

5: Graphs and Charts

Bar Plots:

- Simple Bar Plot
- Horizontal Bar Plot
- Stacked Bar Plot
- Grouped Bar Plot

2 Bar Plot:

- Created by using \rightarrow `barplot()` Func.
- inputs can be vector or matrix.
- plot will have bars with their heights equal to elements in vector.

eg: `temp = c(27, 26, 23, 24, 20)`
`barplot(temp)`

Argument used:

- `main` \rightarrow used to give heading * `xlim`
- `xlab` \rightarrow x-axis name * `ylim`
- `ylab` \rightarrow y-axis name
- `col` \rightarrow give color to bar
- `horiz` \rightarrow TRUE
- `name.arg` \rightarrow name of each bar

eg: `temp = c(`
`barplot(temp,`
`main = "max Temp in a week",`
`xlab = "Degree Celsius",`
`ylab = "Dry "`
`col = "Blue",`

* `density` \rightarrow give lines inside bars

* `border` \rightarrow border to bars

`density = 20, border = "red", col = "green"`

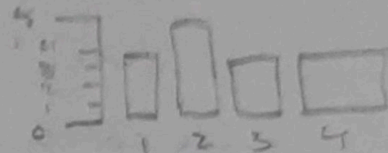
* width \rightarrow size of bars
 * space \rightarrow space b/w bars

* $x \leftarrow c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4)$
 $table(x)$

x	1	2	3	4
	4	5	3	3

plotting of categorical data

* $x \leftarrow c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4)$
 $y = table(x)$
 $barplot(height = y, width = c(3, 4, 5, 6))$



* $x \leftarrow c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4)$
 $y = table(x)$
 $barplot(height = y, space = 5)$

* $x \leftarrow c(1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 3, 4, 4, 4)$
 $y = table(x)$

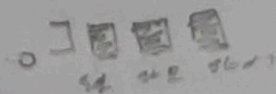
$barplot(height = y, names.arg = LETTERS[1:4])$

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

* $x \leftarrow c(1, 1, 1, 1, 2, 1, 2, 2, 2, 3, 3, 3, 1, 1, 2, 2, 3, 3)$

$y = table(x)$

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



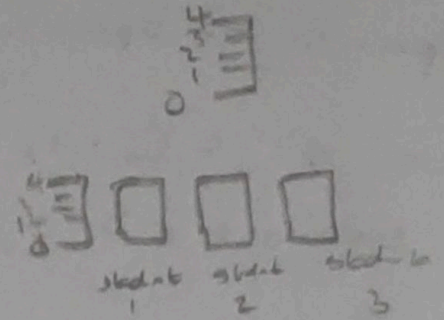
legend.text \rightarrow is a vector of text used to construct a legend for plot

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

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

$\text{barplot}(\text{height} = y, \text{las} = 1)$

$\text{barplot}(\text{height} = y, \text{las} = 2)$



• Stacked Bar Plots:

matrix is given as input

* $> \text{data}(\text{"mtcars"})$

$> \text{names}(\text{mtcars})$

[1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear" "carb"

$> \text{mtcars}$

$> \text{mtcars} \& \text{cyl}$

[1]

