Assignment\_2

Jitesh Rawat

2/4/2021

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

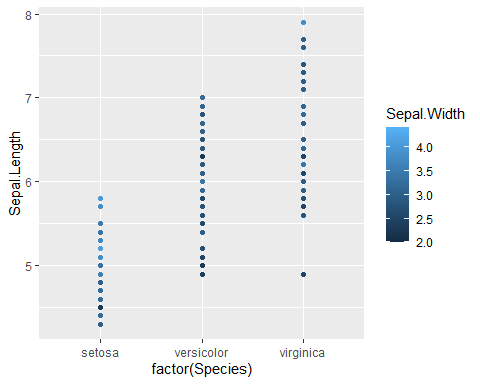
## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

idata = as\_tibble(iris)#loading iris dataset  
head(idata)#taking a look at data

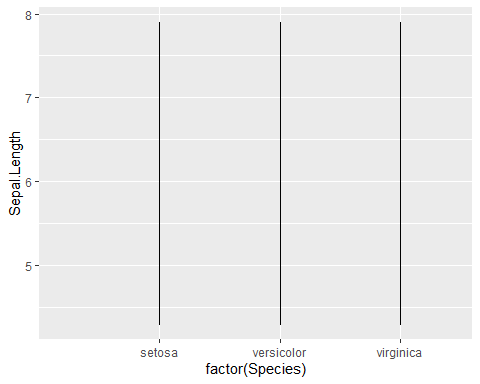
## # A tibble: 6 x 5  
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## <dbl> <dbl> <dbl> <dbl> <fct>   
## 1 5.1 3.5 1.4 0.2 setosa   
## 2 4.9 3 1.4 0.2 setosa   
## 3 4.7 3.2 1.3 0.2 setosa   
## 4 4.6 3.1 1.5 0.2 setosa   
## 5 5 3.6 1.4 0.2 setosa   
## 6 5.4 3.9 1.7 0.4 setosa

plot = ggplot(data = idata, aes(y = Sepal.Length, x = factor(Species),col= Sepal.Width))#creating ggplot   
plot2= ggplot(data = idata, aes(y = Petal.Length, x = factor(Species),col= Petal.Width))

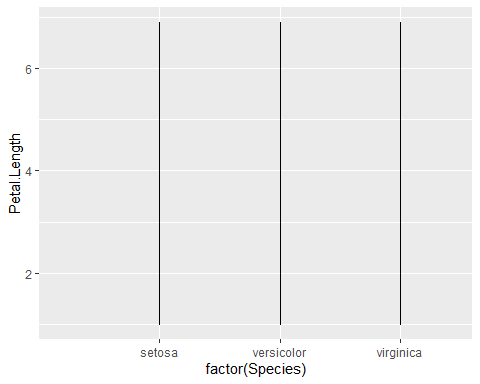
plot+geom\_point()#the use of geom\_point() function to plot the datas sepal length/sepalwidth



plot+geom\_density()



plot2+geom\_density()#Ploting density

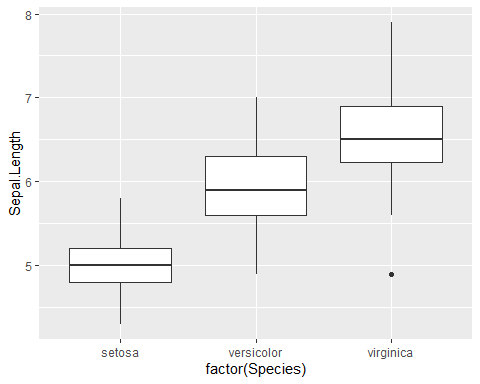


ggplot(idata, aes(x= Sepal.Length))+geom\_histogram()#representing data in terms of histograms

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



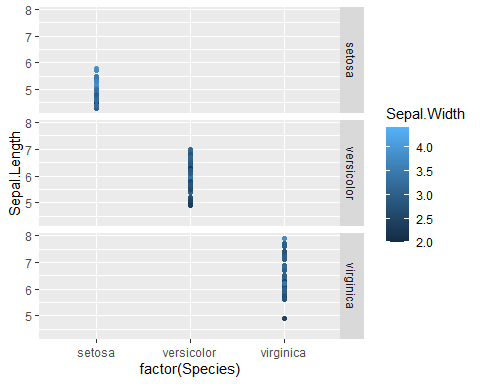
plot+geom\_boxplot()#representing data in terms of boxplot



