AS474\_Automobile\_Analysis

Group06

2023-07-19

#### Import the libraries

library(tidyverse)

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

library(dplyr)  
library(moments)  
library(repr)  
library(corrplot)

## corrplot 0.92 loaded

library(dplyr)  
library(purrr)

#### Loading the data set

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)

#### Displaying the first 6 rows of data

head(autoMobile)

## 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

#### Getting a description about the dataset

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> 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"…

sum(is.na(autoMobile))

## [1] 59

#### Check the missing values in each column

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

## 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

1. **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**

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.

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" ...

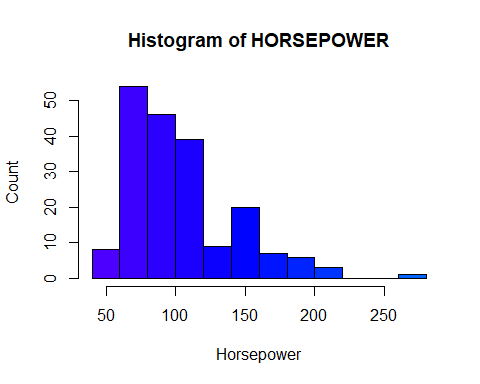
* Checking whether there is any duplicate value.

#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

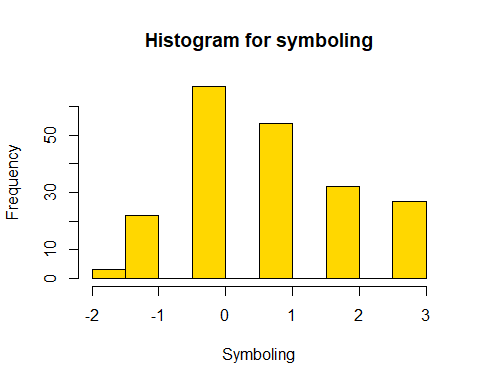
summary(autoMobile)

## 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.

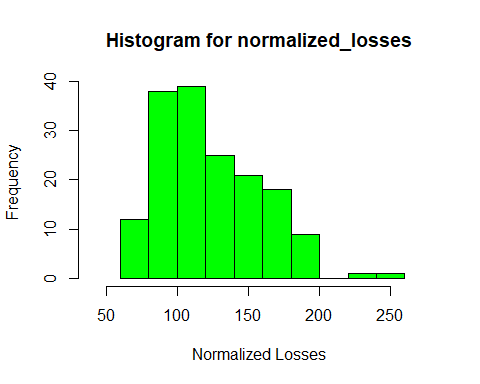
hist(symboling,main= "Histogram for symboling",xlab= "Symboling",ylab= "Frequency", col= "gold",border= "black")



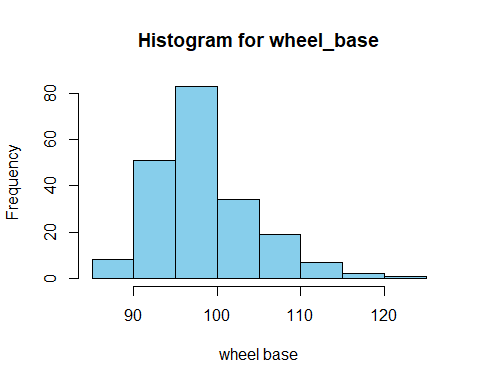
norm\_loss <- as.numeric(normalized\_losses)

## Warning: NAs introduced by coercion

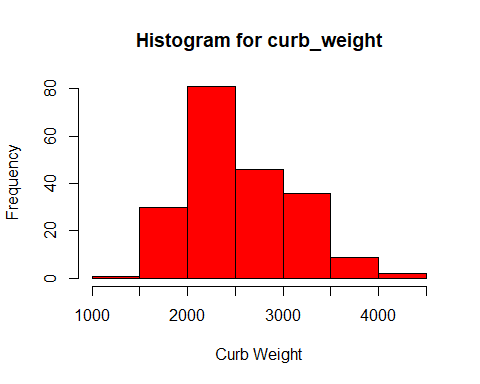
hist(norm\_loss, main= "Histogram for normalized\_losses",xlab= "Normalized Losses",xlim = c(40,275), ylab= "Frequency", col= "green",border= "black")



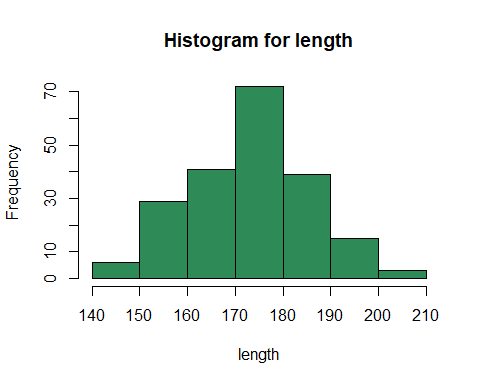
hist(wheel\_base, main= "Histogram for wheel\_base",xlab= "wheel base", ylab= "Frequency", col= "skyblue",border= "black")



