Part 1. Solutions to Python portfolio tasks

Task 1 solutions:

In [1]:
```
#import required libraries
import pandas as pd
from pandas.plotting import parallel_coordinates
import statistics
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:
```
#Read dataset and print the first 10 rows
df = pd.read_csv('Iris.csv')
print(df.head(10))
```

```
   Sepal length  Sepal width  Petal length  Petal width Species
0           5.1          3.5           1.4          0.2  Setosa
1           4.9          3.0           1.4          0.2  Setosa
2           4.7          3.2           1.3          0.2  Setosa
3           4.6          3.1           1.5          0.2  Setosa
4           5.0          3.6           1.4          0.2  Setosa
5           5.4          3.9           1.7          0.4  Setosa
6           4.6          3.4           1.4          0.3  Setosa
7           5.0          3.4           1.5          0.2  Setosa
8           4.4          2.9           1.4          0.2  Setosa
9           4.9          3.1           1.5          0.1  Setosa
```

In [3]:
```
#Measures of Central Tendency

#Mean of Petal Width
petalwidth = df["Petal width"]
petalwidth_mean = statistics.mean(petalwidth)
print("Mean is :", petalwidth_mean)

#Median of Petal Width
petalwidth_median = statistics.median(petalwidth)
print("Median is :", petalwidth_median)
```

```
Mean is : 1.1986666666666668
Median is : 1.3
```

In [4]:
```
# Measures of Dispersion

#Standard deviation of Petal Width
petalwidth_stddev = statistics.stdev(petalwidth)
print("Standand Deviation is :", petalwidth_stddev)

#Range of Petal Width
print("Range is:",min(petalwidth),"to",max(petalwidth))
```

```
Standand Deviation is : 0.7631607417008411
Range is: 0.1 to 2.5
```

Task 2 solutions:

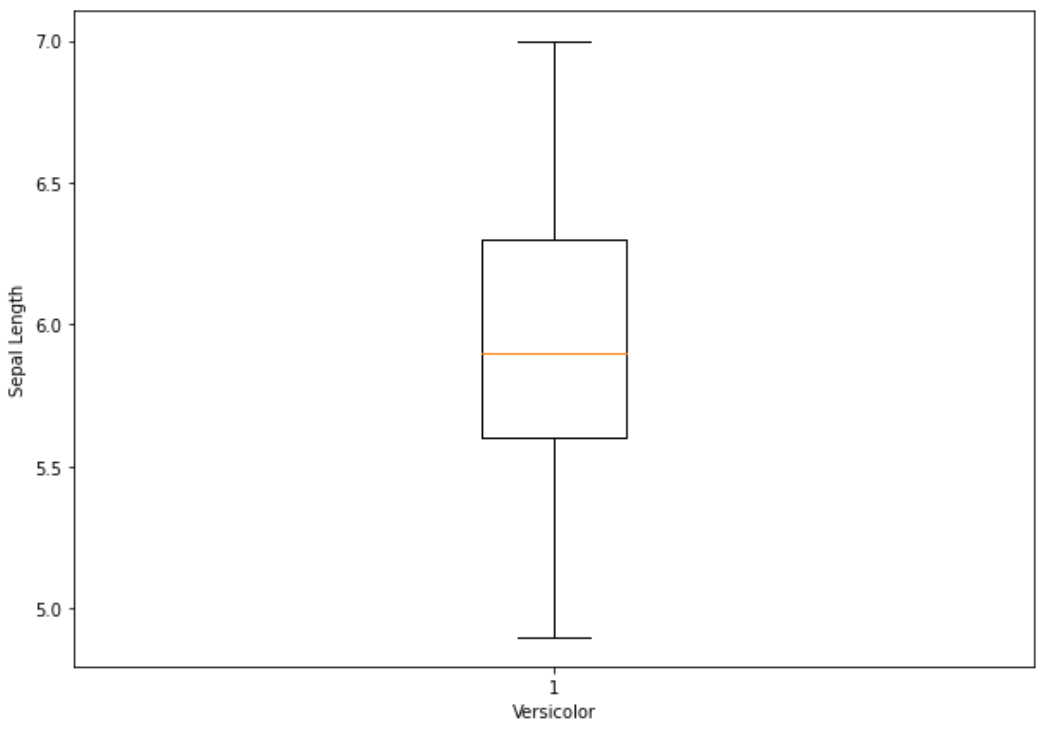
Identification:

The box plot chart is the optimal visualisation for a single attribute of a single species from the Iris dataset.

Justification:

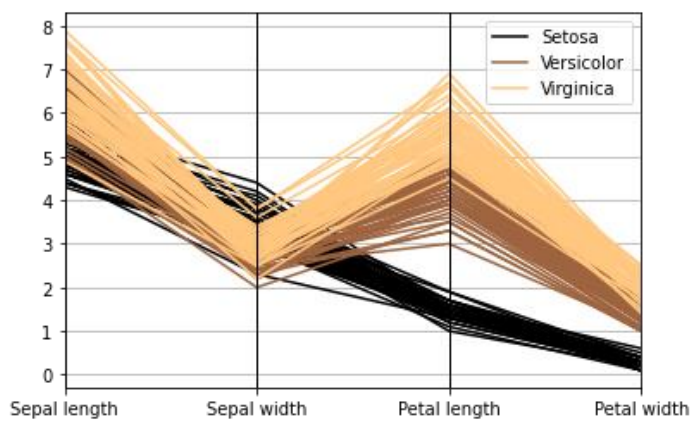
The box plot was chosen as an optimal visualisation for this purpose because it is a standardized way of distributing the data. It is easy to identify properties like minimum, first quartile, median, third quartile and maximum. Moreover, it can project the outliers and their values. Hence, Box plot chart is chosen over Line chart and Bar chart.

In [5]:
```
# Data Visualisation considering Versicolor as species and Sepal Length as the attribute

