Module - 5

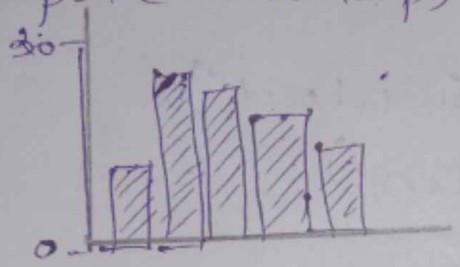
Graphs and charts

1. R barplot

can be created using function barplot() we can supply a matrix or vector to it.

eg: max.temp ← c(22, 27, 26, 24, 23)

barplot(max.temp)



the arguments

main = used to name the plot ("maximum temp

xlab = horizontal description

ylab = vertical description

names.arg = c("sun", "mon", "wed", "thu")

col = "dark red"

horiz = TRUE

plotting categorical data

age ← c(17, 18, 16, 17, 18, 19, 18, 16, 18, 18)

table(age)

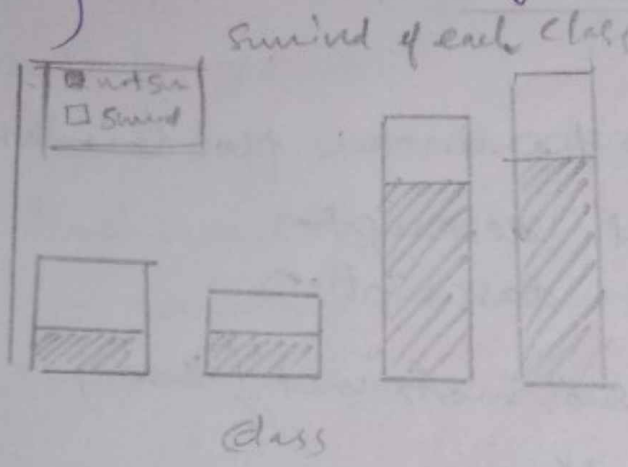
16	17	18	19
1	2	6	1

eg: barplot(table(age), main = "age of 10 students", xlab = "age", ylab = "count", border = "red", col = "blue", density = 10)

higher diamensional tables

we have a built in dataset called Titanic. It have 4 diamensions. we can plot according to the data.

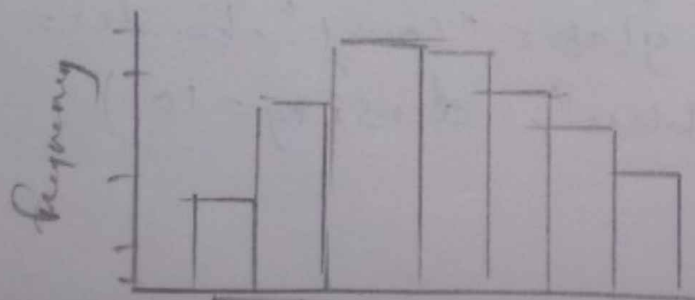
```
barplot(Titanic.data,
    main="Survival of each class",
    xlab = "class",
    col =c("red","green")
    )
legend ("top left",
    c("not survived","survived"),
    fill = c("red","green"),
)
```



2. R Histograms

Created using hist() function
This takes a vector value.
Creating a histogram using dataset airquality

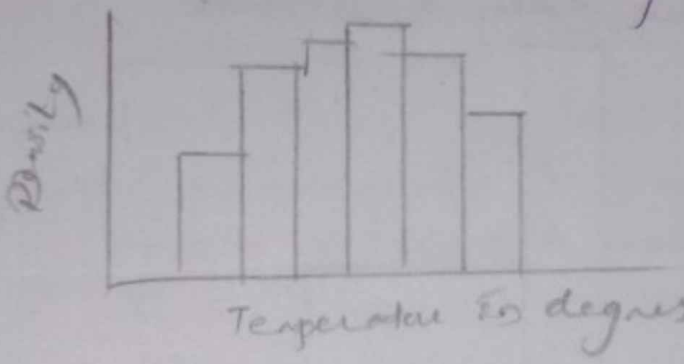
```
Temperature ← airquality $Temp
hist (Temperature)
```



Parameters can be added like

```
hist (Temperature,
main = "maximum daily temperature",
xlab = "Temperature in degrees",
xlim = c(0,100),
col = "darkblue",
freq = FALSE,
)
```

