Home Work-1

Section-A

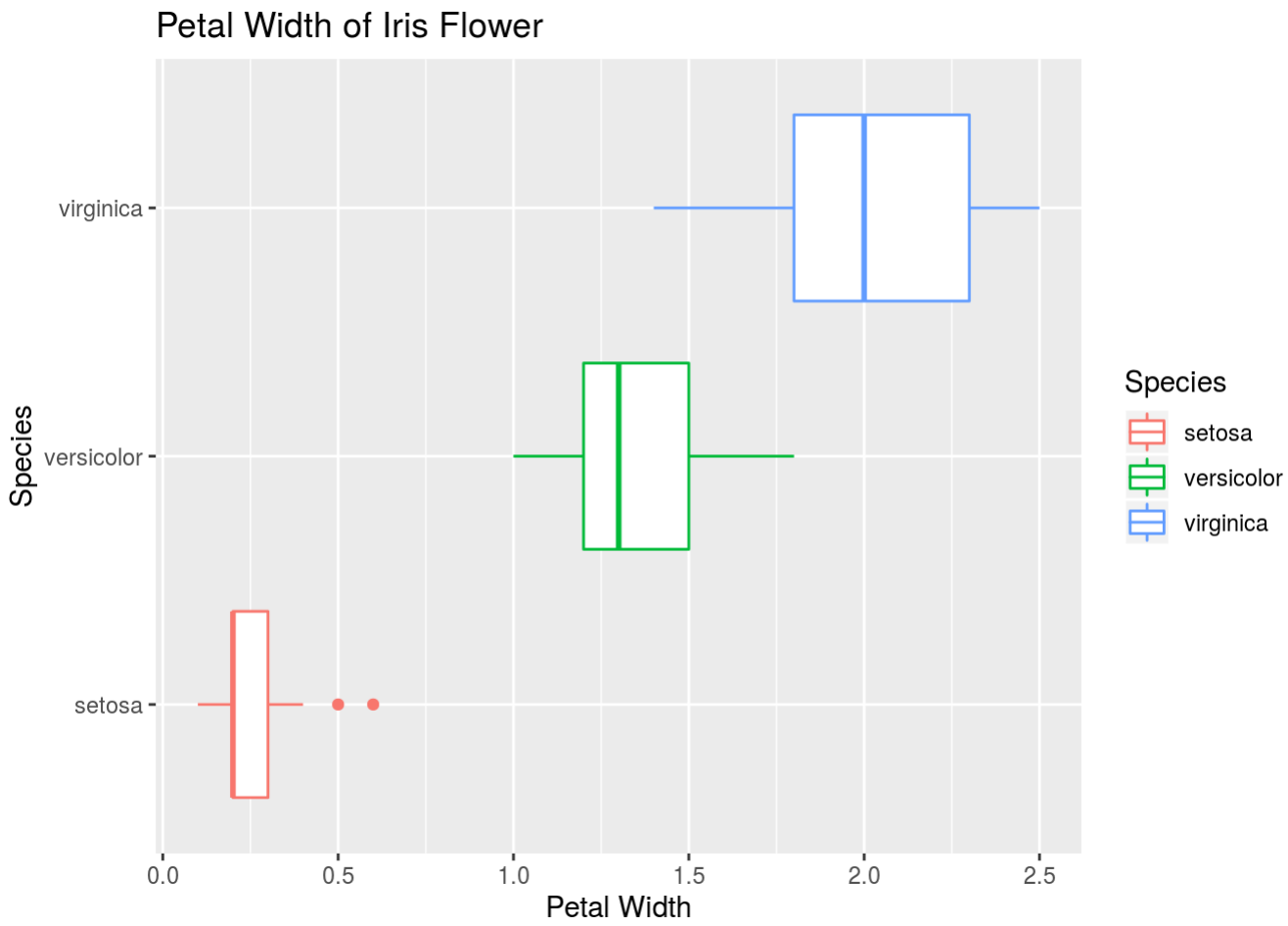
Problem 1 Working on the iris dataset. Create a boxplot to visualize the width of petal among species ordered by virginica, versicolor, setosa. What do you notice about the relationship among different species?

```
##### Loading libraries

library(ggplot2)
```

```
##### Loading Iris dataset

iris<-datasets::iris
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
##### Creating a box plot

box<-ggplot(data=iris,mapping=aes(x=Species,y=Petal.Width,color=Species))+geom_boxplot(stat='box
plot')+xlab('Species')+ylab('Petal Width')+ggtitle('Petal Width of Iris Flower')+coord_flip()

box
```



