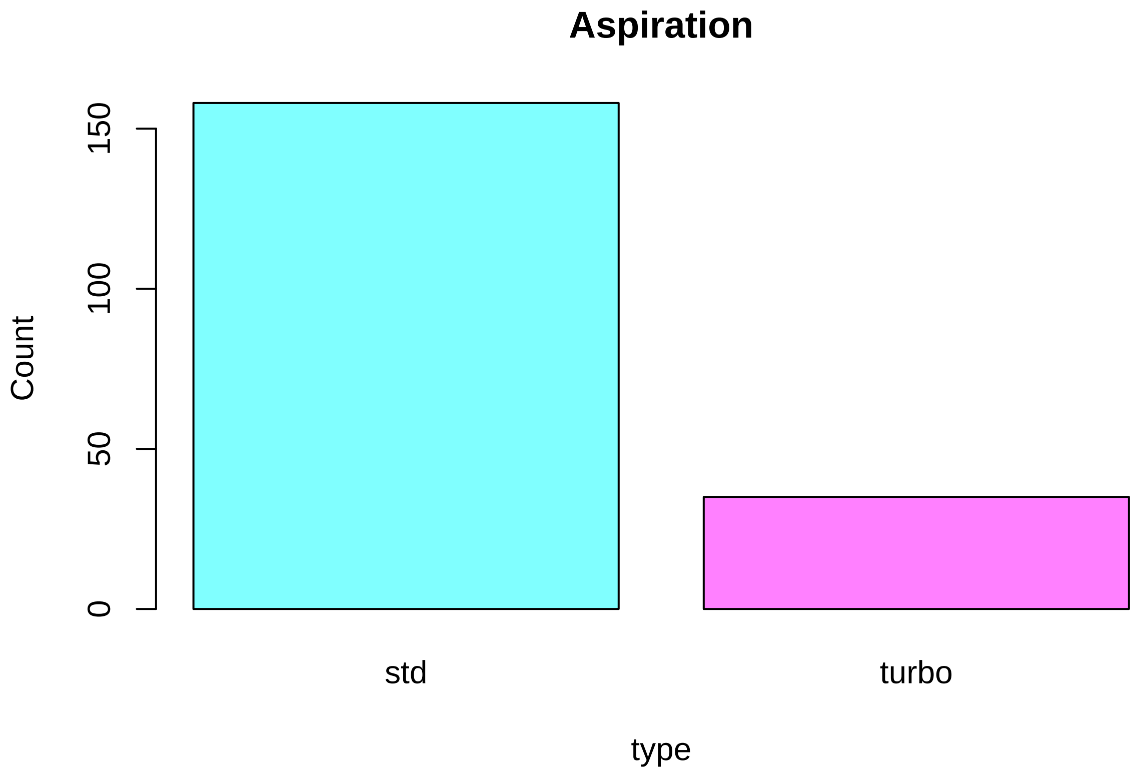
- maximum number of Vehicals are Toyota and the minimum is mercury

```
fuel_type_Tbl <- table(autoMobile$fuel_type)
barplot(fuel_type_Tbl, main = "Fuel Type", xlab = "type", ylab = "Count", col = cm.colors(2),  border =
```



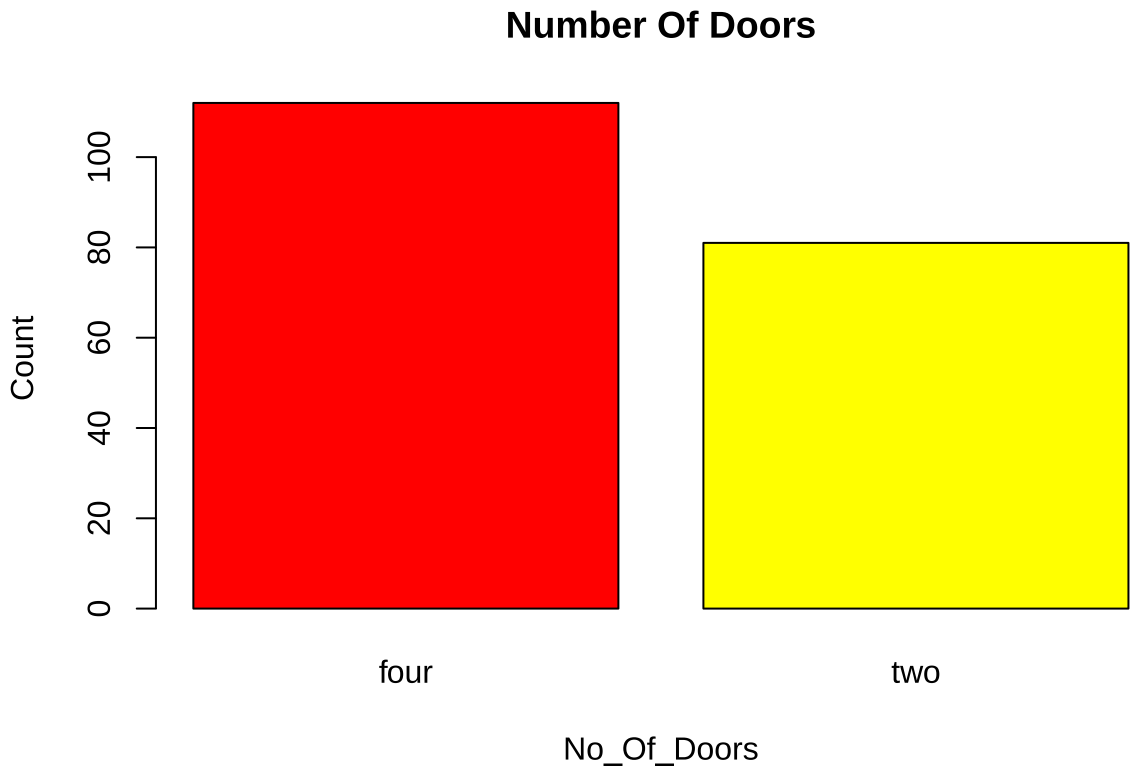
- Mostly used fuel type is gas

```
aspiration_Tbl <- table(autoMobile$aspiration)
barplot(aspiration_Tbl, main = "Aspiration", xlab = "type", ylab = "Count",col = cm.colors(2),  border =
```



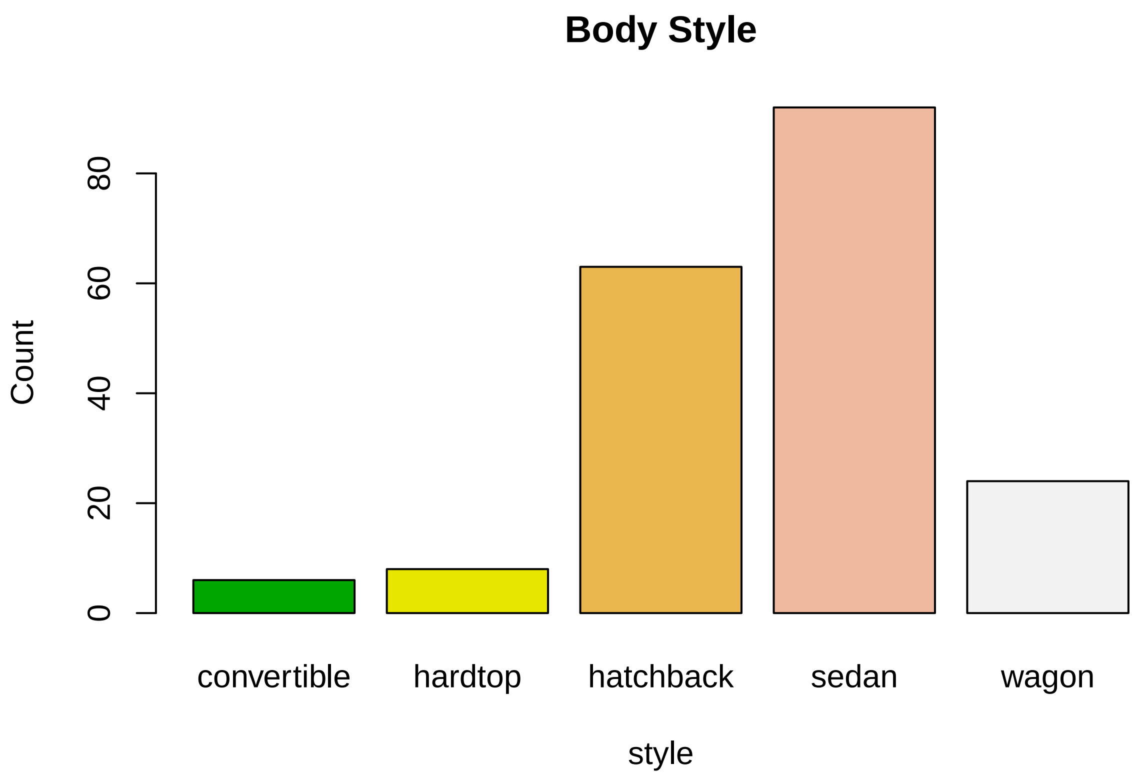
- most of them are Standard vehicles

```
num_of_doors_Tbl <- table(autoMobile$num_of_doors)
barplot(num_of_doors_Tbl, main = "Number Of Doors", xlab = "No_Of_Doors", ylab = "Count", col = heat.co
```



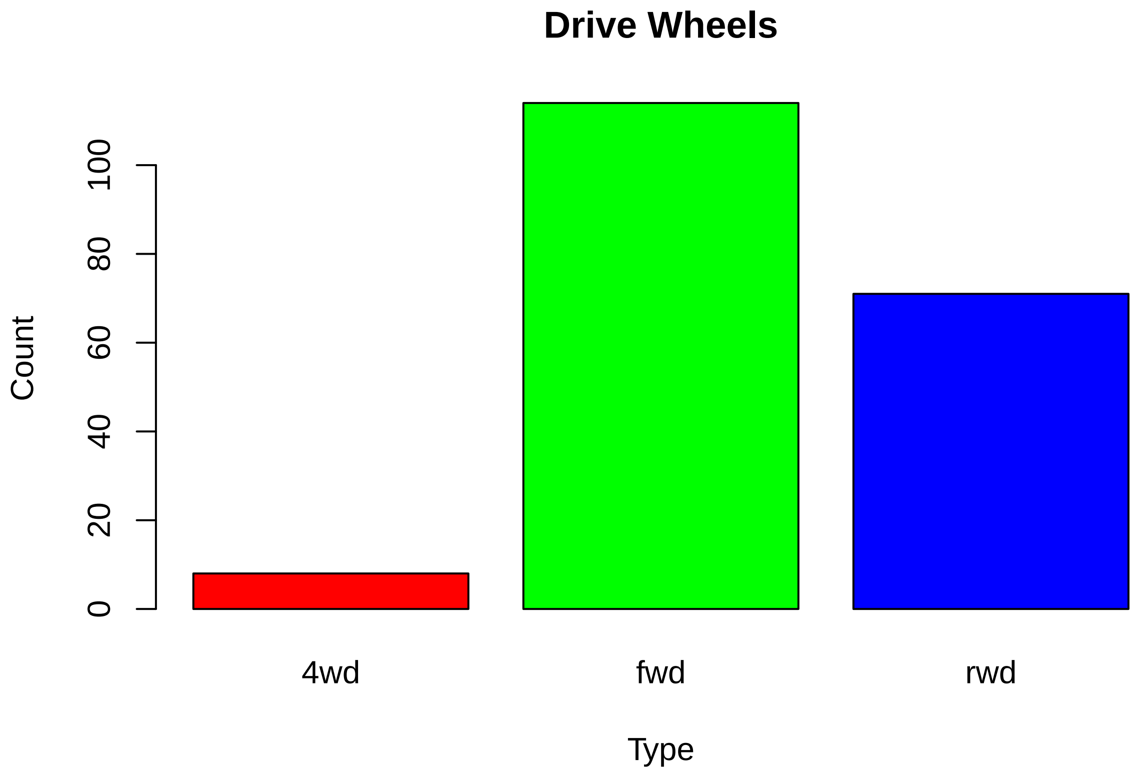
- four doors vehicles are Higher than two no of door vehicles

```
body_style_Tbl <- table(autoMobile$body_style)
barplot(body_style_Tbl, main = "Body Style", xlab = "style", ylab = "Count",col = terrain.colors(5), bo
```