maximum daily Temperature



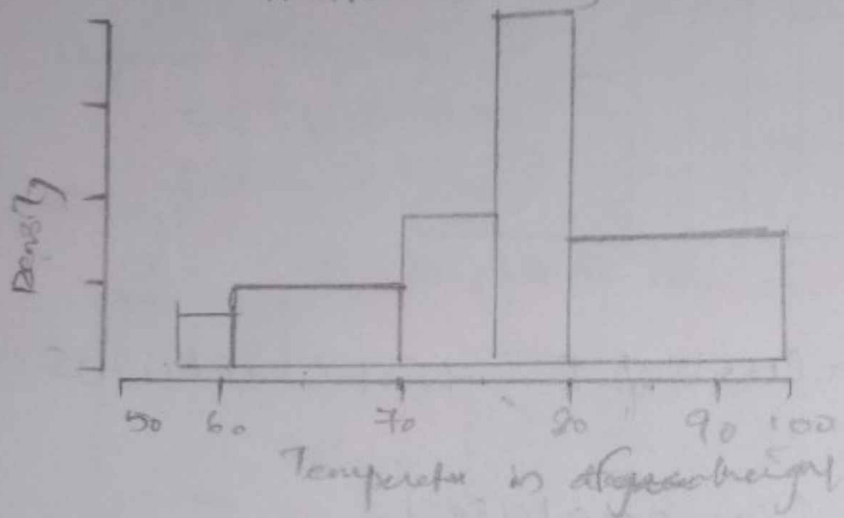
Returns values of R, hist() returns 6 components

```
>h <- hist(Temperature)
>h
$breaks, $counts, $density, $mids
$xname, $attr("class"), $equidist
```

* breaks - places where the breaks occur
* counts - no: of observations falling in that cell
* density - the density of cells
* mids - the midpoint of cells
* xname - the x argument name
* equidist - a logical value indicating the breaks are equally spaced or n

Non uniform widths:

```
hist (Temperature,
main = "maximum daily temperature",
xlab = "Temperature in farenheight",
xlim = c(50,100),
col = "chocolate",
border = "brown",
breaks = c(55,60,70,75,80,100)
)
```



3. R pie chart

created using function pie().

let > expenditure

housing	food	clothes	Entertainment	other
600	300	150	100	200

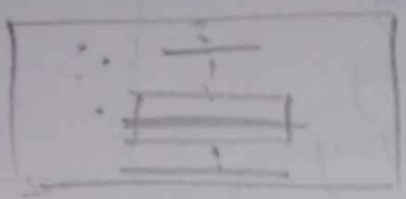
pie (expenditure), with parameters

```
pie (expenditure,
labels = as.character(expenditure),
main = "monthly expenditure",
col = c("red", "orange", "blue", "pink", "grey"),
border = "brown"
,clockwise=TRUE
)
```

4. R boxplot

Created using boxplot() function. We can pass a list or dataframe with numeric vectors to it. We can create a boxplot with dataset airquality

boxplot(airquality$Ozone)



parameters are,

boxplot(airquality$Ozone,
main = "mean ozone in ~~part~~ ppt",
xlab = "parts per billion",
ylab = "ozone",
col = "orange",
border = "brown",
horizontal = TRUE,
notch = TRUE
)

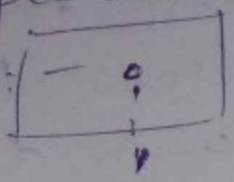
Return values are =>

* n : number of observations is drawn.
* conf - upper/lower extremes of notch
* group - a vector of same length.
* names - a vector names for the group

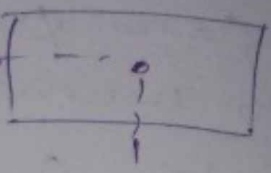
5. Scatterplot

plot(), simple scatter plot

eg: plot(1)

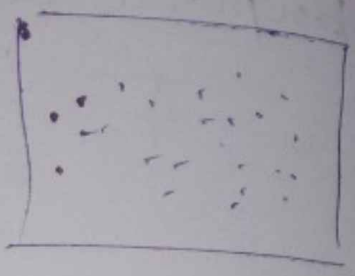


plot(3)



eg: head(airquality)

```
temp = airquality $ Day
day = airquality $ Day
Temp = airquality $ Day
plot (day, temp)
```



with arguments

```
y = 21:30
plot x,y)
plot(x,y, main = "Scatter plot",
    Xlab = "x values",
    Ylab = "y values",
    col = 1:10)
```

what type should be drawn are

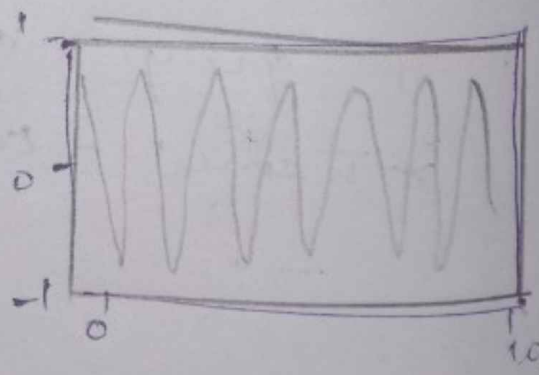
p - point, l - lines, b - both, o - overplotted

s - steps, c - without-bubble, h - histogram

n - none to display

eg:

```
X = 1:100
Y = sin(x)
plot(x,y)
plot(x,y, type = "l")
```



eg:

```
x = seq(0.10, 0.01
y = sin(x)
plot(x,y, type = "l")
```