Understanding ggplot using Iris

K.V.Raghava 20241

04/02/2021

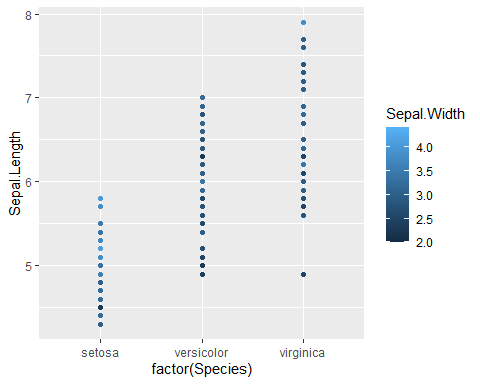
# loading library  
library(tidyverse)  
# Data Layer   
p\_data = as\_tibble(iris)  
##Generating the Normal Graph Empty one  
head(p\_data)

## # A tibble: 6 x 5  
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## <dbl> <dbl> <dbl> <dbl> <fct>   
## 1 5.1 3.5 1.4 0.2 setosa   
## 2 4.9 3 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 3.6 1.4 0.2 setosa   
## 6 5.4 3.9 1.7 0.4 setosa

plot = ggplot(data = p\_data, aes(y = Sepal.Length, x = factor(Species),   
col = Sepal.Width))  
#Can be done like this aes function is for adding the X-axis and Y-axis

*Using geom\_point()*

plot + geom\_point()

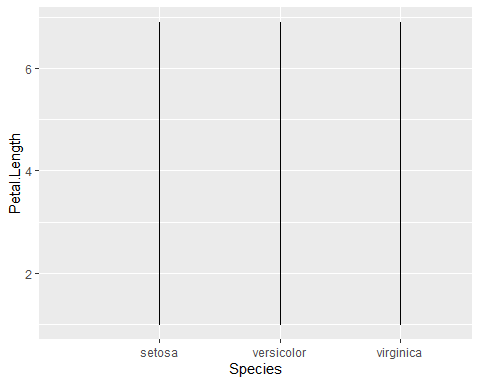


##Adding the layer where the actula plotting is done  
##Plotting is done between Species and Sepal Length

This graph suggest that Virginica is having the highest Sepal Width

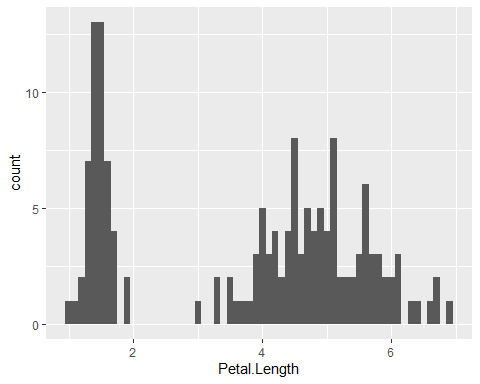
*Using geom\_density()*

p\_data = as\_tibble(iris)  
plot = ggplot(data = p\_data,aes(y=Petal.Length,x=Species,col = Petal.Width))  
plot+geom\_density()

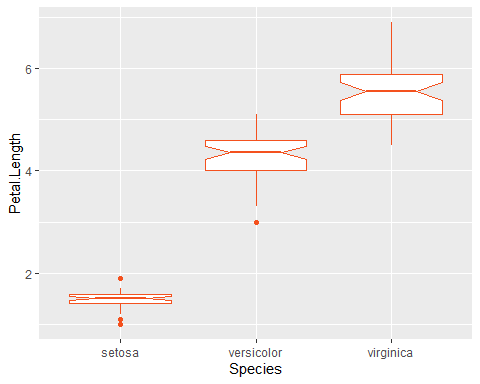


This Graph says that there is a uniformity with respect to the petal length and Species *Using geom\_histogram()*

