MODULE - 5

Graphs & charts

▷ Bar plots in R.

- (inputs can be vectors, matrix).

types : 1) simple Bar plot
2) Horizontal Bar plot.
3) Stacked Bar plot.
4) Grouped Bar plot.

▷ R Bar plot. [in R-studio]

- created by using → barplot () function.
- Inputs can be vectors / Matrix.
- if we supply a vector, the plot will have bars with their height equal to the elements in the vectors/matrix.

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

barplot (temp)

- Argument used

1) main → used to give heading
2) xlab → x-axis name.
3) ylab → y-axis name
4) col → Give colour to bars.
5) horiz → TRUE
6) xlim
7) ylim.

Eg: temp = c(

barplot (temp,

main = "max temp in a week",
xlab = "Degree celcius",
ylab = "Day"
col = "blue",)

8) density → Give lines inside bars
9) borders → borders to bars.

eg: density = 20, borders = "red", col = "green")

10) width → size of bars [default = 1].

11) space → space b/w bars.

D Table (x).

x ← c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4).

table (x).

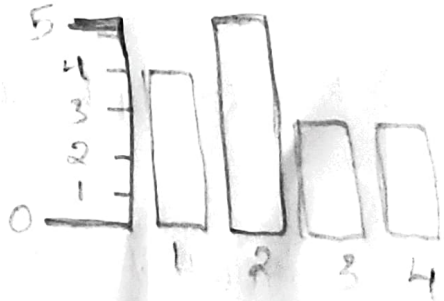
x			
1	2	3	4
↓	↓	↓	↓
4	5	3	3

D Plotting of categorical data.

- x ← c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4)

  y = table (x).

  barplot (height = y, width = c(3, 4, 5, 6))



another examples.

- x ← c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4).

  y → table(x).

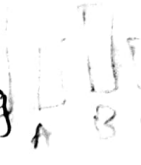
  barplot (height = y, space = 5)

- x ← c(1,1,2,2,2,3,3,1,1,2,2,3,4,4,4)

  y = table(x).

  barplot (height = y, names.arg = LETTERS [1:4])

- barplot (height = y, names.arg = c("Student 1", "Student 2", "Student 3", "Student 4"))



- y = table (x)

  barplot (height = y, names.arg = c("Stdnt 1", "Stdnt 2", "Student 3"), legend.text = T).

1) legend.text ⇒ is a vector of text used to construct a legend for the plot i.e, used to identify what each bar represent.

- barplot (height = y, las = 1) → to straighten numbers.
- barplot (height = y, las = 2) → to slightly rotate
- barplot (height = y, las = 3) ⇒ complete slope.

D Stacked Bar plots

The plot drawn when matrix is given as input.

data ("mtcars")

> names (mtcars) → column headings will be displayed.

[1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec"

"vs" "am" "gear" "carb".

> mt cars (to view datasets of mtcars)

> mtcars $cyl.

[1] . . . .

> table (mtcars $cyl)

4 6 8

11 7 14.

> table (mtcars $gear)