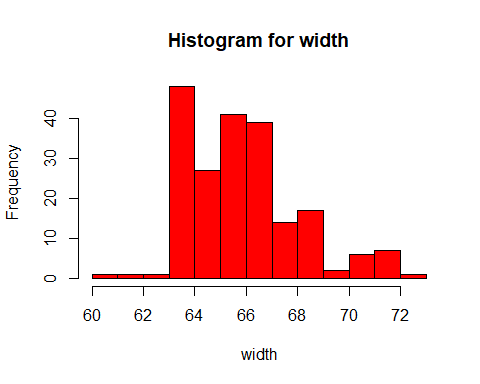
hist(curb\_weight,main= "Histogram for curb\_weight",xlab= "Curb Weight", ylab= "Frequency", col= rainbow(1),border= "black")



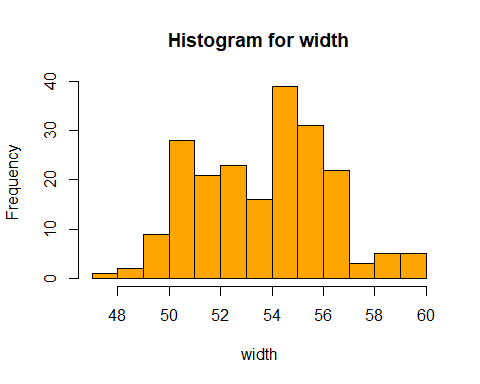
hist(length, main= "Histogram for length",xlab= "length", ylab= "Frequency", col= "seagreen",border= "black")



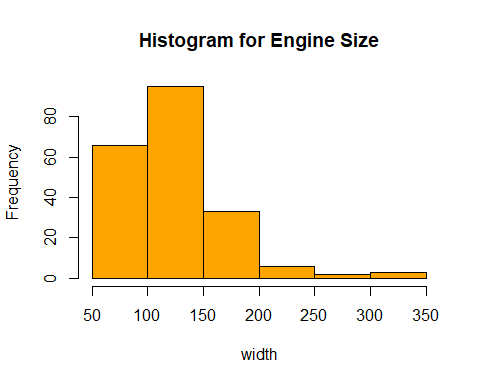
hist(width, main= "Histogram for width",xlab= "width", ylab= "Frequency", col= "red",border= "black")



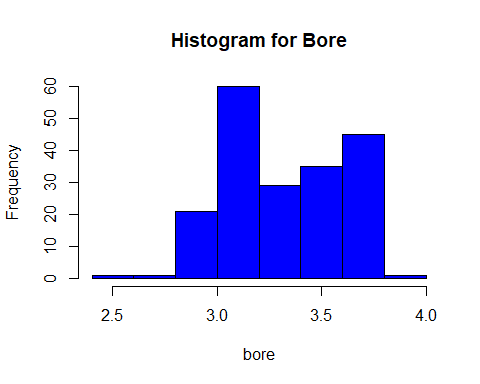
hist(height, main= "Histogram for width",xlab= "width", ylab= "Frequency", col= "orange",border= "black")



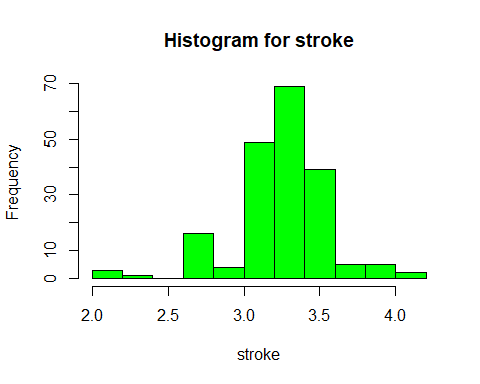
hist(engine\_size, main= "Histogram for Engine Size",xlab= "width", ylab= "Frequency", col= "orange",border= "black")



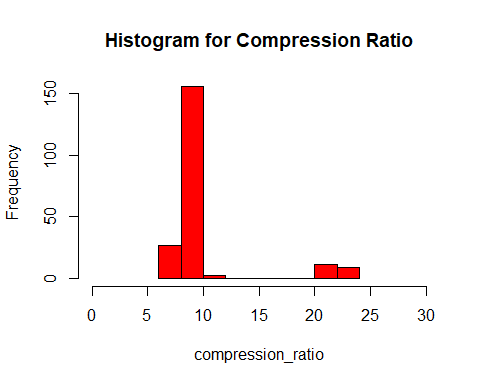
bore\_num <- as.numeric(autoMobile$bore)  
hist(bore\_num, main= "Histogram for Bore",xlab= "bore", ylab= "Frequency", col= "blue",border= "black")