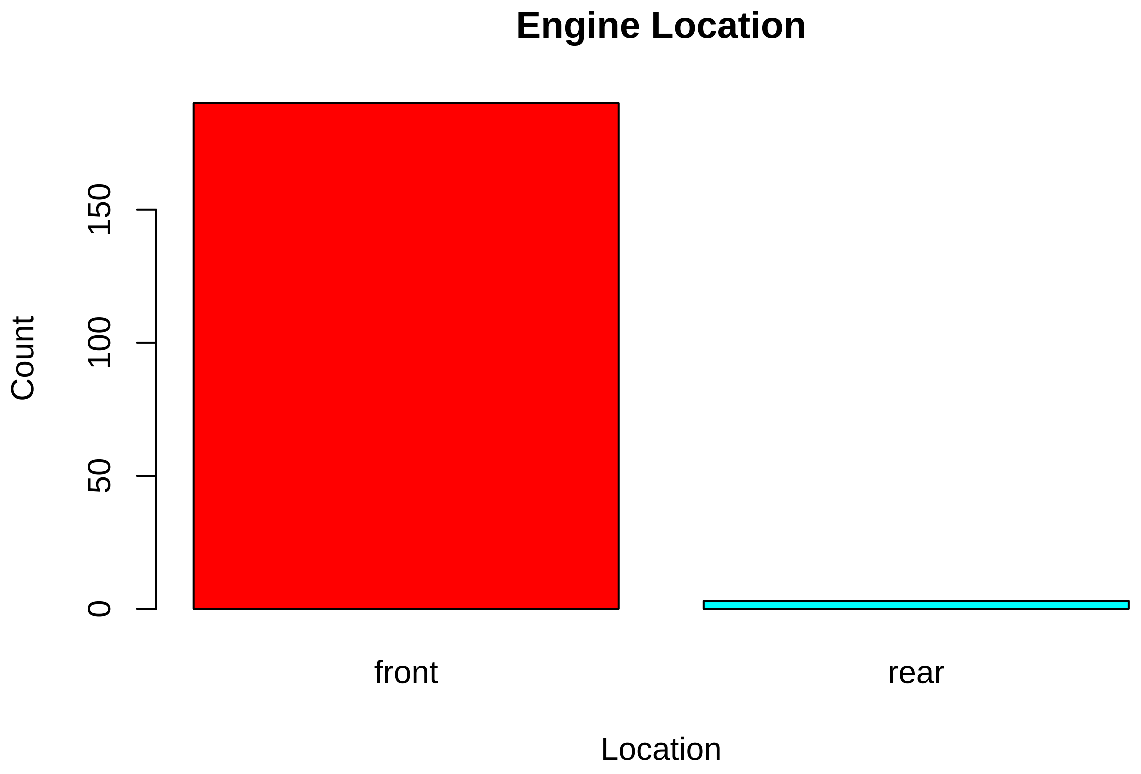
- sedan is the popular vehicle body style

```
drive_wheels_Tbl <- table(autoMobile$drive_wheels)
barplot(drive_wheels_Tbl, main = "Drive Wheels", xlab = "Type", ylab = "Count",col = rainbow(3), border
```



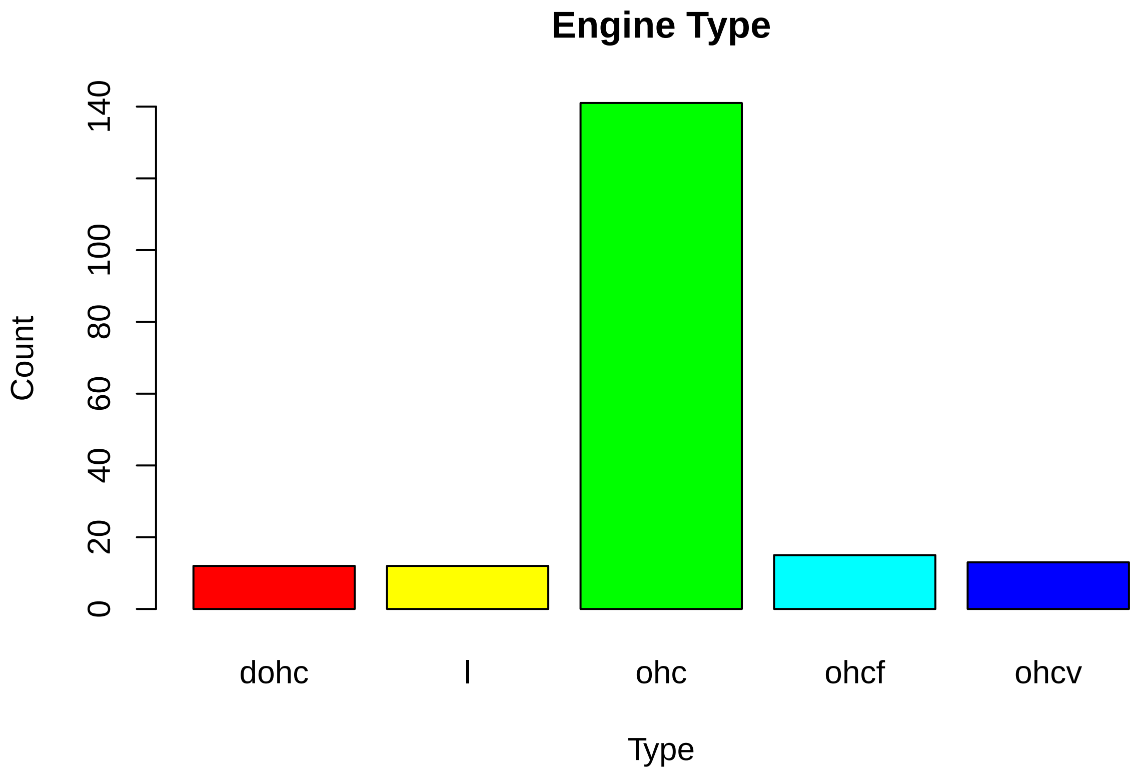
- most of drive wheels are fwd

```
engine_location_Tbl <- table(autoMobile$engine_location)
barplot(engine_location_Tbl, main = "Engine Location", xlab = "Location", ylab = "Count",col = rainbow(
```



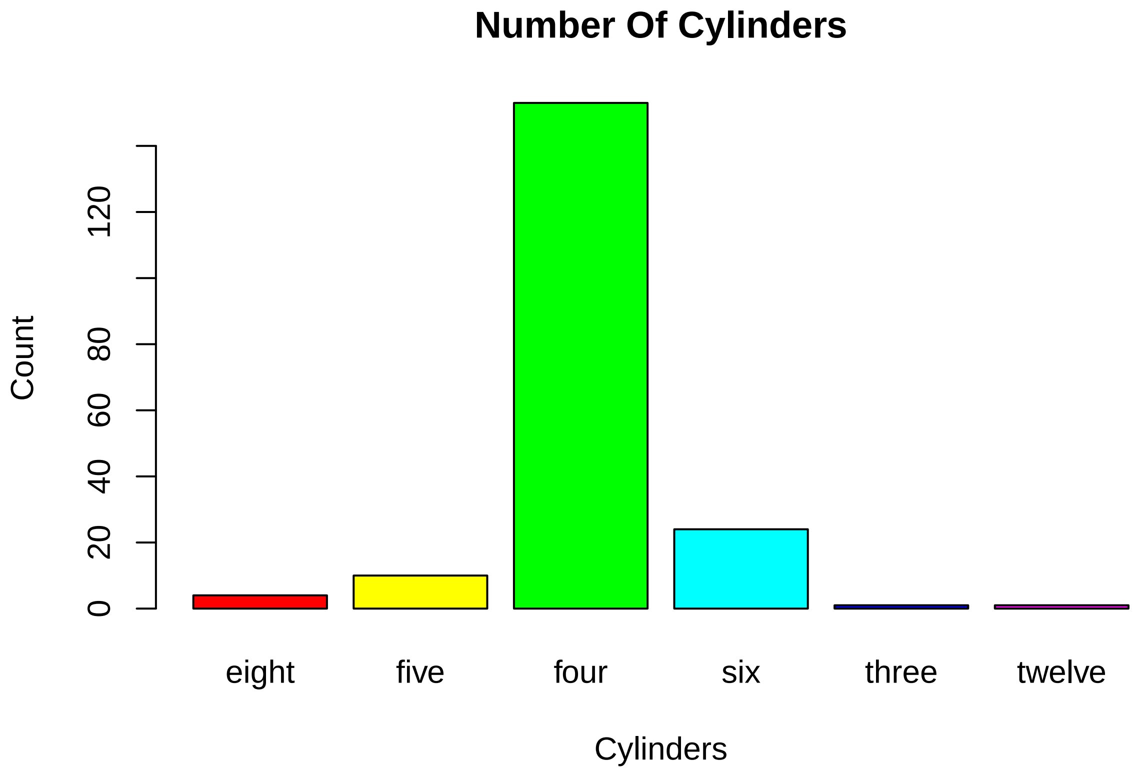
- almost all the vehicle's engine is located in the front

```
engine_type_Tbl <- table(autoMobile$engine_type)
barplot(engine_type_Tbl, main = "Engine Type", xlab = "Type", ylab = "Count", col = rainbow(6), border =
```



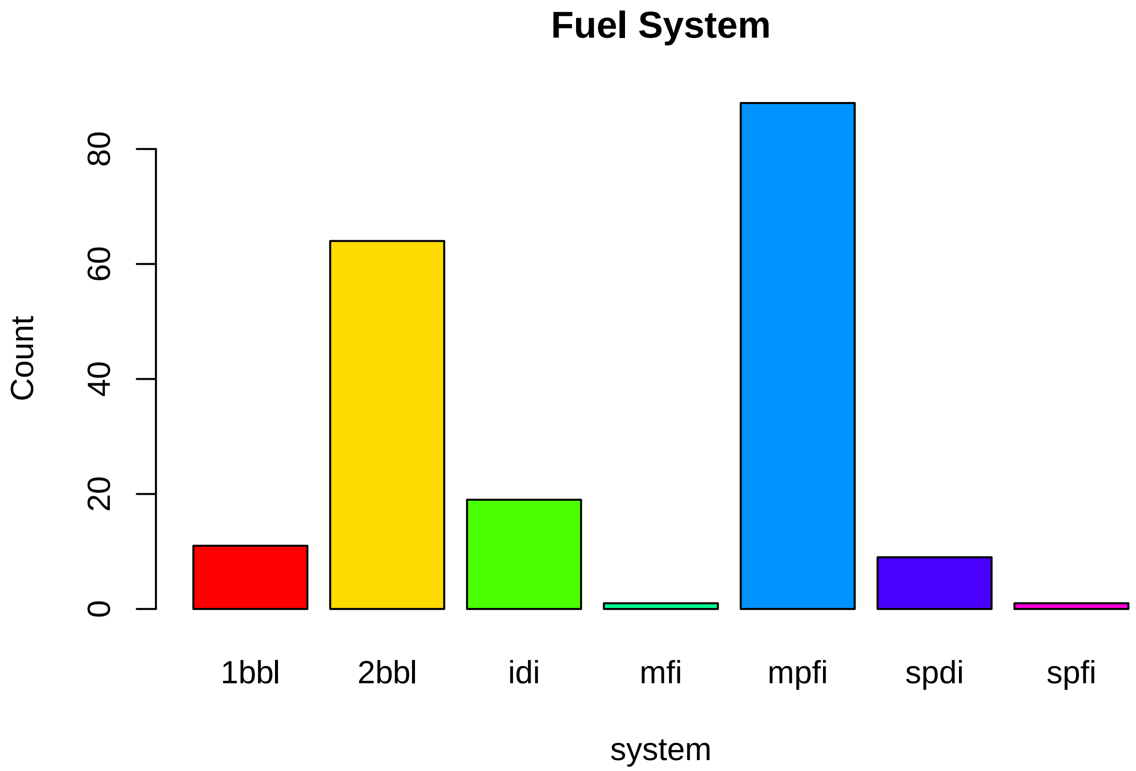
- most popular engine type is ohc

```
num_of_cylinders_Tbl <- table(autoMobile$num_of_cylinders)
barplot(num_of_cylinders_Tbl, main = "Number Of Cylinders", xlab = "Cylinders", ylab = "Count",col = rai
```



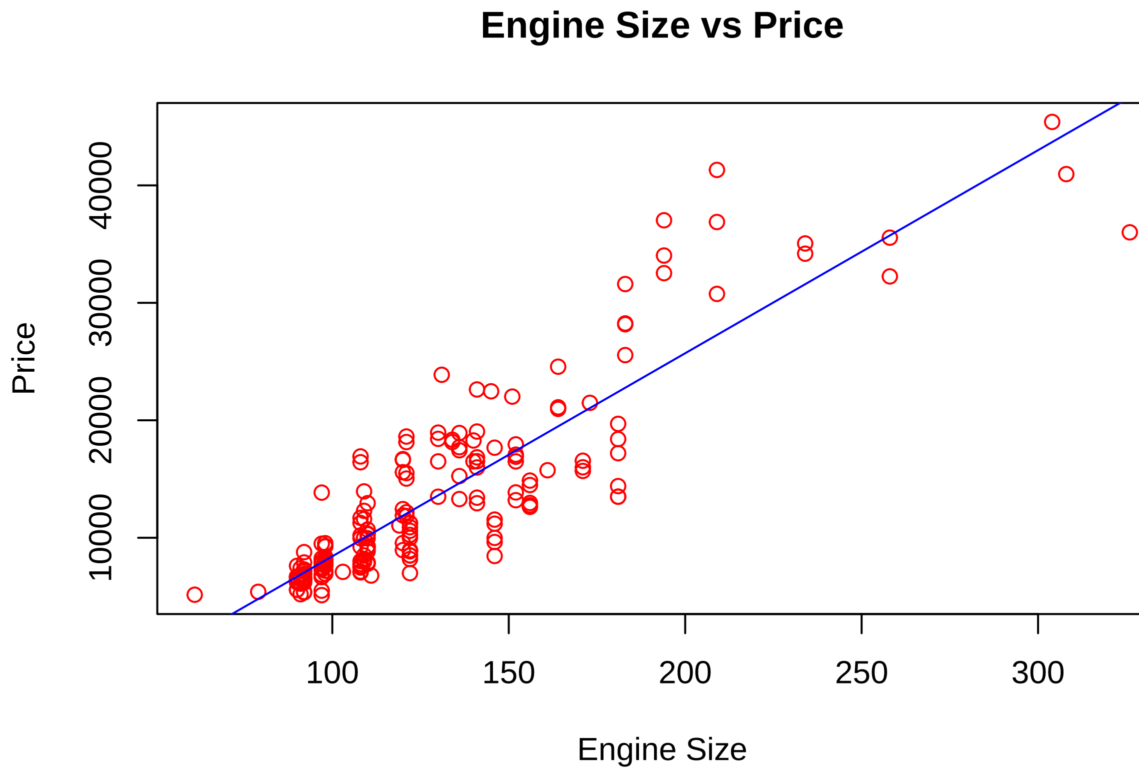
- many vehicles has four cyclinders

```
fuel_system_Tbl <- table(autoMobile$fuel_system)
barplot(fuel_system_Tbl, main = "Fuel System", xlab = "system", ylab = "Count", col = rainbow(7),  bord
```



