**Part 1 – Solution to Python Portfolio tasks**

**Solution to Task 1 Python portfolio**

Reading the dataset into a suitable data structure and printing its first 10 rows on the screen

data=pd.read\_csv("C:/Users/pc/Desktop/001186932/Iris.csv")

data.head(10)

**Measures of Central Tendency (mean and median)**

mean\_and\_median=data.groupby("Species").agg({'Petal width':['mean' ,'median']})

**Values**

Mean Petal width for Setosa=0.244

Median Petal width for Setosa =0.2

Mean Petal width for Versicolor=1.326

Mean Petal width for Versicolor=1.326

Median Petal width for Setosa =1.3

Mean Petal width for Virginia=2.026

Median Petal width for Virginia =2.0

**Measures of Dispersion**

std\_max\_min =data.groupby("Species").agg({'Petal width':[ 'std' ,'max' , 'min' ]})

std\_max\_min\_ = std\_max\_min.iloc[:,:]

std\_max\_min\_.columns=['petal\_width\_std','petal\_width\_max' , 'petal\_width\_min']

std\_max\_min\_['Range']=std\_max\_min\_['petal\_width\_max']-std\_max\_min\_['petal\_width\_min']

std\_max\_min\_range=std\_max\_min\_

**values**

Standard deviation of Petal width for Setosa =0.107210

Range of Petal width for Setosa =0.5

Standard deviation of Petal width for Versicolor =0.197753

Range of Petal width for Versicolor =0.8

Standard deviatiation of Petal width for Virginia =0.274650

Range of Petal width for Virginia =1.1

**Solution to Task 2 Python portfolio**

BoxPlot

It shows the type of distribution of the data

It shows the quartile of the data

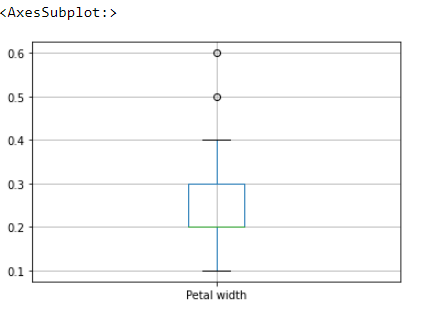
The upper whisker and lower whisker shows the range within which the majority of values fall

It whether there are outliers in the data or not

It also shows the median of the data

#Python code to plot boxplot

petalwidth\_setosa.boxplot(column="Sepal length" , return\_type="axes")



**Solution to Task 3 Python portfolio**

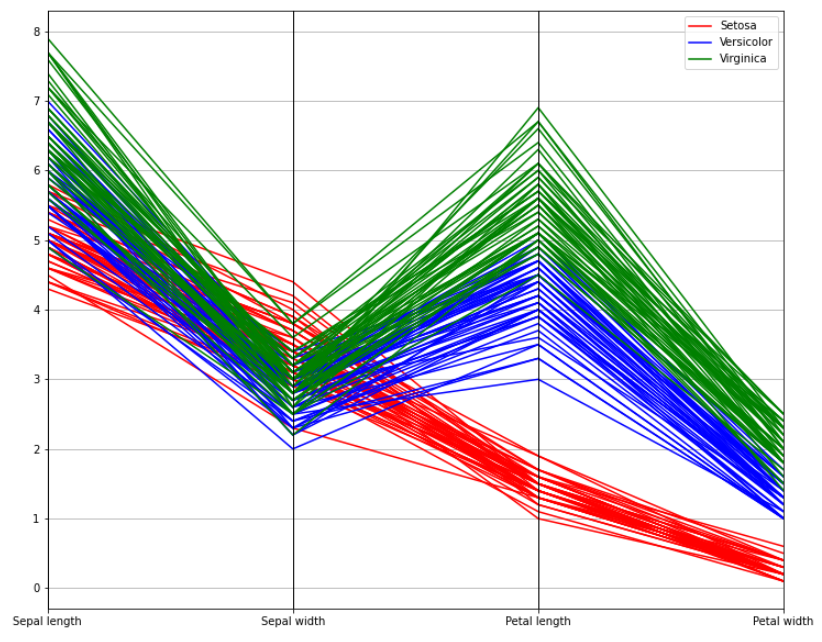
#Python code to plot parallel\_coordinate

