# Everything about plotting in R with ggplot2

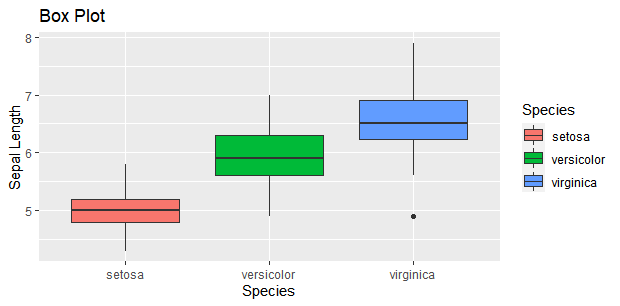
**BASICS**

**​#plotting with ggplot2**

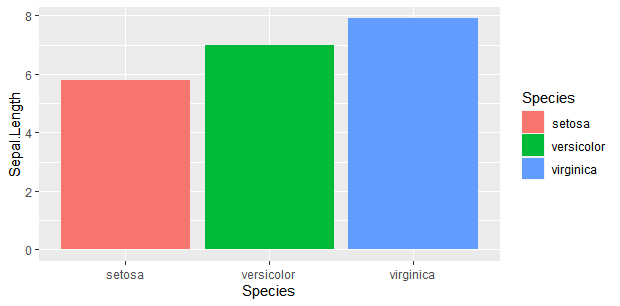
install.packages("ggplot2")

library(ggplot2)

help("ggplot")



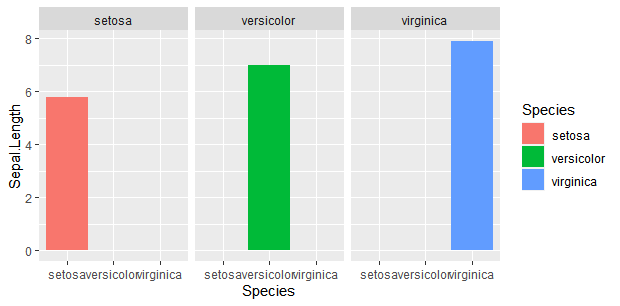
**#barplot**

ggplot(data = x, aes(x=x$Species, y=x$Sepal.Length, fill=x$Species))+ geom\_bar(position="dodge", stat="identity")

ggplot(x,aes(Species,Sepal.Length,fill=Species)) +

geom\_bar(position = 'dodge',stat = 'identity')+

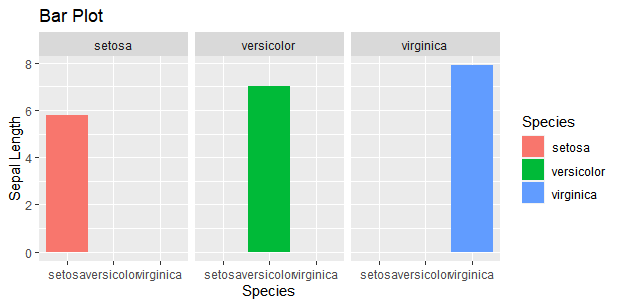
facet\_wrap(x$Species)



ggplot(x,aes(Species,Sepal.Length,fill=Species)) +

geom\_bar(position = 'dodge',stat = 'identity')+

facet\_wrap(x$Species) + labs(title = "Bar Plot",x="Species",y="Sepal Length")



ggplot(x,aes(Species,Sepal.Length,fill=Species)) +

geom\_boxplot()+

labs(title = "Box Plot",x="Species",y="Sepal Length")

| A ggplot2 geom **tells the plot how you want to display your data in R**. ... A geom defines the layout of a ggplot2 layer  fill **defines the colour with which a geom is filled**, whereas colour defines the colour with which a geom is outlined (the shape's "stroke", to use Photoshop language)  Position adjustments **are used to adjust the position of each geom**.  stat\_identity is one possible calculation where y-values are placed at their actual values  Facet wraps are **a useful way to view individual categories in their own graph**. For example, if you wanted to make a separate graph for each cut measuring the price (y axis) for each clarity (x axis), you could add facet\_wrap(~cut) . |
| --- |

**INTERMEDIATE**

install.packages("ggplot2")

library(ggplot2)

data()

View(BOD)

#View(mtcars)

#nrow(mtcars)

#ncol(mtcars)

#str(mtcars)

BOD$demand >- as.numeric(BOD$demand) #changing the type of variables.....numerical,date,etc…

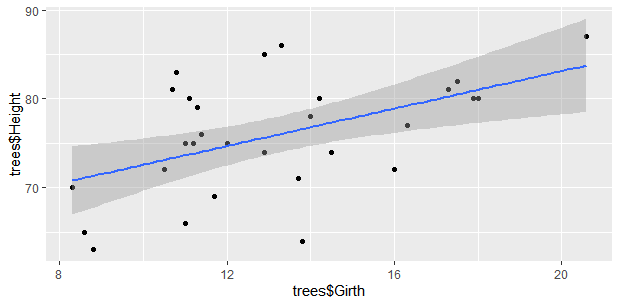
BOD$demand | BOD$Time >- as.numeric(BOD$demand)