- most popular fuel system is mpfi

```
plot(autoMobile$engine_size, autoMobile$price, main = "Engine Size vs Price",
     xlab = "Engine Size", ylab = "Price", col = "red")

lm_fit <- lm(price ~ engine_size, data = autoMobile)

abline(lm_fit, col = "blue")
```



Engine_size vs Price

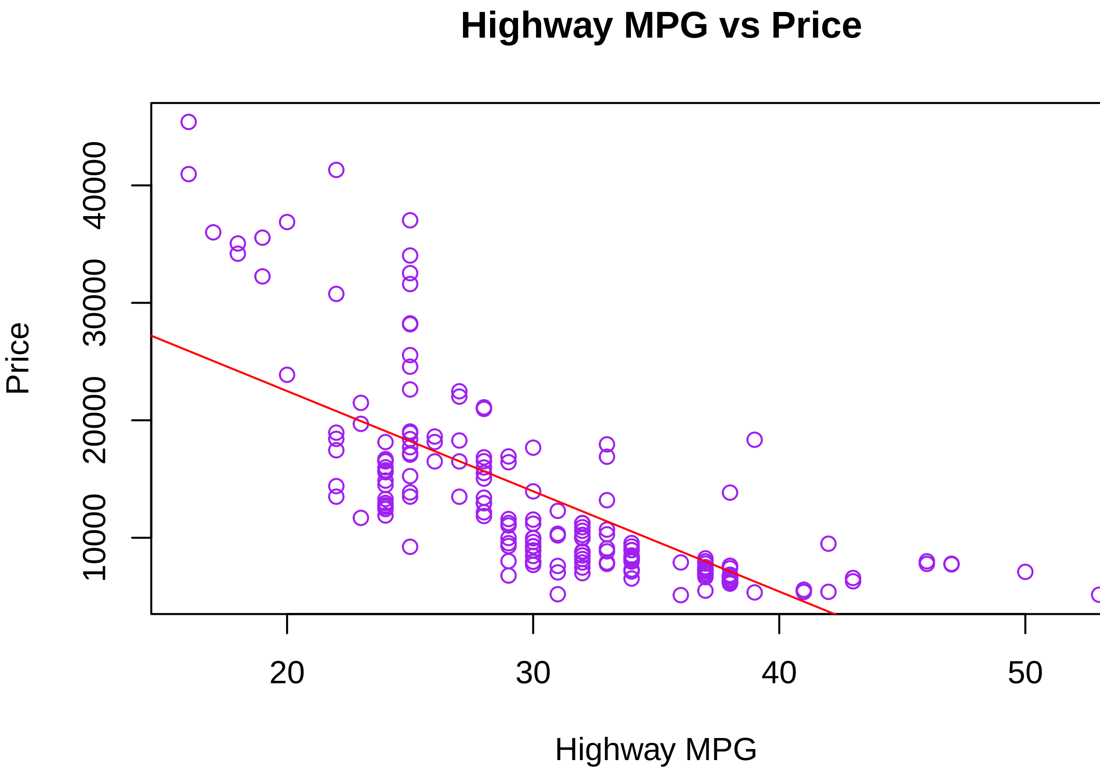
- As the engine-size goes up, the price goes up: this indicates a positive direct correlation between these two variables.
- Engine size seems like a pretty good predictor of price since the regression line is almost a perfect diagonal line.

```
CorEngineS.P <- cor(autoMobile$engine_size, autoMobile$price)
CorEngineS.P
```

```
## [1] 0.8887785
```

- We can examine the correlation between 'engine-size' and 'price' and see it's approximately: 0.8887785

```
plot(autoMobile$highway_mpg, autoMobile$price, main = " Highway MPG vs Price",
     xlab = "Highway MPG", ylab = "Price", col = "purple")

lm_fit <- lm(price ~ highway_mpg, data = autoMobile)
abline(lm_fit, col = "red")
```



Highway_mpg vs Price

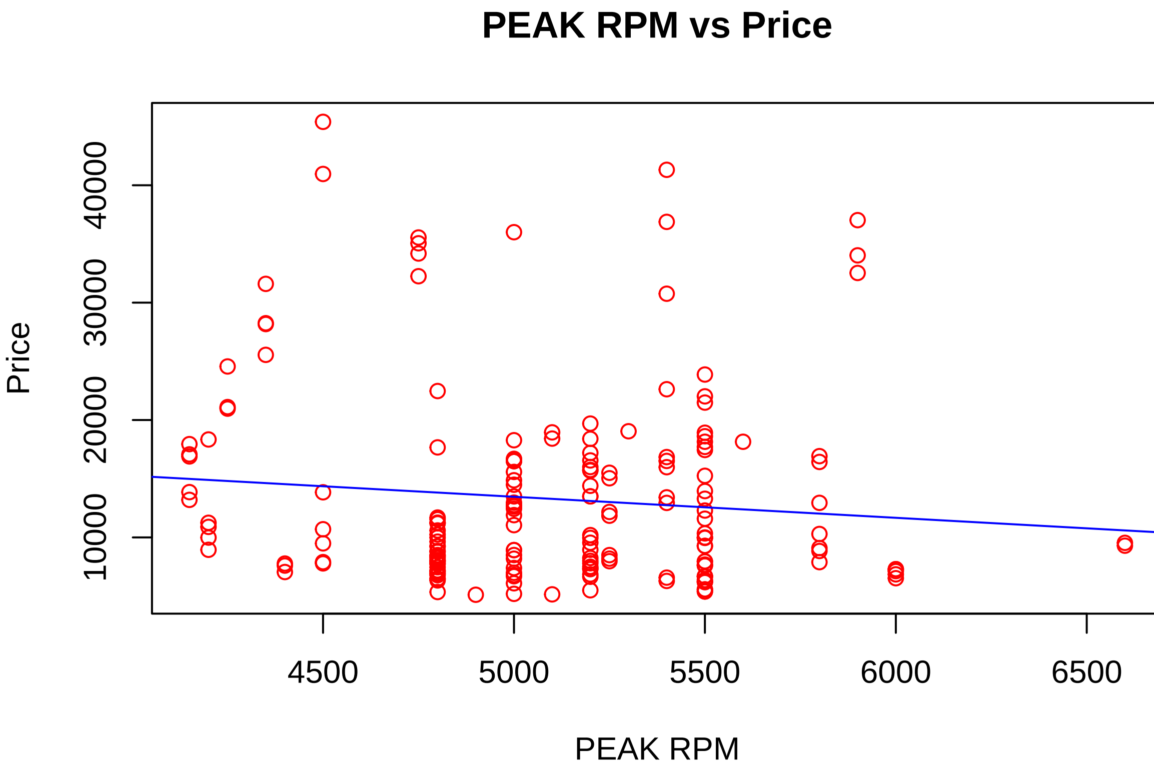
- As the highway-mpg goes up, the price goes down: this indicates an inverse/negative relationship between these two variables.
- Highway mpg could potentially be a predictor of price.

```
CorHighway_mpg.P <- cor(autoMobile$highway_mpg, autoMobile$price)
CorHighway_mpg.P
```

```
## [1] -0.7191777
```

- we can examine the correlation between 'highway_mpg' and 'price' and see it's approximately: **-0.7200901**

```
plot(autoMobile$peak_rpm, autoMobile$price, main = "PEAK RPM vs Price",
     xlab = "PEAK RPM", ylab = "Price", col = "red")

lm_fit <- lm(price ~ peak_rpm, data = autoMobile)
abline(lm_fit, col = "blue")
```



Peak_rpm vs Price

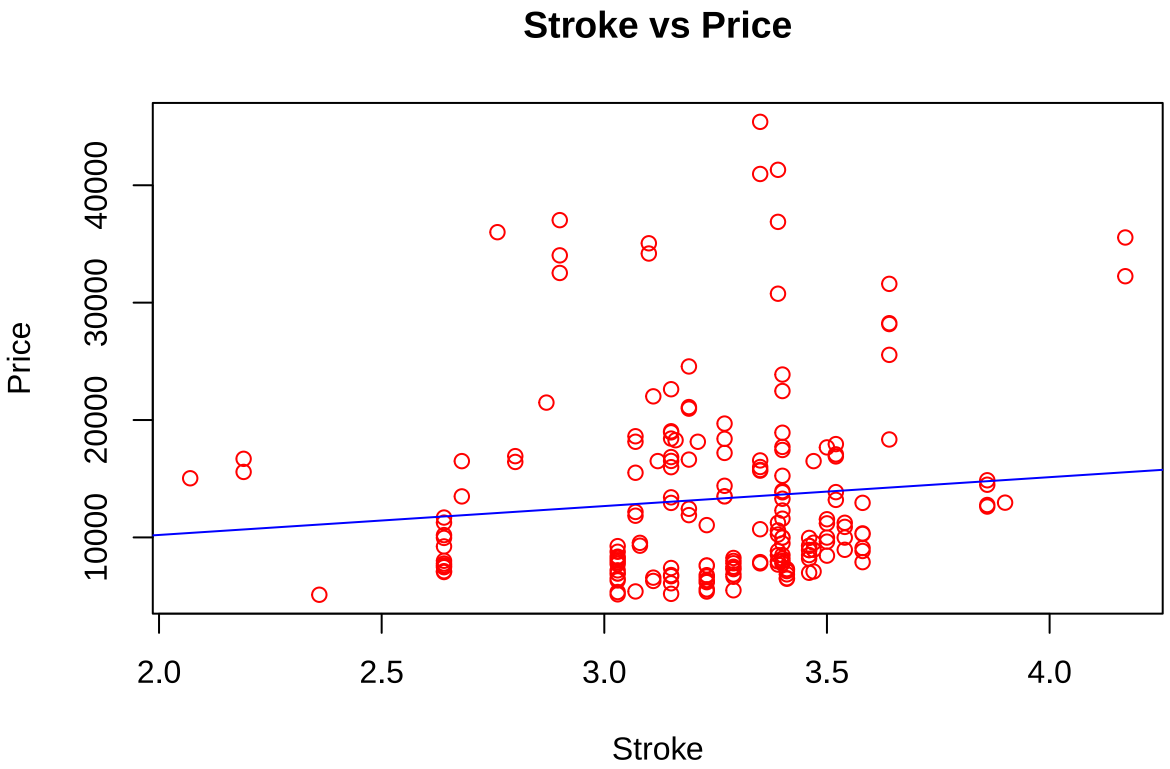
- Peak rpm does not seem like a good predictor of the price at all since the regression line is close to horizontal. Also, the data points are very scattered and far from the fitted line, showing lots of variability. Therefore it's it is not a reliable variable.

```
CorPeak_rpm.P <- cor(autoMobile$peak_rpm, autoMobile$price)
CorPeak_rpm.P
```

```
## [1] -0.1038353
```

- We can examine the correlation between 'peak-rpm' and 'price' and see it's approximately: **-0.1719161**

```


plot(autoMobile$stroke, autoMobile$price, main = "Stroke vs Price",
     xlab = "Stroke", ylab = "Price", col = "red")

lm_fit <- lm(price ~ stroke, data = autoMobile)
abline(lm_fit, col = "blue")
```



Stroke vs Price

```
CorStroke.P <- cor(autoMobile$stroke, autoMobile$price)
CorStroke.P
```

```
## [1] 0.09600668
```

