proj

Suneeth Ravilla

01/12/2019

library(LearnEDAfunctions)

## Loading required package: aplpack

## Loading required package: vcd

## Loading required package: grid

## Loading required package: tidyverse

## -- Attaching packages ------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ---------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

auto.mpg <- read.csv("D:/AAA/Umer/auto-mpg.csv")  
  
data = auto.mpg

outliers

library(tidyverse)  
mpg.five = fivenum(pull(data,mpg))  
mpg.five

## [1] 9.0 17.5 23.0 29.0 46.6

acc.five = fivenum(pull(data,acceleration))  
acc.five

## [1] 8.0 13.8 15.5 17.2 24.8

mpg.step = 1.5 \* (29-17.5)  
acc.step = 1.5 \* (17.2-13.8)  
mpg.step

## [1] 17.25

acc.step

## [1] 5.1

mpg.lower = 17.5 - 2\*mpg.step  
mpg.upper = 29 + 2\*mpg.step  
acc.lower = 13.8 - acc.step  
acc.upper = 17.2 + acc.step  
mpg.upper

## [1] 63.5

mpg.lower

## [1] -17

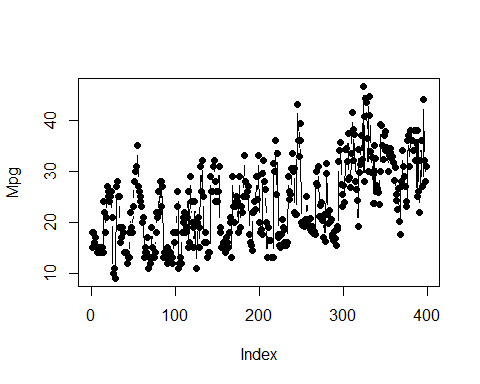
acc.lower

## [1] 8.7

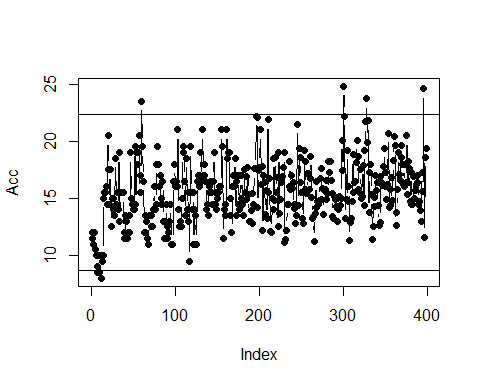
acc.upper

## [1] 22.3

plot(data$mpg, pch =16, type ="b", ylab = "Mpg")  
abline(h=mpg.upper)  
abline(h= mpg.lower)



plot(data$acceleration, pch =16, type ="b",ylab = "Acc")  
abline(h=acc.upper)  
abline(h= acc.lower)



By looking at the boxplots we can say that there is one outlier in mpg value of 46.6, car name mazda glc manufactured by Japan in 1980

By looking at acceleration there are 6 outliers, 2 in lower bound and 4 in upper bound.

# Check how the data was read in

str(data)

## 'data.frame': 398 obs. of 9 variables:  
## $ mpg : num 18 15 18 16 17 15 14 14 14 15 ...  
## $ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...  
## $ displacement: num 307 350 318 304 302 429 454 440 455 390 ...  
## $ horsepower : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...  
## $ weight : int 3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...  
## $ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...  
## $ model.year : int 70 70 70 70 70 70 70 70 70 70 ...  
## $ origin : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ car.name : Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 142 55 224 242 2 ...

# Do data conversions to match up with what it is supposed to be

data$cylinders = as.numeric(data$cylinders)  
data$horsepower = as.numeric(as.character(data$horsepower))

## Warning: NAs introduced by coercion

data$weight = as.numeric(data$weight)  
data$model.year = as.numeric(data$model.year)  
data$origin = as.numeric(data$origin)

# Split the car name column into the brand name, Change some of the brand names so that they match and make the brand name a factor

data$brand\_name = sub("([A-Za-z]+).\*", "\\1", data$car.name)  
data$brand\_name = gsub("chevy", "chevrolet", data$brand\_name)  
data$brand\_name = gsub("chevroelt", "chevrolet", data$brand\_name)  
data$brand\_name = gsub("vw", "volkswagen", data$brand\_name)  
data$brand\_name = gsub("vokswagen", "volkswagen", data$brand\_name)  
data$brand\_name = gsub("maxda", "mazda", data$brand\_name)  
data$brand\_name = gsub("toyouta", "toyota", data$brand\_name)  
data$brand\_name = as.factor(data$brand\_name)