$> \text{table}(\text{mtcars} \& \text{cyl})$

4 6 8

" 7 14

$> \text{table}(\text{mtcars} \& \text{gear})$

3 4 5

15 12 5

$> \text{table}(\text{mtcars} \& \text{cyl}, \text{mtcars} \& \text{gear})$

3 4 5

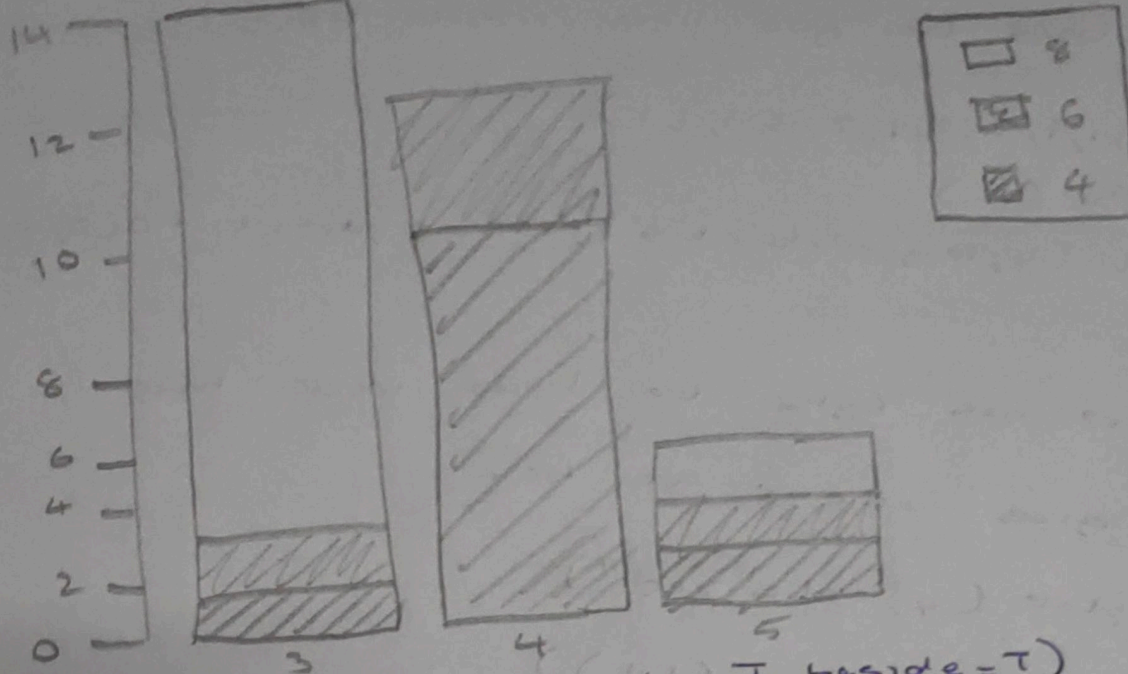
4 1 8 2

6 2 4 1

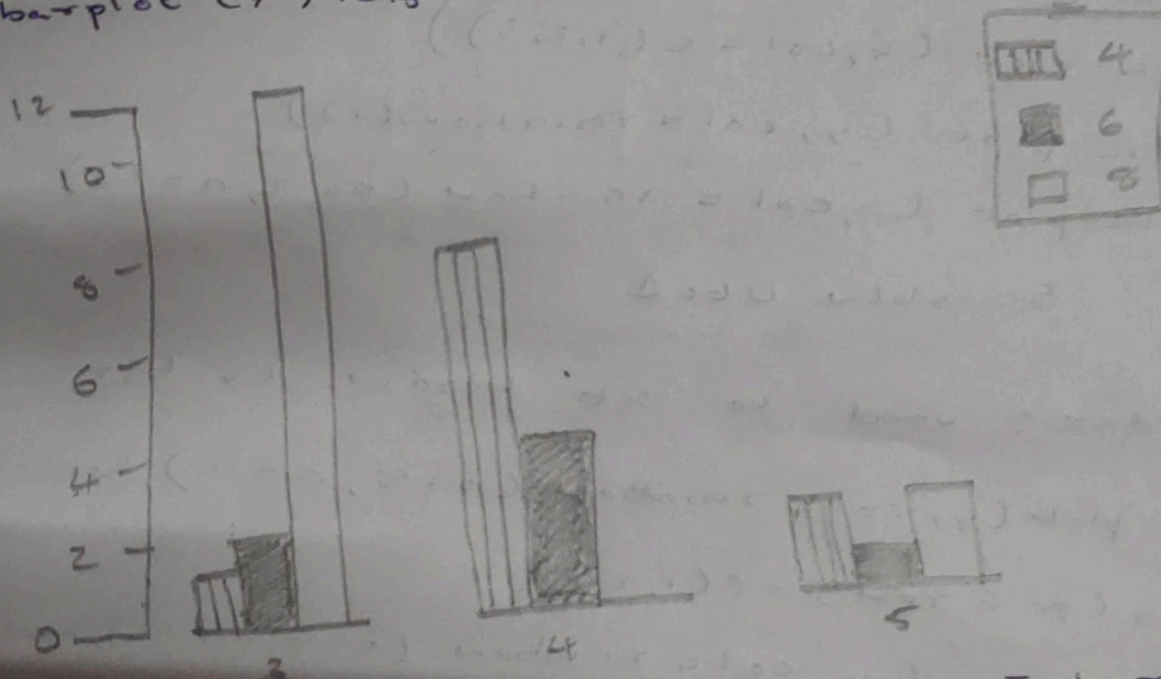
8 12 0 2

$> y_1 = \text{table}(\text{mtcars} \& \text{cyl}, \text{mtcars} \& \text{gear})$

