

MODULE - 5

GRAPHS & CHARTS

I Bar Plot

* Barplot ()

* We can supply a vector or matrix as I/P

* if we supply a vector, the plot will have bars with their heights equal to the elements in the vector.

Eg: temp = c(27, 26, 28, 24, 26, 28, 25)
barplot(temp)

main - heading

xlab - x axis name

ylab - y axis name

name.arg - name of each bar

col - color name of bar

horiz - horizontal graph

density - shading

border - bar border color

legend.text - a vector of text used to construct a legend.

las - to write the text in the y axis horizontally

las = 1 → horizontally.

Stacked bar plot

eg: table (mt cars & cyl)

4	6	8
11	7	14

> table (mt cars & gear)

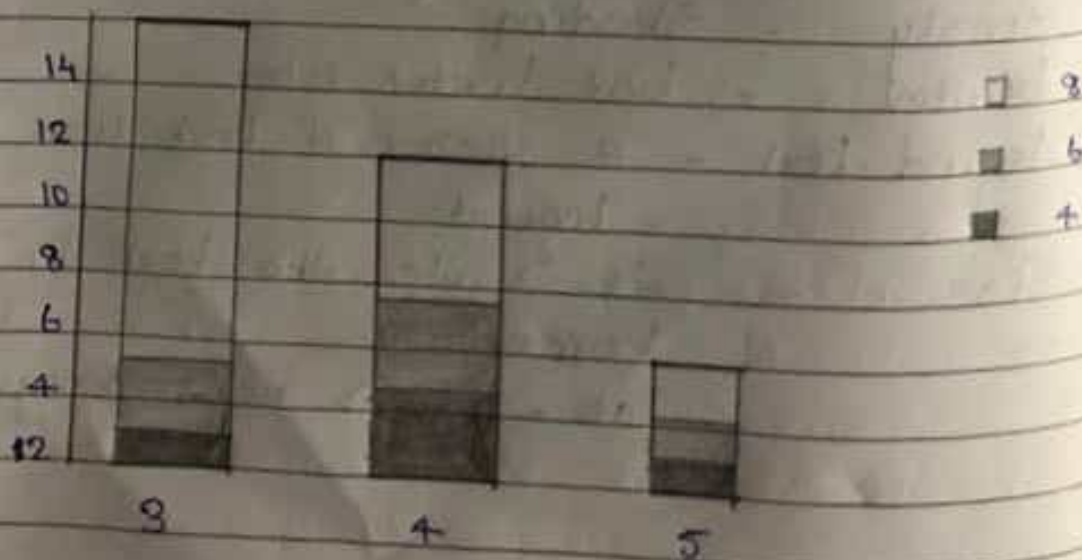
3	4	5
15	12	5

> table (mt cars & cyl, mt cars & gear)

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

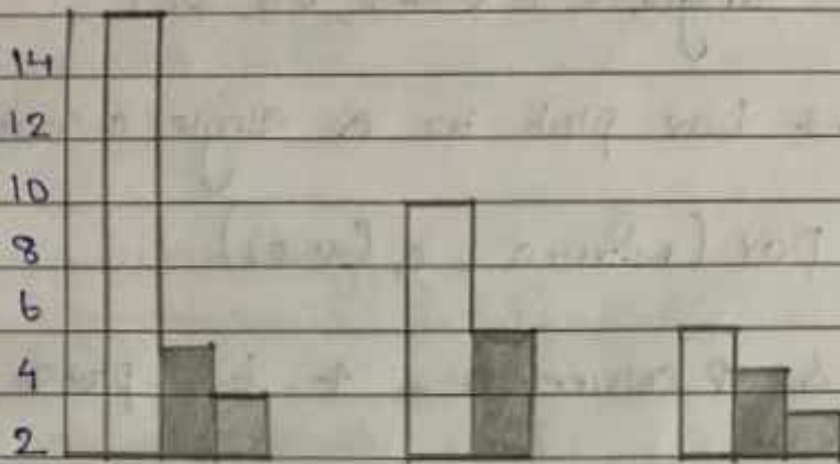
> y = table (mt cars & cyl, mt cars & gear)