# We have 6 missing values in horsepower column

# Impute the mean for the N/A in horsepower

data$horsepower[is.na(data$horsepower)] = round(mean(data$horsepower, na.rm = TRUE),digits=0)

# Look at a data summary to get a flavor of the data after plugging the mean for the missings

summary(data)

## mpg cylinders displacement horsepower   
## Min. : 9.00 Min. :3.000 Min. : 68.0 Min. : 46.0   
## 1st Qu.:17.50 1st Qu.:4.000 1st Qu.:104.2 1st Qu.: 76.0   
## Median :23.00 Median :4.000 Median :148.5 Median : 95.0   
## Mean :23.51 Mean :5.455 Mean :193.4 Mean :104.5   
## 3rd Qu.:29.00 3rd Qu.:8.000 3rd Qu.:262.0 3rd Qu.:125.0   
## Max. :46.60 Max. :8.000 Max. :455.0 Max. :230.0   
##   
## weight acceleration model.year origin   
## Min. :1613 Min. : 8.00 Min. :70.00 Min. :1.000   
## 1st Qu.:2224 1st Qu.:13.82 1st Qu.:73.00 1st Qu.:1.000   
## Median :2804 Median :15.50 Median :76.00 Median :1.000   
## Mean :2970 Mean :15.57 Mean :76.01 Mean :1.573   
## 3rd Qu.:3608 3rd Qu.:17.18 3rd Qu.:79.00 3rd Qu.:2.000   
## Max. :5140 Max. :24.80 Max. :82.00 Max. :3.000   
##   
## car.name brand\_name   
## ford pinto : 6 ford : 51   
## amc matador : 5 chevrolet: 47   
## ford maverick : 5 plymouth : 31   
## toyota corolla: 5 amc : 28   
## amc gremlin : 4 dodge : 28   
## amc hornet : 4 toyota : 26   
## (Other) :369 (Other) :187

Stemplots

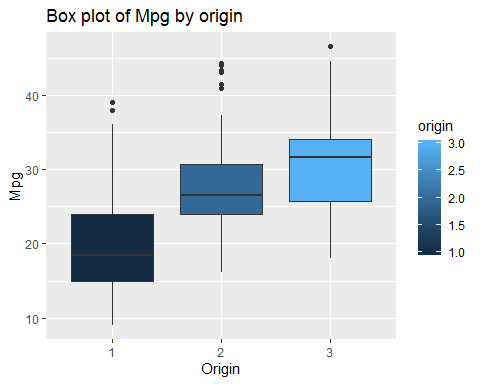
stem.leaf(pull(data, mpg), m=5, depth=FALSE)

## 1 | 2: represents 12  
## leaf unit: 1  
## n: 398  
## 0. | 9  
## 1\* | 001111  
## t | 22222233333333333333333333  
## f | 44444444444444444444555555555555555555555  
## s | 666666666666666666777777777777777  
## 1. | 88888888888888888888888899999999999999999999  
## 2\* | 000000000000000000001111111111111  
## t | 222222222222233333333333333333  
## f | 44444444444444455555555555555555  
## s | 6666666666666666666777777777777777  
## 2. | 88888888888889999999999999  
## 3\* | 000000000001111111111111  
## t | 2222222222222233333333  
## f | 44444444445555  
## s | 666666666777777  
## 3. | 88888999  
## 4\* | 001  
## t | 33  
## f | 444  
## s | 6

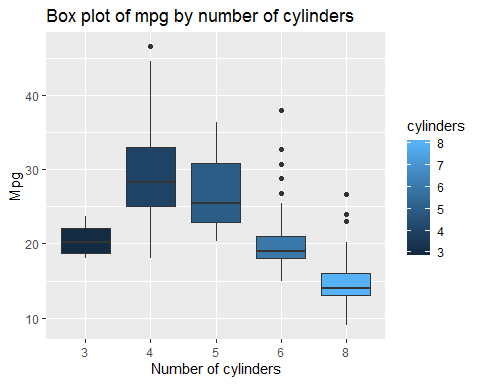
stem.leaf(pull(data, acceleration), m=2, depth=FALSE)