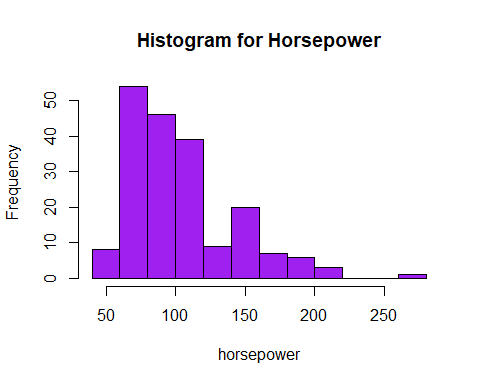
stroke\_num <- as.numeric(autoMobile$stroke)  
hist(stroke\_num, main= "Histogram for stroke",xlab= "stroke", ylab= "Frequency", col= "green",border= "black")



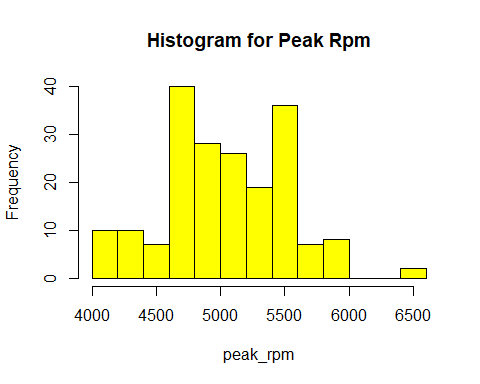
hist(compression\_ratio, main= "Histogram for Compression Ratio",xlab= "compression\_ratio", ylab= "Frequency",xlim = c(0, 30), col= "red",border= "black")



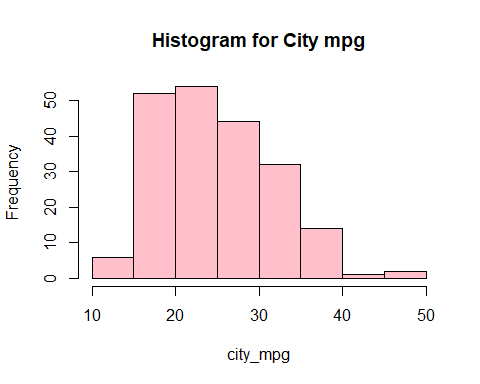
horsepower\_num <- as.numeric(autoMobile$horsepower)  
hist(horsepower\_num, main= "Histogram for Horsepower",xlab= "horsepower", ylab= "Frequency", col= "purple",border= "black")



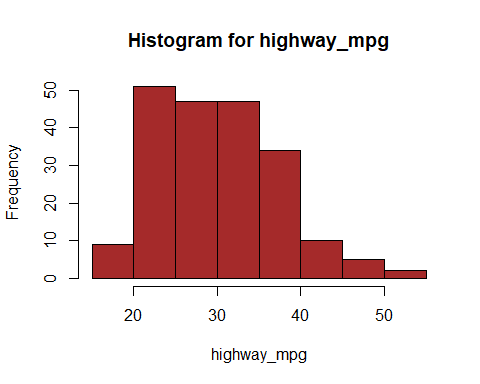
peak\_rpm\_num <- as.numeric(autoMobile$peak\_rpm)  
hist(peak\_rpm\_num, main= "Histogram for Peak Rpm",xlab= "peak\_rpm", ylab= "Frequency", col= "yellow",border= "black")



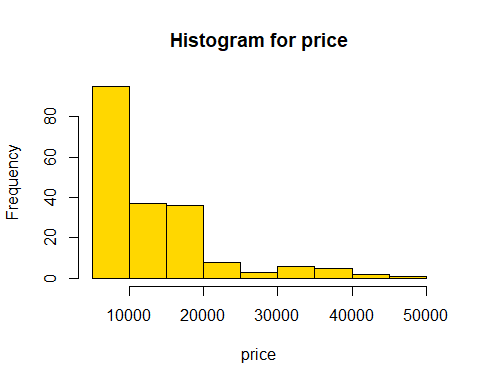
hist(city\_mpg, main= "Histogram for City mpg",xlab= "city\_mpg", ylab= "Frequency", col= "pink",border= "black")



hist(highway\_mpg, main= "Histogram for highway\_mpg",xlab= "highway\_mpg", ylab= "Frequency", col= "brown",border= "black")



price\_num <-as.numeric(autoMobile$price)  
hist(price\_num, main= "Histogram for price",xlab= "price", ylab= "Frequency", col= "gold",border= "black")

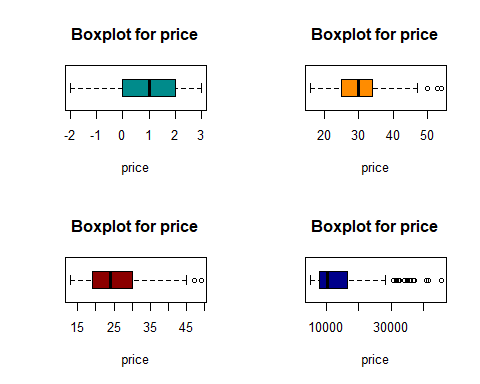


skewness\_automobile = c(skewness(autoMobile$highway\_mpg),  
 skewness(autoMobile$city\_mpg),  
 skewness(autoMobile$price),  
 skewness(autoMobile$peak\_rpm),  
 skewness(autoMobile$horsepower),  
 skewness(autoMobile$compression\_ratio),  
 skewness(autoMobile$bore),  
 skewness(autoMobile$stroke),  
 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