#always name data in small alphabets without any spaces and special character

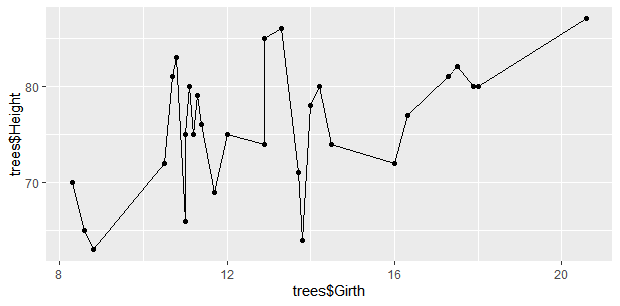
#ggplot tricks and tips

data(trees)

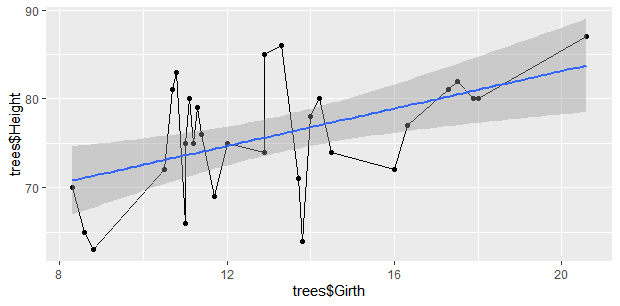
ggplot(trees , aes(trees$Girth , trees$Height)) + geom\_point() + geom\_smooth(method = "lm")



ggplot(trees , aes(trees$Girth , trees$Height)) + geom\_line() + geom\_point()



ggplot(trees , aes(trees$Girth , trees$Height)) + geom\_line() + geom\_point() + geom\_smooth(method = 'lm') #”lm” = linear model



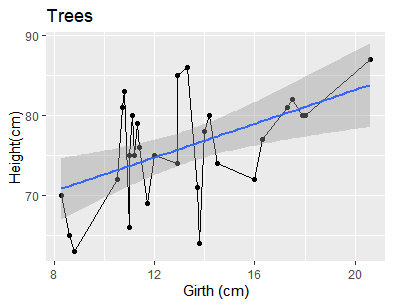
Black line = actual points

Blue line = trends

Shaded area around blue line = confidence interval(standard deviation)

ggplot(trees , aes(trees$Girth , trees$Height)) + geom\_line() + geom\_point() + geom\_smooth(method = 'lm') +

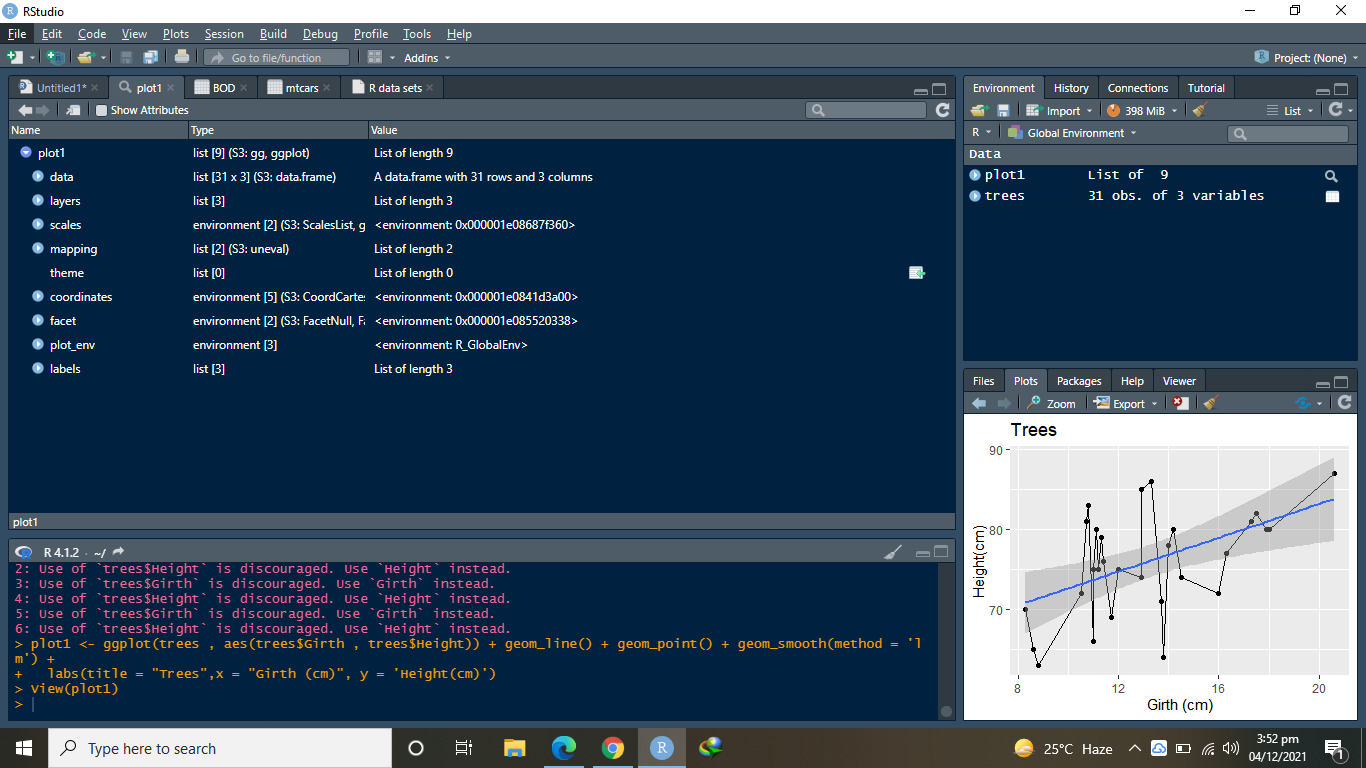
labs(title = "Trees",x = "Girth (cm)", y = 'Height(cm)')