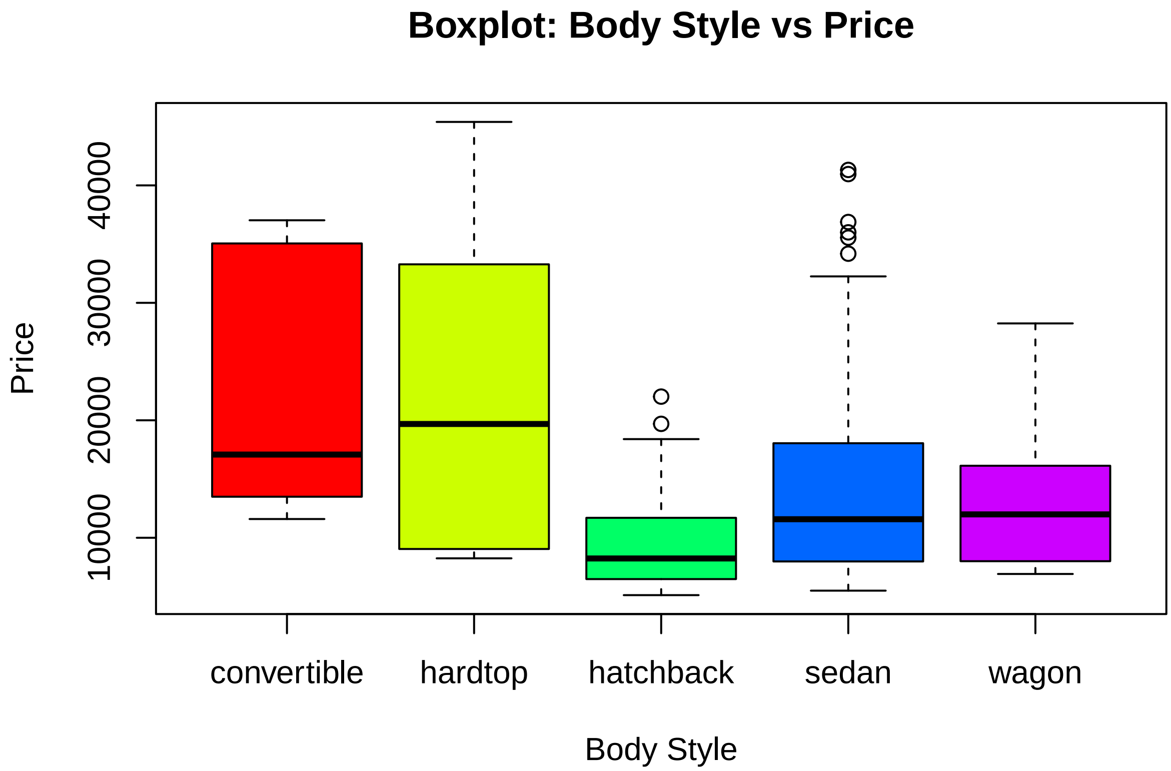
- We can examine the correlation between 'stroke' and 'price' and see it's approximately: **0.09600668**

Categorical Variables

- These are variables that describe a 'characteristic' of a data unit, and are selected from a small group of categories. The categorical variables can have the type "char" or "fact".
- A good way to visualize categorical variables is by using box plots.

1. Relationship between **"body-style"** and **"price"**

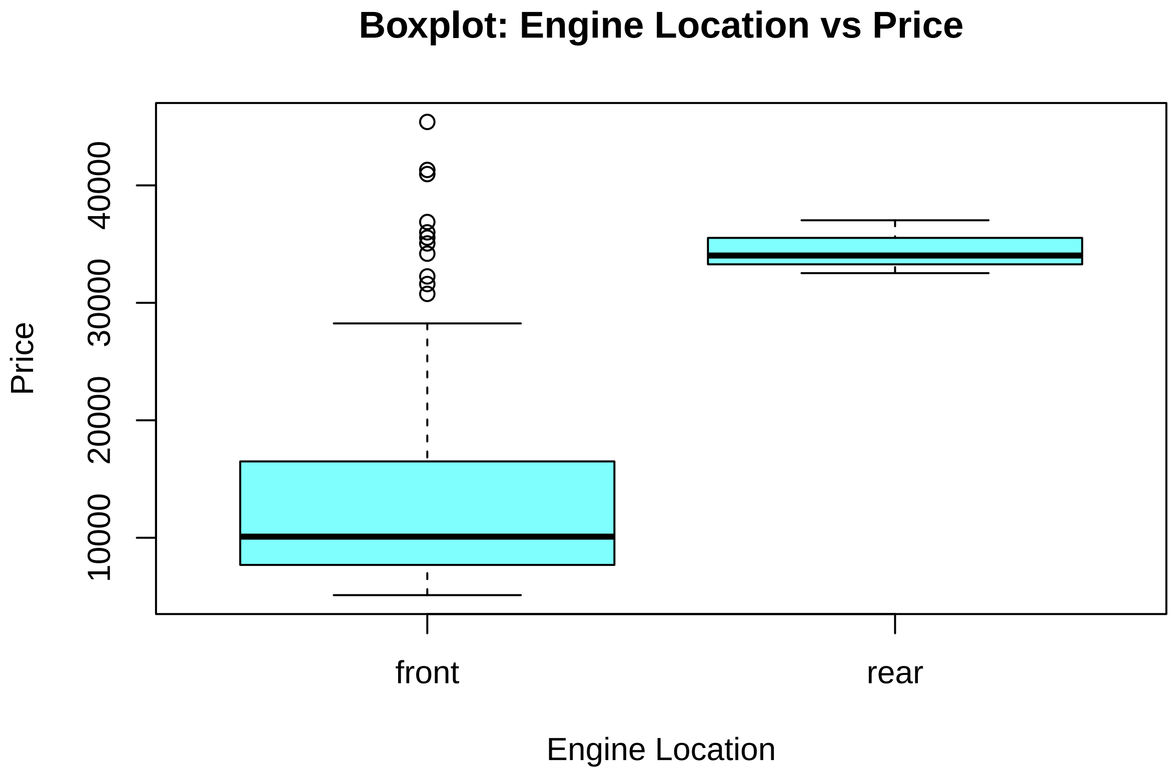
```
boxplot(price ~ body_style, data = autoMobile, col = rainbow(5),
        main = "Boxplot: Body Style vs Price",
        xlab = "Body Style", ylab = "Price")
```



- We see that the distributions of price between the different body-style categories have a significant overlap, and so body-style would not be a good predictor of price.

2. Relationship between **"engine-location"** and **"price"**

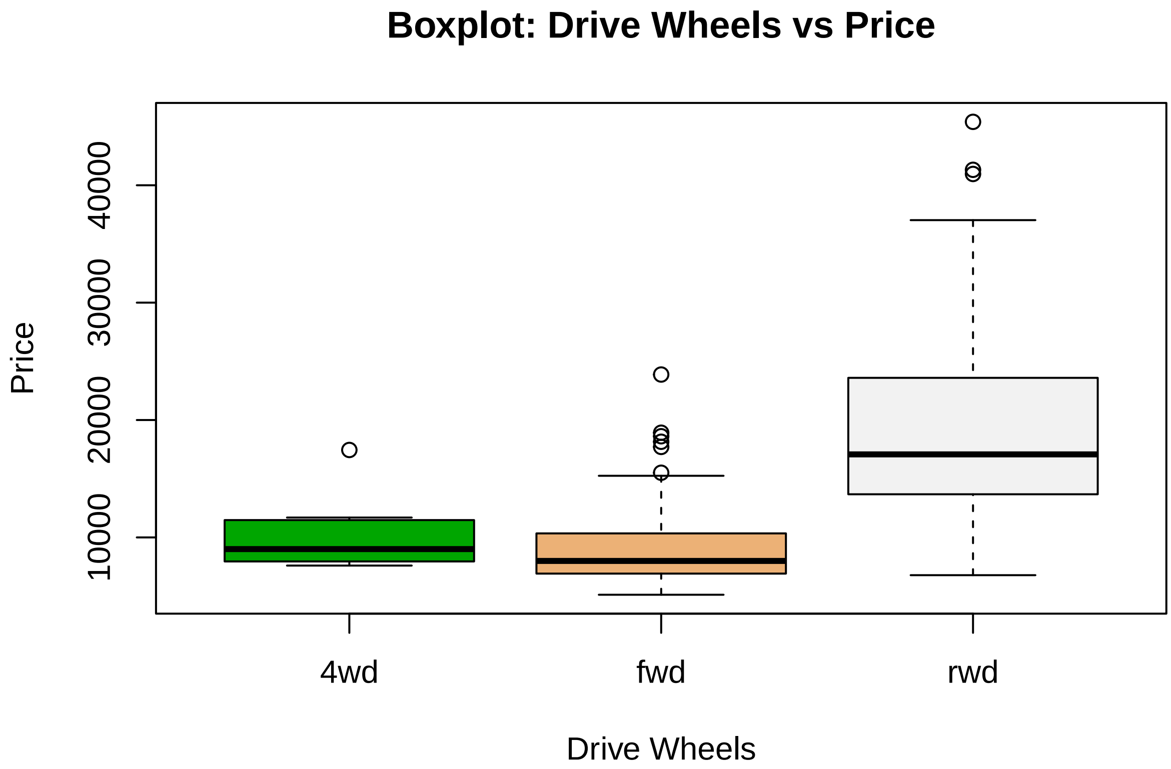
```
boxplot(price ~ engine_location, data = autoMobile, col = cm.colors(1),
        main = "Boxplot: Engine Location vs Price",
        xlab = "Engine Location", ylab = "Price")
```



- Here we see that the distribution of price between these two engine-location categories, front and rear, are distinct enough to take engine-location as a potential good predictor of price.

3.Relationship between **"drive-wheels"** and **"price"**.

```
boxplot(price ~ drive_wheels, data = autoMobile, col = terrain.colors(3),
        main = "Boxplot: Drive Wheels vs Price",
        xlab = "Drive Wheels", ylab = "Price")
```



- Here we see that the distribution of price between the different drive-wheels categories differs; as such drive-wheels could potentially be a predictor of price.

Descriptive Statistical Analysis

1. The summary function automatically computes basic statistics for all continuous variables. Any NA values are automatically skipped in these statistics.

This will show:

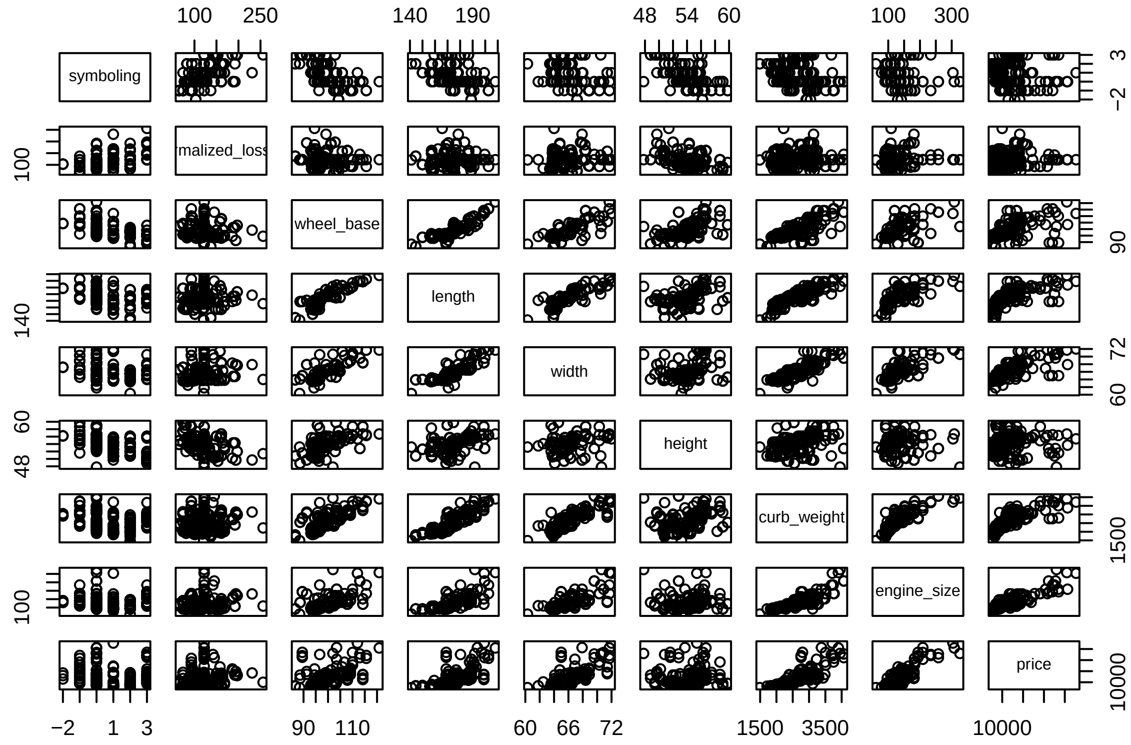
1. The count of that variable
2. The mean
3. The standard deviation (std)
4. The minimum value
5. The IQR (Interquartile Range: 25%, 50% and 75%)
6. The maximum value

```
summary(autoMobile)
```