#Parallel coordinates plot of iris data

plt.figure(figsize=(12,10))

parallel\_coordinates(data, "Species" , color=("red", "blue" , "green"), )

plt.show()



Solution to R part

**Solution to task 1:Data transformation**  
Number 1.  
> library("tidyr")

> who

> who\_=pivot\_longer(data=who,

+ cols=c("new\_sp\_m014", "new\_sp\_m1524", "new\_sp\_m2534", "new\_sp\_m3544", "new\_sp\_m4554", "new\_sp\_m5564", "new\_sp\_m65", "new\_sp\_f014", "new\_sp\_f1524", "new\_sp\_f2534", "new\_sp\_f3544", "new\_sp\_f4554", "new\_sp\_f5564", "new\_sp\_f65", "new\_sn\_m014", "new\_sn\_m1524", "new\_sn\_m2534", "new\_sn\_m3544", "new\_sn\_m4554", "new\_sn\_m5564", "new\_sn\_m65", "new\_sn\_f014", "new\_sn\_f1524", "new\_sn\_f2534", "new\_sn\_f3544", "new\_sn\_f4554", "new\_sn\_f5564", "new\_sn\_f65", "new\_ep\_m014", "new\_ep\_m1524", "new\_ep\_m2534", "new\_ep\_m3544", "new\_ep\_m4554", "new\_ep\_m5564", "new\_ep\_m65", "new\_ep\_f014", "new\_ep\_f1524", "new\_ep\_f2534", "new\_ep\_f3544", "new\_ep\_f4554", "new\_ep\_f5564", "new\_ep\_f65", "newrel\_m014", "newrel\_m1524", "newrel\_m2534", "newrel\_m3544", "newrel\_m4554", "newrel\_m5564", "newrel\_m65", "newrel\_f014", "newrel\_f1524", "newrel\_f2534", "newrel\_f3544", "newrel\_f4554", "newrel\_f5564", "newrel\_f65"),

+ names\_to="key",

+ values\_to="cases")

> who1=who\_%>% drop\_na()

Solution to Task1 number 2.

Who2=who1

> library("stringr")

> who2$key<- str\_replace(who1$key , "newrel" , "new\_rel")

Solution to Task1 number 3.

> who3<-who2%>%separate(key,c("new","type","sexage"),sep="\_")

%>% is used to

Solution to Task1 number 4.

> who4<- who3%>%separate(sexage,c("sex","age"),sep="(?<=[A-Za-z])(?=[0-9])")

Solution to Task1 number 5.

> head(who4 , n=5)

> tail(who4 , n=5)

Solution to Task1 number 6

> write.table(who , file="C:/Users/pc/Desktop/001186932/who4.csv")

Solution to Task 2

Solution to Question 1.

> mean(Nile)

[1] 919.35

> median(Nile)

[1] 893.5

> mode <- function(x) {

+ ux <- unique(x)

+ ux[which.max(tabulate(match(x, ux)))]}

> mode(Nile)

[1] 1160

> var(Nile)

[1] 28637.95

> std<-sqrt(var(Nile))

> std

[1] 169.2275

Solution to Question 2

> min(Nile)

[1] 456

> max(Nile)

[1] 1370

> rangeNile <- max(Nile)-min(Nile)

> print(rangeNile)

[1] 914

Solution to Question 3

> IQR(Nile)

[1] 234

> quantile(Nile)