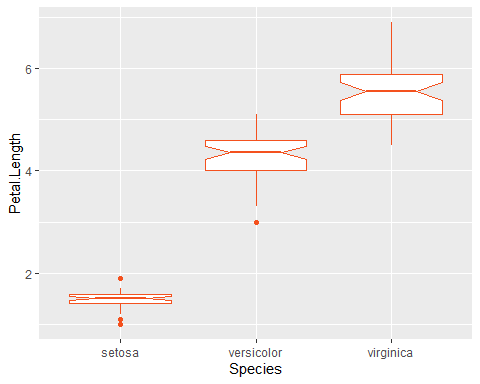
ggplot(data = p\_data,aes(Petal.Length))+  
geom\_histogram(binwidth =0.1)

 This graph infers petal length lies between 0 and 2 has the highest number *Using geom\_boxplot()*

ggplot(data = p\_data,aes(y=Petal.Length,x=Species))+  
geom\_boxplot(fill = "white", color= "#f4511e",notch=TRUE)

 This graph infers that there is less consistency and more outliers for Setosa *Using ggsave()*

ggplot(data = p\_data,aes(y=Petal.Length,x=Species))+  
geom\_boxplot(fill = "white", color= "#f4511e",notch=TRUE)

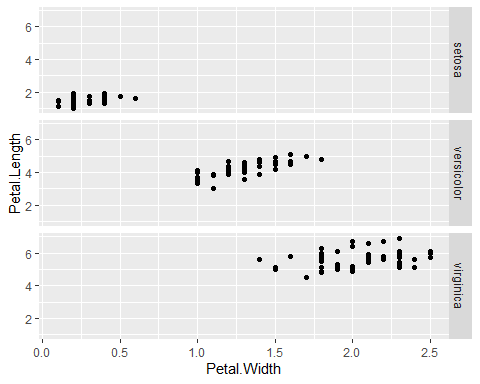


ggsave("Box\_Plot.pdf")#Used to save the plot

## Saving 5 x 4 in image

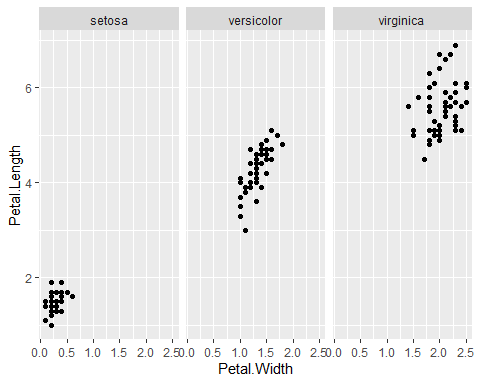
*Using facet\_grid()*

ggplot(data = p\_data,aes(y=Petal.Length,x=Petal.Width,  
))+geom\_point()+facet\_grid(Species~.)

 This graph infers that the Versicolor data is less scattered

*Using facet\_wrap()*

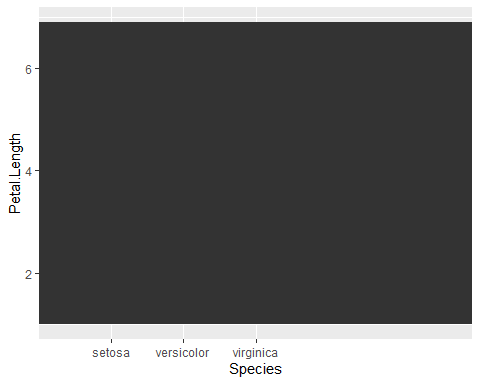
ggplot(data = p\_data,aes(y=Petal.Length,x=Petal.Width,  
))+geom\_point()+facet\_wrap(Species~.)