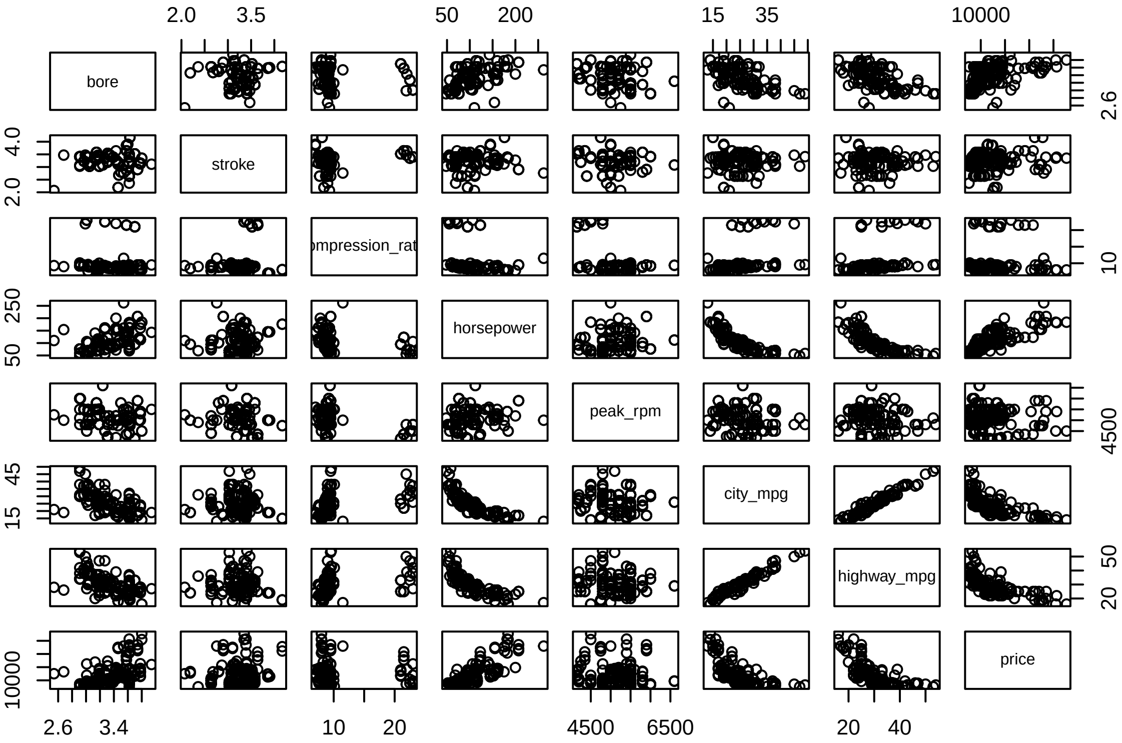
```
##    symboling       normalized_losses         make      fuel_type   aspiration
##  Min.   :-2.0000   Min.   : 65.0     toyota    :32   diesel: 19   std  :158
##  1st Qu.: 0.0000   1st Qu.: 95.0     nissan    :18   gas   :174   turbo: 35
##  Median : 1.0000   Median :122.0     honda     :13
##  Mean   : 0.7979   Mean   :121.3     mitsubishi:13
##  3rd Qu.: 2.0000   3rd Qu.:134.0     mazda     :12
##  Max.   : 3.0000   Max.   :256.0     subaru    :12
##                                      (Other)   :93
##  num_of_doors       body_style drive_wheels engine_location   wheel_base
##  four:112     convertible: 6   4wd:  8      front:190       Min.   : 86.60
##  two : 81     hardtop    : 8   fwd:114      rear :  3       1st Qu.: 94.50
##               hatchback  :63   rwd: 71                      Median : 97.00
##               sedan      :92                                Mean   : 98.92
##               wagon      :24                                3rd Qu.:102.40
##                                                             Max.   :120.90
##
##      length          width          height       curb_weight   engine_type
##  Min.   :141.1   Min.   :60.30   Min.   :47.80   Min.   :1488   dohc: 12
##  1st Qu.:166.3   1st Qu.:64.10   1st Qu.:52.00   1st Qu.:2145   l   : 12
##  Median :173.2   Median :65.40   Median :54.10   Median :2414   ohc :141
##  Mean   :174.3   Mean   :65.89   Mean   :53.87   Mean   :2562   ohcf: 15
##  3rd Qu.:184.6   3rd Qu.:66.90   3rd Qu.:55.70   3rd Qu.:2952   ohcv: 13
##  Max.   :208.1   Max.   :72.00   Max.   :59.80   Max.   :4066
##
##  num_of_cylinders  engine_size    fuel_system      bore          stroke
##  eight :  4       Min.   : 61.0   1bbl:11     Min.   :2.540   Min.   :2.070
##  five  : 10       1st Qu.: 98.0   2bbl:64     1st Qu.:3.150   1st Qu.:3.110
##  four  :153       Median :120.0   idi :19     Median :3.310   Median :3.290
##  six   : 24       Mean   :128.1   mfi : 1     Mean   :3.331   Mean   :3.249
##  three :  1       3rd Qu.:146.0   mpfi:88     3rd Qu.:3.590   3rd Qu.:3.410
##  twelve:  1       Max.   :326.0   spdi: 9     Max.   :3.940   Max.   :4.170
##                                   spfi: 1
##  compression_ratio   horsepower       peak_rpm       city_mpg
##  Min.   : 7.00     Min.   : 48.0   Min.   :4150   Min.   :13.00
##  1st Qu.: 8.50     1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00     Median : 95.0   Median :5100   Median :25.00
##  Mean   :10.14     Mean   :103.5   Mean   :5100   Mean   :25.33
##  3rd Qu.: 9.40     3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00     Max.   :262.0   Max.   :6600   Max.   :49.00
##
##   highway_mpg        price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7738
##  Median :30.00   Median :10245
##  Mean   :30.79   Mean   :13285
##  3rd Qu.:34.00   3rd Qu.:16515
##  Max.   :54.00   Max.   :45400
##
```

- To get a better measure of the important characteristics, we look at the correlation of these variables with the car price, in other words: how is the car price dependent on this variable?

```
df_01 <- autoMobile[, c(1, 2, 10, 11, 12, 13, 14, 17,  26)]
df_02 <- autoMobile[, c(19, 20, 21, 22, 23, 24, 25, 26)]

par(mfrow = c(1,2))

pairs(df_01)
```



```
pairs(df_02)
```



Correlation

- **Correlation:** a measure of the extent of interdependence between variables.

Pearson Correlation

- The Pearson Correlation measures the linear dependence between two variables X and Y.
- The resulting coefficient is a value between -1 and 1 inclusive, where:

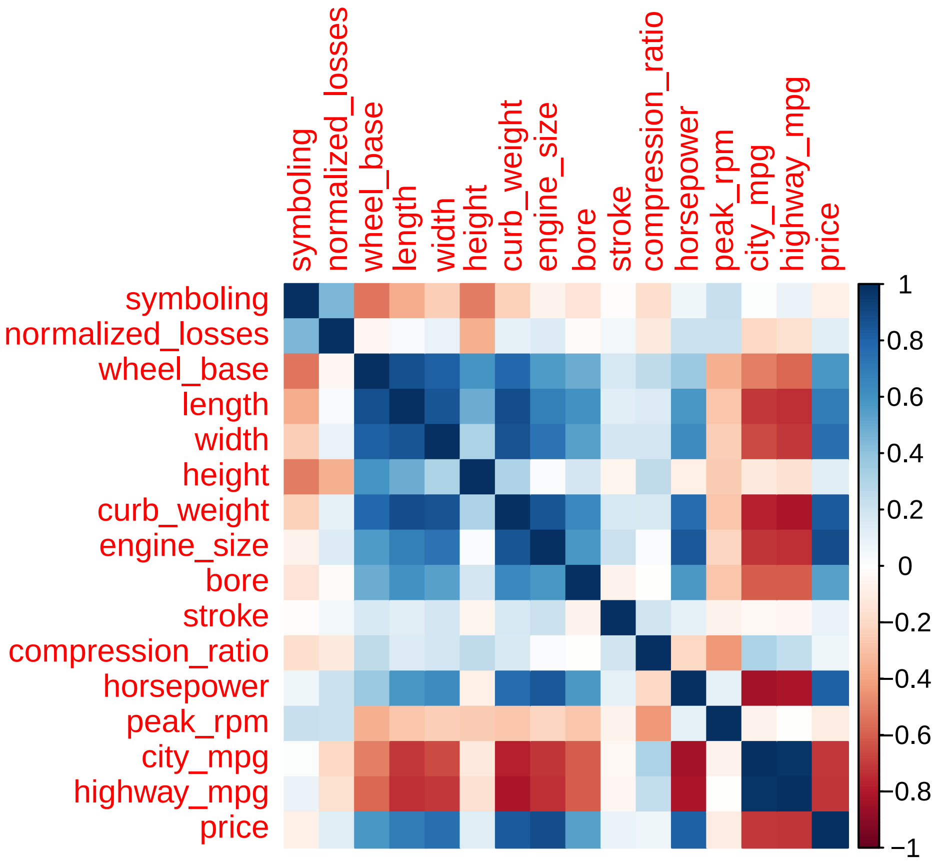
  a. Total positive linear correlation.

  b. No linear correlation, the two variables most likely do not affect each other.

  c. Total negative linear correlation.

- Calculate the correlation between variables of type “int” or “num” using the method “cor”:

```
# Subset the relevant columns for correlation calculation
correlation_matrix <- cor(autoMobile[, c(1, 2, 10, 11, 12, 13, 14, 17,
                                         19, 20, 21, 22, 23, 24, 25, 26)])

# Create the correlation color plot
corrplot(correlation_matrix, method = "color")
```



We can use the stats of this corr table data for creating a model.

- Sometimes we would like to know the significant of the correlation estimate.

P-value

- What is this P-value?

  The P-value is the probability value that the correlation between the two variables is statistically significant. Normally, we choose a significance level of 0.05, which means that we are 95% confident that the correlation between the variables is significant.

- By convention, when the

  a. p-value is < 0.001: We say there is strong evidence that the correlation is significant.

  b. the p-value is < 0.05: There is moderate evidence that the correlation is significant.

  c. the p-value is < 0.1: There is weak evidence that the correlation is significant.

  d. the p-value is < 0.1: There is no evidence that the correlation is significant.

1.Wheel_base vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_wb <- cor.test(autoMobile$wheel_base, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_wb <- result_wb$estimate
```

```r
p_value_wb <- result_wb$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_wb, "with a P-value of P =", p_value_
```

```
## [1] "The Pearson Correlation Coefficient is 0.584950622305816 with a P-value of P = 4.16429781569782e
```

Conclusion:

- Since the p-value is < 0.001, the correlation between wheel-base and price is statistically significant, although the linear relationship isn't extremely strong **(~ 0.585)**

2.Horsepower vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_hp <- cor.test(autoMobile$horsepower, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_hp <- result_hp$estimate
p_value_hp <- result_hp$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_hp, "with a P-value of P =", p_value_
```

```
## [1] "The Pearson Correlation Coefficient is 0.812453204601347 with a P-value of P = 1.24840733993129e
```

Conclusion:

- Since the p-value is < 0.001, the correlation between horsepower and price is statistically significant, and the linear relationship is quite strong (~0.809, close to 1)

3.Lenght vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_L <- cor.test(autoMobile$length, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_L <- result_L$estimate
p_value_L <- result_L$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_L, "with a P-value of P =", p_value_L
```

```
## [1] "The Pearson Correlation Coefficient is 0.695927914443572 with a P-value of P = 2.80926629910332e
```

Conclusion:

- Since the p-value is < 0.001, the correlation between length and price is statistically significant, and the linear relationship is moderately strong (~0.691).

4.Width vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_w <- cor.test(autoMobile$width, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_w <- result_w$estimate
p_value_w <- result_w$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_w, "with a P-value of P =", p_value_
```

```
## [1] "The Pearson Correlation Coefficient is 0.754648894838236 with a P-value of P = 8.44009950371111
```

Conclusion:

- Since the p-value is < 0.001, the correlation between width and price is statistically significant, and the linear relationship is quite strong (~0.751).

5.Curb_weight vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_cw <- cor.test(autoMobile$curb_weight, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_cw <- result_cw$estimate
p_value_cw <- result_cw$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_cw, "with a P-value of P =", p_value_
```

```
## [1] "The Pearson Correlation Coefficient is 0.835367753626223 with a P-value of P = 1.5875863033306e
```

Conclusion:

- Since the p-value is < 0.001, the correlation between curb-weight and price is statistically significant, and the linear relationship is quite strong (~0.834).

6.Engine_size vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_Es <- cor.test(autoMobile$engine_size, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_Es <- result_Es$estimate
p_value_Es <- result_Es$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_Es, "with a P-value of P =", p_value_
```

```
## [1] "The Pearson Correlation Coefficient is 0.888778495310582 with a P-value of P = 1.25250791781395
```

Conclusion:

- Since the p-value is < 0.001, the correlation between engine-size and price is statistically significant, and the linear relationship is very strong (~0.872).

7.Bore vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_B <- cor.test(autoMobile$bore, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_B <- result_B$estimate
p_value_B <- result_B$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_B, "with a P-value of P =", p_value_B
```

```
## [1] "The Pearson Correlation Coefficient is 0.546295274801749 with a P-value of P = 2.0776169810403e
```

Conclusion:

- Since the p-value is < 0.001, the correlation between bore and price is statistically significant, but the linear relationship is only moderate (~0.521).

8.City_mpg vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_Cm <- cor.test(autoMobile$city_mpg, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_Cm <- result_Cm$estimate
p_value_Cm <- result_Cm$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_Cm, "with a P-value of P =", p_value_
```

```
## [1] "The Pearson Correlation Coefficient is -0.706617993498795 with a P-value of P = 1.6533219288194
```

Conclusion:

- Since the p-value is < 0.001, the correlation between city-mpg and price is statistically significant, and the coefficient of ~ -0.687 shows that the relationship is negative and moderately strong.

9.Highway_mpg vs Price

```r
# Calculate the Pearson correlation coefficient and p-value
result_Hm <- cor.test(autoMobile$highway_mpg, autoMobile$price, method = "pearson")

# Extract the correlation coefficient and p-value
pearson_coef_Hm <- result_Hm$estimate
p_value_Hm <- result_Hm$p.value

# Print the results
print(paste("The Pearson Correlation Coefficient is", pearson_coef_Hm, "with a P-value of P =", p_value_
## [1] "The Pearson Correlation Coefficient is -0.719177688383088 with a P-value of P = 5.01527492738632
```

Conclusion:

- Since the p-value is < 0.001, the correlation between highway-mpg and price is statistically significant, and the coefficient of ~ -0.705 shows that the relationship is negative and moderately strong.