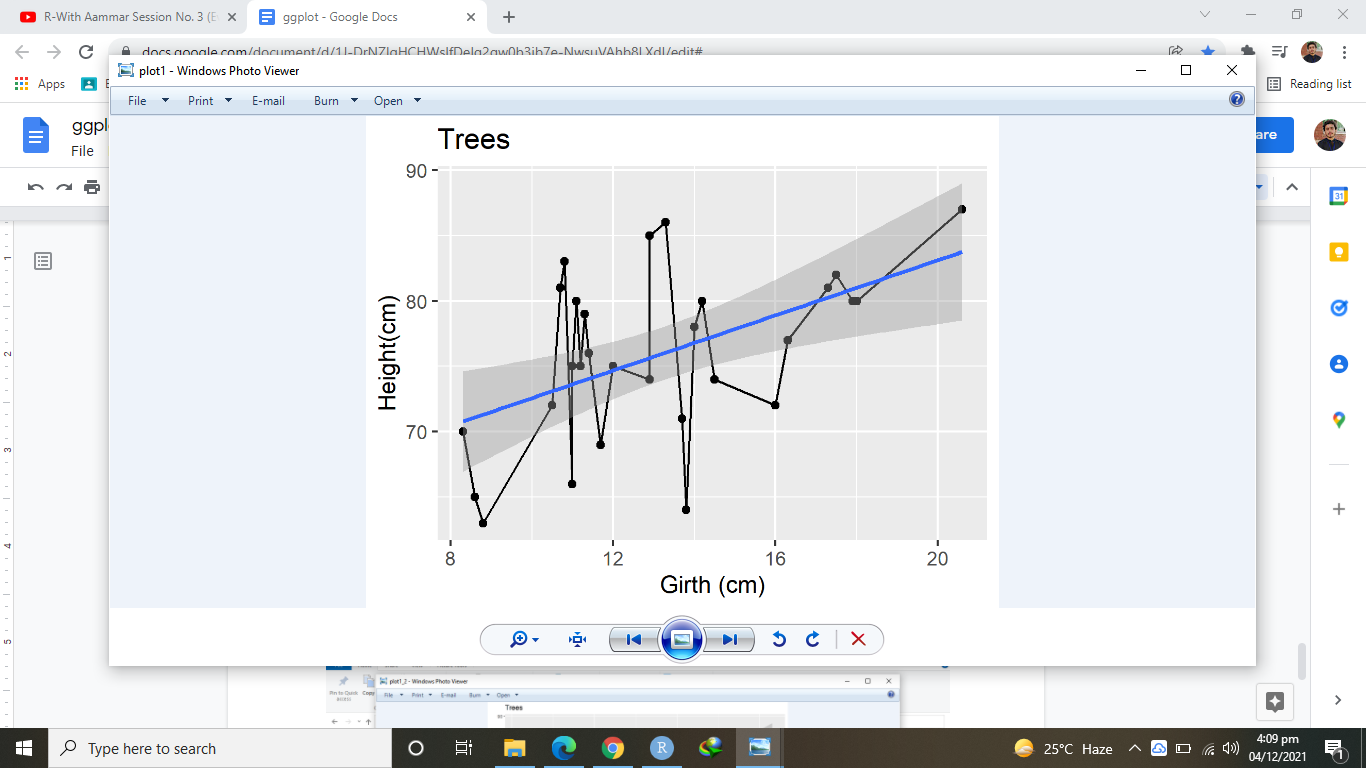
Plot1 <- ggplot(trees , aes(trees$Girth , trees$Height)) + geom\_line() + geom\_point() + geom\_smooth(method = 'lm') +

labs(title = "Trees",x = "Girth (cm)", y = 'Height(cm)')

