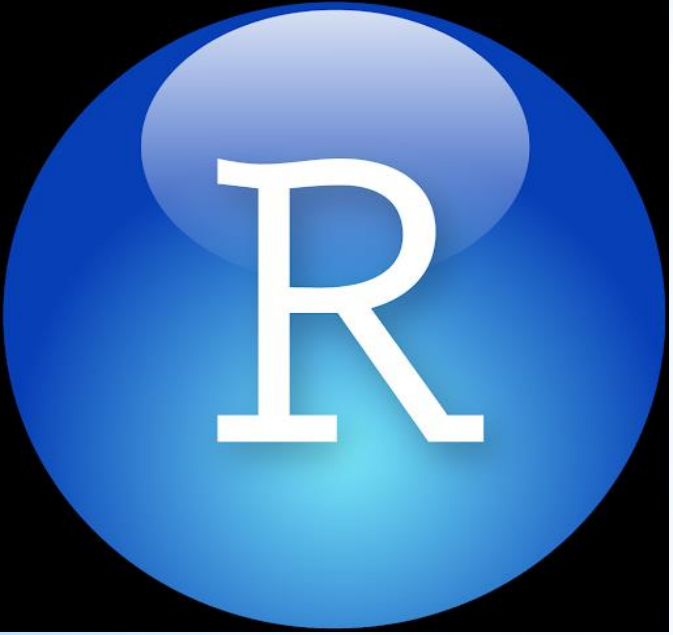


R is a programming language and environment for statistical computing and graphics.

A variety of graphs is available in R, and the use is solely governed by the context. However, exploratory analysis requires the use of certain graphs in R, which must be used for analyzing data.

Some Graphs in R are :

1. Boxplot
2. Scatter Plot
3. Histograms
4. Bar Graphs
5. Line Charts
6. Dot Plot

Now we will focus into the type of graph namely **LINE CHART**

LINE CHART

- **A line chart is a graph that connects a series of points by drawing line segments between them. Line charts are usually used in identifying the trends in data.**

- The **plot()** function in R is used to create the line graph.

- The basic syntax to create a line chart in R is –

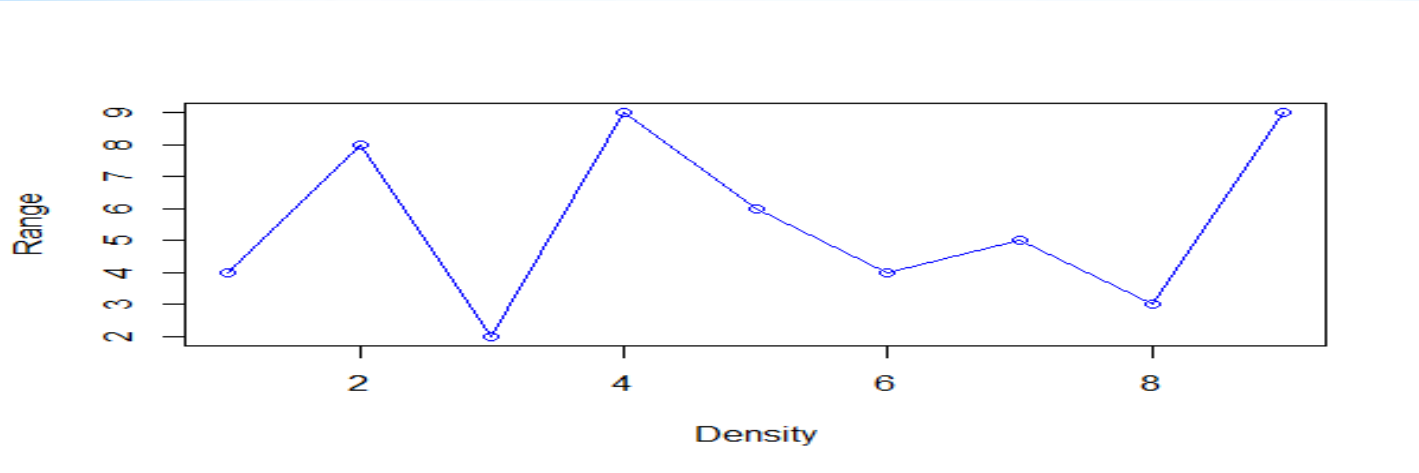
plot(v, type, col, xlab, ylab)