ANOVA

ANOVA: Analysis of Variance The Analysis of Variance (ANOVA) is a statistical method used to test whether there are significant differences between the means of two or more groups. ANOVA returns two parameters:

1. **F-test score:**

   ANOVA assumes the means of all groups are the same, calculates how much the actual means deviate from the assumption, and reports it as the F-test score.

   A larger score means there is a larger difference between the means.

2. **P-value:**
   P-value tells how statistically significant is our calculated score value.

- If our price variable is strongly correlated with the variable we are analyzing, expect ANOVA to return a sizeable F-test score and a small p-value.

Drive Wheels

- Since ANOVA analyzes the difference between different groups of the same variable, the groupby function will come in handy. Because the ANOVA algorithm averages the data automatically, we do not need to take the average before hand.

- Let's see if different types **'drive-wheels'** impact **'price'**, we group the data.

```r
df_gptest <- autoMobile[,c("drive_wheels","price")]
head(df_gptest)
```

```
##   drive_wheels price
## 1          rwd 13495
## 2          rwd 16500
## 3          rwd 16500
## 4          fwd 13950
## 5          4wd 17450
## 6          fwd 15250
```

```r
grouped_test2 <- df_gptest %>%
  select("drive_wheels", "price") %>%
  group_by(drive_wheels)

head(grouped_test2)
```

```
## # A tibble: 6 x 2
## # Groups:   drive_wheels [3]
##   drive_wheels price
##   <fct>        <dbl>
## 1 rwd          13495
## 2 rwd          16500
## 3 rwd          16500
## 4 fwd          13950
## 5 4wd          17450
## 6 fwd          15250
```

```
# Extract the 'price' column of the '4wd' group
price_of_4wd_cars <- grouped_test2 %>%
  filter(drive_wheels == "4wd") %>%
  select(price)
```

```
## Adding missing grouping variables: `drive_wheels`
```

```
price_of_4wd_cars
```

```
## # A tibble: 8 x 2
## # Groups:   drive_wheels [1]
##   drive_wheels price
##   <fct>        <dbl>
## 1 4wd          17450
## 2 4wd           7603
## 3 4wd           9233
## 4 4wd          11259
## 5 4wd           8013
## 6 4wd          11694
## 7 4wd           7898
## 8 4wd           8778
```

ChiSquared Test Use to check the relation between two categorical variables.

```
test1 <- chisq.test(make, fuel_type)
```

```
## Warning in chisq.test(make, fuel_type): Chi-squared approximation may be
## incorrect
```

```
test1
```

```
##
##  Pearson's Chi-squared test
##
## data:  make and fuel_type
## X-squared = 49.043, df = 21, p-value = 0.000495
```

With a p-value of 0.000495, which is smaller than the typical significance level of 0.05, we have enough evidence to reject the null hypothesis. The null hypothesis states that there is no association between the variables make and fuel_type. Therefore, based on the chi-squared test results, we can conclude that there is a significant association between the make and fuel_type variables.

```
test2 <- chisq.test(engine_location, drive_wheels)
```

```
## Warning in chisq.test(engine_location, drive_wheels): Chi-squared approximation
## may be incorrect
```

```
test2
```

```
##
##  Pearson's Chi-squared test
##
## data:  engine_location and drive_wheels
## X-squared = 5.1677, df = 2, p-value = 0.07548
```

With a p-value of 0.07548, which is greater than the typical significance level of 0.05, we do not have enough evidence to reject the null hypothesis. The null hypothesis states that there is no association between the variables engine_location and drive_wheels. Therefore, based on the chi-squared test results, we cannot conclude that there is a significant association between the engine_location and drive_wheels variables.

```
test3 <- chisq.test(engine_type, aspiration)
```

```
## Warning in chisq.test(engine_type, aspiration): Chi-squared approximation may
## be incorrect
```

```
test3
```

```
##
##  Pearson's Chi-squared test
##
## data:  engine_type and aspiration
## X-squared = 10.59, df = 6, p-value = 0.1019
```

With a p-value of 0.1019, which is greater than the typical significance level of 0.05, we do not have enough evidence to reject the null hypothesis. The null hypothesis states that there is no association between the variables engine_type and aspiration Therefore, based on the chi-squared test results, we cannot conclude that there is a significant association between the engine_type and aspiration variables.

Conclusion: Important Variables

- We now have a better idea of what our data looks like and which variables are important to take into account when predicting the car price. We have narrowed it down to the following variables:

A.Continuous numerical variables:

- length
- width
- curb_weight
- engine_size

- horsepower
- city_mpg
- highway_mpg
- wheel_base
- bore

B.Categorical variables:

- drive-wheels

MODEL BUILDING

As we now move into building models to our analysis, feeding the model with variables that meaningfully affect our target variable will improve our model's prediction performance.

Linear Regression and Multiple Linear Regression

Linear Regression One example of a data model that we will be using is

a. Simple Linear regression:

Simple linear regression is a method to help us understand the relationship between two variables:

i.The Predictor/independent variable(X)

ii. The response/dependent variable (that we want to predict)(Y)

The result of Linear Regression is a linear function that predicts the response (dependent) variable as a function of the predictor (independent) variable.

Linear function: Y(hat) = a + b*X

- a = refers to the intercept of the regression line.
- b = refers to the slope of the regression line.

b.Multiple Linear Regression

This method is used to explain the relationship between one continuous response (dependent) variable and two or more predictor (independent) variables. Most of the real-world regression models involve multiple predictors.

Y : Response Variable

X1: Predictor Variable 1

X2: Predictor Variable 2

X3: Predictor Variable 3

X4: Predictor Variable 4

a : intercept

b1: coefficients of Variable 1

b2: coefficients of Variable 2

b3: coefficients of Variable 3

b4: coefficients of Variable 4

Y(hat) = a + b1.X1 + b2.X2 + b3.X3 + b4.X4

- From the previous section we know that other good predictors of price could be: length, width, curb_weight, engine_size, horsepower, city_mpg, highway_mpg, wheel_base, bore, drive_wheels.

Let's develop a model using these variables as the predictor variables.

FULL MODEL

```
F_model <- lm(price ~ ., data = autoMobile)
summary(F_model)
```

```
##
## Call:
## lm(formula = price ~ ., data = autoMobile)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3497.2  -976.4     0.0   871.7  7632.3
##
## Coefficients: (3 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.128e+03  1.830e+04  -0.116 0.907625
## symboling              -4.280e+02  2.636e+02  -1.624 0.106811
## normalized_losses      -1.144e+01  7.702e+00  -1.485 0.139918
## makeaudi                3.196e+03  2.591e+03   1.234 0.219439
## makebmw                 6.597e+03  2.463e+03   2.678 0.008317 **
## makechevrolet          -4.721e+03  2.293e+03  -2.059 0.041431 *
## makedodge              -4.901e+03  1.963e+03  -2.497 0.013714 *
## makehonda              -2.176e+03  2.232e+03  -0.975 0.331309
## makeisuzu              -3.783e+03  2.473e+03  -1.530 0.128426
## makejaguar             -1.376e+03  2.820e+03  -0.488 0.626491
## makemazda              -1.695e+03  1.805e+03  -0.939 0.349198
## makemercedes-benz       2.535e+03  2.545e+03   0.996 0.320989
## makemercury            -3.250e+03  2.996e+03  -1.084 0.280084
## makemitsubishi         -5.013e+03  2.011e+03  -2.493 0.013879 *
## makenissan             -1.936e+03  1.831e+03  -1.058 0.292143
## makepeugot             -8.116e+03  4.486e+03  -1.809 0.072663 .
## makeplymouth           -4.959e+03  1.931e+03  -2.568 0.011317 *
## makeporsche             4.610e+03  3.074e+03   1.500 0.136043
## makesaab                3.066e+03  2.297e+03   1.335 0.184173
## makesubaru             -1.762e+03  1.962e+03  -0.898 0.370819
## maketoyota             -3.076e+03  1.630e+03  -1.888 0.061221 .
## makevolkswagen         -6.980e+02  2.002e+03  -0.349 0.727931
## makevolvo              -2.107e+03  2.219e+03  -0.950 0.344028
## fuel_typegas           -1.356e+04  6.747e+03  -2.010 0.046464 *
## aspirationturbo         2.055e+03  8.241e+02   2.493 0.013865 *
## num_of_doorstwo         9.464e+01  5.085e+02   0.186 0.852629
## body_stylehardtop      -2.156e+03  1.187e+03  -1.816 0.071540 .
## body_stylehatchback    -3.012e+03  1.106e+03  -2.725 0.007288 **
## body_stylesedan        -2.489e+03  1.205e+03  -2.065 0.040847 *
## body_stylewagon        -2.744e+03  1.297e+03  -2.115 0.036257 *
## drive_wheelsfwd        -7.346e+02  9.323e+02  -0.788 0.432102
## drive_wheelsrwd         4.100e+02  1.268e+03   0.323 0.746859
## engine_locationrear     9.618e+03  2.693e+03   3.572 0.000492 ***
## wheel_base              2.457e+02  9.368e+01   2.623 0.009713 **
## length                 -1.399e+02  5.077e+01  -2.756 0.006664 **
## width                   5.986e+02  2.282e+02   2.623 0.009718 **
## height                 -4.256e+02  1.510e+02  -2.818 0.005560 **
## curb_weight             6.513e+00  1.687e+00   3.860 0.000175 ***
## engine_typel            3.796e+03  4.187e+03   0.907 0.366161
## engine_typeohc          6.195e+02  1.228e+03   0.504 0.614748
## engine_typeohcf                NA         NA      NA       NA
## engine_typeohcv        -2.610e+03  1.240e+03  -2.105 0.037144 *
## num_of_cylindersfive   -6.021e+03  2.850e+03  -2.113 0.036466 *
## num_of_cylindersfour   -2.803e+03  3.533e+03  -0.793 0.428885
## num_of_cylinderssix    -3.443e+03  2.686e+03  -1.282 0.202168
## num_of_cylindersthree          NA         NA      NA       NA
## num_of_cylinderstwelve -4.791e+03  5.243e+03  -0.914 0.362435
## engine_size             9.427e+01  2.550e+01   3.697 0.000316 ***
## fuel_system2bbl         2.527e+03  1.483e+03   1.704 0.090659 .
## fuel_systemidi                 NA         NA      NA       NA
## fuel_systemmfi          4.385e+01  2.689e+03   0.016 0.987015
## fuel_systemmpfi         1.278e+03  1.566e+03   0.816 0.415977
## fuel_systemspdi         1.642e-01  1.855e+03   0.000 0.999930
## fuel_systemspfi         2.308e+03  3.063e+03   0.754 0.452350
## bore                   -3.769e+03  1.861e+03  -2.025 0.044834 *
## stroke                 -1.180e+03  9.943e+02  -1.187 0.237397
## compression_ratio      -9.400e+02  5.013e+02  -1.875 0.062929 .
## horsepower             -1.740e+00  2.508e+01  -0.069 0.944798
## peak_rpm                2.382e+00  6.488e-01   3.671 0.000347 ***
## city_mpg               -3.915e+01  1.337e+02  -0.293 0.770183
## highway_mpg             1.414e+02  1.145e+02   1.234 0.219176
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1735 on 135 degrees of freedom
## Multiple R-squared:  0.9677, Adjusted R-squared:  0.954
## F-statistic: 70.84 on 57 and 135 DF,  p-value: < 2.2e-16
```

- Considering the NA values we reduce some variables.