#assigning variable to code for getting grammar of graphics



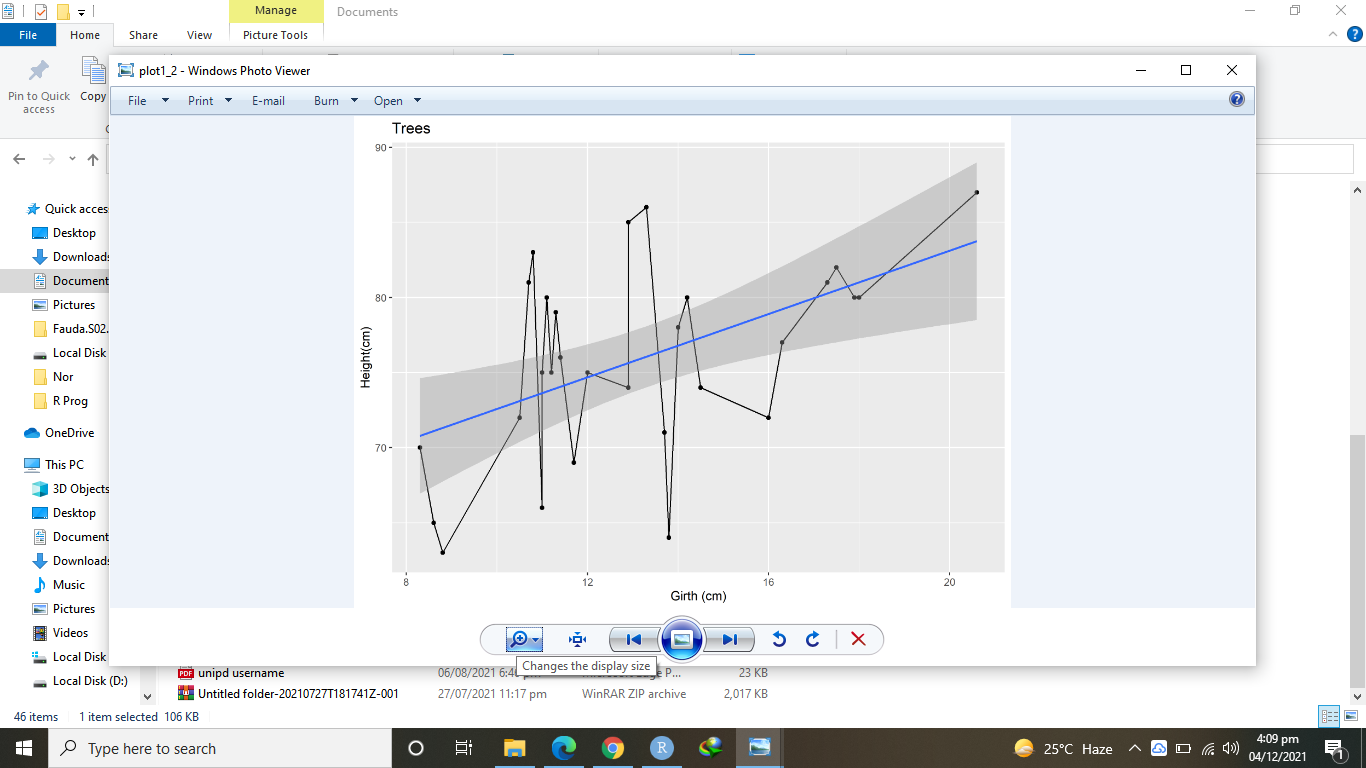
plot1 + ggsave('plot1.tiff')



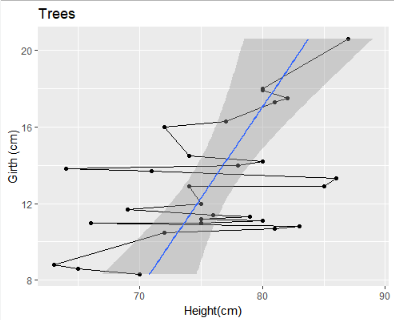
#tiff file not supported so screenshots were taken

plot1 + ggsave('plot1\_2.tiff', units = 'in', width = 8, height = 6, dpi = 300, compression = 'lzw')

#dpi=resoution



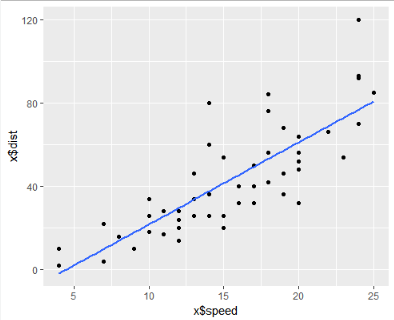
plot1 + coord\_flip()

****

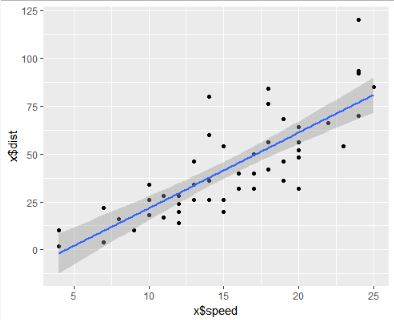
#scatter\_plot

x <- cars

ggplot(x,aes(x$speed,x$dist)) + geom\_point() + geom\_smooth(method = "lm" , se = F,level = .95) #se = standard error



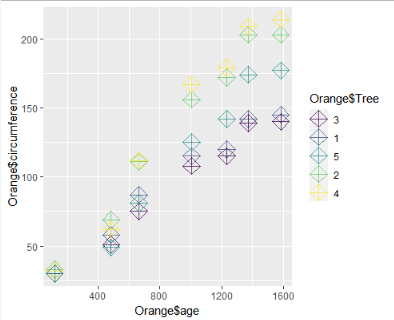
ggplot(x,aes(x$speed,x$dist)) + geom\_point() + geom\_smooth(method = "lm" , se = T,level = .95) #se = standard error

