

GRAPHICAL REPRESENTATION

Bar Plots in R

Inputs can be vector, matrix etc

4 types are there:

- * Simple Bar Plot

- * Horizontal Bar Plot

- * Stacked Bar Plot

- * Grouped Bar Plot

R Bar Plot

- * Created by using \Rightarrow `barplot()` Function

- * If we supply a vector, the plot will have bars with their height equal to the elements in the vector

eg: `temp = c(27, 26, 23, 24, 30)`

`barplot(temp)`

`barplot()` arguments:

`main`: heading

`xlab`: x-axis title

`ylab`: y-axis title

`names.arg`: name of each bar

`col`: gives colors

`horiz`: TRUE (Represents as horizontal bar)

`density`: Splits each bar

\rightarrow Horizontal Bar Plot

border: Gives border

height: describing the bar which makes up the plot

width: gives width to each bar

space: space b/w each bar

legend.text: gives logical indication whether a legend should be included

las: las = 1, 2, 3

names[] gives the column heading

eg: data("mtcars")

names(mtcars)

table(mtcars\$cyl)

gives the table of cylinders in mtcars

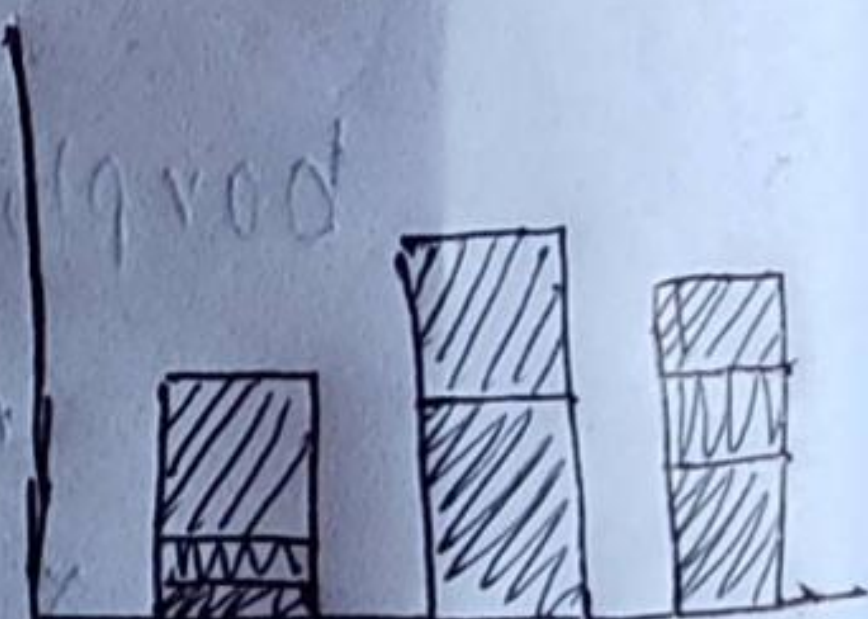
Stacked Bar Plot (bar plot which has matrix input)

eg: y1 = table(mtcars\$cyl, mtcars\$gear)

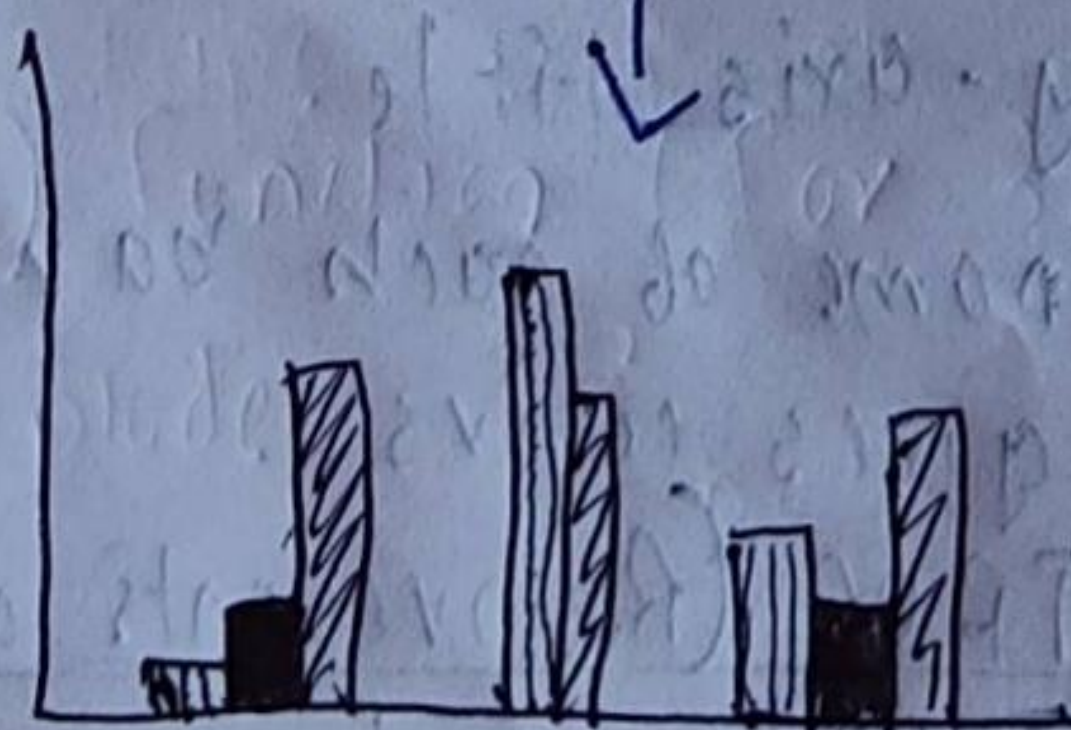
bar plot(y1)