Then,

```
Full_Model <- lm(price ~ peak_rpm + bore + engine_size + curb_weight + engine_location +
                      length + height + width + wheel_base + engine_location + body_style +
                      aspiration + fuel_type + symboling + normalized_losses,
                    data = autoMobile)

summary(Full_Model)
```

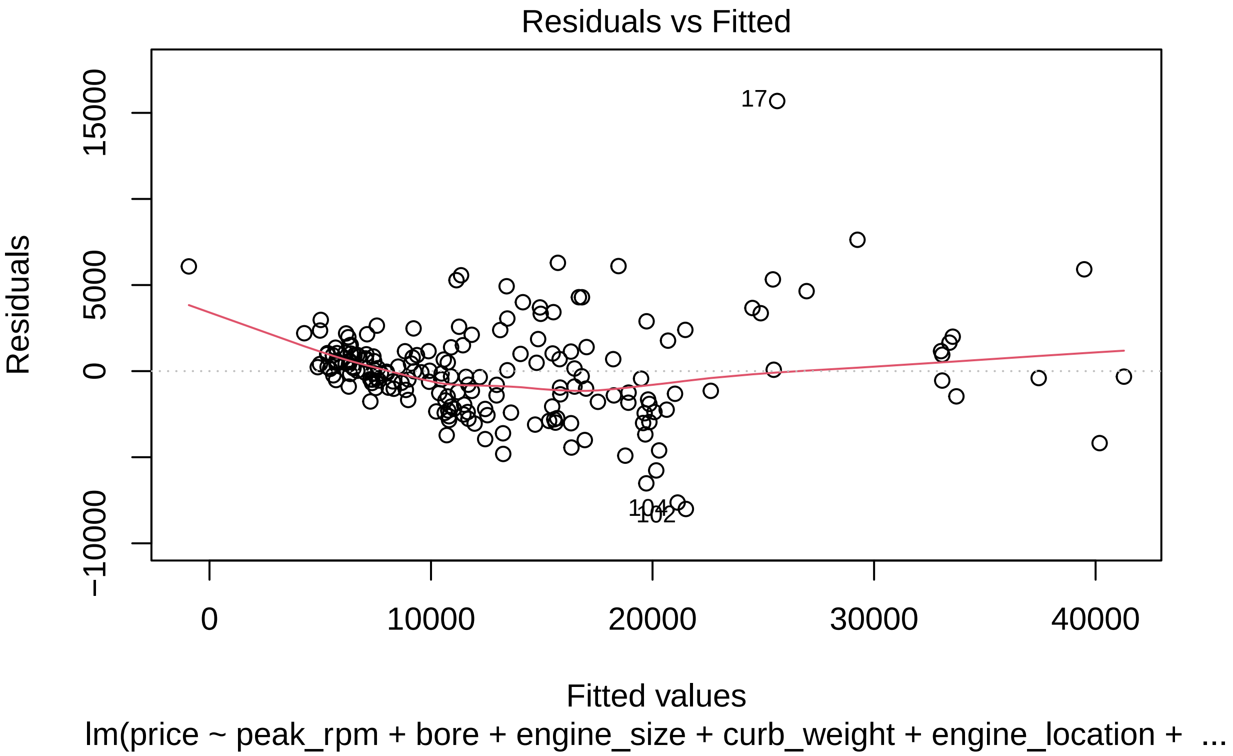
```
##
## Call:
## lm(formula = price ~ peak_rpm + bore + engine_size + curb_weight +
##     engine_location + length + height + width + wheel_base +
##     engine_location + body_style + aspiration + fuel_type + symboling +
##     normalized_losses, data = autoMobile)
##
## Residuals:
```

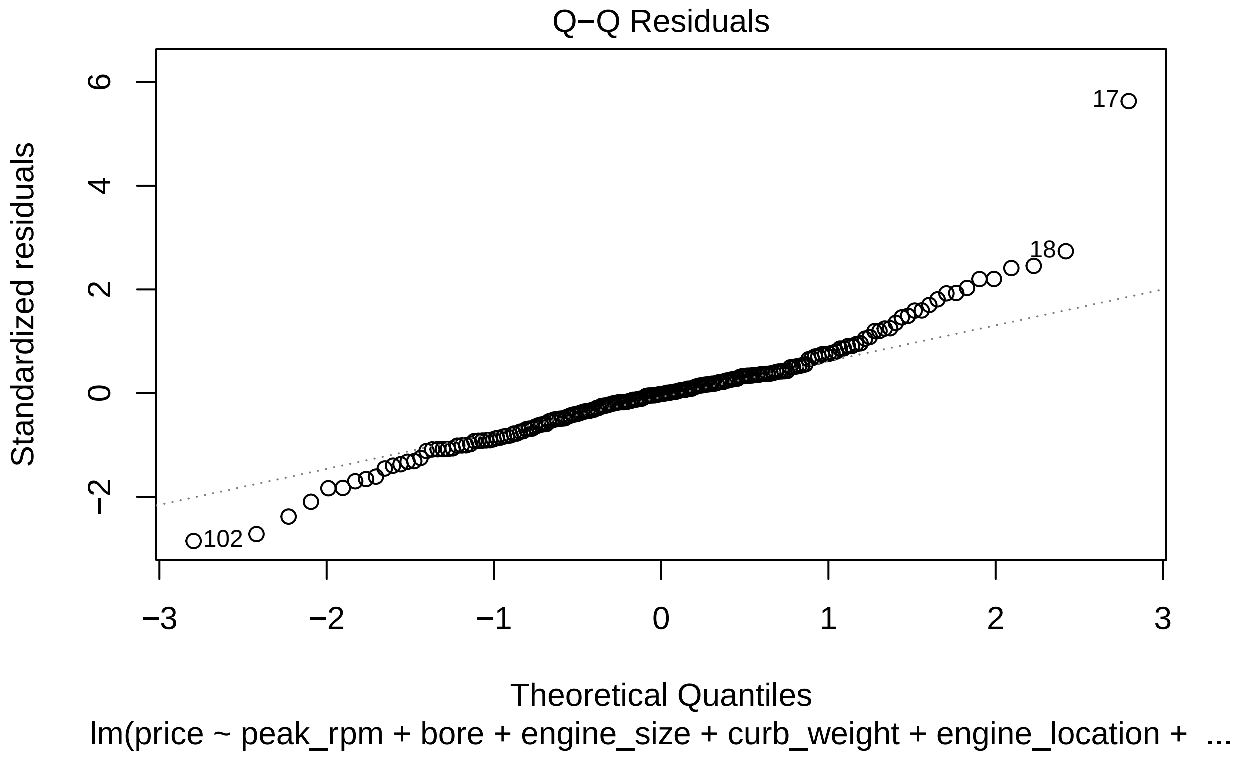
```
##     Min      1Q  Median      3Q     Max 
## -8005.5 -1473.1   -44.6  1046.8 15687.8 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         -6.964e+04  1.381e+04  -5.044 1.13e-06 ***
## peak_rpm             1.486e+00  5.891e-01   2.523 0.012533 *  
## bore                -1.718e+03  1.118e+03  -1.537 0.126176    
## engine_size          1.166e+02  1.406e+01   8.291 2.87e-14 ***
## curb_weight          2.523e+00  1.627e+00   1.551 0.122803    
## engine_locationrear  1.343e+04  2.163e+03   6.211 3.72e-09 ***
## length              -4.800e+01  5.552e+01  -0.865 0.388440    
## height               3.175e+02  1.495e+02   2.124 0.035050 *  
## width                6.810e+02  2.442e+02   2.789 0.005869 ** 
## wheel_base           9.128e+01  1.028e+02   0.888 0.375790    
## body_stylehardtop   -4.244e+03  1.676e+03  -2.532 0.012211 *  
## body_stylehatchback -4.984e+03  1.398e+03  -3.566 0.000468 ***
## body_stylesedan     -4.082e+03  1.488e+03  -2.743 0.006727 ** 
## body_stylewagon     -5.961e+03  1.612e+03  -3.698 0.000291 ***
## aspirationturbo      1.199e+03  7.192e+02   1.666 0.097408 .  
## fuel_typegas         3.207e+02  9.600e+02   0.334 0.738718    
## symboling           -1.039e+02  2.777e+02  -0.374 0.708742    
## normalized_losses    7.238e+00  8.543e+00   0.847 0.398001    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 2877 on 175 degrees of freedom
## Multiple R-squared:  0.8847, Adjusted R-squared:  0.8735 
## F-statistic: 78.97 on 17 and 175 DF,  p-value: < 2.2e-16
```

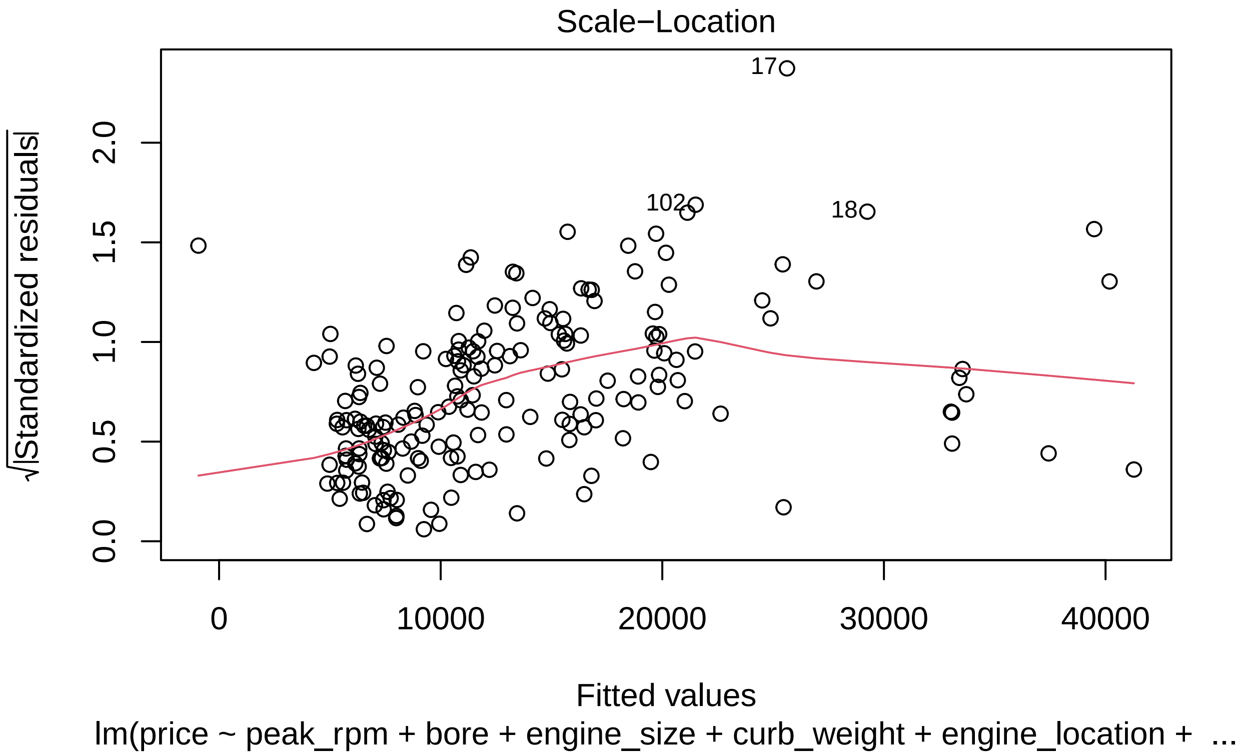
The overall model shows a good fit with an adjusted R-squared of 0.8735, indicating that around 87.35% of the variation in the price can be explained by the included predictor variables.

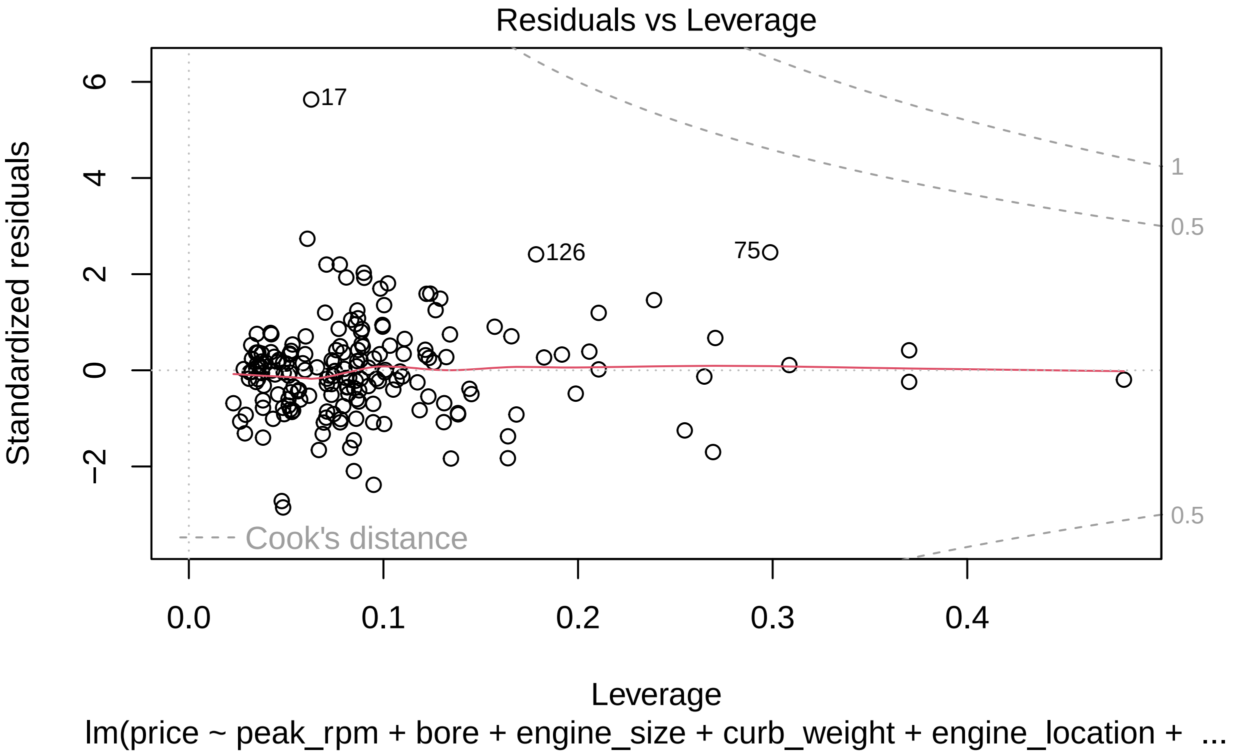
Multicolinearity by the plot

```r
#par(mfrow = c(2,2))
plot(Full_Model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(price ~ peak_rpm + bore + engine_size + curb_weight + engine_location + ...
15000
5000
0
−10000
0
10000
20000
30000
40000
17
104
102


Q−Q Residuals
Standardized residuals
6
4
2
0
−2
17
18
102
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ peak_rpm + bore + engine_size + curb_weight + engine_location + ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
10000
20000
30000
40000
Fitted values
lm(price ~ peak_rpm + bore + engine_size + curb_weight + engine_location + ...
17
102
18



1. Residuals vs fitted and residuals A random scatter of points around the horizontal line at 0 suggests that the model has met the assumption. If there's a pattern or funnel shape, it indicates potential heteroscedasticity. **Heteroscedasticity**

2. Q-Q residuals The points should closely follow the straight line, suggesting that the residuals are approximately normally distributed. Departure from the straight line indicates non-normality of residuals. **Non-normality**

3. Cook's Distance Plot: This plot identifies influential observations that may significantly affect the model. Large Cook's distance values suggest potential outliers or observations that significantly impact the regression coefficients. **There are some extreme values**

4. Influence Plot: This plot helps identify influential observations in terms of their leverage and residuals. Observations outside the dashed lines have high leverage and may affect the regression coefficients. **No leverage points**

To address issues like heteroscedasticity, non-normality, and extreme values in our regression analysis, it is essential to consider building a reduced model. A reduced model involves selecting a subset of the most relevant and least collinear predictor variables. By doing so, we can simplify the model and potentially improve its stability and interpretability. The reduced model can help mitigate the impact of outliers and non-normality by focusing on the most influential predictors. Additionally, it may reduce multicollinearity, leading to more reliable coefficient estimates. Overall, a reduced model provides a practical approach to tackle these challenges and enhance the quality of our regression analysis.

REDUCED MODEL