Following is the description of the parameters used –
v is a vector containing the numeric values.
type takes the value "p" to draw only the points, "l" to draw only the lines and "o" to draw both points and lines.
xlab is the label for x axis.
ylab is the label for y axis.
main is the Title of the chart.
col is used to give colors to both the points and lines.

NOTE : Data considered is mostly numeric
Line Chart is usually produced for time series data

Example : x<-c(4,8,2,9,6,4,5,3,9)
plot(x,type='o',col='blue',xlab='Density',ylab='Range')

OUTPUT



2. y<-c("ram","flower","tin")
plot(y,type='o',col='blue',xlab='Density',ylab='Range')

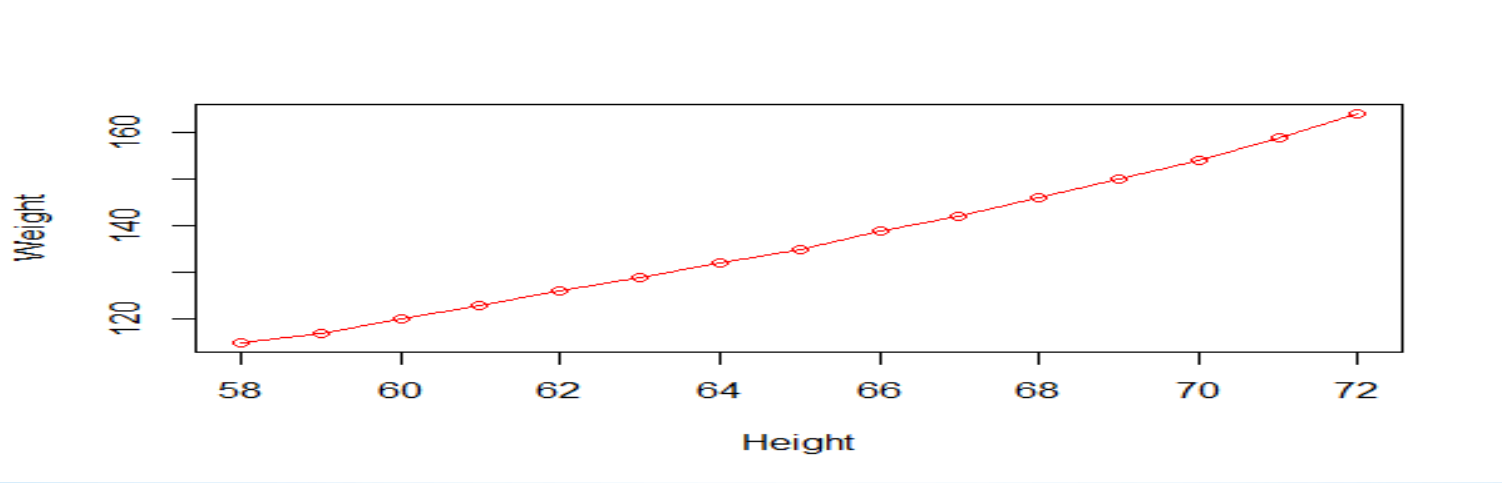
OUTPUT

Error in plot.window(...) : need finite 'ylim' values
In addition: Warning messages:
1: In xy.coords(x, y, xlabel, ylabel, log) : NAs introduced by coercion
2: In min(x) : no non-missing arguments to min; returning Inf
3: In max(x) : no non-missing arguments to max; returning –Inf