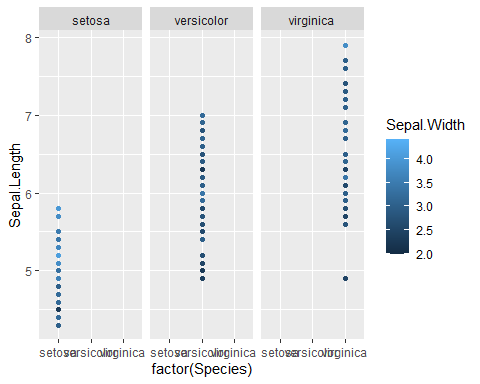
p <- ggplot(iris, aes(sample = Sepal.Width))  
p + geom\_qq() + stat\_qq\_line()#geom qq for checking the normality of data



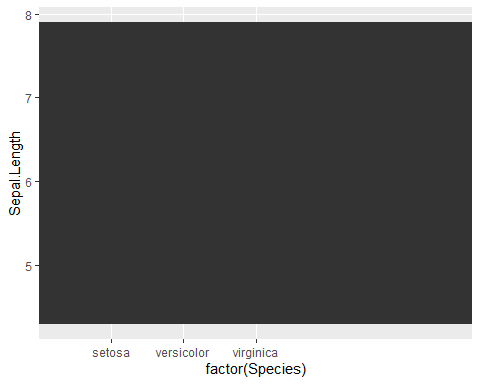
q <- plot+geom\_point()  
q + facet\_grid(Species~.)#faceting using facet grid



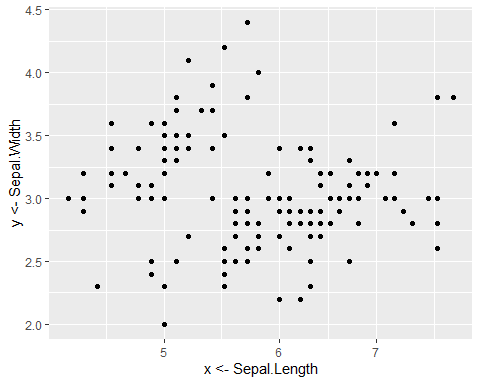
q+facet\_wrap(vars(Species))#facteing using facet wrap



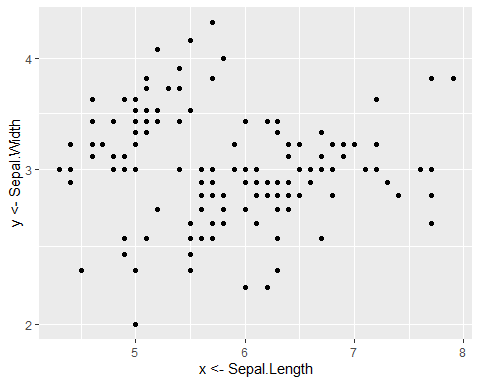
plot + stat\_density()#plotting the statistical density of plot



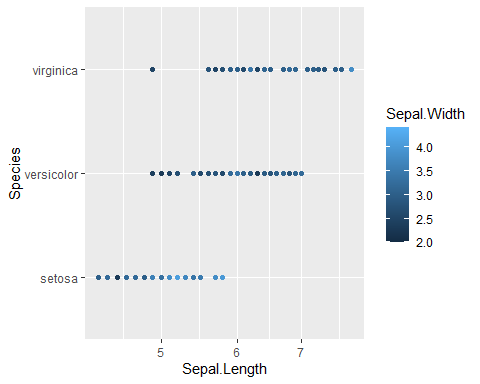
#applying logarithm to coorodinates  
plotx <- ggplot(iris, aes(x <- Sepal.Length,y <- Sepal.Width)) + geom\_point()  
plotx + scale\_x\_log10()



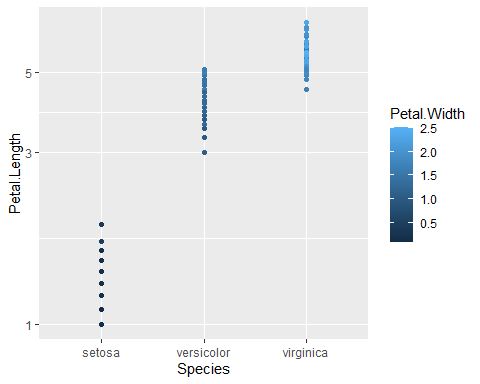
plotx + scale\_y\_log10()