> barplot (y, legend.text = T,



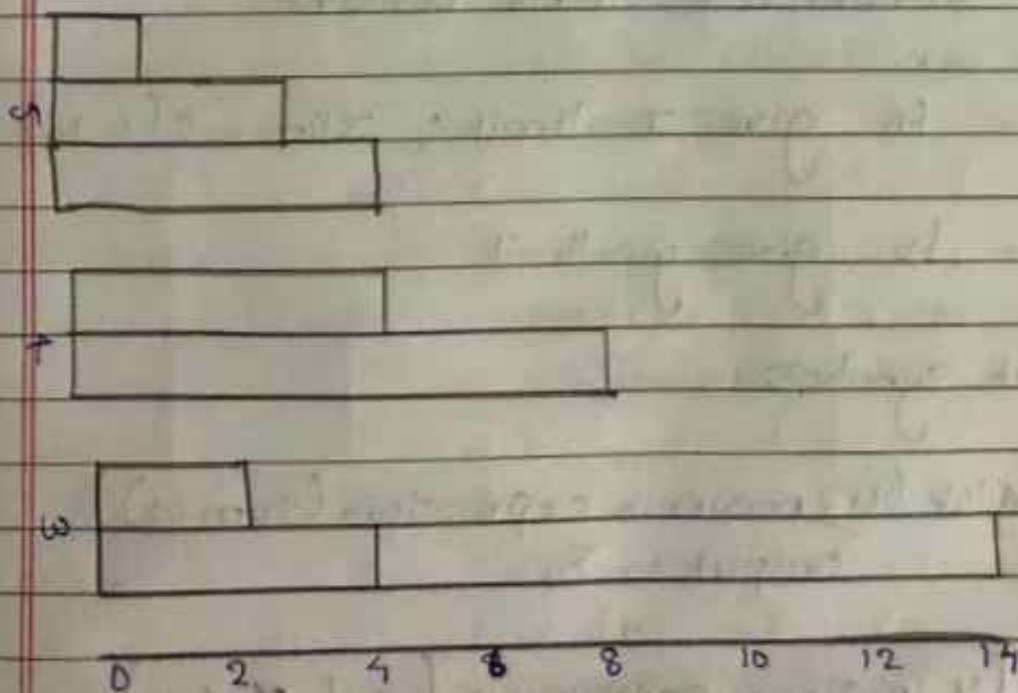
Grouped barplot

> barplot (y, legend.text = T, beside = T)



Horizontal barplot

> barplot (y, legend.text = T, beside = T, las = 1, horiz = T)



* density - shading
density = c(5, 15, 20)

* angle - angle of shading
angle = c(45, 90, 20)

different bar plot in a single window;

par(mfrow = c(2, 2))

2 row & 2 columns - 4 bar plots

* col - coloring bars

Eg: col = c(1, 2, 3)

col = rainbow(n = 15)

col = rainbow(s = 0.2, n = 15)

* border -

Eg: border = F (no border)

* xlim - to give x limit; xlim = c(0, 10)

* ylim - to give y limit

* TO insert symbols;

Eg: barplot(y, main = expression(sum(x)))
output: Σ

> barplot(y, main = expression(x / x / y))
output: $\frac{x}{x} \cdot \frac{1}{y}$

pie chart

Function : `pie(x)`

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

- * `main` :- heading
 `> pie(y, main = "First")`
- * `x` - input values
- * `labels` - to give labels names for slides
- * `edges` - circular o/p of pie is approximated by a polygon with many edges [default: 200]
- * `radius` - to change radius, default - 0.8, max - 1
- * `Clockwise` - to label in clockwise direction
 (`Clockwise = T`)
- * `density` - to shade pie
 eg: `density = c(10, 20, 30, 40) ->`
 different shading each slide
- * `col` - to give colors
 `col = rainbow(15)`
- * `border` - to give border
 `border = F`

• We can make 3D by installing plotrix.
eg: pie 3D(y)

> pie 3D(y, explode = 2)

It makes the pie chart into pieces.