df_Versicolor =df.loc[df['Species'] == "Versicolor"]
fig = plt.figure(figsize =(10, 7))
plt.boxplot(df_Versicolor["Sepal length"])

plt.xlabel("Versicolor")
plt.ylabel("Sepal Length")
plt.show()
```



Task 3 solutions:

In [6]:
```
# Plot that represents different attributes of the three Species (Setosa, Versicolor, Virginica)

parallel_coordinates(df, 'Species', colormap=plt.get_cmap("copper"))
plt.show()
```



Justification:

Petal length is the best attribute to identify different species. As seen in the plot, Petal Length has different ranges for all the 3 species. Hence, it is easy to identify the species based on this attribute's range:

a. Setosa - [1,2]

b. Versicolor - [3,4.5]

c. Virginica - [4.5,7]

Part 2: R Portfolio Tasks

Task 1: Data Transformation

```
library("tidyr")
library("stringr")
```

Task 1.1:

Gather together all the columns from new_sp_m014 to newrel_f65. Drop columns containing missing values and name the new dataset as who1

```
who <- data.frame(who,package="tidyr")
print(head(who,2))
```

```
##       country iso2 iso3 year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
## 1 Afghanistan   AF  AFG 1980          NA           NA           NA           NA
## 2 Afghanistan   AF  AFG 1981          NA           NA           NA           NA
##   new_sp_m4554 new_sp_m5564 new_sp_m65 new_sp_f014 new_sp_f1524 new_sp_f2534
## 1           NA           NA         NA          NA           NA           NA
## 2           NA           NA         NA          NA           NA           NA
##   new_sp_f3544 new_sp_f4554 new_sp_f5564 new_sp_f65 new_sn_m014 new_sn_m1524
## 1           NA           NA           NA         NA          NA           NA
## 2           NA           NA           NA         NA          NA           NA
##   new_sn_m2534 new_sn_m3544 new_sn_m4554 new_sn_m5564 new_sn_m65 new_sn_f014
## 1           NA           NA           NA           NA         NA          NA
## 2           NA           NA           NA           NA         NA          NA
##   new_sn_f1524 new_sn_f2534 new_sn_f3544 new_sn_f4554 new_sn_f5564 new_sn_f65
## 1           NA           NA           NA           NA           NA         NA
## 2           NA           NA           NA           NA           NA         NA
##   new_ep_m014 new_ep_m1524 new_ep_m2534 new_ep_m3544 new_ep_m4554 new_ep_m5564
## 1          NA           NA           NA           NA           NA           NA
## 2          NA           NA           NA           NA           NA           NA
##   new_ep_m65 new_ep_f014 new_ep_f1524 new_ep_f2534 new_ep_f3544 new_ep_f4554
## 1         NA          NA           NA           NA           NA           NA
## 2         NA          NA           NA           NA           NA           NA
##   new_ep_f5564 new_ep_f65 newrel_m014 newrel_m1524 newrel_m2534 newrel_m3544
## 1           NA         NA          NA           NA           NA           NA
## 2           NA         NA          NA           NA           NA           NA
##   newrel_m4554 newrel_m5564 newrel_m65 newrel_f014 newrel_f1524 newrel_f2534
## 1           NA           NA         NA          NA           NA           NA
## 2           NA           NA         NA          NA           NA           NA
##   newrel_f3544 newrel_f4554 newrel_f5564 newrel_f65 package
## 1           NA           NA           NA         NA   tidyr
## 2           NA           NA           NA         NA   tidyr
```