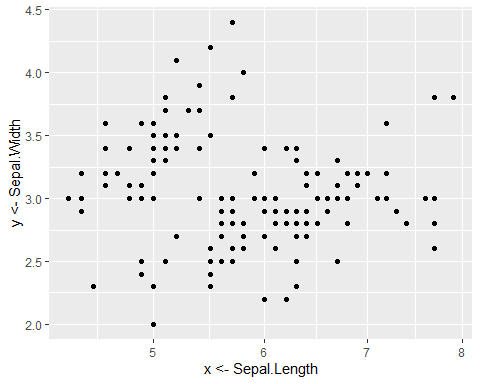
ggplot(data = idata,aes(x=Sepal.Length,y=Species,col = Sepal.Width))+geom\_point()+scale\_x\_log10()



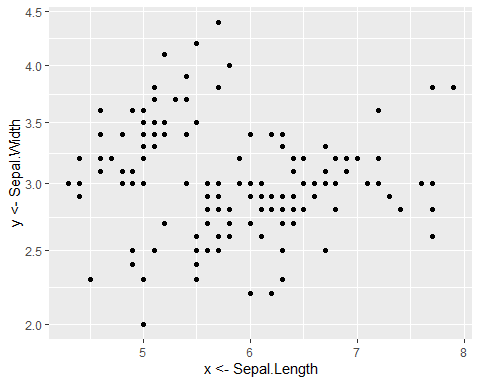
ggplot(data = idata,aes(y=Petal.Length,x=Species,col = Petal.Width))+geom\_point()+scale\_y\_log10()



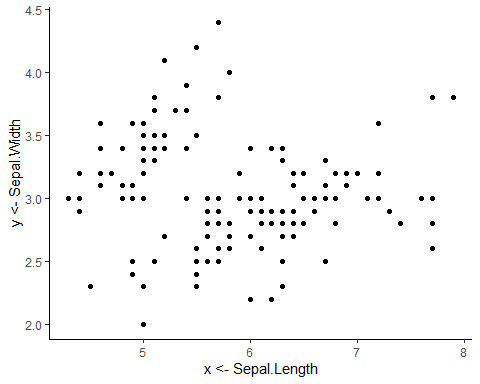
#applying square root on coordinate axes  
plotx+scale\_x\_sqrt()



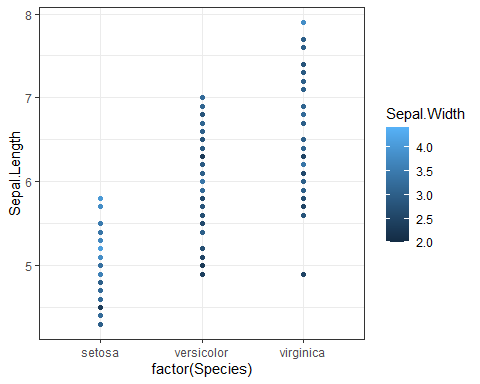
plotx+scale\_y\_sqrt()



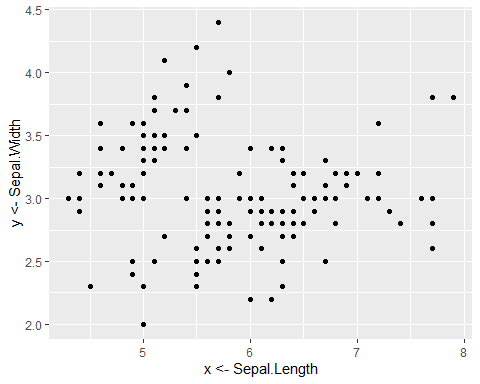
plotx + theme\_classic()#theming



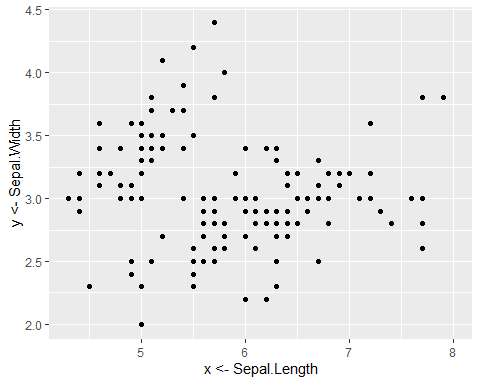
plot + geom\_point() + theme\_bw()#theming



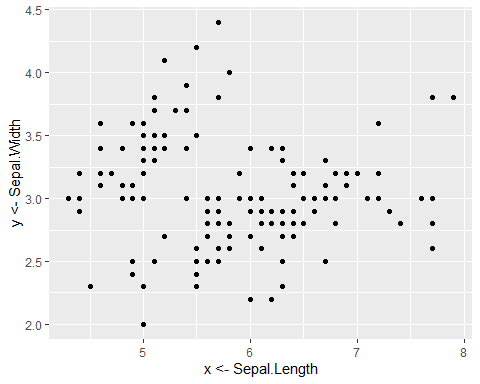
plotx + scale\_x\_continuous()#scaling but his is default scaling that geom point or ggplot applies