3 4 5

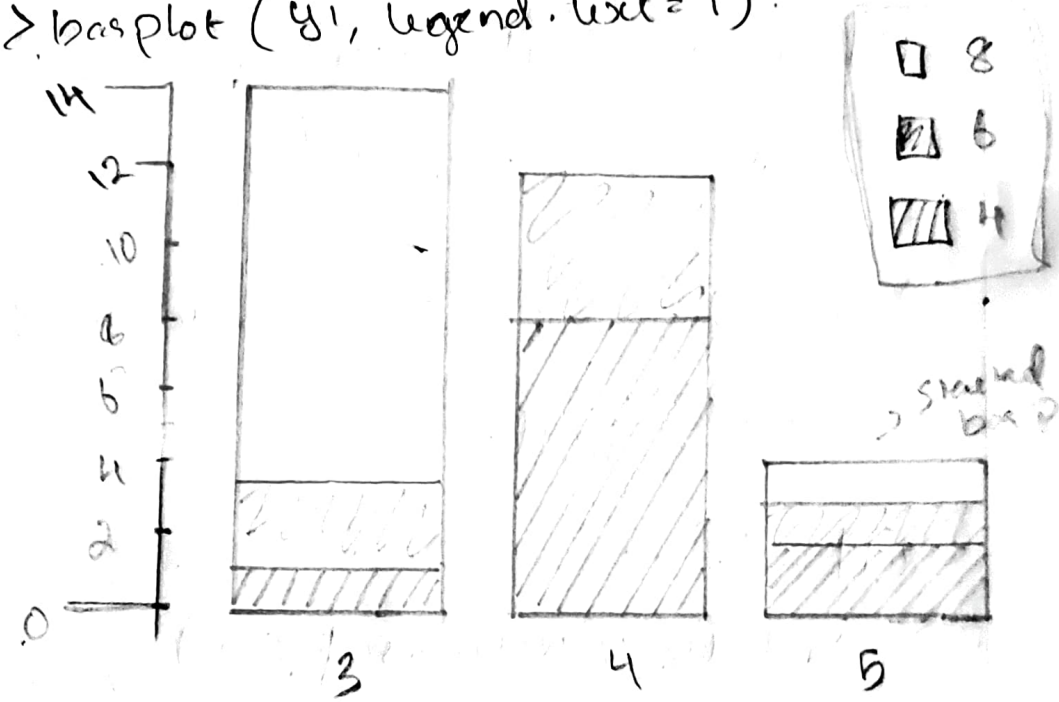
15 12 5

> table (mtcars $ cyl, mtcars $ gear).

	3	4	5
4	1	8	2
6	2	4	1
8	12	0	2

> y1 = table (mtcars $ cyl, mtcars $ gear)

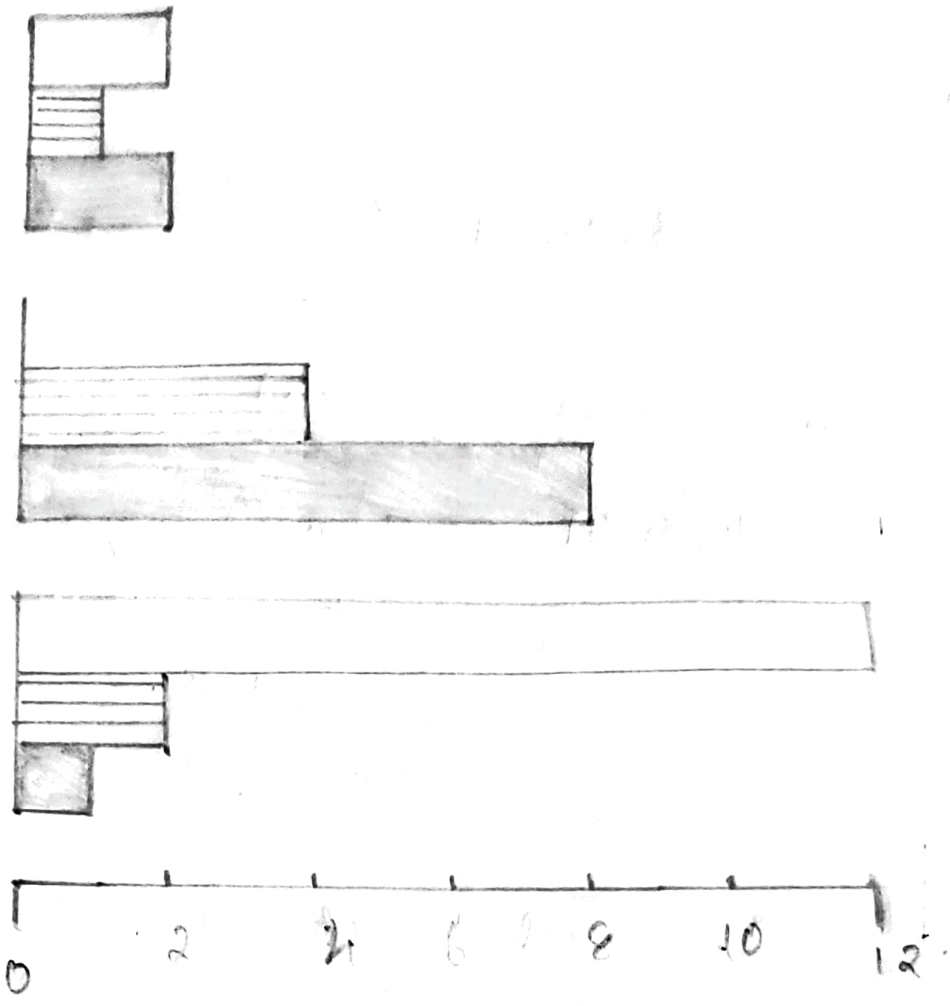
> barplot (y1, legend.text = T).