0% 25% 50% 75% 100%

456.0 798.5 893.5 1032.5 1370.0

Inter quartile range measures the spread of the middle 50% of the data, while quartile shows the percentage of data below some values. For example, 50% quartile = 893.5 , This means that 50% of the data are less than 893.5

Solution to question 4

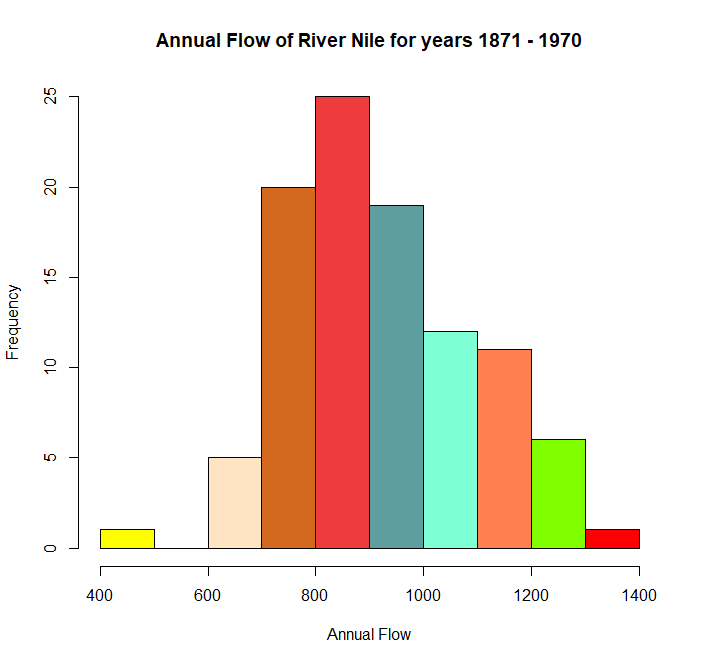
#Plot Histogram

> hist(Nile , main= "Histogram for Annual Flow of River Nile for years 1871 - 1970" , xlab="Annual Flow" , ylab="Frequency")

> nile\_breaks <- hist(Nile)$breaks

> nile\_colors <- c("yellow", "blue1","bisque", "chocolate", "brown2","cadetblue", "aquamarine", "coral","chartreuse","red")

>hist(Nile , main= "Annual Flow of River Nile for years 1871 - 1970" , xlab="Annual Flow" , ylab="Frequency", breaks=nile\_breaks , col=nile\_colors)

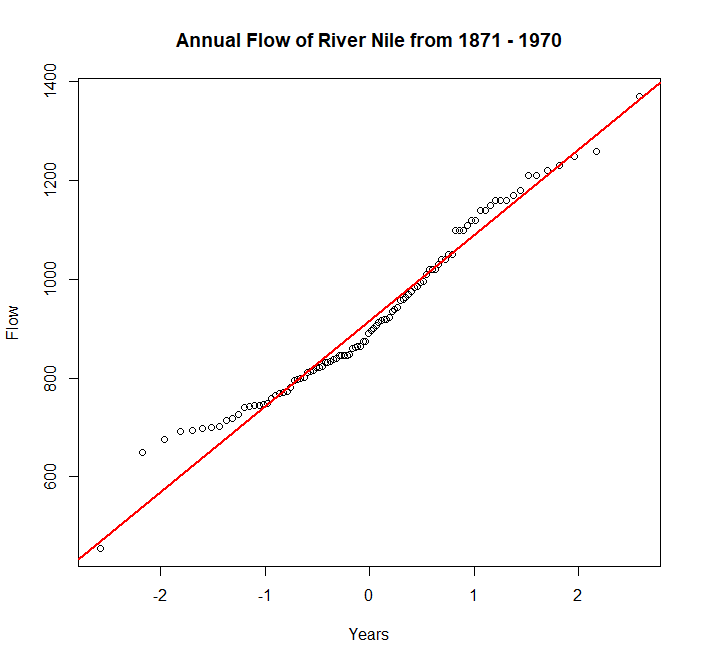


Solution to question 5

#Plot QQNORM

>qqnorm(Nile, main="Annual Flow of River Nile from 1871 - 1970", xlab="Years" , ylab="Flow" )