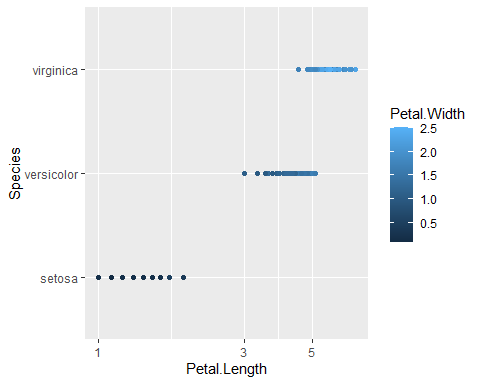
##This is same as facet\_grid but ensure that the screen is used in better way

This graph infers that the ascending order of size is Setosa,Versicolor,Virginica *Using stat\_density*

plot = ggplot(data = p\_data,aes(y=Petal.Length,x=Species,col = Petal.Width))  
plot+stat\_density()

 *Using scale\_x\_log10()*

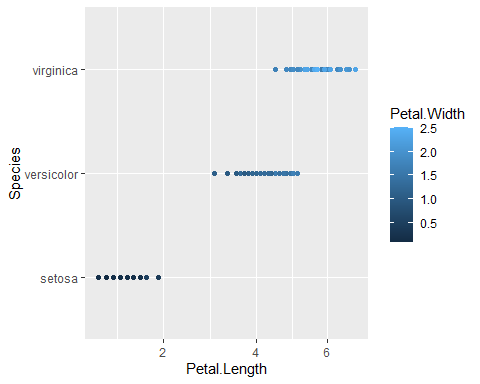
ggplot(data = p\_data,aes(x=Petal.Length,y=Species,col = Petal.Width))+geom\_point()+scale\_x\_log10()



##Used for applying the Log\_10 on x-axis

This graph infers that on log10scale the setosa scatter is less and size is also less than 0.5 *Using scale\_x\_sqrt()*

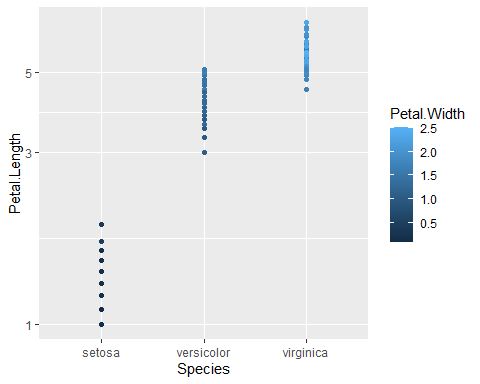
ggplot(data = p\_data,aes(x=Petal.Length,y=Species,  
col = Petal.Width))+geom\_point()+scale\_x\_sqrt()



##Used for applying the sqrt on x-axis  
##Nothing much different may be the values are too small

*Using scale\_y\_log10()*

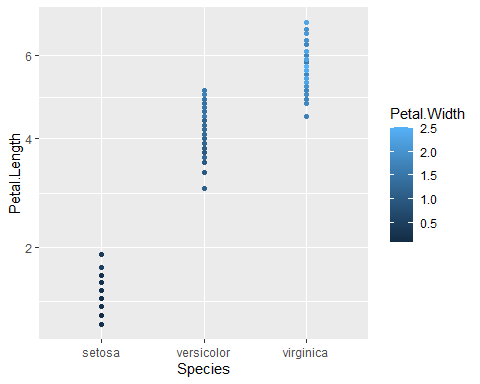
ggplot(data = p\_data,aes(y=Petal.Length,x=Species,col = Petal.Width))+geom\_point()+scale\_y\_log10()



##Used for applying the Log\_10 on y-axis

*Using scale\_y\_sqrt()*

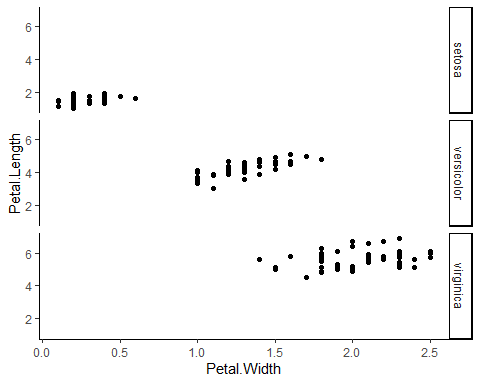
ggplot(data = p\_data,aes(y=Petal.Length,x=Species,  
col = Petal.Width))+geom\_point()+scale\_y\_sqrt()



##Used for applying the sqrt on x-axis  
##Nothing much difference I think it might be my wrong implementation  
##But the datapoints are getting nearer to each other  
##This graph infers that on log10scale the setosa scatter is less and size is also less than 0.5