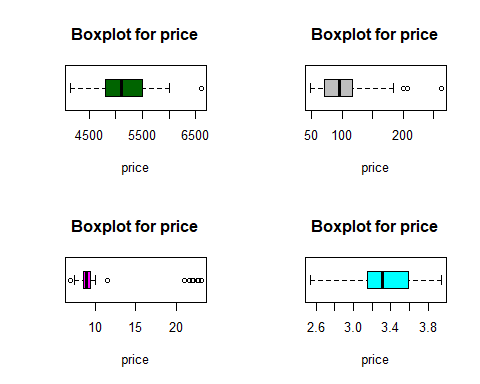
## skew.Names skewness\_automobile  
## 1 highway\_mpg 0.52925492  
## 2 city\_mpg 0.66937963  
## 3 price 1.74745011  
## 4 peak\_rpm 0.09637108  
## 5 horsepower 1.12744523  
## 6 compression\_ratio 2.58226566  
## 7 bore -0.02563592  
## 8 stroke -0.74139863  
## 9 engine\_size 1.99891286  
## 10 height 0.03465855  
## 11 width 0.85722465  
## 12 length 0.13679913  
## 13 wheel\_base 0.96786876  
## 14 curb\_weight 0.66042442

BOX PLOTS

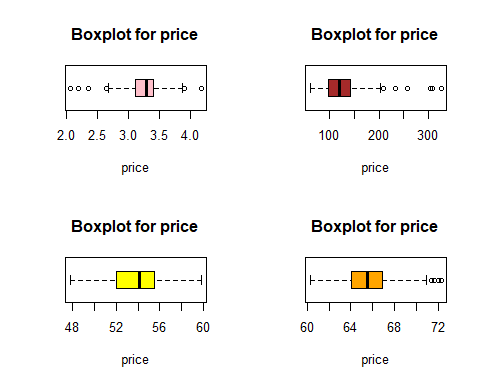
par(mfrow = c(2,2))  
  
boxplot(symboling, main= "Boxplot for price", xlab= "price", col= "darkcyan", horizontal = TRUE)  
boxplot(highway\_mpg, main= "Boxplot for price", xlab= "price", col= "darkorange", horizontal = TRUE)  
boxplot(city\_mpg, main= "Boxplot for price", xlab= "price", col= "darkred", horizontal = TRUE)  
boxplot(price\_num, main= "Boxplot for price", xlab= "price", col= "darkblue", horizontal = TRUE)



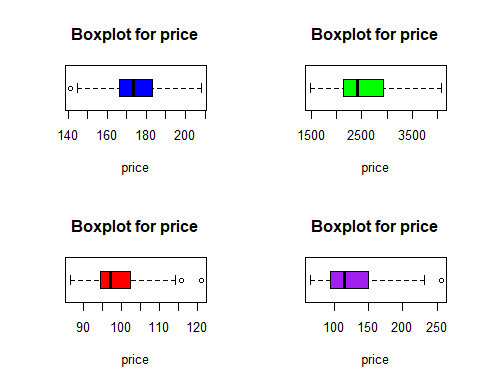
par(mfrow = c(2,2))  
boxplot(peak\_rpm\_num, main= "Boxplot for price", xlab= "price", col= "darkgreen", horizontal = TRUE)  
boxplot(horsepower\_num, main= "Boxplot for price", xlab= "price", col= "gray", horizontal = TRUE)  
boxplot(compression\_ratio, main= "Boxplot for price", xlab= "price", col= "magenta", horizontal = TRUE)  
boxplot(bore\_num, main= "Boxplot for price", xlab= "price", col= "cyan", horizontal = TRUE)



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)

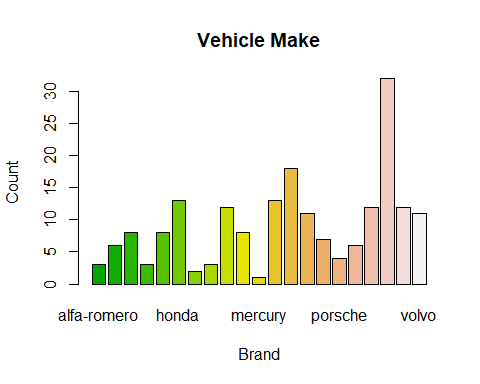


par(mfrow = c(2,2))  
boxplot(length, main= "Boxplot for price", xlab= "price", col= "blue", horizontal = TRUE)  
boxplot(curb\_weight, main= "Boxplot for price", xlab= "price", col= "green", horizontal = TRUE)  
boxplot(wheel\_base, main= "Boxplot for price", xlab= "price", col= "red", horizontal = TRUE)  
boxplot(norm\_loss, main= "Boxplot for price", xlab= "price", col= "purple", horizontal = TRUE)



