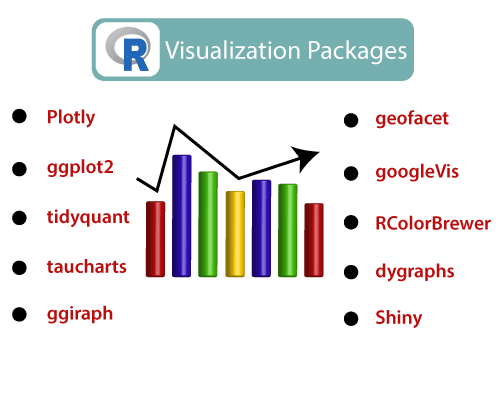
Data Handling and Visualization:

R Visualization Packages



1. **R Pie Charts:**

pie(X, Labels, Radius, Main, Col, Clockwise)

# Creating data for the graph.

x <- c(20, 65, 15, 50)

labels <- c("India", "America", "Shri Lanka", "Nepal")

# Giving the chart file a name.

png(file = "Country.jpg")

# Plotting the chart.

pie(x,labels)

# Saving the file.

dev.off()

1. **Title and color**

# Creating data for the graph.

x <- c(20, 65, 15, 50)

labels <- c("India", "America", "Shri Lanka", "Nepal")

# Giving the chart file a name.

png(file = "title\_color.jpg")

# Plotting the chart.

pie(x,labels,main="Country Pie chart",col=rainbow(length(x)))

# Saving the file.

dev.off()

1. **Slice Percentage & Chart Legend**

# Creating data for the graph.

x <- c(20, 65, 15, 50)

labels <- c("India", "America", "Shri Lanka", "Nepal")

pie\_percent<- round(100\*x/sum(x), 1)

# Giving the chart file a name.

png(file = "per\_pie.jpg")

# Plotting the chart.

pie(x, labels = pie\_percent, main = "Country Pie Chart",col = rainbow(length(x)))

legend("topright", c("India", "America", "Shri Lanka", "Nepal"), cex = 0.8,

fill = rainbow(length(x)))

#Saving the file.

dev.off()

1. **3 Dimensional Pie Chart**

# Getting the library.

library(plotrix)

# Creating data for the graph.

x <- c(20, 65, 15, 50,45)

labels <- c("India", "America", "Shri Lanka", "Nepal","Bhutan")

# Give the chart file a name.

png(file = "3d\_pie\_chart1.jpg")

# Plot the chart.

pie3D(x,labelslabels = labels,explode = 0.1, main = "Country Pie Chart")

# Save the file.

dev.off()

# Getting the library.

library(plotrix)

# Creating data for the graph.

x <- c(20, 65, 15, 50,45)

labels <- c("India", "America", "Shri Lanka", "Nepal","Bhutan")

pie\_percent<- round(100\*x/sum(x), 1)

# Giving the chart file a name.

png(file = "three\_D\_pie.jpg")

# Plotting the chart.

pie3D(x, labels = pie\_percent, main = "Country Pie Chart",col = rainbow(length(x)))

legend("topright", c("India", "America", "Shri Lanka", "Nepal","Bhutan"), cex = 0.8,

fill = rainbow(length(x)))

#Saving the file.

dev.off()

1. **R BAR Chart :**

barplot(h,x,y,main, names.arg,col)

# Creating the data for Bar chart

H<- c(12,35,54,3,41)

# Giving the chart file a name

png(file = "bar\_chart.png")

# Plotting the bar chart

barplot(H)

# Saving the file

dev.off()

1. **Labels, Title & Colors**

# Creating the data for Bar chart

H <- c(12,35,54,3,41)

M<- c("Feb","Mar","Apr","May","Jun")

# Giving the chart file a name

png(file = "bar\_properties.png")

# Plotting the bar chart

barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="Green",

main="Revenue Bar chart",border="red")

# Saving the file

dev.off()

1. **Group Bar Chart & Stacked Bar Chart**

library(RColorBrewer)

months <- c("Jan","Feb","Mar","Apr","May")

regions <- c("West","North","South")

# Creating the matrix of the values.

Values <- matrix(c(21,32,33,14,95,46,67,78,39,11,22,23,94,15,16), nrow = 3, ncol = 5, byrow = TRUE)

# Giving the chart file a name

png(file = "stacked\_chart.png")

# Creating the bar chart

barplot(Values, main = "Total Revenue", names.arg = months, xlab = "Month", ylab = "Revenue", ccol =c("cadetblue3","deeppink2","goldenrod1"))