Histogram

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

```
y = table(x)
```

```
> y
```

```
> x
```

1	2	3	4
5	4	3	2

```
> hist(x)
```

to see grouping;

```
> cut(x, 5)
```

* main - heading

* xlab - x axis name

* ylab - y axis name

* xlim - x limit

* ylim - y limit

* col - color

* density - shading. density = c(20, 30, 40)

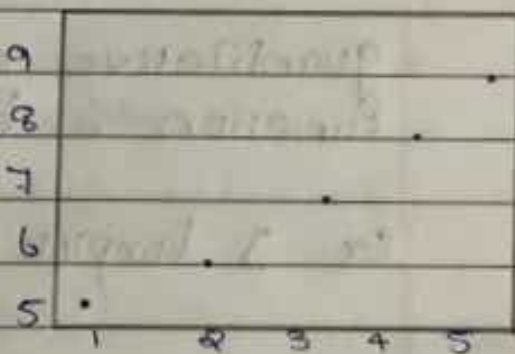
* Freq - get the probability distri instead of Freq.

Freq = FALSE

- * las = to show the limit values horizontally.
las = TRUE
- * border = Set border
border = F
- * breaks = no. of cells we want.
place where the breaks occur
- * counts = no. of observations falling in that cell.

Scatter plot-

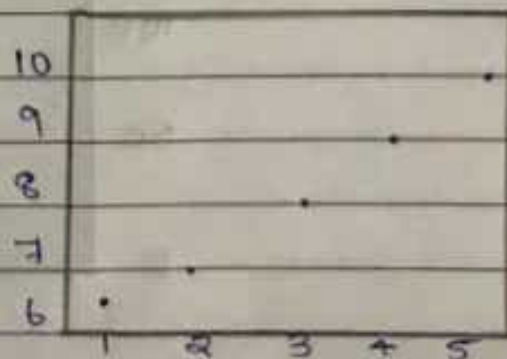
> plot (c(5, 6, 7, 8, 9))



> x = 1:5

> y = 6:10

> plot (x, y)

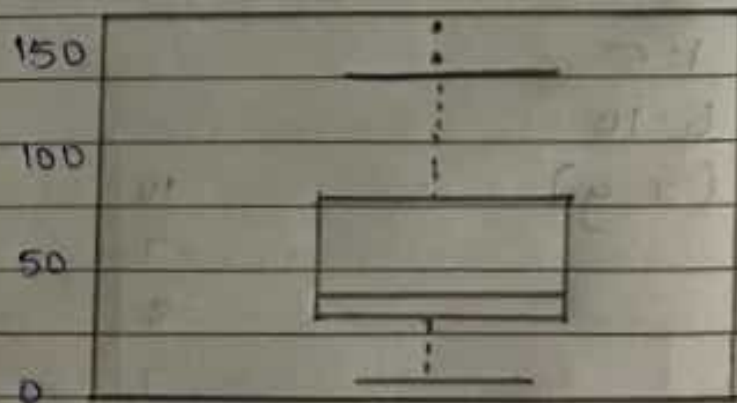


- * main - heading
- * x lab -
- * y lab -
- * col - color
- * type - 'p' : points
- 'l' : lines
- 'b' : both line & point — • —
- 'c' : lines part alone of 'b'
- 'o' : over plotted — • —
- 'h' : histogram
- 's' : box stair
- 'n' : no plotting

Box plot:

- quantitative data plotting
- function - boxplot

Ex: 1 boxplot (airquality & ozone)



- * main
- * x lab
- * y lab

* col

* notch - notch in the plot = notch = T

* horizontal - horizontal = T

display box plot horizontally

multiple box plot;

> oz = airquality & ozone

> temp = airquality & temp

> wind = airquality & wind

> boxplot = (oz, temp, wind)

* varwidth :

- changes the box width

- varwidth = 1

* border - it changes border color.