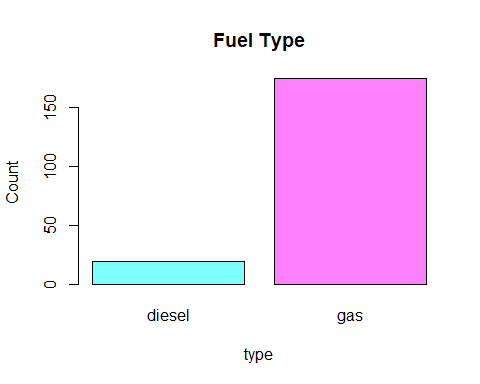
BAR PLOT

Make\_Tbl <- table(autoMobile$make)  
  
barplot(Make\_Tbl, main = "Vehicle Make", xlab = "Brand", ylab = "Count", col = terrain.colors(21), border = "black")



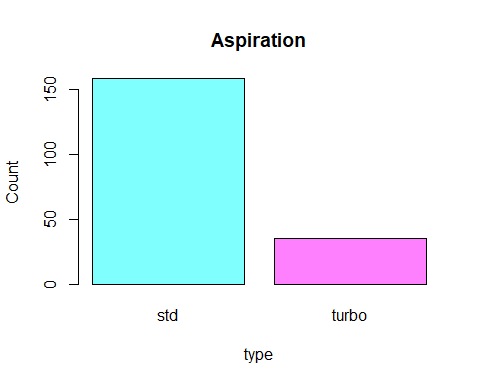
* maximum number of Vehicals are Toyota and the minimum is mercury

fuel\_type\_Tbl <- table(autoMobile$fuel\_type)  
barplot(fuel\_type\_Tbl, main = "Fuel Type", xlab = "type", ylab = "Count", col = cm.colors(2), border = "black")



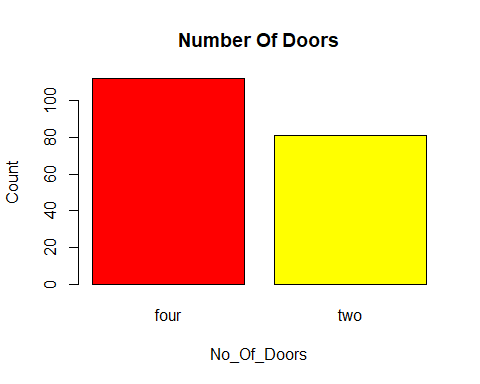
* Mostly used fuel type is gas

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



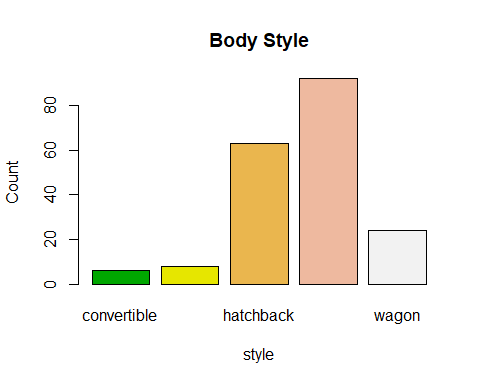
* most of them are Standard vehicles

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.colors(2), border = "black")



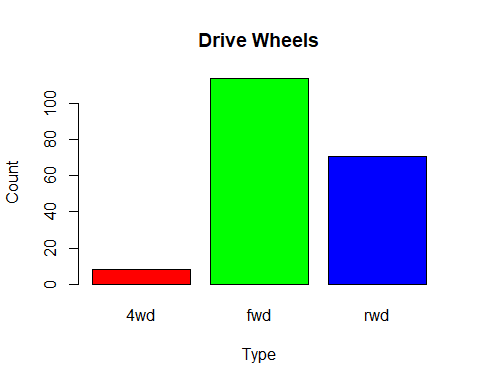
* four doors vehicles are Higher than two no of door vehicles

body\_style\_Tbl <- table(autoMobile$body\_style)  
barplot(body\_style\_Tbl, main = "Body Style", xlab = "style", ylab = "Count",col = terrain.colors(5), border = "black")



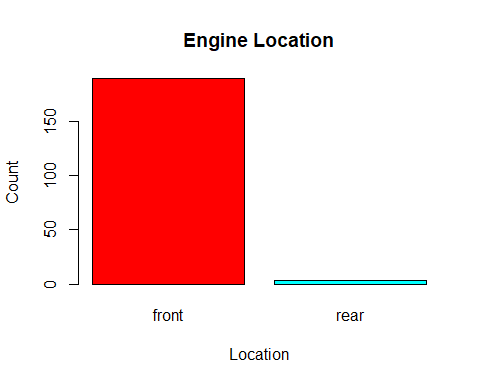
* sedan is the popular vehicle body style

drive\_wheels\_Tbl <- table(autoMobile$drive\_wheels)  
barplot(drive\_wheels\_Tbl, main = "Drive Wheels", xlab = "Type", ylab = "Count",col = rainbow(3), border = "black")