*Using theme\_classic()*

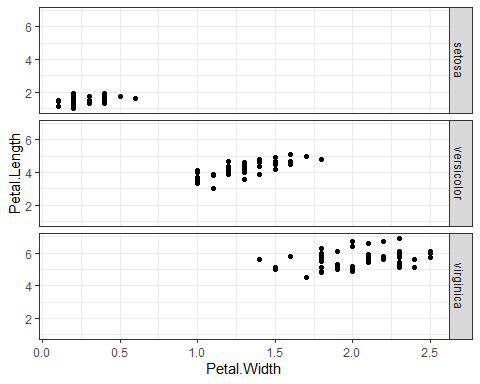
ggplot(data = p\_data,aes(y=Petal.Length,x=Petal.Width,  
))+geom\_point()+facet\_grid(Species~.)+theme\_classic()



##It gives the classic(old) look for the plot

*Using theme\_classic()*

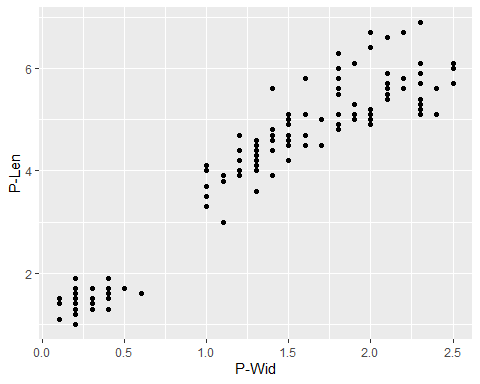
ggplot(data = p\_data,aes(y=Petal.Length,x=Petal.Width,  
))+geom\_point()+facet\_grid(Species~.)+theme\_bw()



##It gives the minimilistic look for the plot with gridlines

*Using scale\_x\_continuous and scale\_y\_continuous*

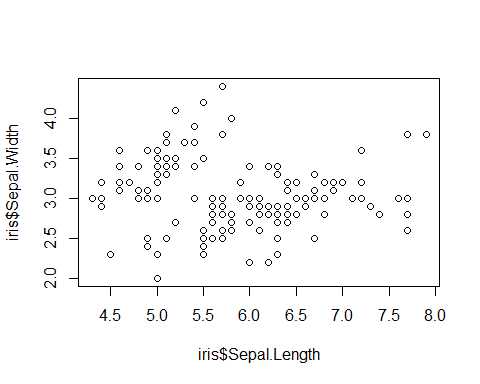
ggplot(data = p\_data,aes(y=Petal.Length,x=Petal.Width,  
))+geom\_point()+scale\_x\_continuous("P-Wid")+scale\_y\_continuous("P-Len")



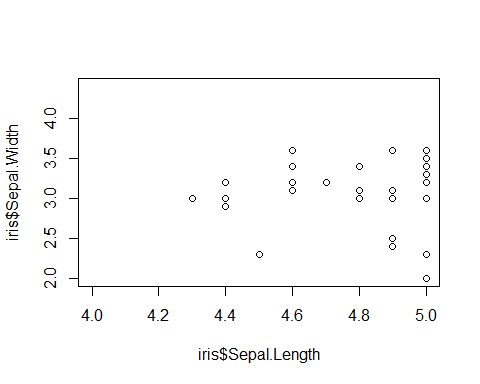
##scale\_x\_continuous() and scale\_y\_continuous() are the default scales for   
##continuous x and y aesthetics. There are three variants that set the trans   
##argument for commonly used transformations:  
## scale\_\*\_log10(), scale\_\*\_sqrt() and scale\_\*\_reverse().