```
who1 <- pivot_longer(data = who,names_to = "key", values_to = "cases", cols = c("new_sp_m014":"newrel_f65"), values_drop_na = TRUE)
print(head(who1,2))
```

```
## # A tibble: 2 x 7
##   country     iso2  iso3   year package key          cases
##   <chr>       <chr> <chr> <int> <chr>   <chr>        <int>
## 1 Afghanistan AF    AFG    1997 tidyr   new_sp_m014      0
## 2 Afghanistan AF    AFG    1997 tidyr   new_sp_m1524    10
```

Task 1.2:

Make variable names consistent and name the new dataset as who2

```
who2 <- who1
who2$key <- str_replace(who1$key, "newrel", "new_rel")
print(head(who2,2))
```

```
## # A tibble: 2 x 7
##   country     iso2  iso3   year package key          cases
##   <chr>       <chr> <chr> <int> <chr>   <chr>        <int>
## 1 Afghanistan AF    AFG    1997 tidyr   new_sp_m014      0
## 2 Afghanistan AF    AFG    1997 tidyr   new_sp_m1524    10
```

Task 1.3:

'%>%.' takes the output of one function and passes it into another function as an argument. This allows us to link a sequence of analysis steps.Name the new dataset as who3 and comment this code.

```
who3 <- who2
who3 <- who2 %>% separate(key, c("new", "type","sexage"), sep="_")
print(head(who3,10))
```

```
## # A tibble: 10 x 9
##    country     iso2  iso3   year package new   type  sexage cases
##    <chr>       <chr> <chr> <int> <chr>   <chr> <chr> <chr>  <int>
##  1 Afghanistan AF    AFG    1997 tidyr   new   sp    m014       0
##  2 Afghanistan AF    AFG    1997 tidyr   new   sp    m1524     10
##  3 Afghanistan AF    AFG    1997 tidyr   new   sp    m2534      6
##  4 Afghanistan AF    AFG    1997 tidyr   new   sp    m3544      3
##  5 Afghanistan AF    AFG    1997 tidyr   new   sp    m4554      5
##  6 Afghanistan AF    AFG    1997 tidyr   new   sp    m5564      2
##  7 Afghanistan AF    AFG    1997 tidyr   new   sp    m65        0
##  8 Afghanistan AF    AFG    1997 tidyr   new   sp    f014       5
##  9 Afghanistan AF    AFG    1997 tidyr   new   sp    f1524     38
## 10 Afghanistan AF    AFG    1997 tidyr   new   sp    f2534     36
```

Task 1.4

Separating sexage into sex and age: Use the function separate(). Name the new dataset who4

```
who4 <- who3
who4 <- who3 %>% separate(sexage, c("sex", "age"), sep="(?<=[mf])(?=[0-9])")
print(head(who4,10))
```

```
## # A tibble: 10 x 10
##    country     iso2  iso3   year package new   type  sex   age   cases
##    <chr>       <chr> <chr> <int> <chr>   <chr> <chr> <chr> <chr> <int>
##  1 Afghanistan AF    AFG    1997 tidyr   new   sp    m     014       0
##  2 Afghanistan AF    AFG    1997 tidyr   new   sp    m     1524     10
##  3 Afghanistan AF    AFG    1997 tidyr   new   sp    m     2534      6
##  4 Afghanistan AF    AFG    1997 tidyr   new   sp    m     3544      3
##  5 Afghanistan AF    AFG    1997 tidyr   new   sp    m     4554      5
##  6 Afghanistan AF    AFG    1997 tidyr   new   sp    m     5564      2
##  7 Afghanistan AF    AFG    1997 tidyr   new   sp    m     65        0
##  8 Afghanistan AF    AFG    1997 tidyr   new   sp    f     014       5
##  9 Afghanistan AF    AFG    1997 tidyr   new   sp    f     1524     38
## 10 Afghanistan AF    AFG    1997 tidyr   new   sp    f     2534     36
```

Task 1.5

Print the first 5 rows and the last 5 rows of the dataset who4