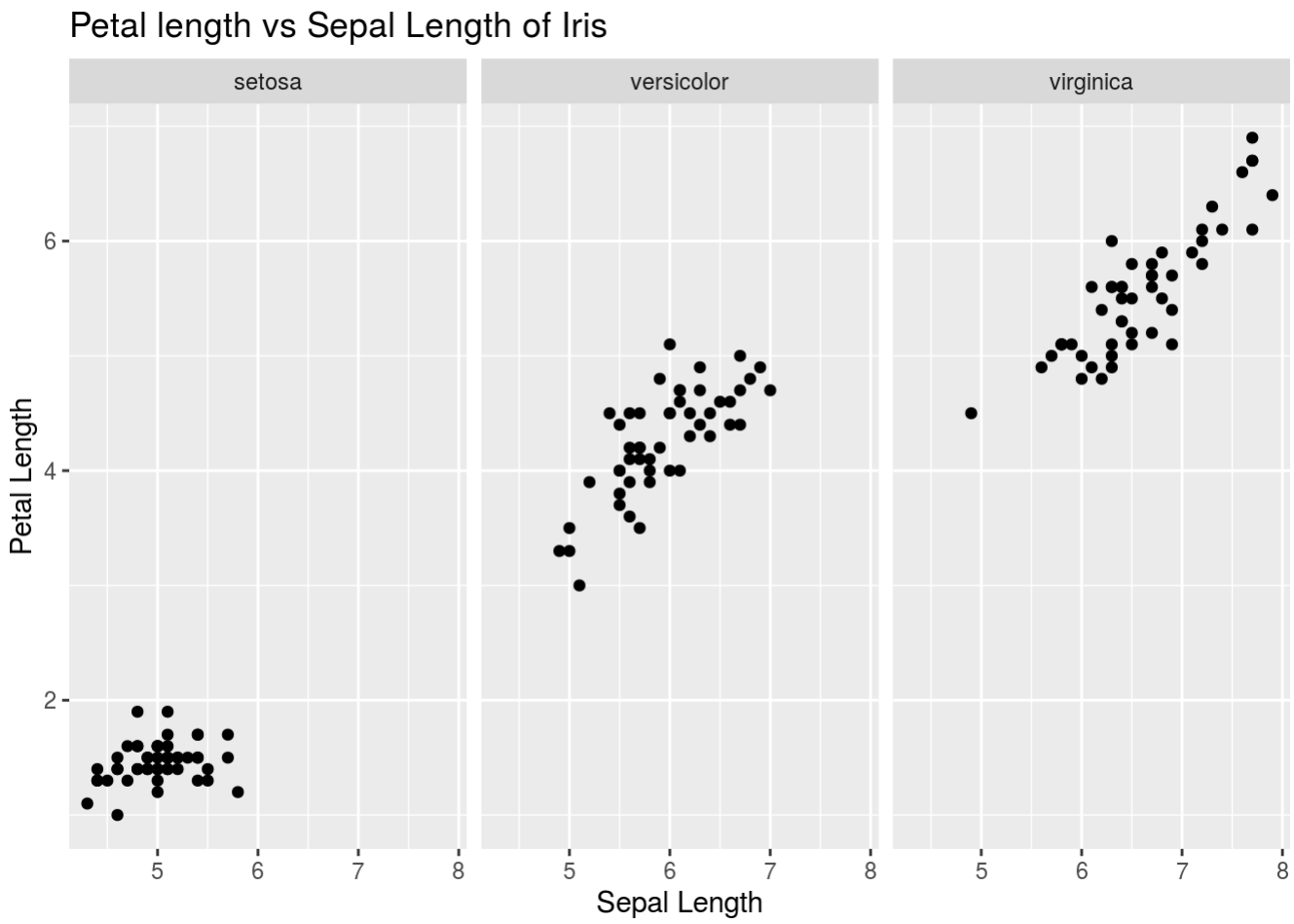
Interpretation:

The boxplot shows the varition of petal width of iris flower across their species.From the above boxplot the setosa species of iris flower has the smallest width with the median around 0.20 cm which also contain two outliers, the next largest petal width of the iris species belong to the versicolor with median of with around 1.3 cm, the largest of petal of width iris species belong veriginica with petal width around 2 cm. We can interpret from the box plot that iris flower with larger petal width might belong to virginica,smaller width might belong to setosa and width of medium range might belong to vericolor. Most of width of the setosa species of iris are right skewed, the petal width of versicolor are more likely to be right skewed, whereas the virginica has most of the petal length are symmetric, and is slightly right skewed.

Problem 2

Create a scatter plot to visualize the length of Sepal versus the length of Petal and facet this scatter plot by Species. What do you notice about the relationship between these two variables?

```
##### Creating a scatter plot

scatter<-ggplot(data=iris,aes(x=Sepal.Length,y=Petal.Length))+geom_point(stat='identity')+facet_
grid(.~Species)+xlab('Sepal Length')+ylab('Petal Length')+ggtitle('Petal length vs Sepal Length
 of Iris')

scatter
```

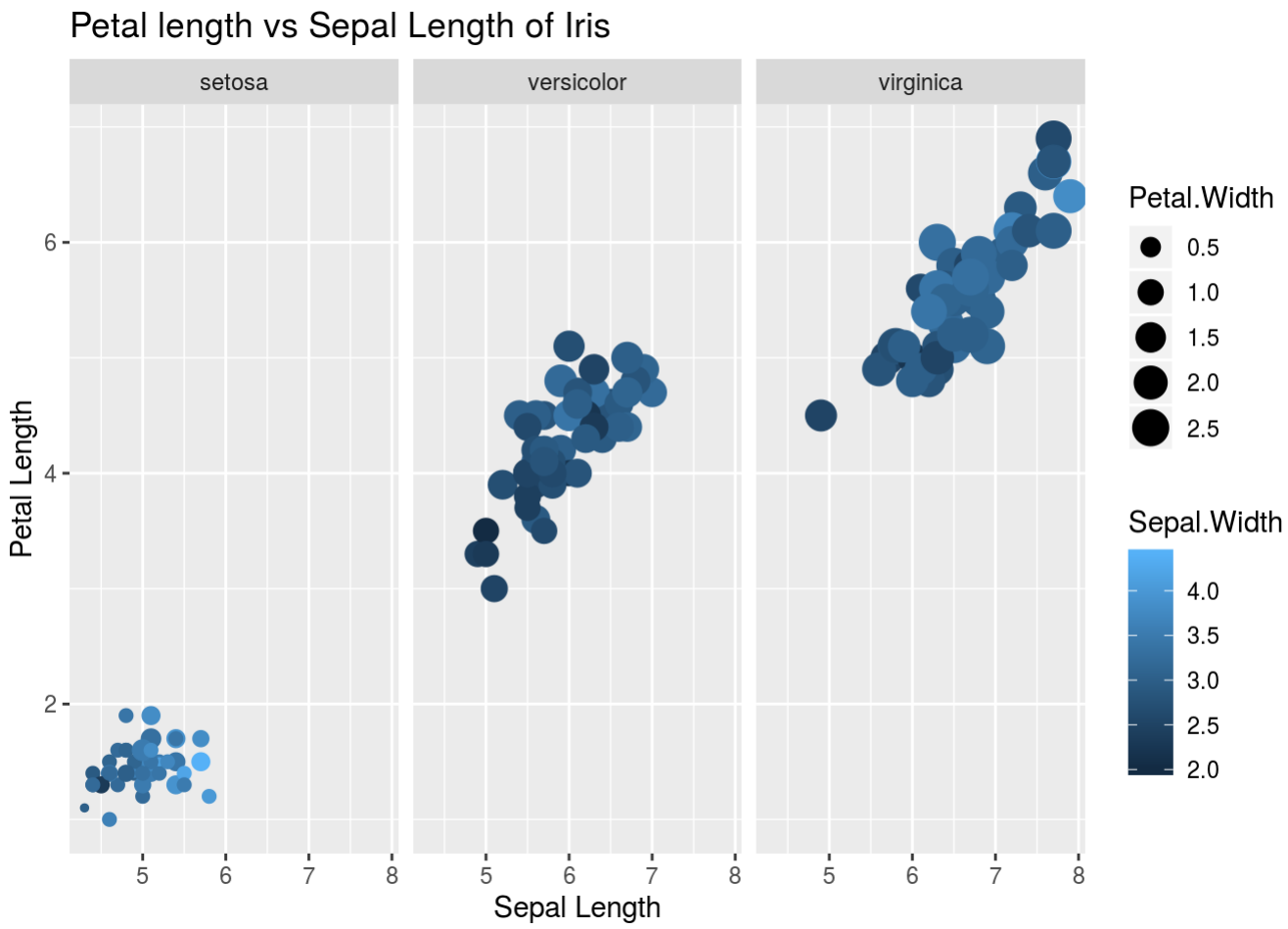


Interpretation:

The scatter plot represents Petal Width vs Sepal length of Iris species. The majorly classified species are setosa, versicolor and virginica. The flower with smaller petal length belong to setosa with maximum sepal length around 6cm. The viriginica flower has the maximum sepal and petal length among three species of iris flower, whereas versicolor has the sepal and petal length around 6 cm. The sepal length vs petal length of setosa shows a non-linear relationship ,whereas the versicolor and virginica indcates a postive linear relationship between sepal length and petal length.

Problem 3 Still working on the above scatter plot. Mapping Sepal.Width and Petal.Width as color and size ,respectively, in the same plot. Interpret the relationship among the four variables.

```
##### Plotting scatter plot with color and size
scatter_color<-ggplot(data=iris,aes(x=Sepal.Length,y=Petal.Length))+geom_point(stat='identity',a
es(color=Sepal.Width,size=Petal.Width))+facet_grid(.~Species)+xlab('Sepal Length')+ylab('Petal L
ength')+ggtitle('Petal length vs Sepal Length of Iris')

scatter_color
```



Interpretation

The scatter plot represents Petal Width vs Sepal length of Iris species. The majorly classified species are setosa, versicolor and virginica. The cirles represents the size of petal width across three species, whereas lighter color represents bigger sepal length, while darker color represents bigger sepal length of iris species. The circles with lighter in color and bigger in size has the maximum petal and sepal length whereas smaller and darker circle represents smaller speal and petal length. Most of the smaller circles belong to setosa species of iris flower, whereas the larger petal width belong to virginica species. The versicolor has the sepal length and petal length around 6 to 7 cms.

