**R - Histograms**

A histogram represents the frequencies of values of a variable bucketed into ranges. Histogram is similar to bar chat but the difference is it groups the values into continuous ranges. Each bar in histogram represents the height of the number of values present in that range.

R creates histogram using hist() function. This function takes a vector as an input and uses some more parameters to plot histograms.

Syntax

The basic syntax for creating a histogram using R is −

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

Following is the description of the parameters used −

v is a vector containing numeric values used in histogram.

main indicates title of the chart.

col is used to set color of the bars.

border is used to set border color of each bar.

xlab is used to give description of x-axis.

xlim is used to specify the range of values on the x-axis.

ylim is used to specify the range of values on the y-axis.

breaks is used to mention the width of each bar.

Example

A simple histogram is created using input vector, label, col and border parameters.

The script given below will create and save the histogram in the current R working directory.

# Create data for the graph.

v <- c(9,13,21,8,36,22,12,41,31,33,19)

# Give the chart file a name.

png(file = "histogram.png")

# Create the histogram.

hist(v,xlab = "Weight",col = "yellow",border = "blue")

# Save the file.

dev.off()



Range of X and Y values

To specify the range of values allowed in X axis and Y axis, we can use the xlim and ylim parameters.

The width of each of the bar can be decided by using breaks.

# Create data for the graph.

v <- c(9,13,21,8,36,22,12,41,31,33,19)

# Give the chart file a name.

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

# Create the histogram.

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

breaks = 5)

# Save the file.

dev.off()

When we execute the above code, it produces the following result −



**R - Boxplots**

Boxplots are a measure of how well distributed is the data in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.

Boxplots are created in R by using the boxplot() function.

Syntax

The basic syntax to create a boxplot in R is −

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

Following is the description of the parameters used −

x is a vector or a formula.

data is the data frame.

notch is a logical value. Set as TRUE to draw a notch.

varwidth is a logical value. Set as true to draw width of the box proportionate to the sample size.

names are the group labels which will be printed under each boxplot.

main is used to give a title to the graph.

Example

We use the data set "mtcars" available in the R environment to create a basic boxplot. Let's look at the columns "mpg" and "cyl" in mtcars.

Live Demo

input <- mtcars[,c('mpg','cyl')]

print(head(input))

When we execute above code, it produces following result −

mpg cyl

Mazda RX4 21.0 6

Mazda RX4 Wag 21.0 6

Datsun 710 22.8 4

Hornet 4 Drive 21.4 6

Hornet Sportabout 18.7 8

Valiant 18.1 6

Creating the Boxplot

The below script will create a boxplot graph for the relation between mpg (miles per gallon) and cyl (number of cylinders).

Live Demo

# Give the chart file a name.

png(file = "boxplot.png")

# Plot the chart.

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

ylab = "Miles Per Gallon", main = "Mileage Data")

# Save the file.

dev.off()

When we execute the above code, it produces the following result −

Box Plot using R



Boxplot with Notch

We can draw boxplot with notch to find out how the medians of different data groups match with each other.

The below script will create a boxplot graph with notch for each of the data group

# Give the chart file a name.

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

# Plot the chart.

boxplot(mpg ~ cyl, data = mtcars,

xlab = "Number of Cylinders",

ylab = "Miles Per Gallon",

main = "Mileage Data",

notch = TRUE,

varwidth = TRUE,

col = c("green","yellow","purple"),

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

)

# Save the file.

dev.off()