## 1 | 2: represents 1.2  
## leaf unit: 0.1  
## n: 398  
## LO: 8 8.5 8.5  
## 9\* | 0  
## 9. | 55  
## 10\* | 0000  
## 10. | 5  
## 11\* | 000000012344  
## 11. | 55555556  
## 12\* | 0000000000122  
## 12. | 555555556688899  
## 13\* | 00000000000022222244  
## 13. | 55555555555555566778899  
## 14\* | 000000000000000012223344444  
## 14. | 55555555555555555555555777778889999999  
## 15\* | 00000000000000112223334444  
## 15. | 55555555555555555555567777888888899  
## 16\* | 000000000000000012222444444444  
## 16. | 5555555555555666777889999  
## 17\* | 000000000000001223333344  
## 17. | 55556666777889  
## 18\* | 000000001222223  
## 18. | 555556666778  
## 19\* | 000000000000222444  
## 19. | 555555669  
## 20\* | 114  
## 20. | 5557  
## 21\* | 00000  
## 21. | 5789  
## 22\* | 122  
## HI: 23.5 23.7 24.6 24.8

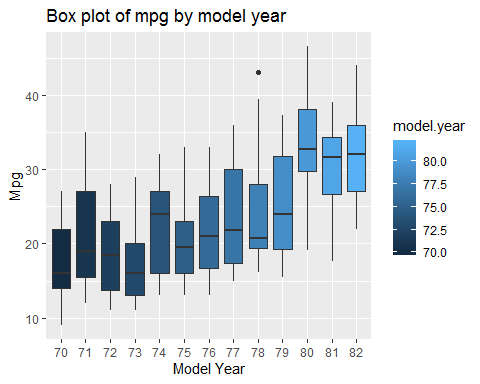
ggplot(data,aes(x=factor(origin),y = mpg,fill = origin))+geom\_boxplot()+  
xlab("Origin")+ylab("Mpg")+ggtitle("Box plot of Mpg by origin")



ggplot(data,aes(x=factor(cylinders),y=mpg,fill = cylinders))+geom\_boxplot()+  
xlab("Number of cylinders")+ylab("Mpg")+ggtitle("Box plot of mpg by number of cylinders")



ggplot(data,aes(x=factor(model.year),y=mpg,fill = model.year))+geom\_boxplot()+  
xlab("Model Year")+ylab("Mpg")+ggtitle("Box plot of mpg by model year")



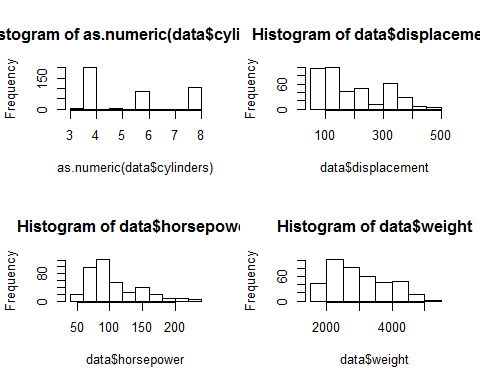
I am not sure what the corresponding actual names for the places of origin are but it seems like cars that came from region 3 had the best mpg with region 1 having the worst of the three regions

The general trend for the mpg as the number of cylinders increased was down. The more cylinders you had the worse off your mpg was. The best number of cylinders to have seems like 4, 4 cylinders had the best mpg

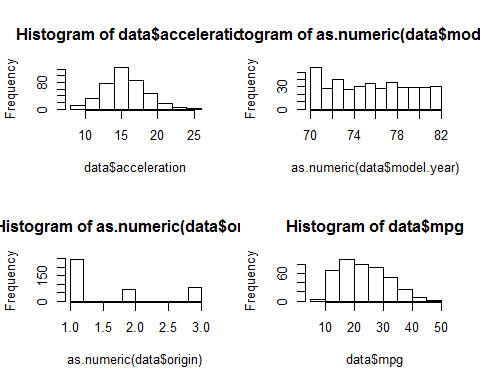
It seems as if as the years progressed the general mpg values increased too. I am guessing with improved engineering the mpg would have been expected to improve with time.