# Adding the legend to the chart

legend("topleft", regions, cex = 1.3, fill = c("cadetblue3","deeppink2","goldenrod1"))

# Saving the file

dev.off()

1. **R Boxplot**

boxplot(x, data, notch, varwidth, names, main)

# Giving a name to the chart file.

png(file = "boxplot.png")

# Plotting the chart.

boxplot(mpg ~ cyl, data = mtcars, xlab = "Quantity of Cylinders",

ylab = "Miles Per Gallon", main = "R Boxplot Example")

# Save the file.

dev.off()

1. **Boxplot using notch**

# Giving a name to our chart.

png(file = "boxplot\_using\_notch.png")

# Plotting the chart.

boxplot(mpg ~ cyl, data = mtcars,

xlab = "Quantity of Cylinders",

ylab = "Miles Per Gallon",

main = "Boxplot Example",

notch = TRUE,

varwidth = TRUE,

ccol = c("green","yellow","red"),

names = c("High","Medium","Low")

)

# Saving the file.

dev.off()

1. **Violin Plots**

# Loading the vioplot package

library(vioplot)

# Giving a name to our chart.

png(file = "vioplot.png")

#Creating data for vioplot function

x1 <- mtcars$mpg[mtcars$cyl==4]

x2 <- mtcars$mpg[mtcars$cyl==6]

x3 <- mtcars$mpg[mtcars$cyl==8]

#Creating vioplot function

vioplot(x1, x2, x3, names=c("4 cyl", "6 cyl", "8 cyl"),

col="green")

#Setting title

title("Violin plot example")

# Saving the file.

dev.off()

1. **R Histogram**

hist(v,main,xlab,ylab,xlim,ylim,breaks,col,border)

# Creating data for the graph.

v <- c(12,24,16,38,21,13,55,17,39,10,60)

# Giving a name to the chart file.

png(file = "histogram\_chart.png")

# Creating the histogram.

hist(v,xlab = "Weight",ylab="Frequency",col = "green",border = "red")

# Saving the file.

dev.off()

1. **Use of xlim & ylim parameter**

# Creating data for the graph.

v <- c(12,24,16,38,21,13,55,17,39,10,60)

# Giving a name to the chart file.

png(file = "histogram\_chart\_lim.png")

# Creating the histogram.

hist(v,xlab = "Weight",ylab="Frequency",col = "green",border = "red",xlim = c(0,40), ylim = c(0,3), breaks = 5)

# Saving the file.

dev.off()

1. **R Line Graphs**

plot(v,type,col,xlab,ylab)

# Creating the data for the chart.

v <- c(13,22,28,7,31)

# Giving a name to the chart file.

png(file = "line\_graph.jpg")

# Plotting the bar chart.

plot(v,type = "o")

# Saving the file.

dev.off()

1. **Line Chart Title, Color, and Labels**

# Creating the data for the chart.

v <- c(13,22,28,7,31)

# Giving a name to the chart file.

png(file = "line\_graph\_feature.jpg")

# Plotting the bar chart.

plot(v,type = "o",col="green",xlab="Month",ylab="Temperature")

# Saving the file.

dev.off()

1. **Line Charts Containing Multiple Lines**

# Creating the data for the chart.

v <- c(13,22,28,7,31)

w <- c(11,13,32,6,35)

x <- c(12,22,15,34,35)

# Giving a name to the chart file.

png(file = "multi\_line\_graph.jpg")

# Plotting the bar chart.

plot(v,type = "o",col="green",xlab="Month",ylab="Temperature")

lines(w, type = "o", col = "red")

lines(x, type = "o", col = "blue")

# Saving the file.

dev.off()

1. **Line Graph using ggplot2**

library(ggplot2)

#Creating data for the graph

data\_frame<- data.frame(dose=c("D0.5", "D1", "D2"),

len=c(4.2, 10, 29.5))

head(data\_frame)

png(file = "multi\_line\_graph2.jpg")

# Basic line plot with points

ggplot(data=data\_frame, aes(x=dose, y=len, group=1)) +geom\_line()+geom\_point()

# Change the line type

ggplot(data=df, aes(x=dose, y=len, group=1)) +geom\_line(linetype = "dashed")+geom\_point()

# Change the color

ggplot(data=df, aes(x=dose, y=len, group=1)) +geom\_line(color="red")+geom\_point()

dev.off()

1. **R Scatterplots**

plot(x, y, main, xlab, ylab, xlim, ylim, axes)