```
Reduced_Model <- lm(price ~ peak_rpm + engine_size + curb_weight + engine_location + width + engine_loca
                    data = autoMobile)

summary(Reduced_Model)
```

```
##
## Call:
## lm(formula = price ~ peak_rpm + engine_size + curb_weight + engine_location +
##     width + engine_location, data = autoMobile)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7325.2 -1630.4   -73.2  1311.8 15315.1
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -6.810e+04  1.217e+04  -5.598 7.65e-08 ***
## peak_rpm             1.327e+00  4.971e-01   2.670 0.008253 **
## engine_size          1.019e+02  1.066e+01   9.562  < 2e-16 ***
## curb_weight          3.268e+00  1.092e+00   2.993 0.003135 **
## engine_locationrear  1.372e+04  1.933e+03   7.094 2.60e-11 ***
## width                8.039e+02  2.085e+02   3.855 0.000159 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3007 on 187 degrees of freedom
## Multiple R-squared:  0.8654, Adjusted R-squared:  0.8618
## F-statistic: 240.4 on 5 and 187 DF,  p-value: < 2.2e-16
```

- The overall model shows a good fit with an adjusted R-squared of 0.8618, indicating that around 86.18% of the variation in the price can be explained by the included predictor variables.

Multiple Linear Function Price = -0.0006810 + {(0.08039)x width} + {(5.283)x curb_weight} + {(0.01019)x engine_size} + {(0.0001372)x engine_location} + {(1.327)x peak_rpm}

Model Evaluation using Visualization

- To evaluate our models and to choose the best one? One way to do this is by using visualization.
- The variable "highway_mpg" has a stronger correlation with "price", it is approximate -0.72009010 compared to "peak_rpm" which is approximate -0.1719161

Residual Plot

- A good way to visualize the variance of the data is to use a residual plot.

Residual:

- The difference between the observed value (y) and the predicted value. It is the distance from the data point to the fitted regression line.

- Y(hat) is called the residual Residual plot:

  It is a graph that shows the residuals on the vertical y-axis and the independent variable on the horizontal x-axis. We should always look at the spread of the residuals.

If the points in a residual plot are randomly spread out around the x-axis, then a linear model is appropriate for the data (Randomly spread out residuals means that the variance is constant, and thus the linear model is a good fit for this data)

- We can see from this residual plot - residuals are not randomly spread around the x-axis,thus a non-linear model is more appropriate for this data.

Decision Making:

- Determining a Good Model Fit *Model with the higher R-squared value is a better fit for the data.
- Model with the smallest MSE value is a better fit for the data.

Multiple Linear Regression Visualizing a model for Multiple Linear Regression Distribution plot : Compare the distribution of the fitted values that result from the model and distribution of the actual values.

The following assumptions should be satisfied by a Linear Regression model. i. x and y should have a linear relationship. - The 1st assumption should be checked before fitting the regression model. - Identify the independent variable and the dependent variable - For a simple linear regression, R is the square of the Pearson correlation coefficient.It ranges from 0 to 1. A large value of R indicates a better fit.

ii. Residuals are normally distributed.

- Residuals are normally distributed.
- using shapiro.Test If $p < 0.05$ we can say that residuals do not follow a normal distribution.

iii. Residuals have a zero mean.

- significant value is 0. randomly distributed.

iv. Residuals have a constant variance.

- randomly distributed plot means the constant variance

v. Residuals are independently distributed.

- randomly distributed plot means the independent distributed

residuals

```r
residuals_RM <- Reduced_Model$residuals

head(residuals_RM, 10)
```

```
##          1          2          3          4          5          6          7 
##  1850.5781  4855.5781   589.4250  2783.3634  1779.4487   695.9292 -2045.3613 
##          8          9         11 
## -1194.8884  3838.2000  5902.9372
```

2nd Assumption

```r
shapiro.test(residuals_RM)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  residuals_RM
## W = 0.95179, p-value = 4.094e-06
```

the test statistic (W) is 0.95179. The associated p-value is 4.094e-06, which is extremely small.

The null hypothesis for the Shapiro-Wilk test assumes that the residuals are normally distributed. In this case, since the p-value is significantly smaller than the conventional significance level of 0.05, there is strong evidence to reject the null hypothesis. This suggests that the residuals are not normally distributed

3rd Assumption

```r
mean(residuals_RM)
```

```
## [1] 6.669392e-14
```

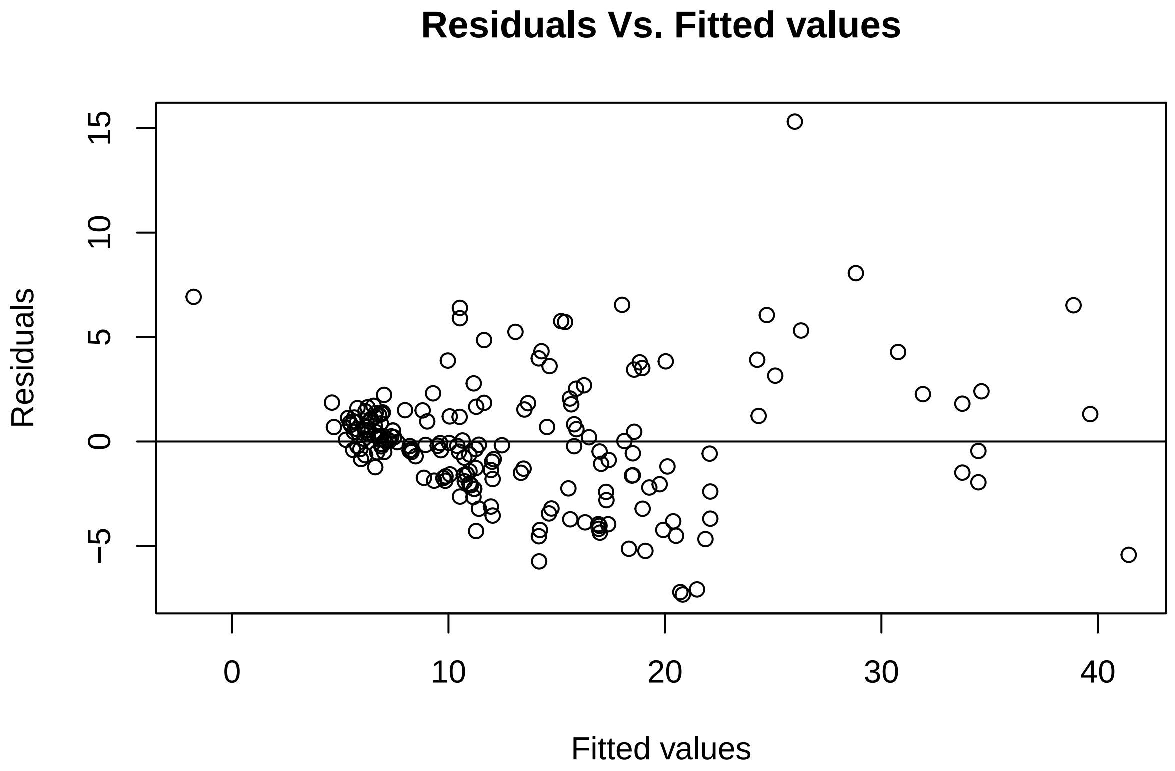
- mean residuals nearly goes to zero. therefore we can take this as mean val = 0.

4th Assumption

```r
predictor_RM <- Reduced_Model$fitted.values

head(predictor_RM)
```

```
##        1        2        3        4        5        6 
## 11644.42 11644.42 15910.57 11166.64 15670.55 14554.07
```

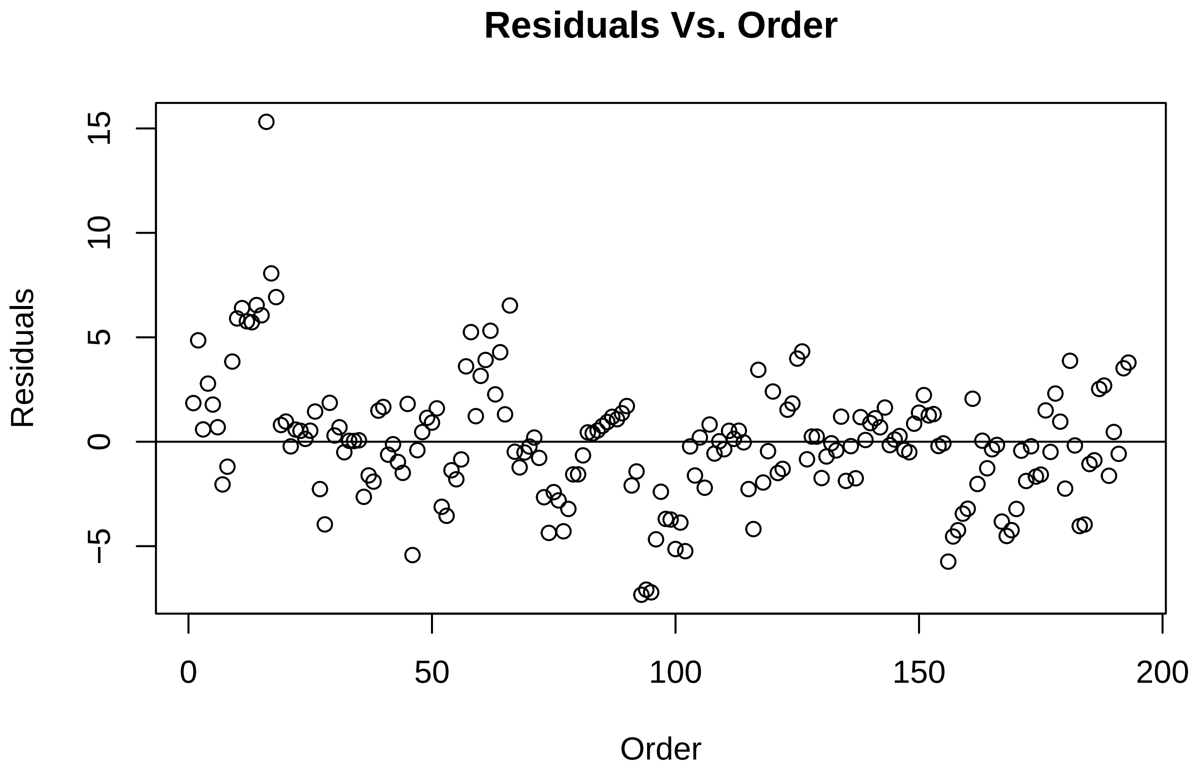
```r
plot(predictor_RM/1000, residuals_RM/1000, main = "Residuals Vs. Fitted values", xlab = "Fitted values"
abline(h=0)
```



A random scatter plot without any discernible pattern indicates that the model captures the underlying variability and randomness of the data. It suggests that the linear regression model is a reasonable fit for the data and that the assumptions of linearity and constant variance of residuals are met.

5th Assumption

```
#Residuals vs Order
plot(residuals_RM/1000, main = "Residuals Vs. Order", xlab = "Order", ylab = "Residuals")
abline(h=0)
```



Not randomly distributed, residuals are not independently distributed

Prediction Accuracy

MAE

```
mae = mean(abs(residuals_RM))
mae
```

```
## [1] 2112.683
```

These value should be close to zero. Then the difference between the of predictive value and actual value nearly zero.

RMSE

```
rmse = sqrt(mean(residuals_RM^2))
rmse
```

```
## [1] 2960.364
```

These value should be close to zero. Then the difference between the of predictive value and actual value nearly zero.

```
#library(MASS)
#library(car)

#lambda <- boxcox(Reduced_Model)$lambda
#lambda

#transformed_response <- powerTransform(price, lambda = lambda)

#lm_transformed <- lm(transformed_response ~ peak_rpm + engine_size + curb_weight + engine_location + w
```

CONCLUSION

- Comparing these, we conclude that the Reduced MLR model is the best model to be able to predict price from our data set. This result makes sense.