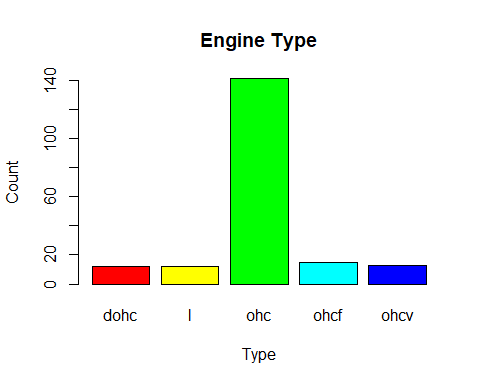
* most of drive wheels are fwd

engine\_location\_Tbl <- table(autoMobile$engine\_location)  
barplot(engine\_location\_Tbl, main = "Engine Location", xlab = "Location", ylab = "Count",col = rainbow(2), border = "black")



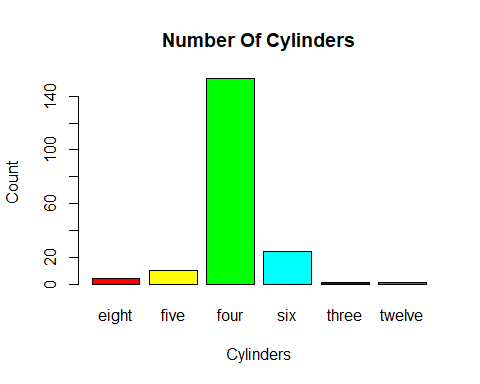
* almost all the vehicle’s engine is located in the front

engine\_type\_Tbl <- table(autoMobile$engine\_type)  
barplot(engine\_type\_Tbl, main = "Engine Type", xlab = "Type", ylab = "Count", col = rainbow(6), border = "black")



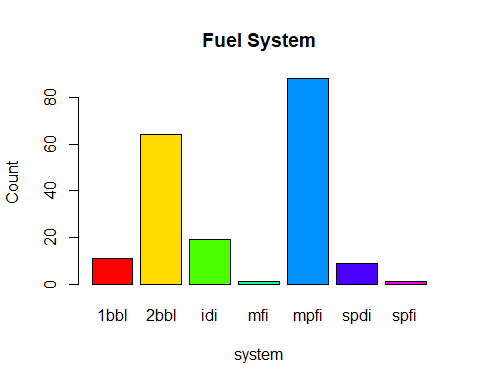
* most popular engine type is ohc

num\_of\_cylinders\_Tbl <- table(autoMobile$num\_of\_cylinders)  
barplot(num\_of\_cylinders\_Tbl, main = "Number Of Cylinders", xlab = "Cylinders", ylab = "Count",col = rainbow(6), border = "black")



* many vehicles has four cyclinders

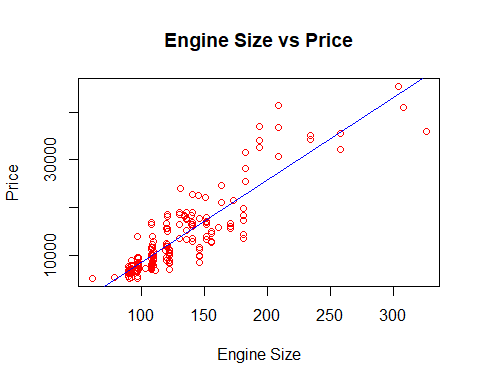
fuel\_system\_Tbl <- table(autoMobile$fuel\_system)  
barplot(fuel\_system\_Tbl, main = "Fuel System", xlab = "system", ylab = "Count", col = rainbow(7), border = "black")



* most popular fuel system is mpfi

#### Engine\_size vs Price

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")



* 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.

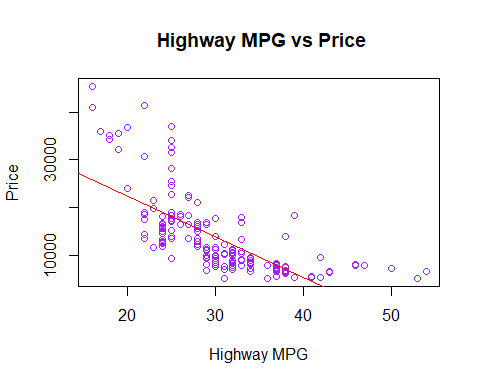
CorEngineS.P <- cor(autoMobile$engine\_size, autoMobile$price)  
CorEngineS.P

## [1] 0.8887785

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

#### Highway\_mpg vs Price

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")



* 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.

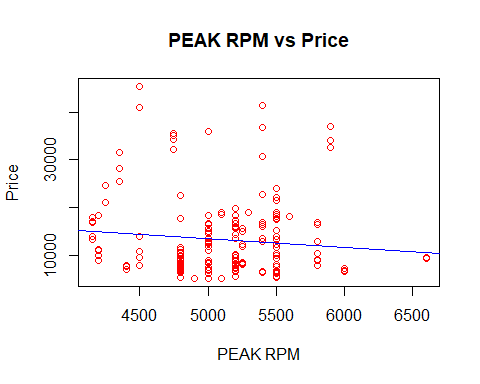
CorHighway\_mpg.P <- cor(autoMobile$highway\_mpg, autoMobile$price)  
CorHighway\_mpg.P

## [1] -0.7191777

* we can examine the correlation between ‘highway\_mpg’ and ‘price’ and see it’s approximately: **-0.7200901**

#### Peak\_rpm vs Price

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 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.