#Fetching two columns from mtcars

data <-mtcars[,c('wt','mpg')]

# Giving a name to the chart file.

png(file = "scatterplot.png")

# Plotting the chart for cars with weight between 2.5 to 5 and mileage between 15 and 30.

plot(x = data$wt,y = data$mpg, xlab = "Weight", ylab = "Milage", xlim = c(2.5,5), ylim = c(15,30), main = "Weight v/sMilage")

# Saving the file.

dev.off()

1. **Scatterplot using ggplot2**

#Loading ggplot2 package

library(ggplot2)

# Giving a name to the chart file.

png(file = "scatterplot\_ggplot.png")

# Plotting the chart using ggplot() and geom\_point() functions.

ggplot(mtcars, aes(x = drat, y = mpg)) +geom\_point()

# Saving the file.

dev.off()

1. **Scatterplot with groups**

#Loading ggplot2 package

library(ggplot2)

# Giving a name to the chart file.

png(file = "scatterplot1.png")

# Plotting the chart using ggplot() and geom\_point() functions.

#The aes() function inside the geom\_point() function controls the color of the group.

ggplot(mtcars, aes(x = drat, y = mpg)) +

geom\_point(aes(color=factor(gear)))

# Saving the file.

dev.off()

1. **Changes in axis**

#Loading ggplot2 package

library(ggplot2)

# Giving a name to the chart file.

png(file = "scatterplot2.png")

# Plotting the chart using ggplot() and geom\_point() functions.

#The aes() function inside the geom\_point() function controls the color of the group.

ggplot(mtcars, aes(x = log(mpg), y = log(drat))) +geom\_point(aes(color=factor(gear)))

# Saving the file.

dev.off()

1. **Scatterplot with fitted values**

#Loading ggplot2 package

library(ggplot2)

# Giving a name to the chart file.

png(file = "scatterplot3.png")

#Creating scatterplot with fitted values.

# An additional function stst\_smooth is used for linear regression.

ggplot(mtcars, aes(x = log(mpg), y = log(drat))) +geom\_point(aes(color = factor(gear))) + stat\_smooth(method = "lm",col = "#C42126",se = FALSE,size = 1)

#in above example lm is used for linear regression and se stands for standard error.

# Saving the file.

dev.off()

1. **Adding title**

#Loading ggplot2 package

library(ggplot2)

# Giving a name to the chart file.

png(file = "scatterplot4.png")

#Creating scatterplot with fitted values.

# An additional function stst\_smooth is used for linear regression.

new\_graph<-ggplot(mtcars, aes(x = log(mpg), y = log(drat))) +geom\_point(aes(color = factor(gear))) +

stat\_smooth(method = "lm",col = "#C42126",se = FALSE,size = 1)

#in above example lm is used for linear regression and se stands for standard error.

new\_graph+

labs(

title = "Scatterplot with more information"

)

# Saving the file.

dev.off()

1. **Adding a sub-title**

#Loading ggplot2 package

library(ggplot2)

# Giving a name to the chart file.

png(file = "scatterplot6.png")

#Creating scatterplot with fitted values.

# An additional function stst\_smooth is used for linear regression.

new\_graph<-ggplot(mtcars, aes(x = log(mpg), y = log(drat))) +geom\_point(aes(color = factor(gear))) +

stat\_smooth(method = "lm",col = "#C42126",se = FALSE,size = 1)

#in above example lm is used for linear regression and se stands for standard error.

#Adding title with dynamic name

new\_graph + labs(

title =

"Relation between Mile per hours and drat",

subtitle =

"Relationship break down by gear class",

caption = "Authors own computation"

)

# Saving the file.

dev.off()

1. **Changing name of x-axis and y-axis**

#Loading ggplot2 package

library(ggplot2

# Giving a name to the chart file.

png(file = "scatterplot7.png")

#Creating scatterplot with fitted values.

# An additional function stst\_smooth is used for linear regression.

new\_graph<-ggplot(mtcars, aes(x = log(mpg), y = log(drat))) +geom\_point(aes(color = factor(gear))) +

stat\_smooth(method = "lm",col = "#C42126",se = FALSE,size = 1)

#in above example lm is used for linear regression and se stands for standard error.

#Adding title with dynamic name

new\_graph + labs(

x = "Drat definition",

y = "Mile per hours",

color = "Gear",

title = "Relation between Mile per hours and drat",

subtitle = "Relationship break down by gear class",

caption = "Authors own computation"

)

# Saving the file.

dev.off()