```
print(head(who4,5))
```

```
## # A tibble: 5 x 10
##   country     iso2  iso3   year package new   type  sex   age   cases
##   <chr>       <chr> <chr> <int> <chr>   <chr> <chr> <chr> <chr> <int>
## 1 Afghanistan AF    AFG    1997 tidyr   new   sp    m     014       0
```

```
## 2 Afghanistan AF    AFG    1997 tidyr   new   sp    m     1524     10
## 3 Afghanistan AF    AFG    1997 tidyr   new   sp    m     2534      6
## 4 Afghanistan AF    AFG    1997 tidyr   new   sp    m     3544      3
## 5 Afghanistan AF    AFG    1997 tidyr   new   sp    m     4554      5
```

```
print(tail(who4,5))
```

```
## # A tibble: 5 x 10
##   country  iso2  iso3   year package new   type  sex   age   cases
##   <chr>    <chr> <chr> <int> <chr>   <chr> <chr> <chr> <chr> <int>
## 1 Zimbabwe ZW    ZWE    2013 tidyr   new   rel   f     2534   4649
## 2 Zimbabwe ZW    ZWE    2013 tidyr   new   rel   f     3544   3526
## 3 Zimbabwe ZW    ZWE    2013 tidyr   new   rel   f     4554   1453
## 4 Zimbabwe ZW    ZWE    2013 tidyr   new   rel   f     5564    811
## 5 Zimbabwe ZW    ZWE    2013 tidyr   new   rel   f     65      725
```

Task 1.6

Export who4 as an csv file and save it in the local directory.

```
write.csv(who4,"who", row.names = FALSE)
```

Task 2

Task 2.1

Mean, Median, Mode, Variance and Standard deviation of the dataset

```
print(Nile)
```

```
## Time Series:
## Start = 1871
## End = 1970
## Frequency = 1
##   [1] 1120 1160  963 1210 1160 1160  813 1230 1370 1140  995  935 1110  994 1020
##  [16]  960 1180  799  958 1140 1100 1210 1150 1250 1260 1220 1030 1100  774  840
##  [31]  874  694  940  833  701  916  692 1020 1050  969  831  726  456  824  702
##  [46] 1120 1100  832  764  821  768  845  864  862  698  845  744  796 1040  759
##  [61]  781  865  845  944  984  897  822 1010  771  676  649  846  812  742  801
##  [76] 1040  860  874  848  890  744  749  838 1050  918  986  797  923  975  815
##  [91] 1020  906  901 1170  912  746  919  718  714  740
```

```
mean(Nile)
```

```
## [1] 919.35
```

```
median(Nile)
```

```
## [1] 893.5
```

```
mode(Nile)
```

```
## [1] "numeric"
```

```
var(Nile)
```

```
## [1] 28637.95
```

```
sd(Nile)
```

```
## [1] 169.2275
```

Task 2.2

Minimum, Maximum and Range of the dataset

```
min(Nile)
```

```
## [1] 456
```

```
max(Nile)
```

```
## [1] 1370
```

```
range(Nile)
```

```
## [1]  456 1370
```

Task 2.3

Interquartile (IQR) range and quantile() function

