

GRAPH AND CHART

Bar Plots in R

Input can be vector, matrix etc.
Types are three:

- * Simple Bar plot
- * Horizontal Bar Plot
- * Stacked Bar Plot
- * Grouped Bar Plot

R Bar Plot

Created by using `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:

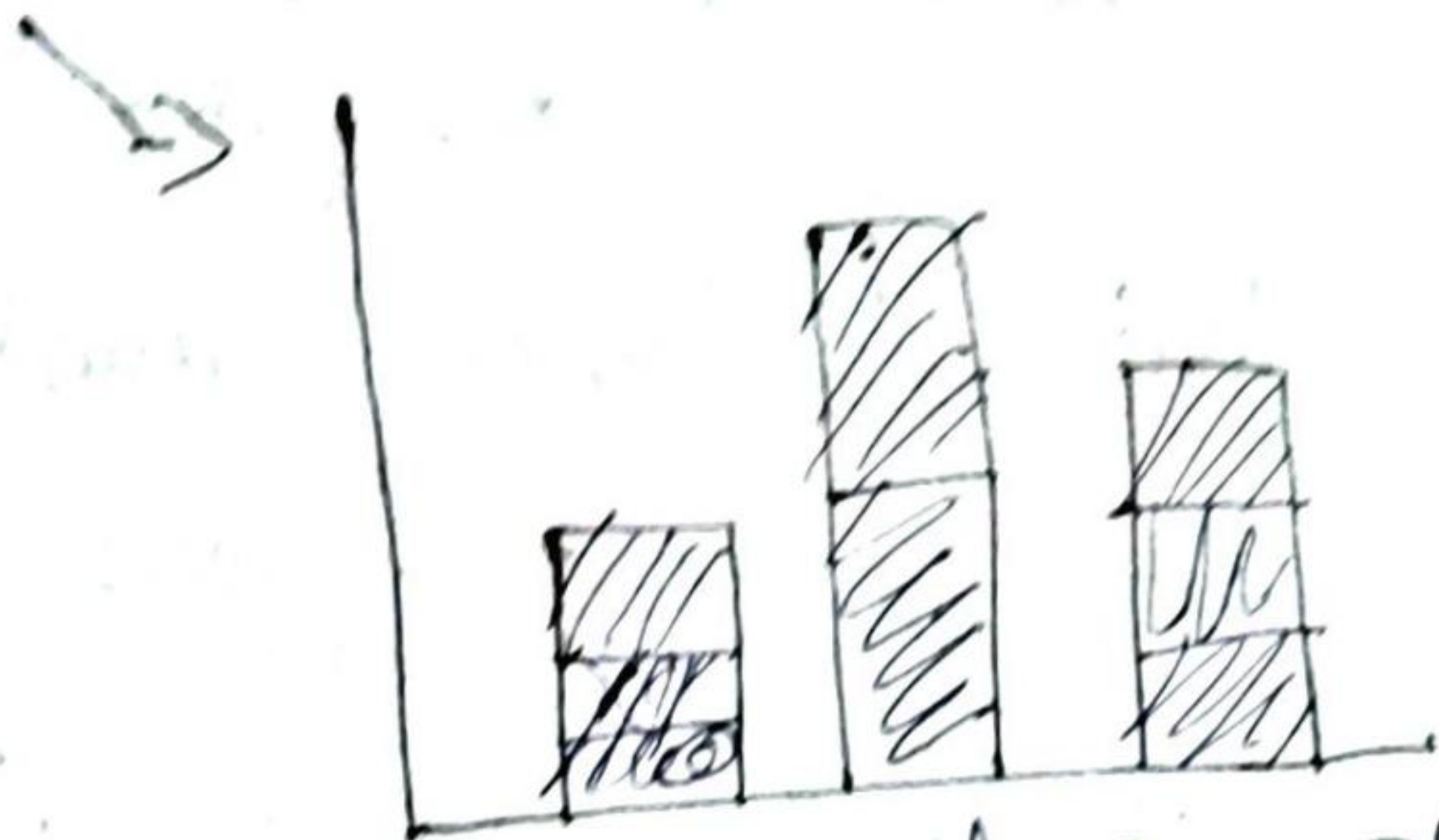
`main`: heading.

`xlab`: x-axis title.