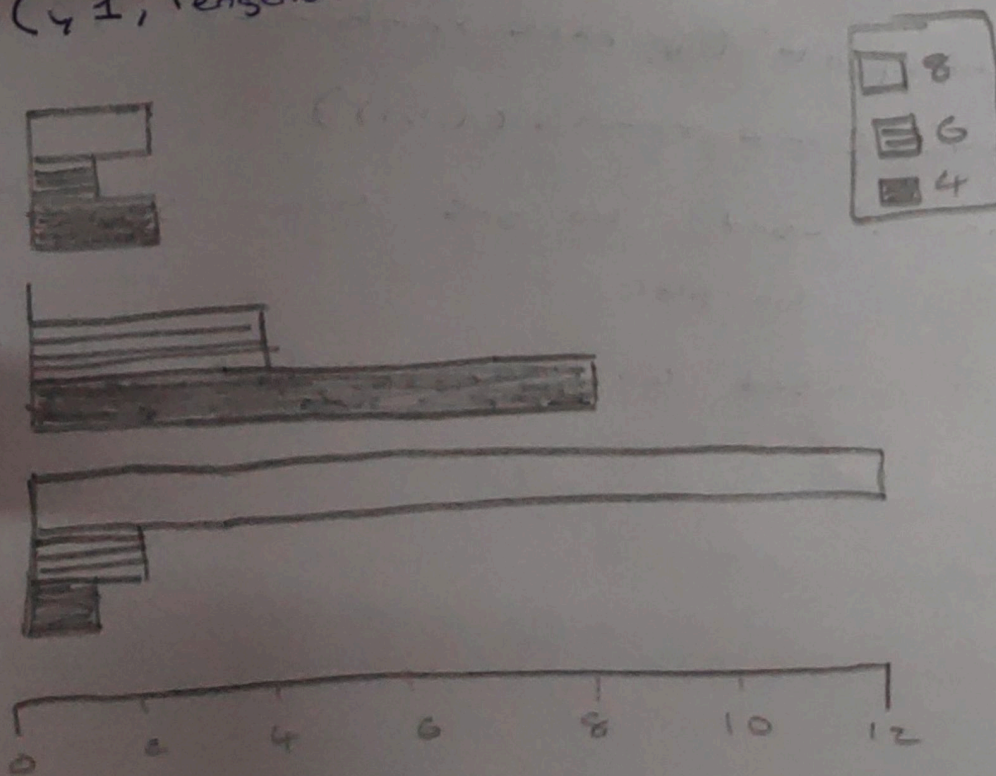
$\text{barplot}(y_1, \text{legend} = \text{TRUE})$



> barplot(y1, legend, las=c(1, 2, 3), beside=T)



> barplot(y1, legend, las=c(1, 2, 3), beside=T, horiz=T)



• 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.beside=T, las=1, density=c(5,10,15))`

• Angle: used to give angle to lines inside bars

• col: give colour to bar

⑧ `> barplot(y, col="red")`

⑨ `> bar(mfrow=c(1,1))`

`> barplot(y, col=c(1,2,3))`

⑩ `> barplot(y, col=rainbow(1))`

⑪ `> barplot(y, col=rainbow(s=2, n=2))`

5 is the output

⑫ border: used to set border to bar

`> barplot(y, col=rainbow(s=5, n=15), border=T)`

• `> bar(mfrow=c(1,2))`

`> barplot(y, col=rainbow(s=5, n=13), border=T)`

`> barplot(y, col=rainbow(s=5, n=13), border=T)`

`> bar(mfrow=c(1,1))`

* main: used to give heading to particular barplot.

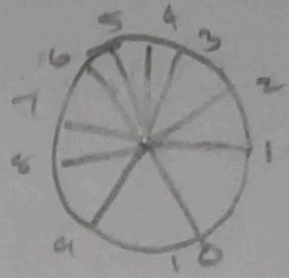
sub: used to give heading at bottom.