```
IQR(Nile)
```

```
## [1] 234
```

```
quantile(Nile)
```

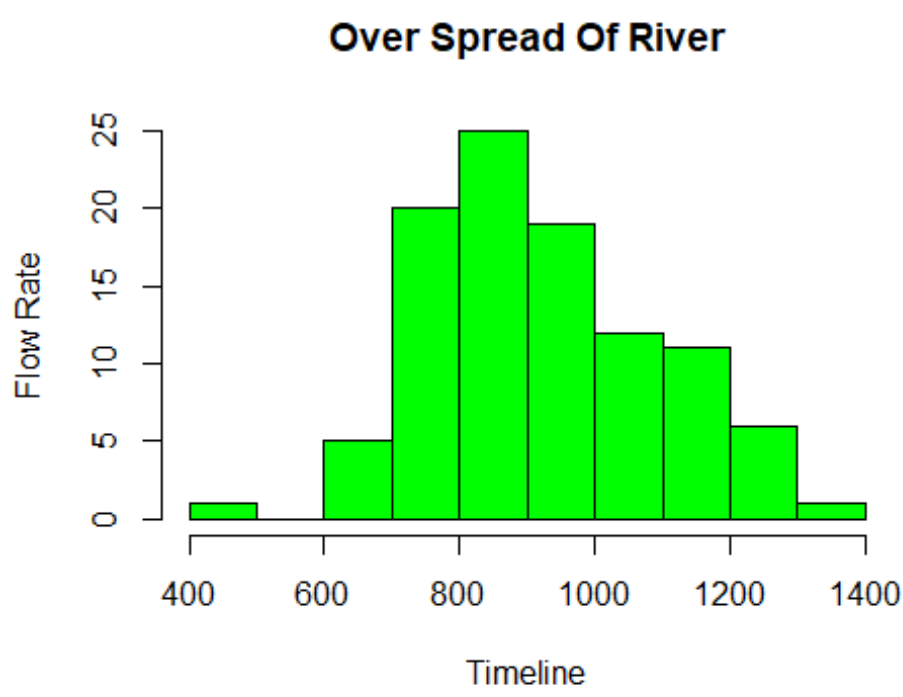
```
##     0%    25%    50%    75%   100%
##  456.0  798.5  893.5 1032.5 1370.0
```

The difference between the first quartile Q1 and the third quartile Q3 is called the interquartile range. A quantile is a sample that is divided into equal groups or sizes.

Task 2.4

Histogram

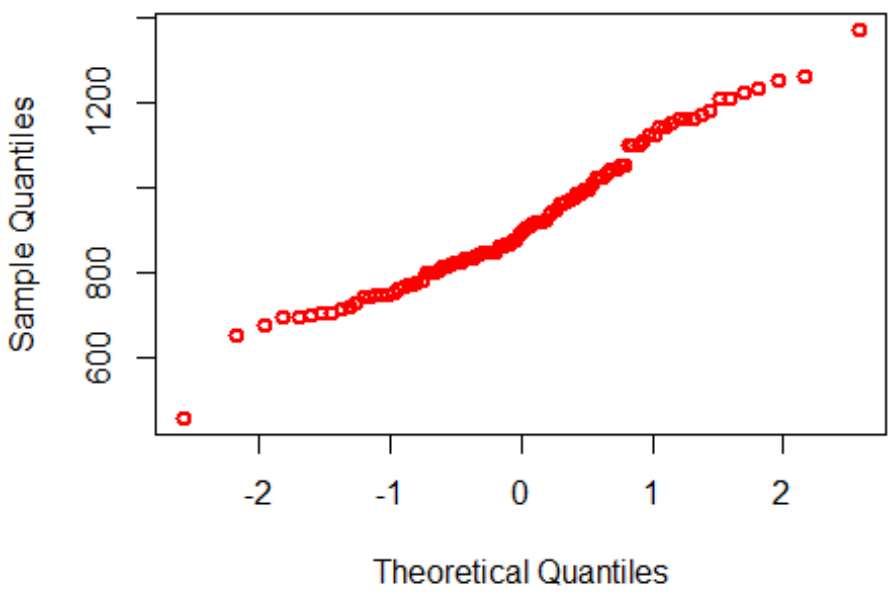
```
hist(Nile,main="Over Spread Of River",xlab="Timeline",ylab="Flow Rate",col="green",border="black")
```



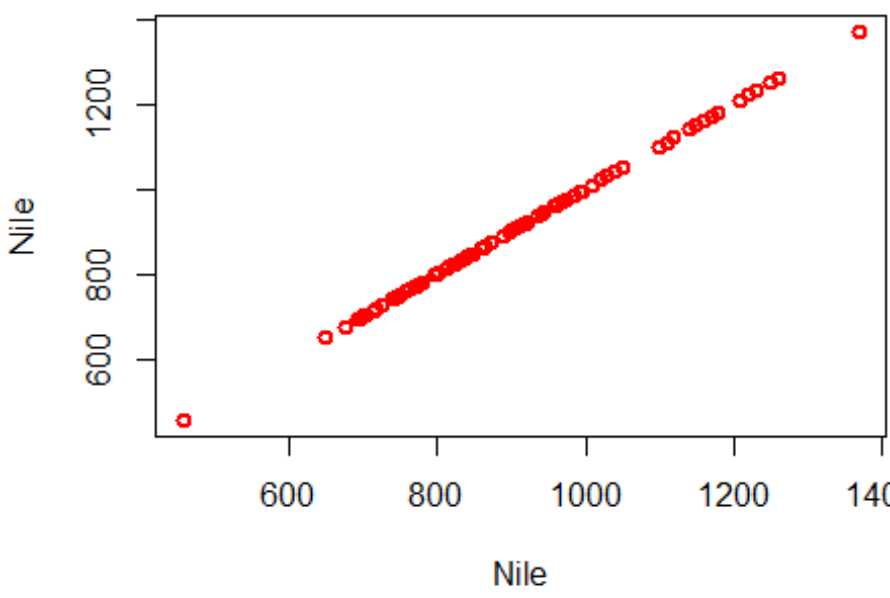
Task 2.5

quantile-quantile plot using qqnorm() function