Y lab : J - axis title
 names. avg : Name of each bar
 col : Gives colors
 horiz : TRUE (Represents as horizontal bar)
 density : Splits each bar.
 border : Gives border.
 height : describing the bar which makes up the plot
 width : gives width to each other
 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

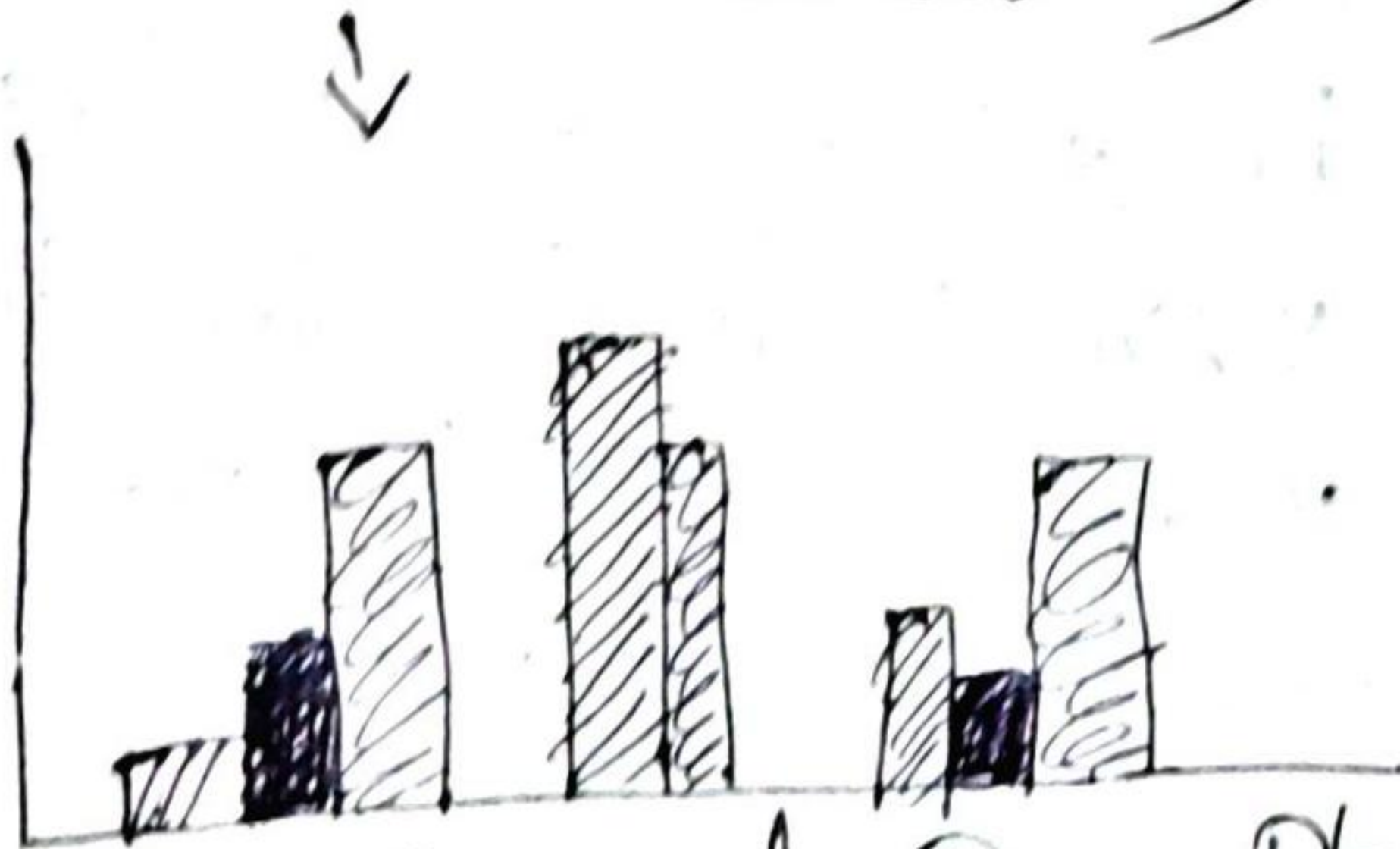
Stacked Bar Plot (bar plot which has
matrix input)

eg: y_1 : table (mt car & cyl, mt cars & gears)
bar plot (y_1)



Stacked Bar Plot

Bar plot (y_1 , legend.text = 1,
beside = 1)



Grouped Bar Plot

angle : the slope of shading lines
 $\text{Paa}(\text{m} \times \text{row} = c(2, 2))$: gives two

$\text{Paa}(\text{m} \times \text{row} = c(1, 1))$: default

$\text{barplot}(y, \text{col} = c(10, 12, 13))$

\Rightarrow gives each bar each color.