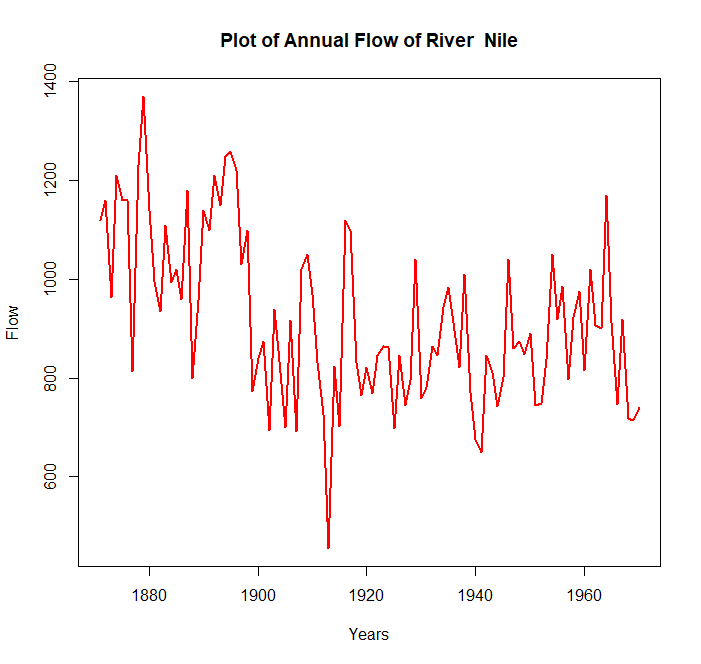
> qqline(Nile , col="red" , lwd=2)



The diagram above shows that the annual flow of river Nile has a high degree of normality. Though, with fluctuations, many data points are concentrated at the centre.

Solution to question 6

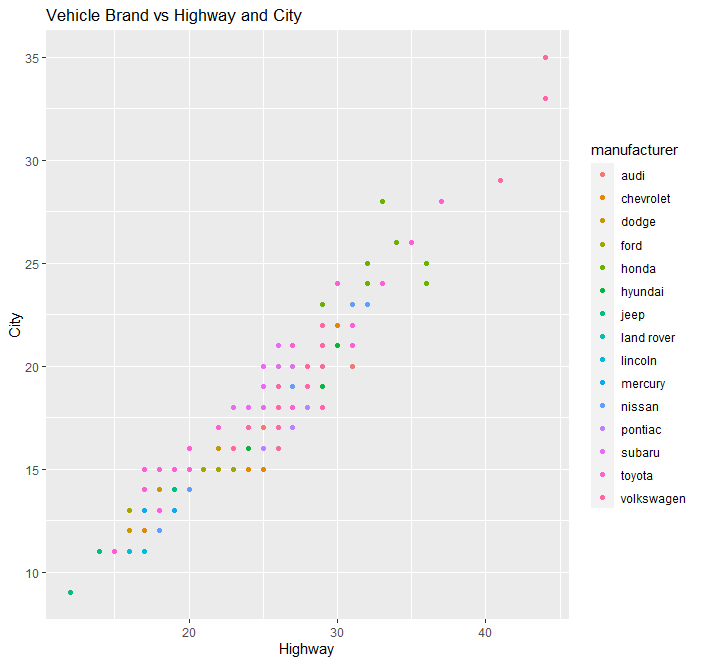
>plot(Nile , main="Plot of Annual Flow of River Nile",type="l" , xlab="Years" , ylab="Flow" , col="red" , lwd=2)



Solution to task 3 Question 1

> print(ggplot(data=mpg, aes(x=hwy , y=cty ,color= manufacturer )) +geom\_point() + ggtitle("Vehicle Brand vs Highway and City")+