****

#scatter\_plot with multiple lines

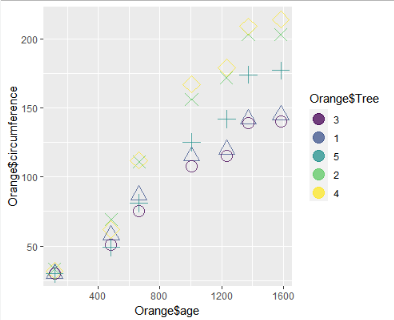
ggplot(Orange,aes(Orange$age,Orange$circumference, col= Orange$Tree))+geom\_point(size=5, alpha=.75,shape=9)

#alpha =transparent



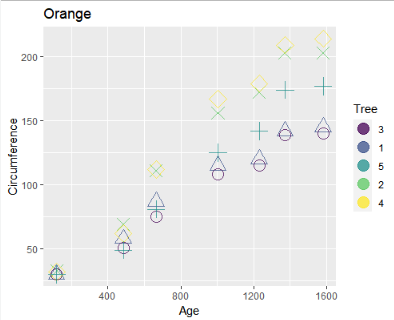
#scatter\_plot with multiple lines

ggplot(Orange,aes(Orange$age,Orange$circumference, col= Orange$Tree))+geom\_point(size=5, alpha=.75 ,shape=Orange$Tree)



ggplot(Orange,aes(Orange$age,Orange$circumference, col= Orange$Tree))+geom\_point(size=5, alpha=.75,shape=Orange$Tree)+

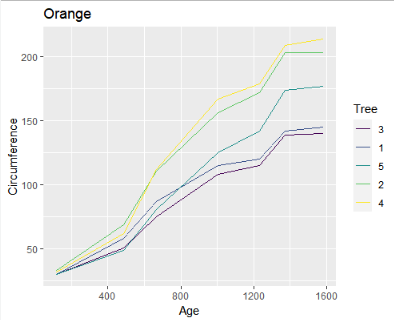
labs(title = "Orange",x="Age", y="Circumference",col="Tree")



#line\_plot

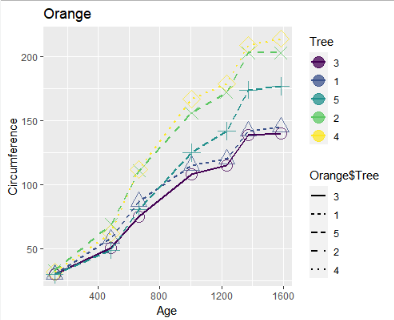
ggplot(Orange,aes(Orange$age,Orange$circumference, col= Orange$Tree))+geom\_line()+

labs(title = "Orange",x="Age", y="Circumference",col="Tree")



ggplot(Orange,aes(Orange$age,Orange$circumference, col= Orange$Tree))+geom\_point(size=5, alpha=.75,shape=Orange$Tree)+

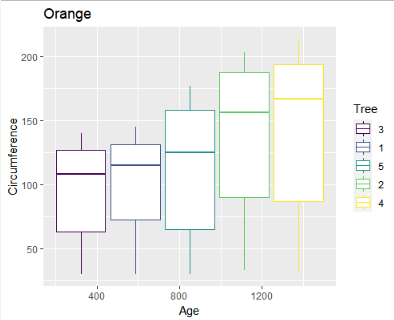
labs(title = "Orange",x="Age", y="Circumference",col="Tree")+ geom\_line(aes(linetype = Orange$Tree),size=1)



#box\_plot

ggplot(Orange,aes(Orange$age,Orange$circumference, col= Orange$Tree)) + geom\_boxplot()+

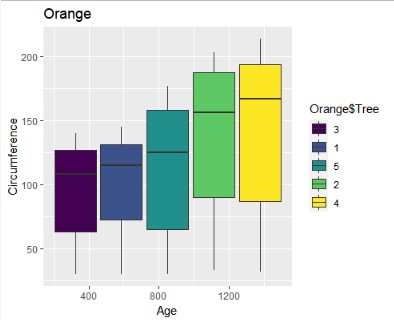
labs(title = "Orange",x="Age", y="Circumference",col="Tree")



#box\_plot

ggplot(Orange,aes(Orange$age,Orange$circumference, fill = Orange$Tree)) + geom\_boxplot()+

labs(title = "Orange",x="Age", y="Circumference",col="Tree")



#bubble\_plot

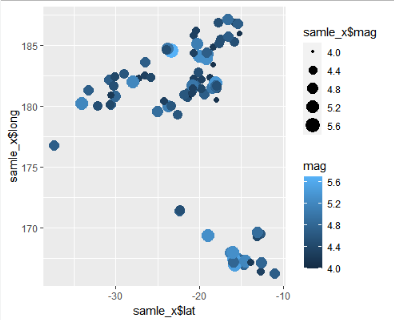
install.packages('viridis')

library(viridis)

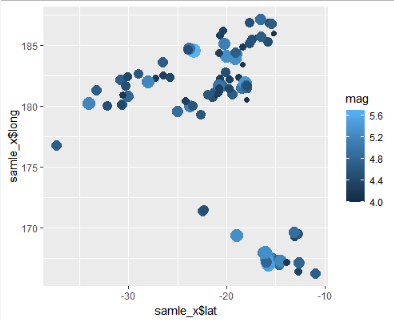
x = quakes

samlpe\_x <- x [seq(from=1,to=1000,by=10),]

ggplot(samle\_x , aes(samle\_x$lat , samle\_x$long)) + geom\_point(aes(size=samle\_x$mag , col=mag))