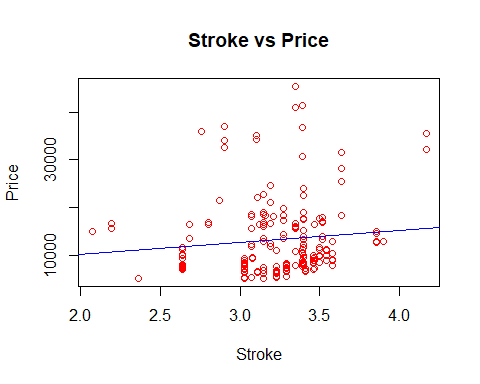
CorPeak\_rpm.P <- cor(autoMobile$peak\_rpm, autoMobile$price)  
CorPeak\_rpm.P

## [1] -0.1038353

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

#### Stroke vs Price

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")



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

## [1] 0.09600668

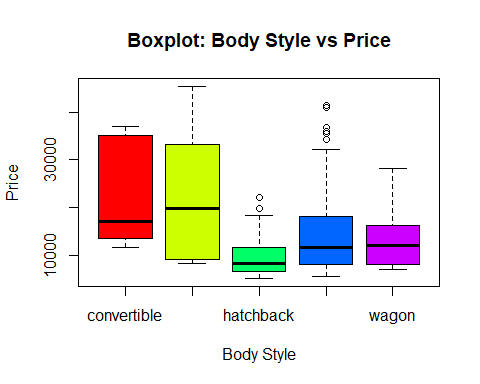
* We can examine the correlation between ‘stroke’ and ‘price’ and see it’s approximately: **0.09600668**

#### **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”**

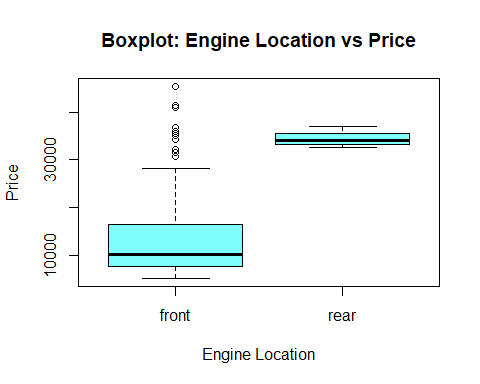
boxplot(price ~ body\_style, data = autoMobile, col = rainbow(5),  
 main = "Boxplot: Body Style vs Price",  
 xlab = "Body Style", ylab = "Price")



* 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.

1. Relationship between **“engine-location”** and **“price”**

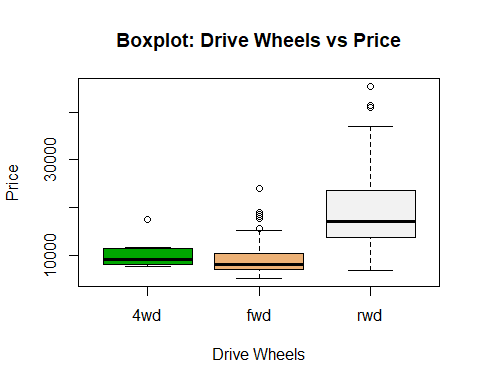
boxplot(price ~ engine\_location, data = autoMobile, col = cm.colors(1),  
 main = "Boxplot: Engine Location vs Price",  
 xlab = "Engine Location", ylab = "Price")



* 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:

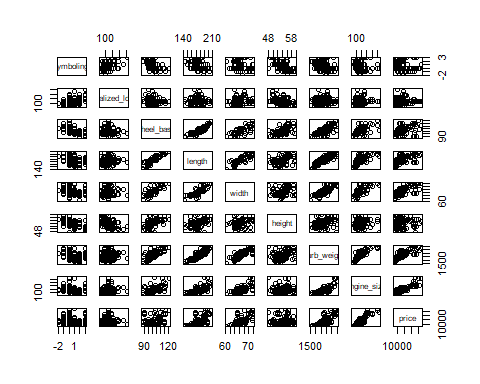
1. The count of that variable
2. The mean
3. The standard deviation (std)
4. The minimum value
5. The IQR (Interquartile Range: 25%, 50% and 75%)
6. The maximum value

summary(autoMobile)