```
qqnorm(Nile,col="red",lwd = 2)
```




```
qqplot(Nile,Nile,col="red",lwd = 2)
```

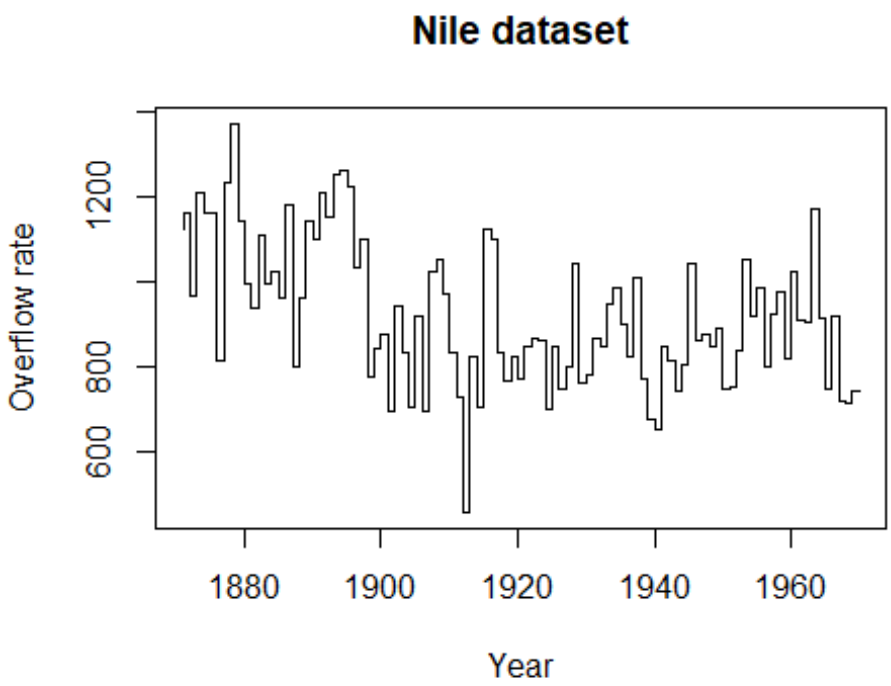


The data is normally distributed, since the points in a Q-Q plot lie on a straight diagonal line. There are not many data points that lie outside the straight diagonal line. Hence, there is not much deviation.

Task 2.6

plot() function to further explore the dataset including arguments such as xlab, ylab, main and type

```
plot(Nile,xlab= "Year", ylab="Overflow rate", main="Nile dataset", type="S")
```



Task 3

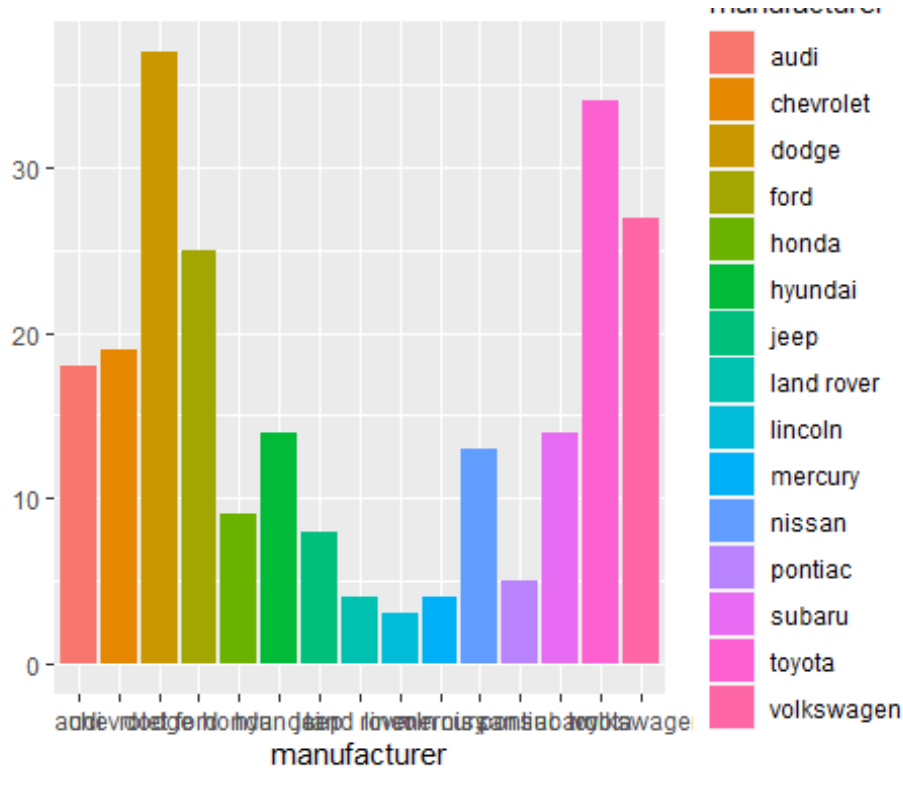
```
library(ggplot2)
```

Task 3.1

Plot to explain which vehicle brand offers the best mpg in both city and in the highway.