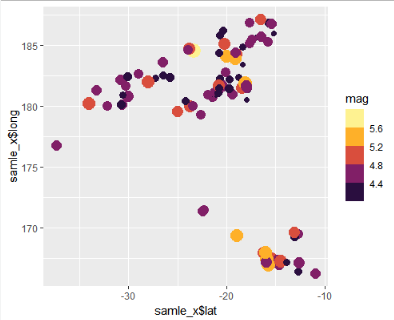


ggplot(samle\_x , aes(samle\_x$lat , samle\_x$long)) + geom\_point(aes(size=samle\_x$mag , col=mag)) + guides(size=FALSE)



ggplot(samle\_x , aes(samle\_x$lat , samle\_x$long)) + geom\_point(aes(size=samle\_x$mag , col=mag)) + guides(size=FALSE) +

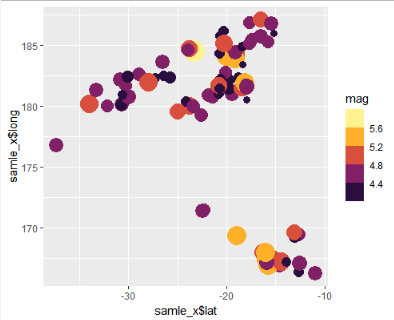
scale\_color\_viridis\_b(option = "B")



ggplot(samle\_x , aes(samle\_x$lat , samle\_x$long)) + geom\_point(aes(size=samle\_x$mag , col=mag)) + guides(size=FALSE) +

scale\_color\_viridis\_b(option = "B") +

scale\_size\_continuous(range = c(1,9))

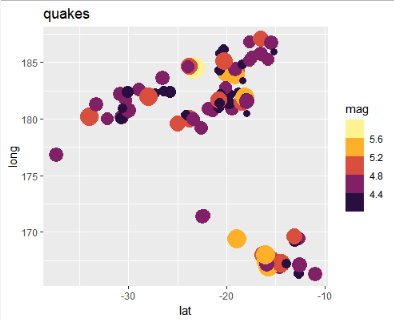


ggplot(samle\_x , aes(samle\_x$lat , samle\_x$long)) + geom\_point(aes(size=samle\_x$mag , col=mag)) + guides(size=FALSE) +

scale\_color\_viridis\_b(option = "B") +

scale\_size\_continuous(range = c(1,9)) + labs(title = "quakes",x="lat", y="long") +

ggsave("Quakes.tiff")



library(ggplot2)

#violin\_plot

data('diamonds')

x <- diamonds

nrow(x)

head(x)

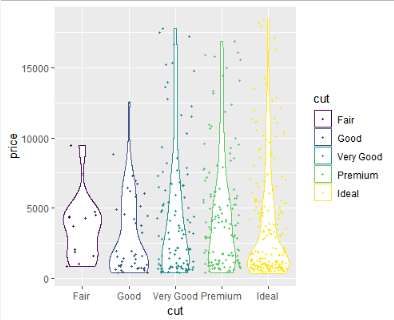
str(x)

sample\_x <- x[seq(1,50000,100),]

nrow(sample\_x)

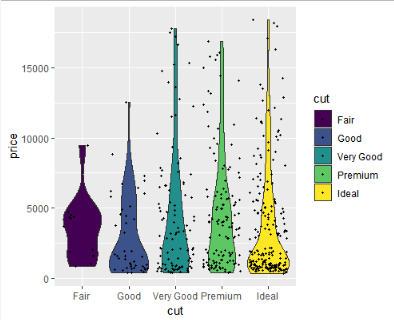
ggplot(sample\_x,aes(cut,price,col=cut))+geom\_violin() +

geom\_jitter(size=.6)

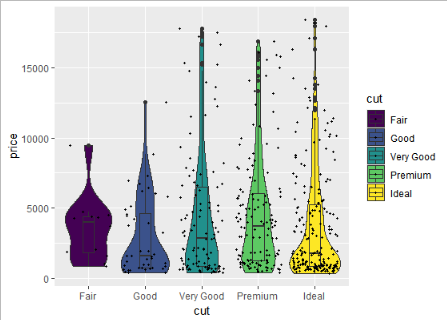


ggplot(sample\_x,aes(cut,price,fill=cut))+geom\_violin() +

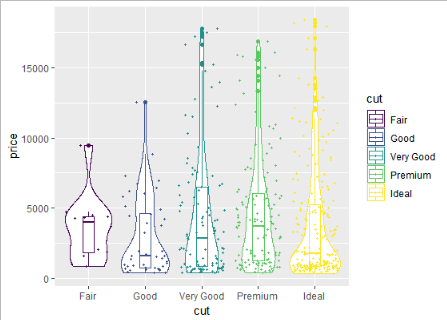
geom\_jitter(size=.6)



ggplot(sample\_x,aes(cut,price,fill=cut))+geom\_violin() + geom\_boxplot (width=0.2) + geom\_jitter(size=.6)



ggplot(sample\_x,aes(cut,price,col=cut))+geom\_violin() + geom\_boxplot (width=0.2) + geom\_jitter(size=.6) [+ facet\_wrap(\_)]