Histogram

par(mfrow = c(2, 2))  
hist(as.numeric(data$cylinders))  
hist(data$displacement)  
hist(data$horsepower)  
hist(data$weight)



par(mfrow = c(2, 2))  
hist(data$acceleration)  
hist(as.numeric(data$model.year))  
hist(as.numeric(data$origin))  
hist(data$mpg)



We see how Acceleration distribution is close to normal but others such as HorsePower or displacement or Weight or weight are displaced very far to the left skewed, this was to be expected since there will be few cars with a lot of power or with a lot of weight, but most will be located near the initial values.

So after the first exploration – our data in a nutshell:

398 different cars from 1970 to 1982 296 distinct models from 30 manufacturers, a few of them dominating like Ford, Chevrolet. Cars mainly from USA (249), much less from Japan (79) and Europe (70) 1973 and 1978 seem a little bit stronger years with more samples.

stem.leaf(data$mpg)

## 1 | 2: represents 12  
## leaf unit: 1  
## n: 398  
## 1 0. | 9  
## 7 1\* | 001111  
## 33 t | 22222233333333333333333333  
## 74 f | 44444444444444444444555555555555555555555  
## 107 s | 666666666666666666777777777777777  
## 151 1. | 88888888888888888888888899999999999999999999  
## 184 2\* | 000000000000000000001111111111111  
## (30) t | 222222222222233333333333333333  
## 184 f | 44444444444444455555555555555555  
## 152 s | 6666666666666666666777777777777777  
## 118 2. | 88888888888889999999999999  
## 92 3\* | 000000000001111111111111  
## 68 t | 2222222222222233333333  
## 46 f | 44444444445555  
## 32 s | 666666666777777  
## 17 3. | 88888999  
## 9 4\* | 001  
## 6 t | 33  
## 4 f | 444  
## 1 s | 6

This plot represents 2 leaf stem plot. It looks like Right-Skewed

mpg.mids = lval(data$mpg)  
select(mpg.mids,mids)

## mids  
## M 23.000  
## H 23.250  
## E 23.750  
## D 24.550  
## C 25.050  
## B 26.250  
## A 27.500  
## Z 27.225  
## Y 27.800

Median = 23

roots

roots.mpg = sqrt(pull(data,mpg))  
logs.mpg = log(pull(data,mpg))  
recr.mpg =-1/sqrt(pull(data,mpg))

hinkley method

hinkley(data$mpg)

## h   
## 0.04474547

hinkley(roots.mpg)

## h   
## -0.01118955

hinkley(logs.mpg)

## h   
## -0.06753192

hinkley(recr.mpg)

## h   
## -0.1251619

With Hinkley method we can see high postive value with p = 1, negative when p = 0.5. So we would get actual symmetric when we take value between p=0 and p=-1/2. But Logs gives symmetric than all for mpg data

dat= subset(data, select = -c(car.name,brand\_name)) #  
cor(dat)

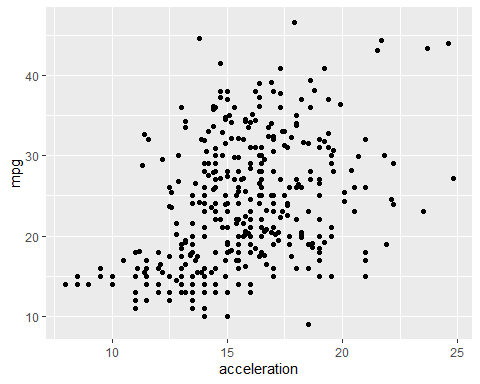
## mpg cylinders displacement horsepower weight  
## mpg 1.0000000 -0.7753963 -0.8042028 -0.7715428 -0.8317409  
## cylinders -0.7753963 1.0000000 0.9507214 0.8390609 0.8960168  
## displacement -0.8042028 0.9507214 1.0000000 0.8937600 0.9328241  
## horsepower -0.7715428 0.8390609 0.8937600 1.0000000 0.8606759  
## weight -0.8317409 0.8960168 0.9328241 0.8606759 1.0000000  
## acceleration 0.4202889 -0.5054195 -0.5436841 -0.6843761 -0.4174573  
## model.year 0.5792671 -0.3487458 -0.3701642 -0.4117505 -0.3065643  
## origin 0.5634504 -0.5625433 -0.6094094 -0.4536133 -0.5810239  
## acceleration model.year origin  
## mpg 0.4202889 0.5792671 0.5634504  
## cylinders -0.5054195 -0.3487458 -0.5625433  
## displacement -0.5436841 -0.3701642 -0.6094094  
## horsepower -0.6843761 -0.4117505 -0.4536133  
## weight -0.4174573 -0.3065643 -0.5810239  
## acceleration 1.0000000 0.2881370 0.2058730  
## model.year 0.2881370 1.0000000 0.1806622  
## origin 0.2058730 0.1806622 1.0000000