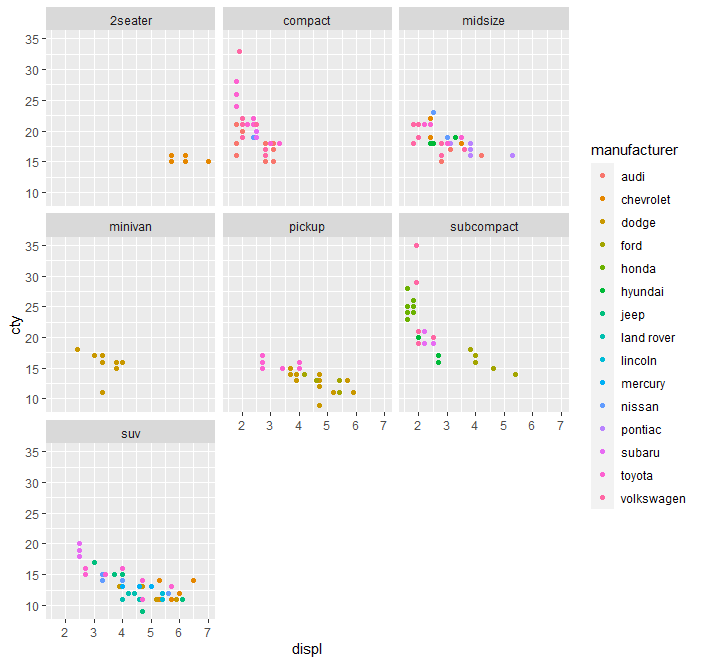
+ labs(y="City", x="Highway"))



Comment: audi has the best mpg , it has highest city mileage and Highway mileage

Solution to task 3 question 2

>ggplot(mpg, aes(displ, cty,hwy,color=manufacturer))+ geom\_point()+ facet\_wrap(vars(class))

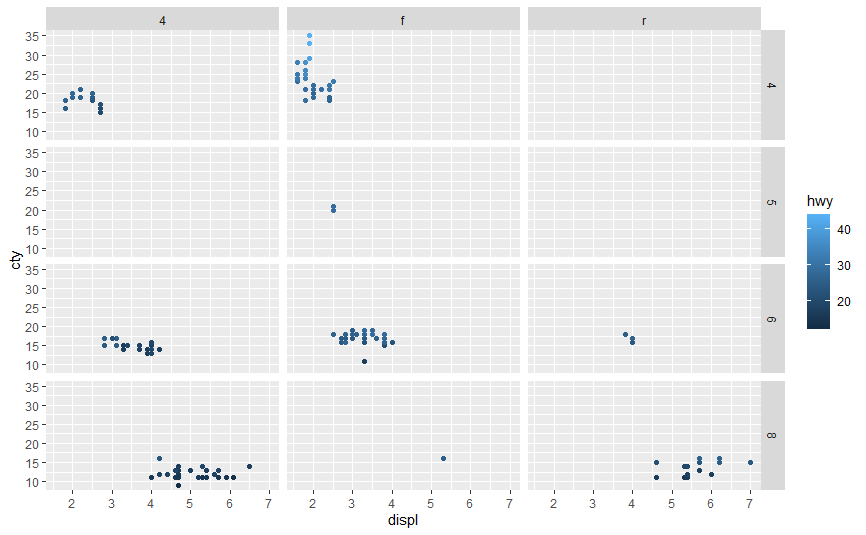


Comment:SubCompact has the best mpg in the city when categorized by vehicle type.

Solution to question 3

# Plot of type of car, regarding their *displ* range(size of engine) ,*mpg* performance in city and highway, categorized by the number of cylinders and the drive type.