bar plot(y1, legend.text=T,

beside=T)



Stacked Bar Plot



Grouped Bar Plot

angle: the slope of shading lines

par(mfrow = c(2, 2)): gives two row and 2 columns

par(mfrow = c(1, 1)): default

barplot(y, col = c(10, 12, 13))

⇒ gives each bar each color

barplot(y, col = rainbow(1))

⇒ gives a color in the window

barplot(y, col = rainbow(s = 1, n = 15))

⇒ gives the lightest ~~white~~ color

border = F: without border

border = T: with border

xlim: limit for the x-axis

ylim: limit for the y-axis

expression(sum(i)) gives the heading Σ

⇒ barplot(y, main = expression(x + y))

(alpha)

(beta)

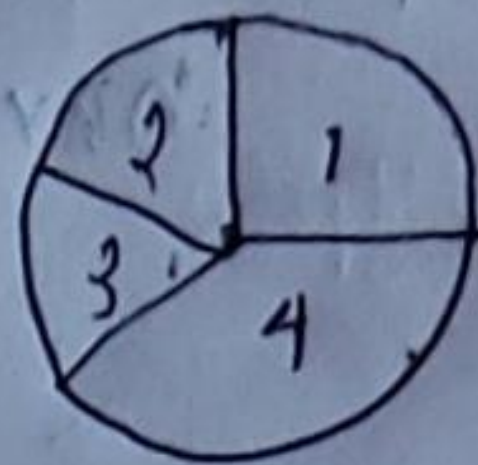
Piechart

pie() function

eg: $x = c(1, 1, 1, 2, 2, 3, 3, 4, 4, 4)$

$y = table(x)$

pie(y)



Arguments

x : input values

labels: giving names for the slices

edges: the circular outline of the pie is

approximated by a polygon with

this many edges (default = 200)

radius: gives the radius of the pie

(-1, 1) change the direction

clockwise:

init.angle: changes the angle

density, main, angle, col, border

(plotrix package) → for 3D

⇒ library(plotrix)

?pie3D

pie3D() function (for 3D piechart)

explode:

Histogram

hist() Function

$x = c(1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4)$

hist(x)

cut(x, 6) : shows the interval

data("cars")

head(cars)

⇒ hist(car\$speed)

breaks: one of a vector giving the breakpoints b/w histogram cells

breaks = 22 : 22 ruls (No. of columns)

xlab, ylab, main, xlim, ylim, col

freq: FALSE, we can get the probability distribution instead of frequency

Return value of hist()

$h \leftarrow \text{hist}(\text{temp})$

h

breaks: places where the breaks occur

counts: the no. of observations falling in that cell

density: density of the cells

mids: midpoint of the cell

xname: the x argument name

equidist: a logical value indicating if the breaks are equally spaced or not

Defin

Breaks: we can specify the no. of cells
we want in the histogram and
we can also give values

Scatter Plot

plot() function

Generic X-Y plotting

plot(1, 2, 3, 4): plots the values in
(1, 1), (2, 2), (3, 3), (4, 4)

plot(c(5, 6, 7, 8)): plots the values in
(1, 5), (2, 6), (3, 7), (4, 8)

eg: $x = 1:5$

$y = 6:10$

plot(x, y)

values are plotted by by

(1, 6), (2, 7), (3, 8), (4, 9), (5, 10)

arguments: main, xlab, ylab, col

↳ type: ~~gives~~

type = "p", points

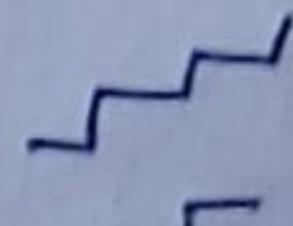
"l", lines


"b", both

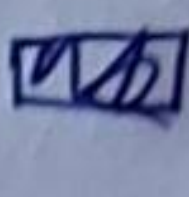
"n", for the line alone

"o", overplotted

"h", histogram | | | |

"as", stair steps 

"S", other steps 

"n", no plotting 

Box Plots

boxplot() function

Arguments: main, xlab,
ylab, col

notch: T 

horizontal: T, represents
as horizontal

boxplot(oz, temp, wind)

↳ gives 3 ^{box} plot in one
plot

varwidth: T

border