#heat\_maps

x<-iris

install.packages('reshape')

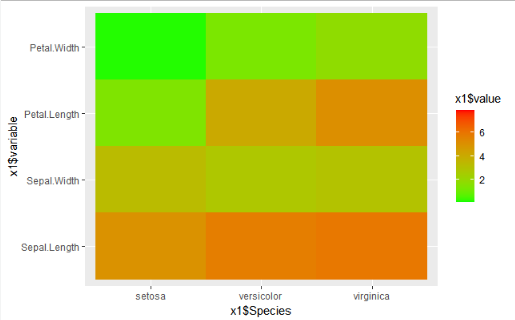
library(reshape)

x1<-melt(x) #stacking of data

ggplot(x1,aes(x1$Species , x1$variable , fill=x1$value)) +

geom\_tile()+

scale\_fill\_gradient(low = "green" , high = 'red')



#marginal\_plots

library(ggplot2)

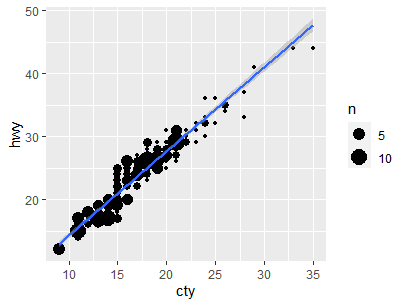
install.packages('ggExtra')

library(ggExtra)

data(mpg)

x <- mpg

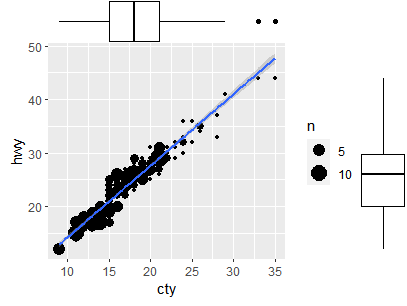
ggplot(x,aes(cty, hwy)) +geom\_count() + geom\_smooth(method = 'lm')



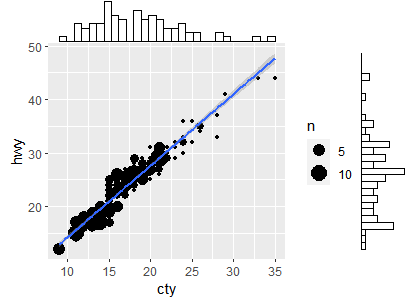
p1 <- ggplot(x,aes(cty, hwy)) +geom\_count() + geom\_smooth(method = 'lm')

p1

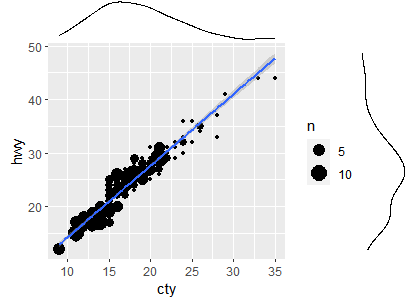
ggMarginal(p1,type='boxplot',fill='transparent')



ggMarginal(p1,type='histogram',fill='transparent')



ggMarginal(p1,type='density',fill='transparent')



# Animated Graphs in Rstudio

#Animated Graphs in Rstudio

library(ggplot2)

install.packages('gganimate')

library(gganimate)

install.packages('gifski')

install.packages('av')

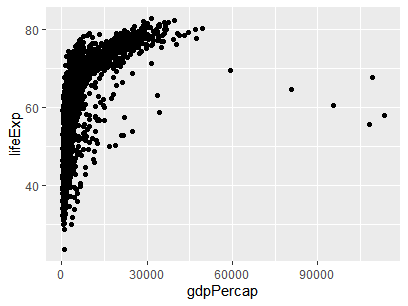
library(gifski)

library(av)

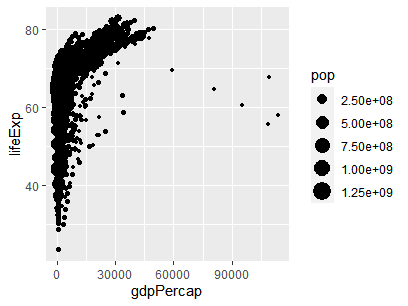
install.packages('gapminder')

library(gapminder)

ggplot(gapminder , aes(gdpPercap,lifeExp)) + geom\_point()



ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop)) + geom\_point()



ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop , col=country)) + geom\_point(show.legend = F , alpha=.8)



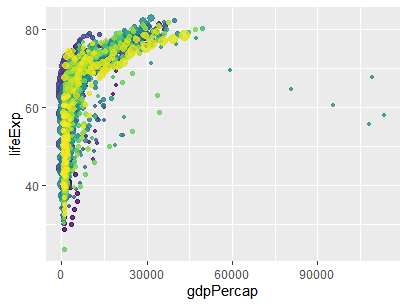
#changing\_color

install.packages('viridis')

library(viridis)