Nothing much difference (I think it might be my wrong implementation) May be the values are too small *Using xlim and ylim*Part of Basic R

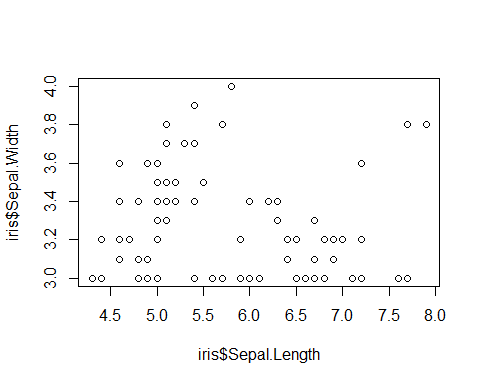
plot(iris$Sepal.Length,iris$Sepal.Width)



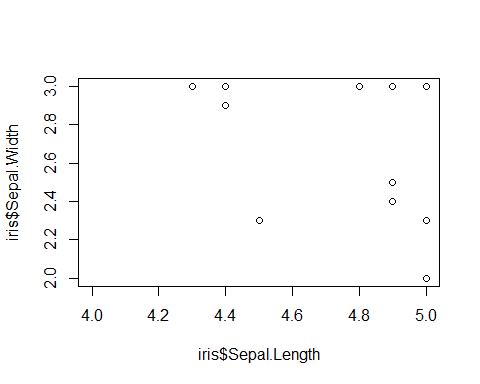
#Normal graph with no limit.Kindly note the x axis is from 4.5 to 8 y-axis 2.0 to 4  
plot(iris$Sepal.Length,iris$Sepal.Width, xlim = c(4,5))



#From this plot, the absolute minimum Sepal Length is 4.2. The x-axis size is decrease  
plot(iris$Sepal.Length,iris$Sepal.Width, ylim = c(3,4))



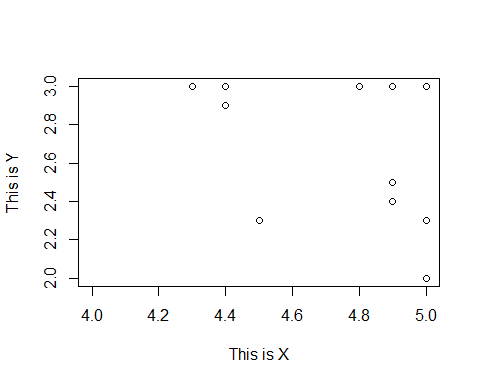
#From this plot,the max sepalwidth has the sepal length between 5.5 to 6.0  
#These both can be used together  
plot(iris$Sepal.Length,iris$Sepal.Width, xlim=c(4,5),ylim = c(2,3))



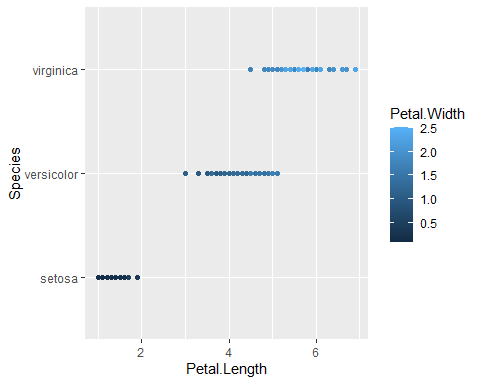
#The smalles size of the flower is 10.55

*Using xlab and ylab* (Generally These are the lables for X-axis and y-axis)Part of Basic R

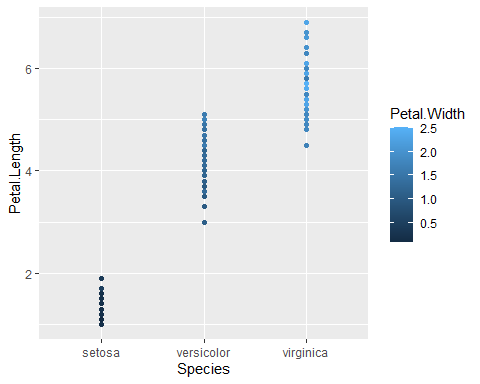
plot(iris$Sepal.Length,iris$Sepal.Width, xlim=c(4,5),ylim = c(2,3),xlab="This is X"  
,ylab="This is Y")

 *Using Coord\_flip function*

plot+geom\_point()+coord\_flip()