> barplot (y1, legend.text = T, beside = T) (we get grouped barplot)

D > barplot(y1, legend.text = T, beside = T, horiz = T)

↓ we get horizontal bar graph.



D density → used to give lines inside bars.

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

y = table(x).

barplot (y, legend.text = T, las = 1, density = c(5, 10, 15).

D Angle is used to give angle to lines inside bar.

D Colour :-

Gives colour to bar.

eg:

- barplot (y, col = "red").

- > bar (mf rows = c(1,1))

> barplot (y, col = c(1,2,3)) ← used to give different colour to each bar.

- barplot (y, col = rainbow(1))

barplot (y, col = rainbow (s=.2, n=2)) ↵ (no: of colors)

used to set darkness to each bars. (b/w 0 to 1)

D borders :- used to set borders to bar.

> barplot (y, col = rainbow (s=.5, n=15), borders = T)

> bar (mf rows = c(1,2))

> barplot (y, col = rainbow. (s=5, n=13), borders = F)

> barplot (y, col = rainbow (s=.5, n=13), borders = T)

> bar (mf rows = c(1,1)).

▷ main :- used to give heading to the particular bar plot.

▷ sub :- used to give heading at bottom.

eg :-

```
barplot (y, main = "header", sub = "Footer")
```

▷ x limit, y limit

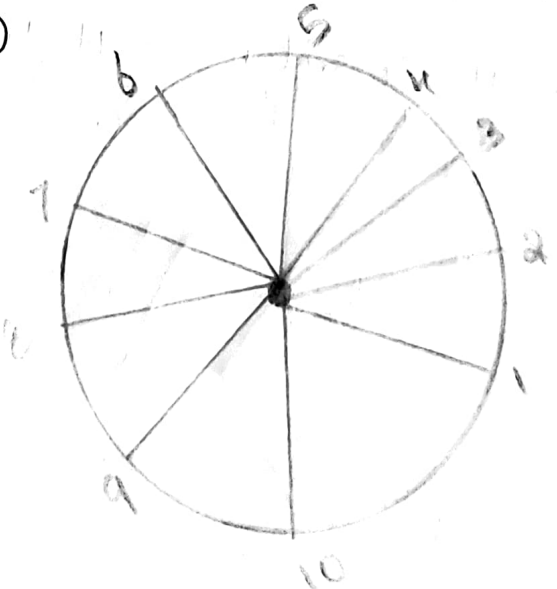
```
barplot (y, ylim = c(0,10))
barplot (y, xlim = c(0,3))
```

▷ PIE CHART (Qualitative Data)

Diagramatic representation of values.