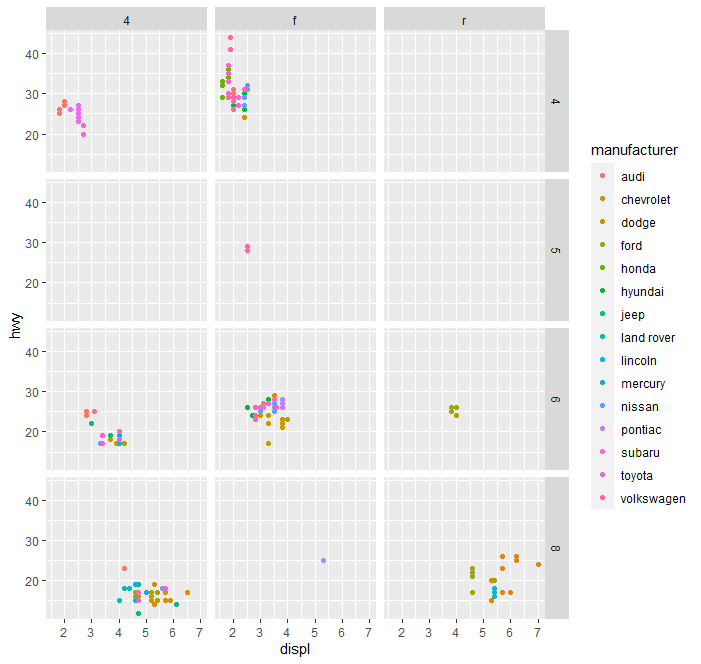
>ggplot(mpg, aes(displ, cty , color=hwy))+ geom\_point()+ facet\_grid(cyl~drv)



Considering the diagram above, front wheel cars have the best mpg performance in both cities and highways.

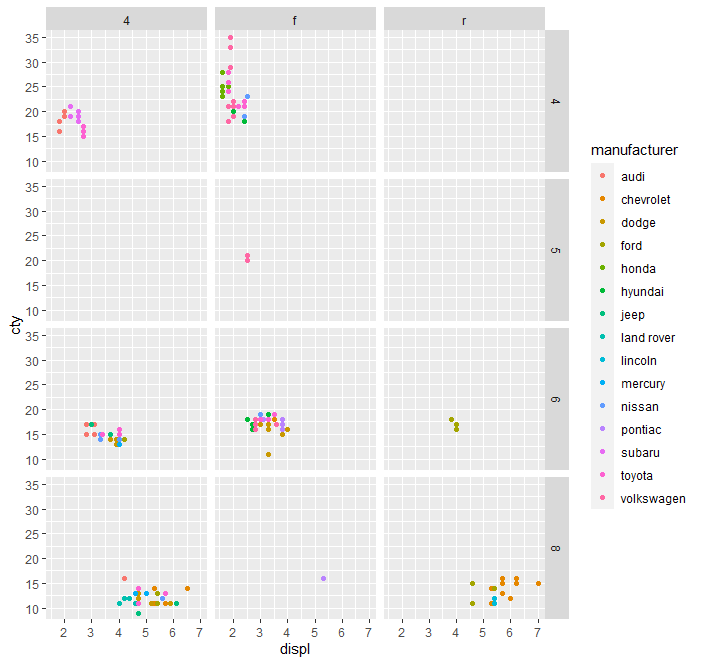
# Plot of type of car, regarding their *displ* range(size of engine) ,*mpg* performance in highway, categorized by the number of cylinders and the drive type.

> ggplot(mpg, aes(displ, hwy))+ geom\_point()+ facet\_grid(cyl~drv)



# Plot of typeof car, regarding their *displ* range(size of engine) ,*mpg* performance in city, categorized by the number of cylinders and the drive type

>ggplot(mpg, aes(displ, cty , color=hwy))+ geom\_point()+ facet\_grid(cyl~drv)



Comment: comparing the two diagrams above, Volkswagen front wheel have the highest mileage in the city and on highway. As a customer, I will buy a Volkswagen front wheel car.