Section B

Problem 1 With the ggplot2 dataset mpg creating a new data frame: mpg1, only contained columns: manufacturer, model, trans, drv, hwy, and class. Then rename drv as driveType, hwy as hwyMPG, trans as TransmissionType.

```
data('mpg')
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(… f        18    29 p     comp…
## 2 audi         a4      1.8  1999     4 manua… f        21    29 p     comp…
## 3 audi         a4      2    2008     4 manua… f        20    31 p     comp…
## 4 audi         a4      2    2008     4 auto(… f        21    30 p     comp…
## 5 audi         a4      2.8  1999     6 auto(… f        16    26 p     comp…
## 6 audi         a4      2.8  1999     6 manua… f        18    26 p     comp…
```

```
m<-c('manufacturer','model','trans','drv','hwy','class')
mpg1<-data.frame(mpg[m])
names(mpg1)[names(mpg1)=='hwy']<-'hwyMPG'
names(mpg1)[names(mpg1)=='drv']<-'driveType'
names(mpg1)[names(mpg1)=='trans']<-'TransmissionType'
head(mpg1)
```

```
##   manufacturer model TransmissionType driveType hwyMPG   class
## 1         audi    a4         auto(l5)         f     29 compact
## 2         audi    a4       manual(m5)         f     29 compact
## 3         audi    a4       manual(m6)         f     31 compact
## 4         audi    a4         auto(av)         f     30 compact
## 5         audi    a4         auto(l5)         f     26 compact
## 6         audi    a4       manual(m5)         f     26 compact
```

Problem 2 Using the above new data frame. Create another new data frame: mpg2, we would like to keep the information only from the manufacturers: ford, honda, hyundai, jeep, nissan, and toyota. Then we would like to find all the suv with 4 wheel drive type, and the highway miles per gallon should be higher than 18. Then put this data frame into a new list “suv”, with the new element name “suv18”.

```
##### Creating a new dataframe from the above dataframe and putting it into list

a<-c('ford','honda','hyundai','jeep','nissan','toyota')
mpg2<- data.frame(mpg1[mpg1$manufacturer==a,])
head(mpg2)
```

```
##     manufacturer           model TransmissionType driveType hwyMPG
## 79          ford    explorer 4wd       manual(m5)         4     19
## 85          ford f150 pickup 4wd       manual(m5)         4     17
## 91          ford         mustang       manual(m5)         r     26
## 97          ford         mustang       manual(m5)         r     23
## 104        honda           civic         auto(l4)         f     32
## 111      hyundai          sonata         auto(l4)         f     30
##          class
## 79         suv
## 85      pickup
## 91  subcompact
## 97  subcompact
## 104 subcompact
## 111    midsize
```

```
mpg2<-mpg2[mpg2$class=='suv'& mpg2$driveType==4 & mpg2$hwyMPG >18,]
mpg2
```

```
##     manufacturer              model TransmissionType driveType hwyMPG
## 79          ford       explorer 4wd       manual(m5)         4     19
## 124         jeep grand cherokee 4wd         auto(l5)         4     19
## 174       toyota        4runner 4wd       manual(m5)         4     20
##     class
## 79    suv
## 124   suv
## 174   suv
```

```
##### creating a list
suv<-list(mpg2)
names(suv)<-('suv18')
suv
```

```
## $suv18
##     manufacturer              model TransmissionType driveType hwyMPG
## 79          ford       explorer 4wd       manual(m5)         4     19
## 124         jeep grand cherokee 4wd         auto(l5)         4     19
## 174       toyota        4runner 4wd       manual(m5)         4     20
##     class
## 79    suv
## 124   suv
## 174   suv
```

Problem 3 (Not allowed to use apply/mapply/sapply/lapply family) Write your own version of apply function family with the following format: applyFun(df, f, …) • df: Selected column/s from one data.frame (this data.frame can be from a list) • f: Proper function applied on the selected column/s

This created function should return a message(vector) as the form: function's name, selected column/s' name, and the result/s.

```
functions<-'mean'
apply<-function(df,func,...)
{
new_func<- match.fun(functions)
dataframe_1<- new_func(df[[1]],...)
dataframe_2<- as.vector(c(functions, 'mileage of cars in', colnames(df),'is',dataframe_1),mode=
'any')
return(dataframe_2)
}

df<-data.frame(mpg$hwy)
df1<-data.frame(suv$suv18$ hwyMPG)
apply(df,func)
```

```
## [1] "mean"              "mileage of cars in" "mpg.hwy"
## [4] "is"                "23.4401709401709"
```

```
apply(df1,func)
```

```
## [1] "mean"               "mileage of cars in" "suv.suv18.hwyMPG"
## [4] "is"                 "19.3333333333333"
```