✔ DATA CONSIDERED IS USUALLY NUMERIC

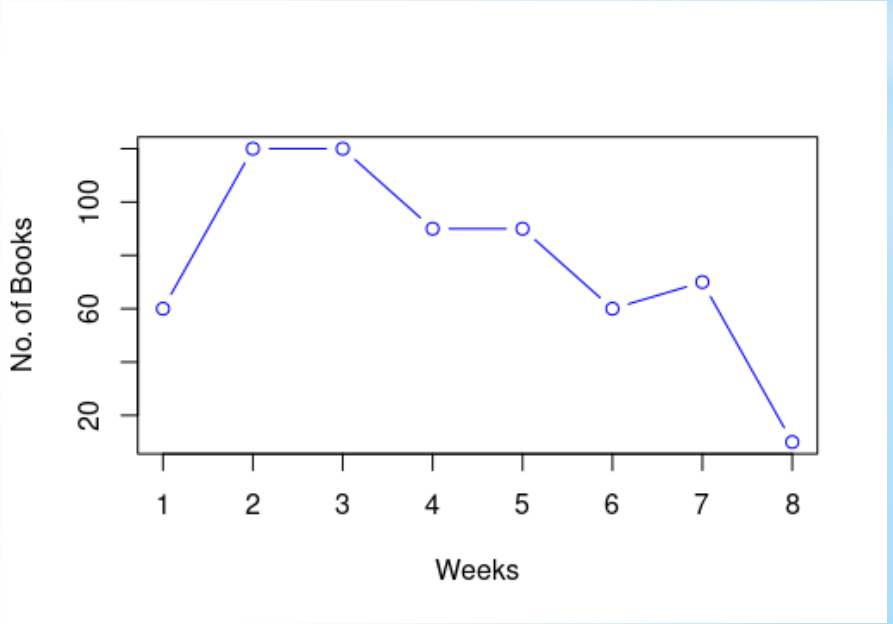
3. Women -> Inbuilt data set in R

```
plot(women,type='o',col='red',xlab='Height',ylab='Weight')
```



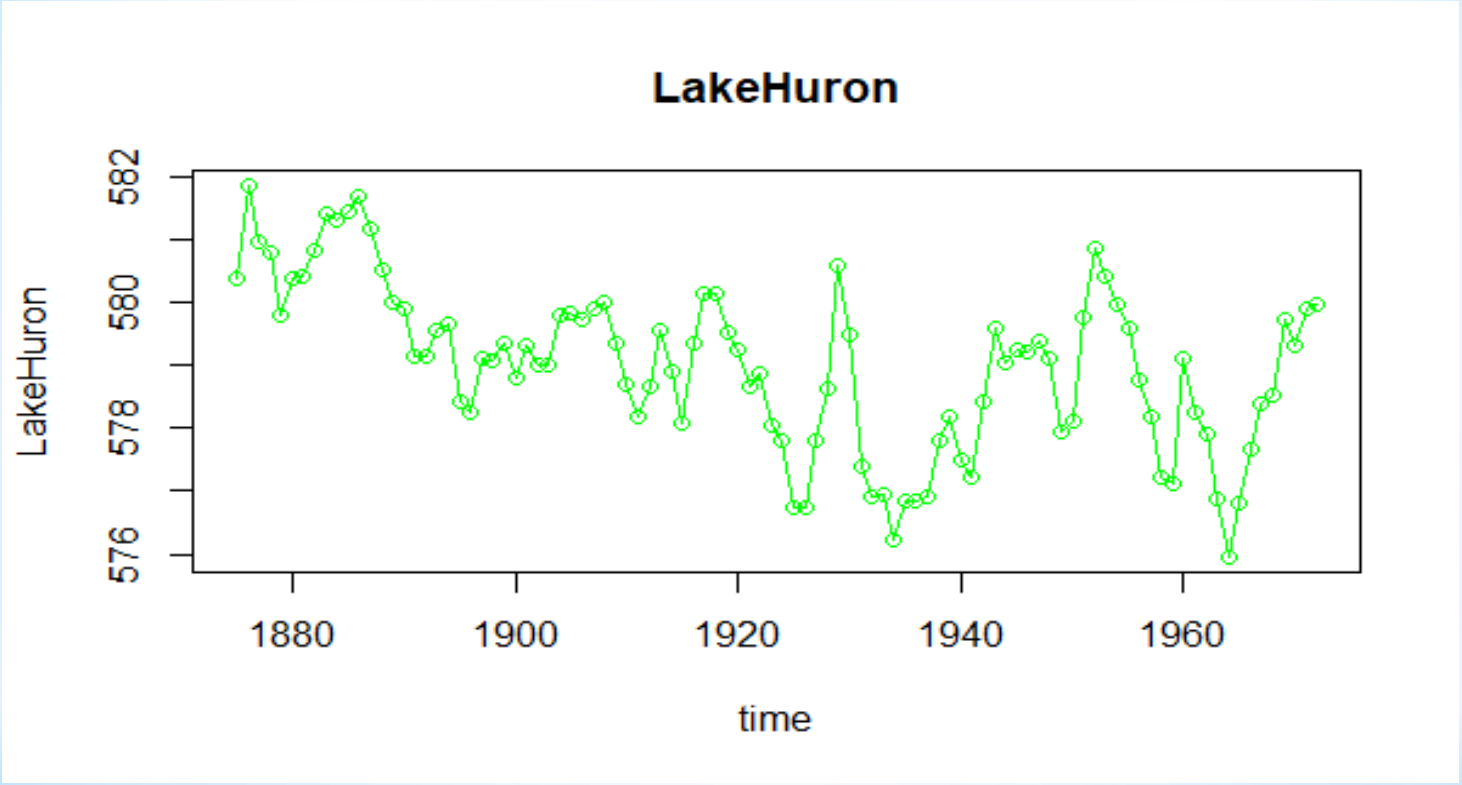
4. A line graph of the number of books sold per week during a certain period.

```
m<-c(1:8)
n<-c(60,120,120,90,90,60,70,10)
plot(m,n,type='b',col='blue',
xlab='Weeks,ylab='No. of Books')
```