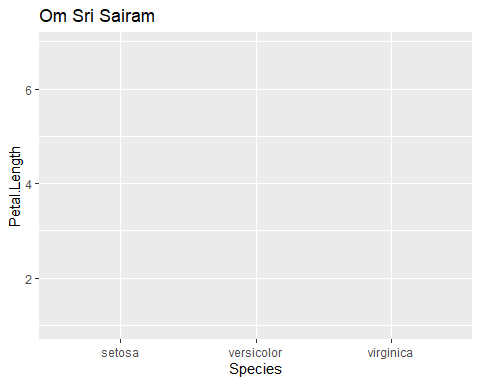
#Simply flips the x and y axis  
plot+geom\_point()+coord\_cartesian()



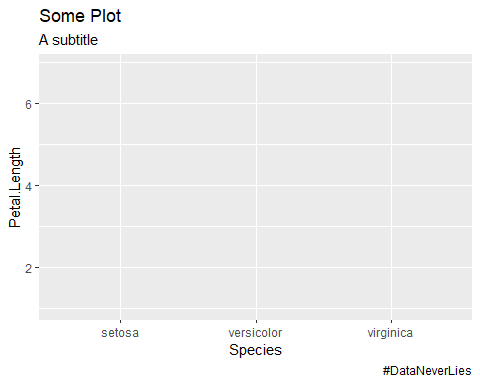
#Used to convert the given grpah to cartesian plane enables us to Zoom in and Out

*Using labs*

plot + labs(title = "Om Sri Sairam")



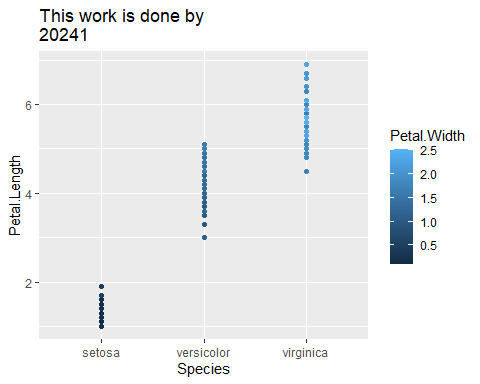
plot + labs(title = "Some Plot ", subtitle = "A subtitle")+labs(caption = "#DataNeverLies")



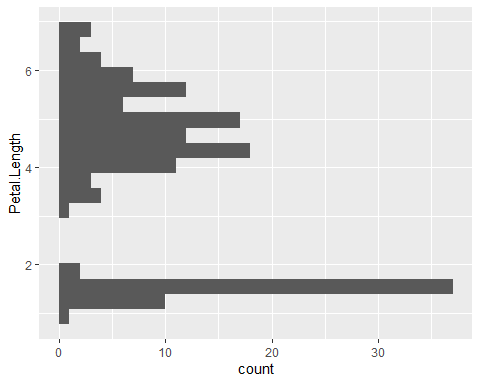
#Used for creating the labels like title,subtitle,caption,tag too

*Using ggtile*

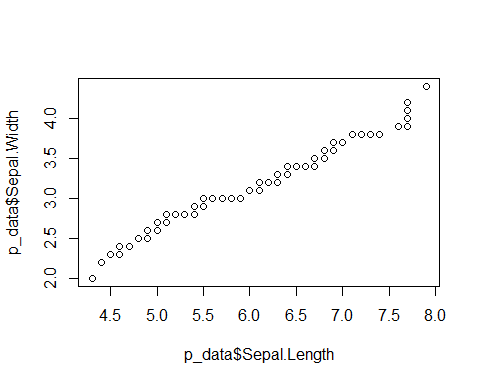
plot+geom\_point()+ggtitle("This work is done by\n\t20241")

 *Using Dodge*

ggplot(data = p\_data,aes(y=Petal.Length))+ geom\_histogram(bins=20,position = "dodge")

 *Using qqplot*

qqplot(p\_data$Sepal.Length,p\_data$Sepal.Width)



##Used for checking if the normality is present

all the points does not fall approximately along this reference line, we cannot assume normality.