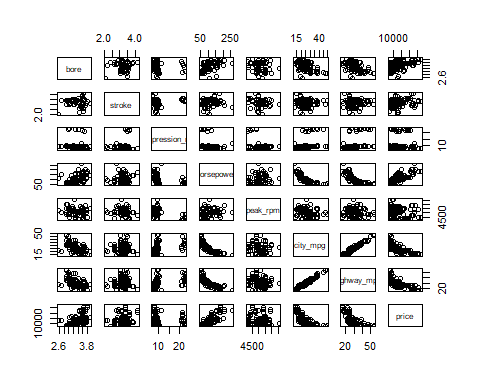
## symboling normalized\_losses make fuel\_type aspiration   
## Min. :-2.0000 Min. : 65.0 toyota :32 diesel: 19 std :158   
## 1st Qu.: 0.0000 1st Qu.: 95.0 nissan :18 gas :174 turbo: 35   
## Median : 1.0000 Median :122.0 honda :13   
## Mean : 0.7979 Mean :121.3 mitsubishi:13   
## 3rd Qu.: 2.0000 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)



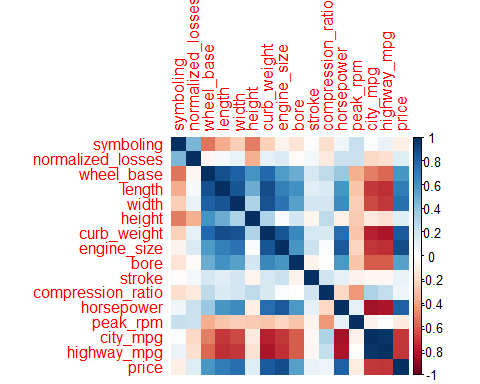
### Correlation

* **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**

# 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$estimate  
p\_value\_wb <- result\_wb$p.value  
  
# Print the results  
print(paste("The Pearson Correlation Coefficient is", pearson\_coef\_wb, "with a P-value of P =", p\_value\_wb))

## [1] "The Pearson Correlation Coefficient is 0.584950622305816 with a P-value of P = 4.16429781569782e-19"

**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**

# 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\_hp))

## [1] "The Pearson Correlation Coefficient is 0.812453204601347 with a P-value of P = 1.24840733993129e-46"

**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**

# 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-29"

**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**

# 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\_w))

## [1] "The Pearson Correlation Coefficient is 0.754648894838236 with a P-value of P = 8.44009950371111e-37"

**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**

# 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\_cw))

## [1] "The Pearson Correlation Coefficient is 0.835367753626223 with a P-value of P = 1.5875863033306e-51"

**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**

# 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\_Es))

## [1] "The Pearson Correlation Coefficient is 0.888778495310582 with a P-value of P = 1.25250791781395e-66"

**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**

# 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-16"

**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**

# 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\_Cm))

## [1] "The Pearson Correlation Coefficient is -0.706617993498795 with a P-value of P = 1.65332192881941e-30"

**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**

# 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\_Hm))

## [1] "The Pearson Correlation Coefficient is -0.719177688383088 with a P-value of P = 5.01527492738632e-32"

**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.

1. **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.

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

grouped\_test2 <- df\_gptest %>%  
 select("drive\_wheels", "price") %>%  
 group\_by(drive\_wheels)  
  
head(grouped\_test2)

## # A tibble: 6 × 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 × 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.

REDUCED MODEL

Reduced\_Model <- lm(price ~ peak\_rpm + engine\_size + curb\_weight + engine\_location + width + engine\_location,  
 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.

1. 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.

1. Residuals have a zero mean.

* significant value is 0. randomly distributed.

1. Residuals have a constant variance.

* randomly distributed plot means the constant variance

1. Residuals are independently distributed.

* randomly distributed plot means the independent distributed

residuals

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

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

mean(residuals\_RM)

## [1] 6.669392e-14

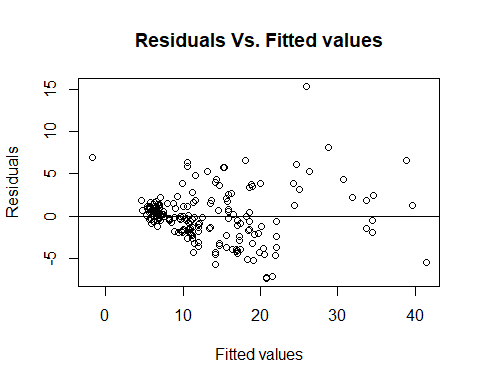
* mean residuals nearly goes to zero. therefore we can take this as mean val = 0.

4th Assumption

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

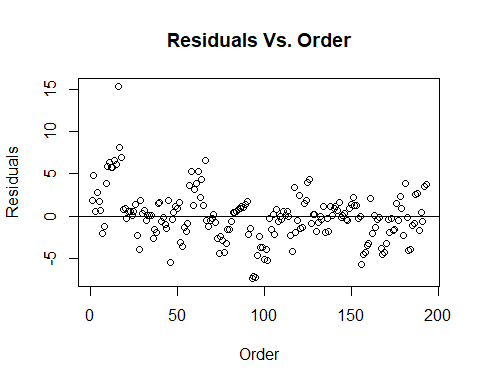
plot(predictor\_RM/1000, residuals\_RM/1000, main = "Residuals Vs. Fitted values", xlab = "Fitted values", ylab = "Residuals")  
abline(h=0)



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 scattered, residuals are not randomly 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 + width + engine\_location, data = autoMobile)

### **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.