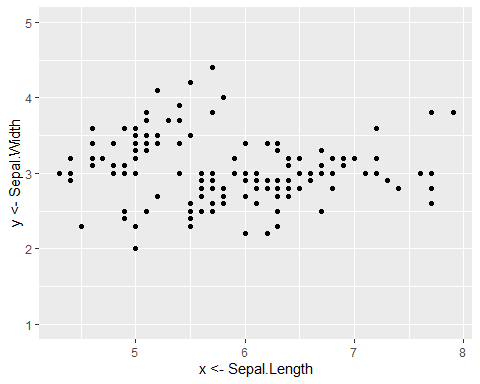
plotx + xlim(NA,8)#applying limitation in x axis up to 8 which covers entire data



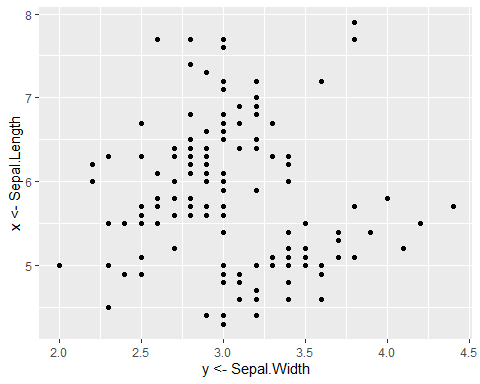
plotx + scale\_y\_continuous()#scaling the y axis



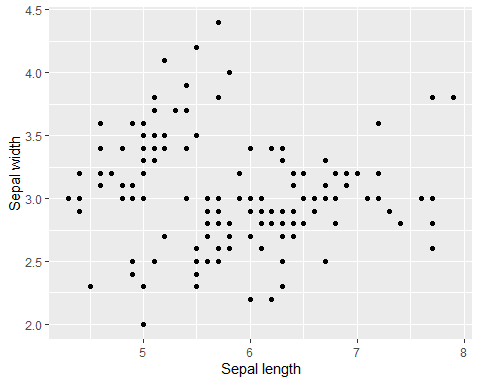
plotx+ ylim(1,5)#applying the limit to y axis



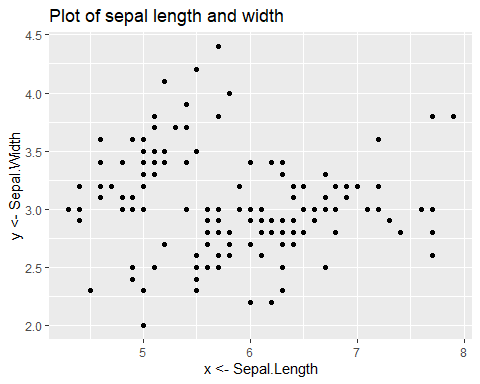
plotx+ coord\_flip()



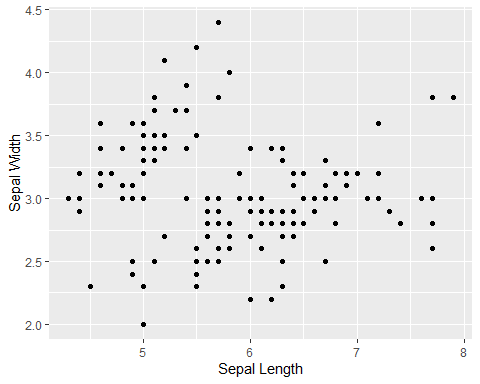
plotx+ labs(x= "Sepal length",y= "Sepal width")#applying label using label function



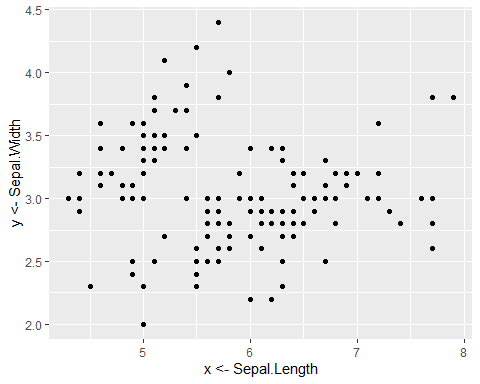
plotx+ ggtitle("Plot of sepal length and width")#giving title to plot



plotx + xlab("Sepal Length") + ylab("Sepal Width")#labelling



plotx+coord\_cartesian(expand = TRUE)#it is often used to zoom plot



ggplot(data = idata,aes(x=Petal.Width, y=Petal.Length, position\_dodge(width= NULL)))+ geom\_point()#