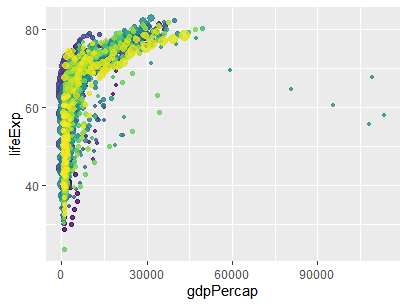
ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop , col=country)) + geom\_point(show.legend = F , alpha=.8) +

scale\_color\_viridis\_d()



+

scale\_size(c(1,11)) #size variations

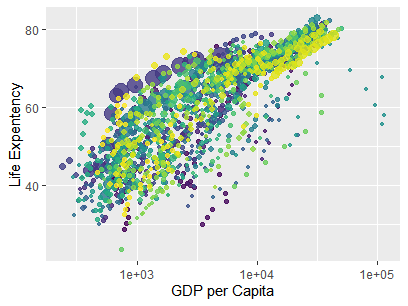


ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop , col=country)) + geom\_point(show.legend = F , alpha=.8) +

scale\_color\_viridis\_d()+

scale\_size(c(1,11)) +

scale\_x\_log10() + labs(x='GDP per Capita', y='Life Expentency' )



AN1 <- ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop , col=country)) + geom\_point(show.legend = F , alpha=.8) +

scale\_color\_viridis\_d()+

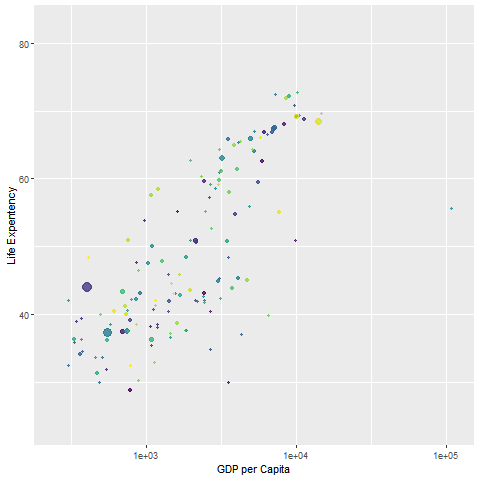
scale\_size(c(1,11)) +

scale\_x\_log10() + labs(x='GDP per Capita', y='Life Expentency' )

AN1 + transition\_time(year)

gganimate(AN2 , 'output.gif') or

anim\_save('ouput.gif' , AN2)



AN1 <- ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop , col=country)) + geom\_point(show.legend = F , alpha=.8) +

scale\_color\_viridis\_d()+

scale\_size(c(1,11)) + #range of different points

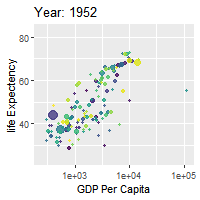
scale\_x\_log10() + labs(title = 'Year: {frame\_time}', x="GDP Per Capita", y= "life Expectency")+ transition\_time(year) +

ease\_aes("linear") # Year :frame\_time: each pic corresponds to specific year

AN1

animate(AN1, duration = 5, fps = 20, width = 200, height = 200, renderer = gifski\_renderer()) #fps: frame per second, renderer : gganimate package for rendering images , duration:total time for graph

anim\_save("output.gif")



AN1 <- ggplot(gapminder , aes(gdpPercap,lifeExp , size=pop , col=country)) + geom\_point(show.legend = F , alpha=.8) +

scale\_color\_viridis\_d()+

scale\_size(c(1,11)) +

scale\_x\_log10() + **facet\_wrap(~continent,nrow = 1)** + labs(title = 'Year: {frame\_time}', x="GDP Per Capita", y= "life Expectency")+ transition\_time(year) + # (shadow\_wake(wake\_lenght=.1 , alpha=F) #for tails

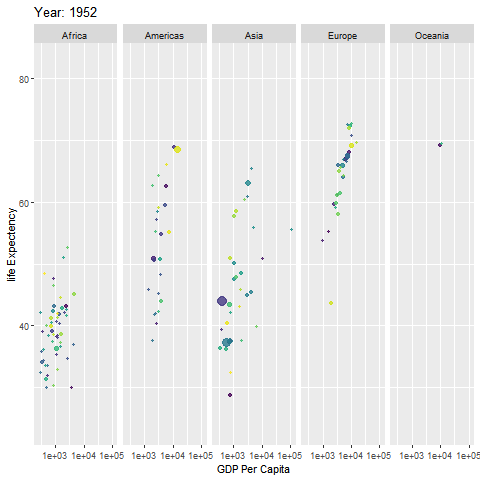
ease\_aes("linear")

ease\_aes("linear")

AN1

animate(AN1, duration = 5, fps = 20, width = 200, height = 200, renderer = gifski\_renderer())

anim\_save("output.gif")



#line\_plot

install.packages('gganimate')

library(gganimate)

install.packages('gifski')

install.packages('av')

library(gifski)

library(av)

data()

p<- ggplot(airquality, aes(Day , Temp, group=Month , col= factor(Month))) +

geom\_line() +

scale\_color\_viridis\_d() +

labs(x='Day of month' , y = 'Temp') +

theme(legend.position = 'top')

p

p1 <- p + transition\_reveal(Day) + geom\_point()

animate(p1, duration = 5, fps = 20, width = 200, height = 200, renderer = gifski\_renderer())

anim\_save("output.gif")