The highest correlation between mpg and any of the other variables is with weight and it is a negative correlation. This makes sense, the more a car weighs the more energy required to move it down the road and the energy is from the gasoline which means your mpg is going to suffer

The highest correlation among all the variables is between cylinder and displacement with correlation of 95% which shows that displacement of the car is dependent on number of cylinders

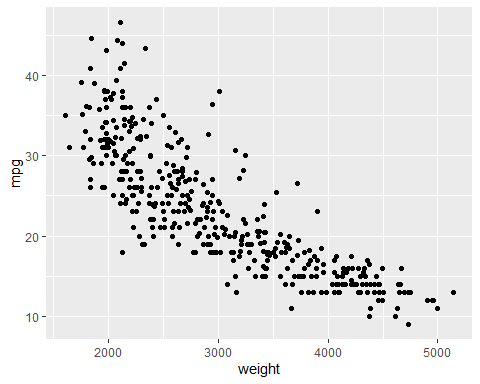
scatterplot

ggplot(data,aes(acceleration,mpg))+geom\_point()+xlab("acceleration")+ ylab("mpg")

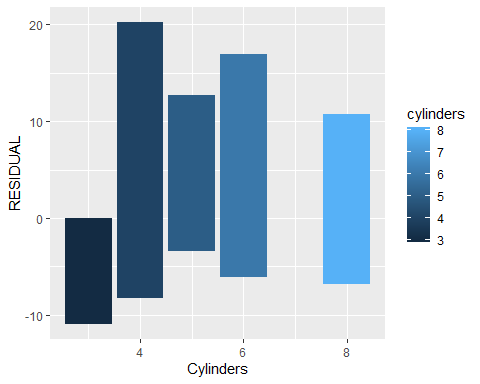


acceleration variable is not adding considerable strength.

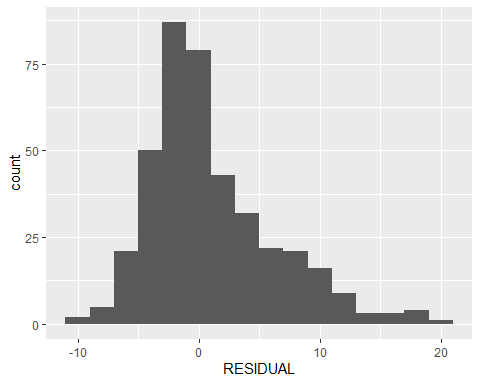
library(ggplot2)  
ggplot(data,aes(weight, mpg))+geom\_point()+xlab("weight")+ ylab("mpg")



library(dplyr)  
originalfit = rline(mpg~cylinders, data)  
dat<- mutate(dat, FIT =pluck(originalfit, "a")+pluck(originalfit, "b")\*(data$cylinders-pluck(originalfit, "xC")),RESIDUAL = data$mpg-FIT)  
ggplot(dat, aes(data$cylinders,RESIDUAL,fill=cylinders))+xlab("Cylinders")+geom\_bar(stat = "identity",position= 'Dodge')#+geom\_hline(yintercept = 0, color = "Blue")



ggplot(data,aes(x =dat$RESIDUAL)) + xlab("RESIDUAL") + geom\_histogram(binwidth = 2,position = 'dodge')



Gaussian fit

m = mean(data$mpg)  
sd = sd(data$mpg)  
m

## [1] 23.51457

sd

## [1] 7.815984