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

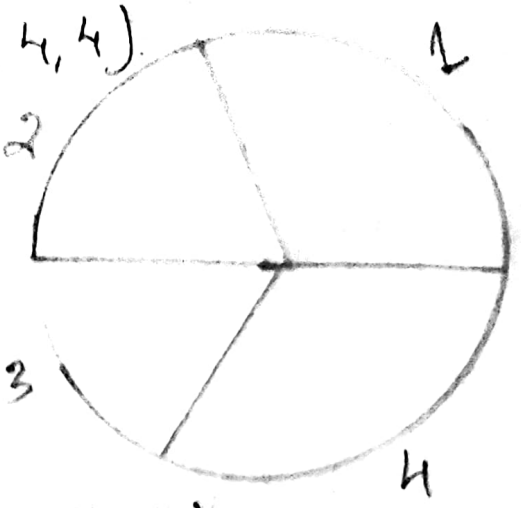
`Pie (x)`



This is wrong because 10 values are displayed.

eg: 2)

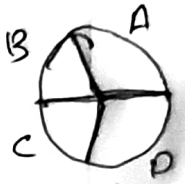
```
> x = c(1, 1, 2, a, 3, 3, 4, 4, 4)
> y = table(x).
> pie(y)
```




```
> pie(y, main = "my first plot")
```

* x → a vector of non negative numerical quantities.

eg 1 → pie(y, labels = LETTERS[1:4])

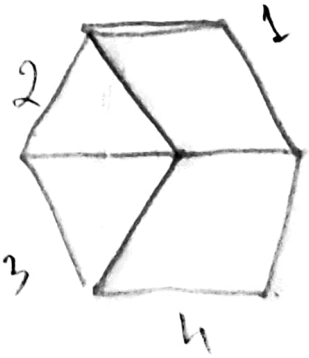


eg 2 → pie(y, labels = c("red", "blue", "green", "orange"))

* labels are name of each slices.

D pie(y, edges = 10) → default = 200.

nb: It doesn't mean it has 10 edges, means to change to shape.



- pie (y, radius = 0.5)
- pie (y, clockwise = T)
- pie (y, density = c(10, 20, 30, 40).
- Density : used to give shading to each slice

- Colours :- [101]
  - pie (y, col = rainbow (15))
  - pie (y, col = 1:4)

- Borders :-
  used to set borders it can be either T or F.

eg:
pie (y, col = 1:4, border = F)

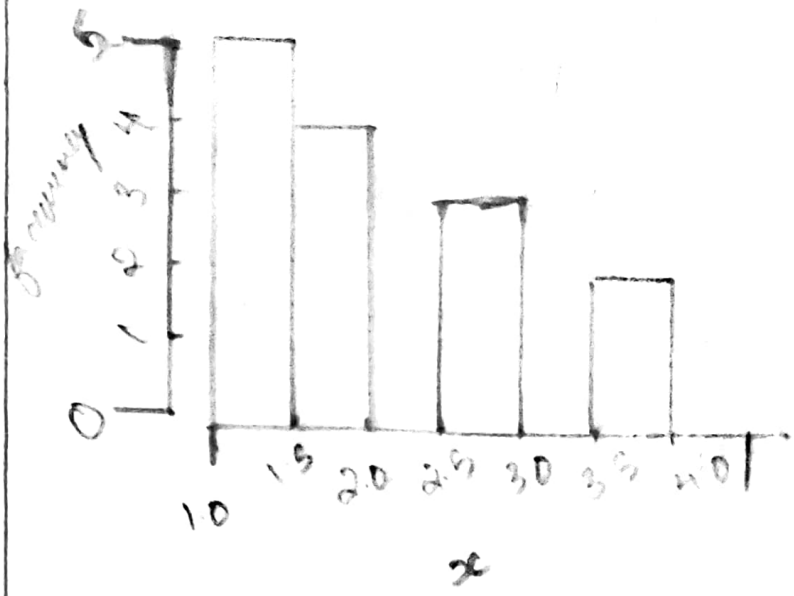
- Histogram

  used to plot quantitative data.

  function : Hist ()

- Inputs are vector inputs.

```
> x = c(1,1,1,1,1,2,2,2,2,3,3,3,4,4).
> Hist (x)
```



▷ for viewing the grouping arrangement use the function, cut ()

```
> cut (x,6).
[1]  (,] , (,]
> data frame (x, cut (x,6))
```

▷

```
data ("cars")
> head (cars).
```

	Speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10

> cars $ speed (To display data of speed).