Piechart (Qualitative Data)

diagrammatic representn of values

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

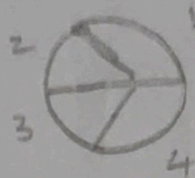
`pie(x)`



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

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

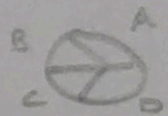
$\rightarrow \text{pie}(y)$



$\rightarrow \text{pie}(y, \text{main} = "my first plot")$

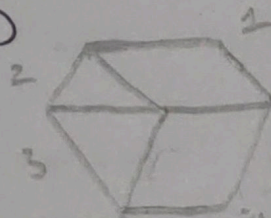
* $x \rightarrow$ a vector of non negative numerical quantities

eg: `pie(y, labels = LETTERS[1:4])`



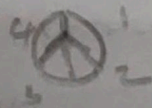
* labels are names of each slices

⑤ `pie(y, edges = 10)`



⑥ `pie(y, radius = .5)`

⑦ `pie(y, clockwise = T)`



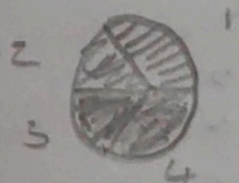
⑧ `pie(y, density = c(10, 20, 30, 40))`

Density: used to give shading to each slice

colour: [col]

* `pie(y, col = rainbow(15))`

* `pie(y, col = 1:4)`



border: set border it can be either

T or F `pie(y, col = 1:4, border = F)`

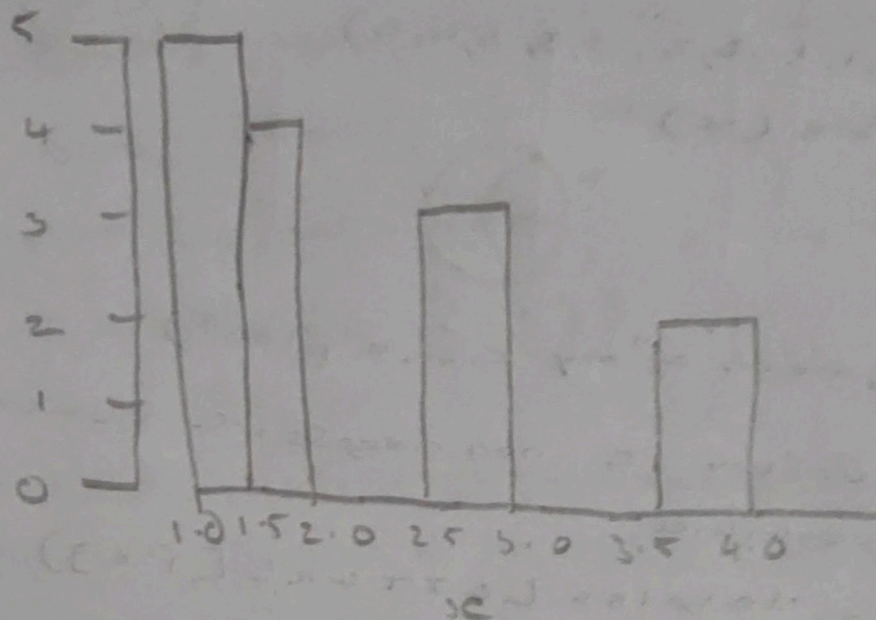
Histogram:

plot quantitative data

Function: hist()

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

```
> hist(x)
```



For viewing grouping arrangement use the

Function, cut()

```
> cut(x, 6)
```

```
[1] [,] [,]
```

```
> data.frame(x, cut(x, 6))
```

```
④ > data("cars")
```

```
> head(cars)
```

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

```
> car & speed
```

```
[1] . . . .
```

```
> hist(car & speed)
```


- breaks: one of a vector giving the breakpoints b/w histogram cells
breaks = 22: 22 cells (no. of columns)

x lab, y lab, main, xlim, ylim, col

freq = FALSE, we can get probability distribution instead of frequency

return value of hist()

h ← hist(temp)

h

break: place where break occur

count: no. of observations falling in that cell

density: density of cells

mids: mid point of cells

xname: the x argument name

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

breaks: we can specify the no. of cells we want in histogram &

we can give vectors

- Scatter Plot:

plot() function

generic x-y plotting

plot(c(1, 2, 3, 4)): plots 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 $(1,6), (2,7), (3,6), (4,4), (5,10)$

arguments: main, xlab, ylab, col

↳ type = "p", points 0.00


"l", lines ———


"b", both — o — o —


"c", for line alone of "b" — — —

"o", over plotted o o o

"h", histogram | | | |

"s", stair steps 

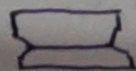
"S", other steps 

"p", no plotting 

• Bar Plots:

barplot() function

Arguments: main, xlab,
ylab, col

notch = T 

horizontal = T, represents
as horizontal

barplot(x, cex.p, wind)
box

↳ gives 3 plot in one
plot

marwidth = 1
border