```
qplot(manufacturer, data=mpg, geom="bar", fill=manufacturer)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```

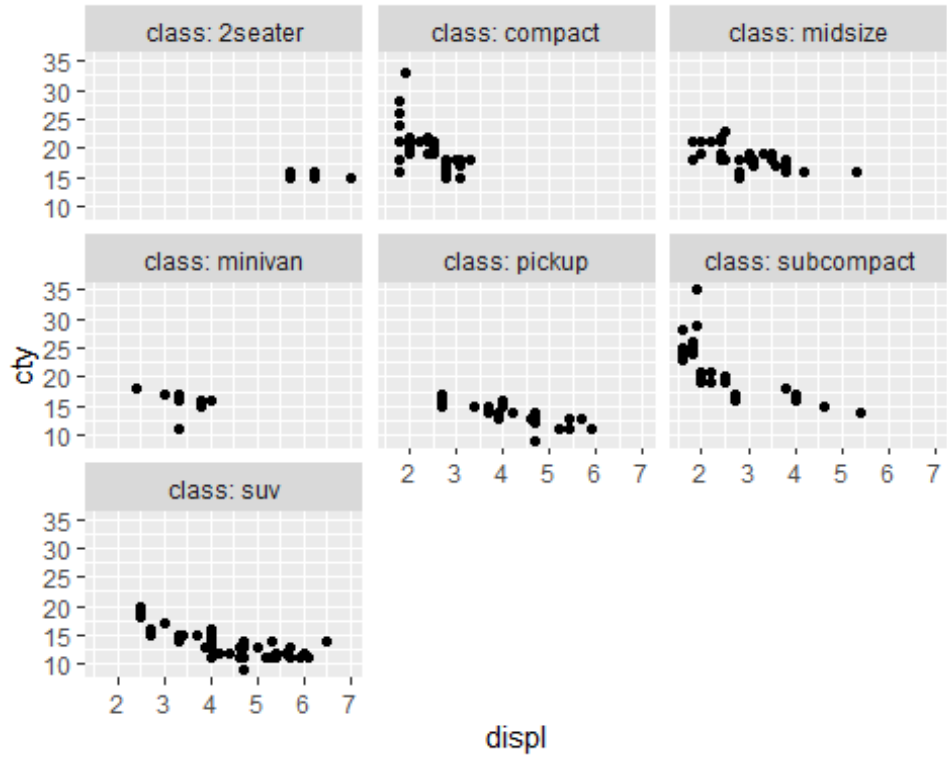


From the above plot, it can be inferred that ‘dodge’ vehicle brand offers the best mpg in both city and highway.

Task 3.2

Plot to explain Which type of car, regarding their displ range (size of engine) has the lowest mpg in the city categorised by the vehicle type

```
ggplot(mpg, aes(displ, cty))+geom_point()+facet_wrap(vars(class),labeller = "label_both")
```