[1] . . . .

> hist (cars $ speed)

Arguments.

- breaks: to up & down no: of columns

> hist (cars $ speed, breaks = 22).

- main :- used to give title.

> hist (cars $ speed, breaks = 5, main = "Histogram")

- xlab, ylab :-

> hist (cars $ speed, xlab = "dist", ylab = "no of times")

eg:

> data (airquality).

> head (airquality).

	ozone	solar	wind
1	.	.	.
2	.	.	.
3	.	.	.
4	.	.	.

> temp = airquality $ Temp.

> hist (temp).

> str (airquality)

str : used to display structure.

▷ xlim, ylim :-

used to provide range of axis

▷ col :-

used to define colors.

▷ With the argument **freq = false** we can get the probability distribution instead of the frequency.

> hist (temp, freq = F)

eg: hist(temp, main = "Maximum Daily Temperature",
xlab = "Temp in Degree Fahrenheit," xlim = c(50,100)
col = rainbow [20], freq = FALSE, las = T)

▷ borders :- (T/F)

▷ density :-

> hist(temp, borders = "blue", density = 20)

▷ Return value of hist()

Displays the value in hist().

* breaks :- places where the breaks occur.
* counts : The no: of observations falling in that

Eg:->x = 1:50
>y = sin(x)
>plot (x,y)

Eg:->x=1:10
>y=21:30
>plot(x,y,main="Scatter Plot", xlab="x-values", ylab="y values",
col=1:10)

*type
specifies what type of plot should be drawn. Possible types are:-
"p" → for points
"l" → for lines
"b" → for both ie, combination of points & lines
"c" → for the lines part alone of b (ie, dashed lines)
"o" → for overplotted (bubbles plotted over lines)
"h" → for histogram like (or high-density) vertical lines
"s" → for scalar steps
"n" → no plotting

Eg:->plot (x,y,main="Scatter plot", type="c")

Eg:->x=1:100
>y=sin(x)
>plot((x,y, type="l")

Eg:->x=seq(0,10,0,0.1)
>y=sin(x)
>polot(x,y)

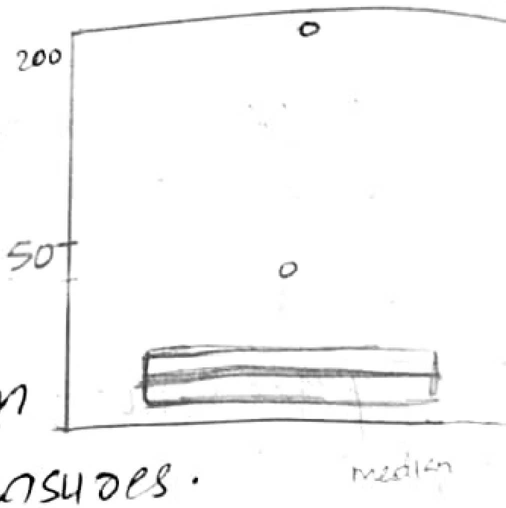
Boxplot:

Used to plot quantitative Data

Eg: x = c(1,1,1,2,2,1,1,3,3,3,4,4,5,6,7,4,4,6,5,7,20,
20,20,25,25,25,45,200)

boxplot(x)

outlayers



* Boxplot can be used to identify median, range, quartile deviation and various other statistical measures.

Eg:- str(airquality)

>barplot(airquality$ozone)

>barplot(airquality$ozone, main="Ozone in parts per billion from 1300 to 1500 hrs at Roosevelt island, xlab="parts per billion", ylab="ozone", col="orange", notch=T, horizontal=T)

For drawing multiple boxplot

>ozone = airquality$ozone
>temp = airquality$temp
> wind = airquality$wind
>boxplot(ozone,temp,wind)

For changing the width of bars

>barplot (ozone, width=12, border="red")