$\text{barplot}(y, \text{col} = \text{rainbow}(1))$

\Rightarrow gives a color in the

$\text{Barplot}(y, \text{col} = \text{rainbow}(S=1, n=13))$

\Rightarrow gives the lightest color.

$\text{border} = f$: without border

$\text{border} = T$: with border.

$x\text{lim}$: limit for the x axis.

$y\text{lim}$: limit for the y axis.

expression $s(\text{sum}(1))$ give & the leading Σ

$\Rightarrow \text{barplot}(y, \text{main} = \text{expression}(x+y))$

(alpha)

beta)

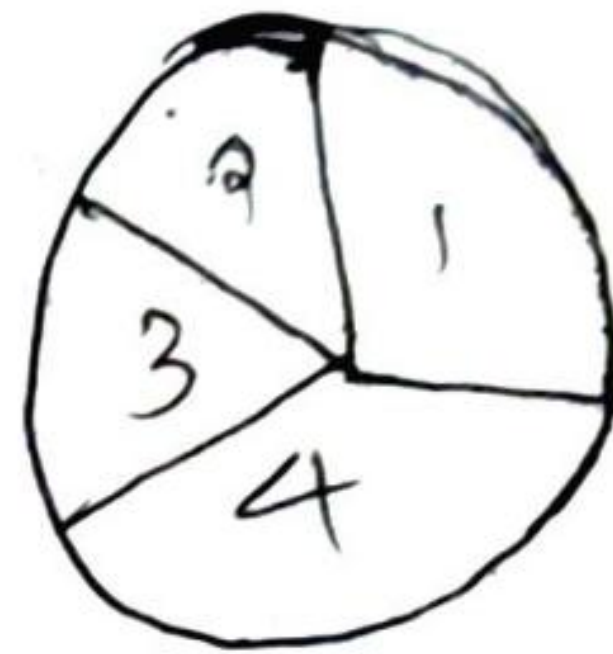
Pie chart

Pie() function

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

$y = \text{table}(x)$

$\text{pie}(4)$



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:

int angle: Changes the angle

density, main, angle, col, border
(plotrix Package) → for 3D

=> library(plotrix)

?pie3D

Pie3D() function (for 3D pie chart)

example:

Histogram

hist() function

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

hist(x)

cut(x, 6) : show the interval

data('cars')

head(cars)

=> as(carspeed,

breaks : one of a vector giving the
break points for histogram cells

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

xlab, ylab, main, xlim, ylim, col

freq = FALSE, we can get the probability
Distribution instead of frequency.

Return value of hist()

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

"

breaks : Places where the break occurs.

counts : the no. of observation falling in that cell

density : Density of cells

mids : Midpoint of cell

xname : that x argument name

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

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

Scatter Plot

plot () function

Generic Y-Y plotting

$\text{plot}(c(1:2, 3:4))$: plots the value
in $(1,1), (2,2), (3,3),$
 $(4,4)$

$\text{plot}(c(5, 6, 7, 8))$: plots the value in
 $(1,5), (2,6), (3,7),$
 $(4,8)$

eg:

$x = 1:3$

$y = 6:10$

$\text{plot}(x, y)$

values are plotted by.


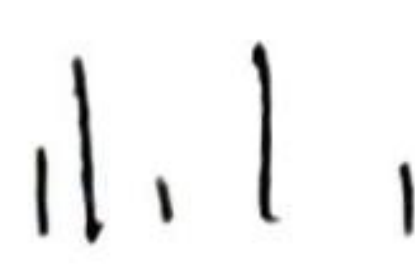



$(1,6), (2,7), (3,8), (4,9)$
 $(3,10)$

Arguments: main, xlab, ylab & col.

type = "p" points o o o

"l" line ———

"b" both —o—o—

"c" for the line along of "b" ---
 "o", overplotted 
 "h", histograms 
 "s", stair steps 
 "S", other steps 
 "n" no plotting 

Box plots

boxplot() function.

Arguments: mean x lab
y lab, col

notch = T



horizontal = T, represent
as horizontal

boxplot(02: temp, wind)
box

↳ gives 3 plot in one.
plot

varwidth = 1
border