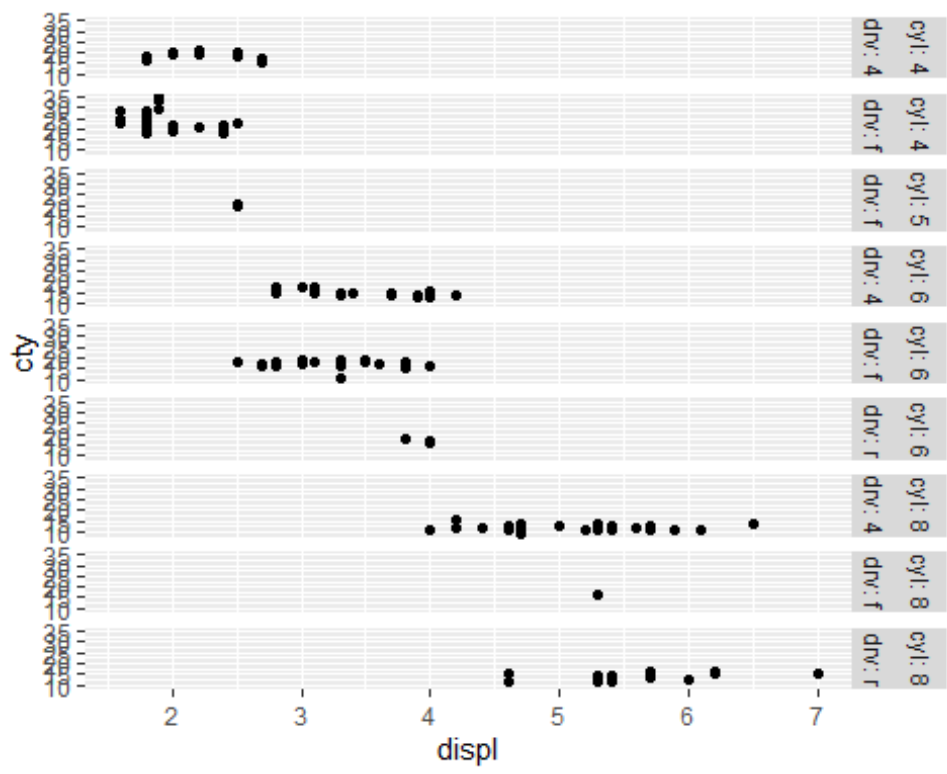


From the above plot,it can be inferred that 'SUV' has the lowest mpg regarding their displ range (size of engine) in city.

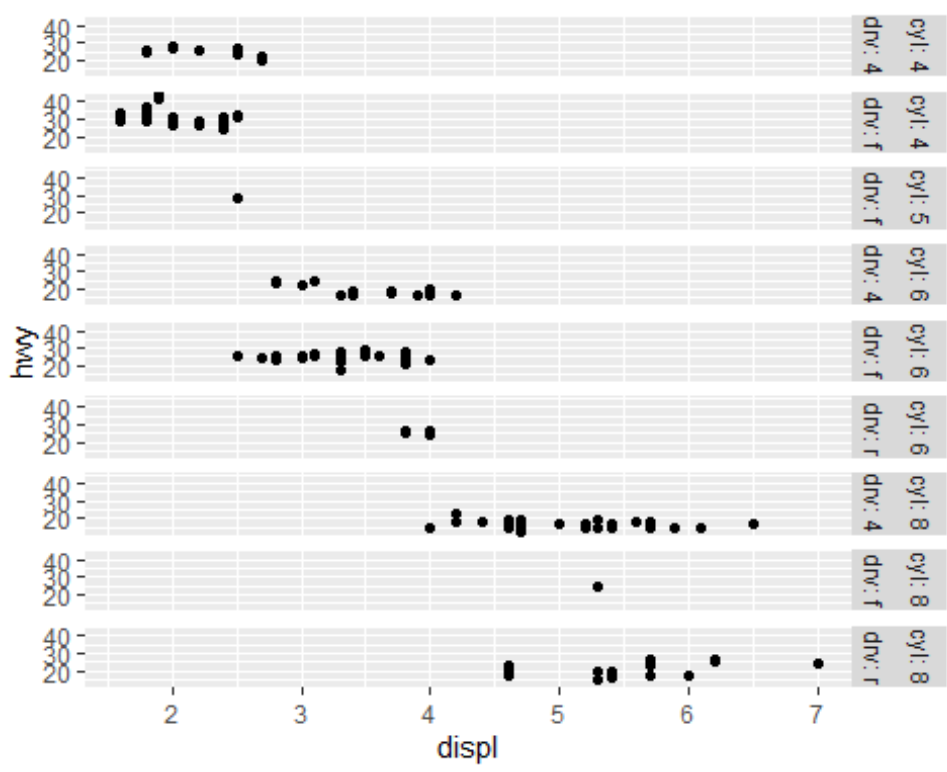
Task 3.3

Plot to explain Which type of car, regarding their displ range (size of engine) has the best mpg performance in both city and highway

```
ggplot(mpg, aes(displ, cty))+geom_point()+facet_grid(vars(cyl,drv),labeller = "label_both")
```




```
ggplot(mpg, aes(displ, hwy))+geom_point()+facet_grid(vars(cyl,drv),labeller = "label_both")
```



From the above plots, it can be inferred that 4-cylinder car with front wheel drive has the highest mpg in both city and highway.