* Consider an inbuilt time series “LakeHuron” in R

* plot(LakeHuron,type='o',col='green',main='LakeHuron',xlab='time',ylab='LakeHuron')

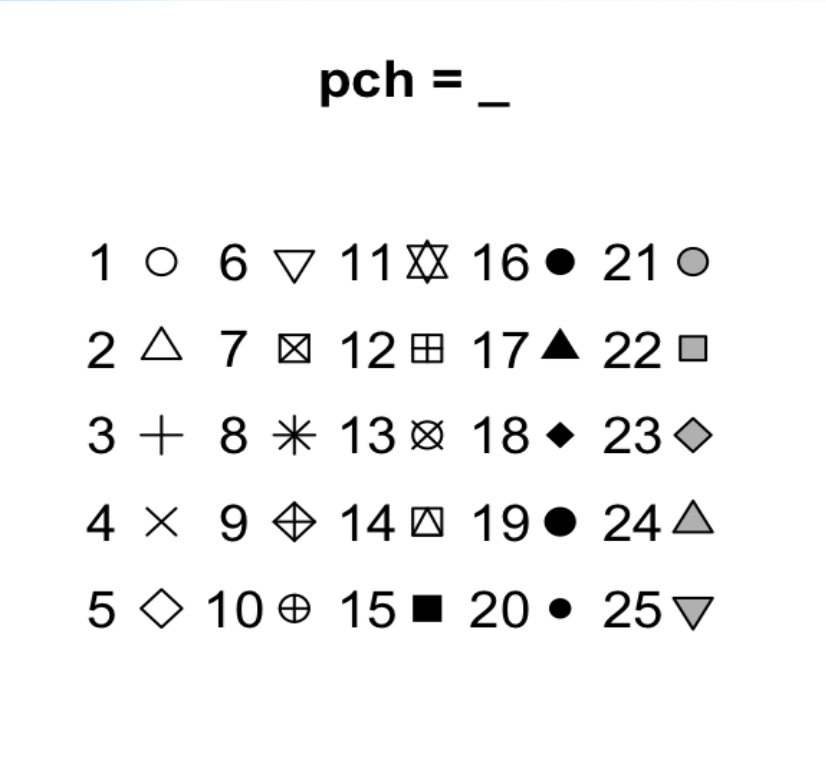


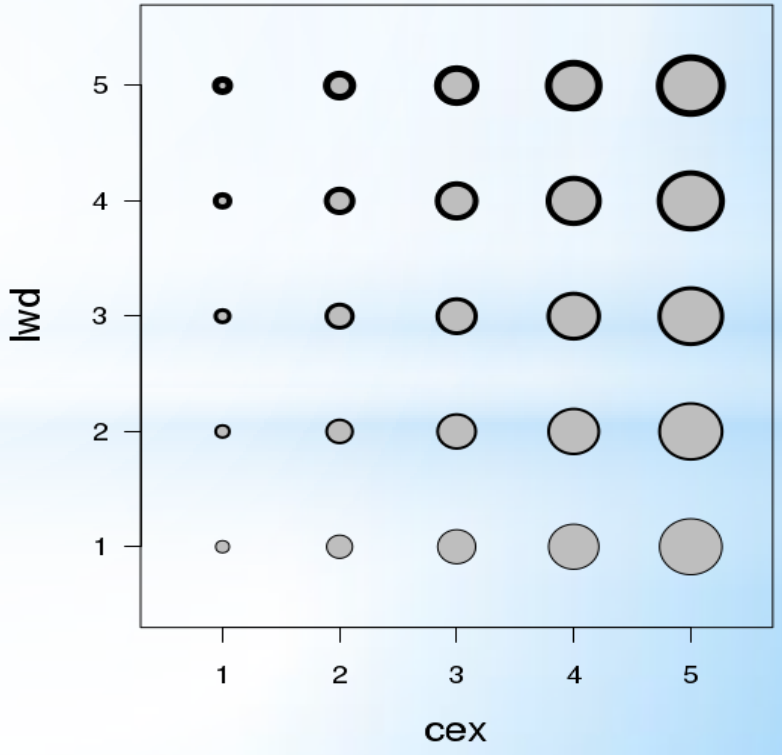
- **It represents the annual measurements of the level of water, in feet, of Lake Huron from 1875–1972.**

Plotting Symbols

We can use many other graphical parameters to modify the plot.This is useful when we want to add more points to the graph.The pch = instruction refers to the plotting characters. We can type an integer value and this code will be reflected in the symbol/character produced. For values from 0 to 25,we get symbols shown below:

We can alter the size of the plotted characters using the cex = instruction; this is a character expansion factor. So setting cex = 2 makes point twice as large as normal and cex = 0.5 makes them half normal size.
To change line width, lwd can be used.



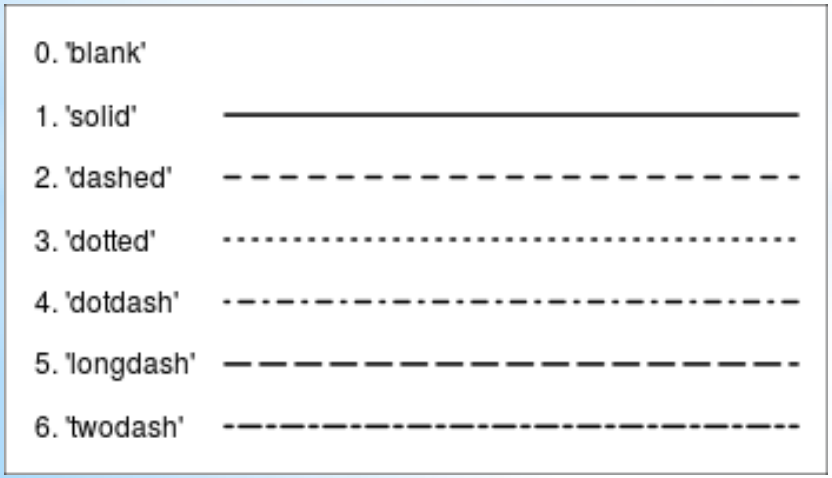


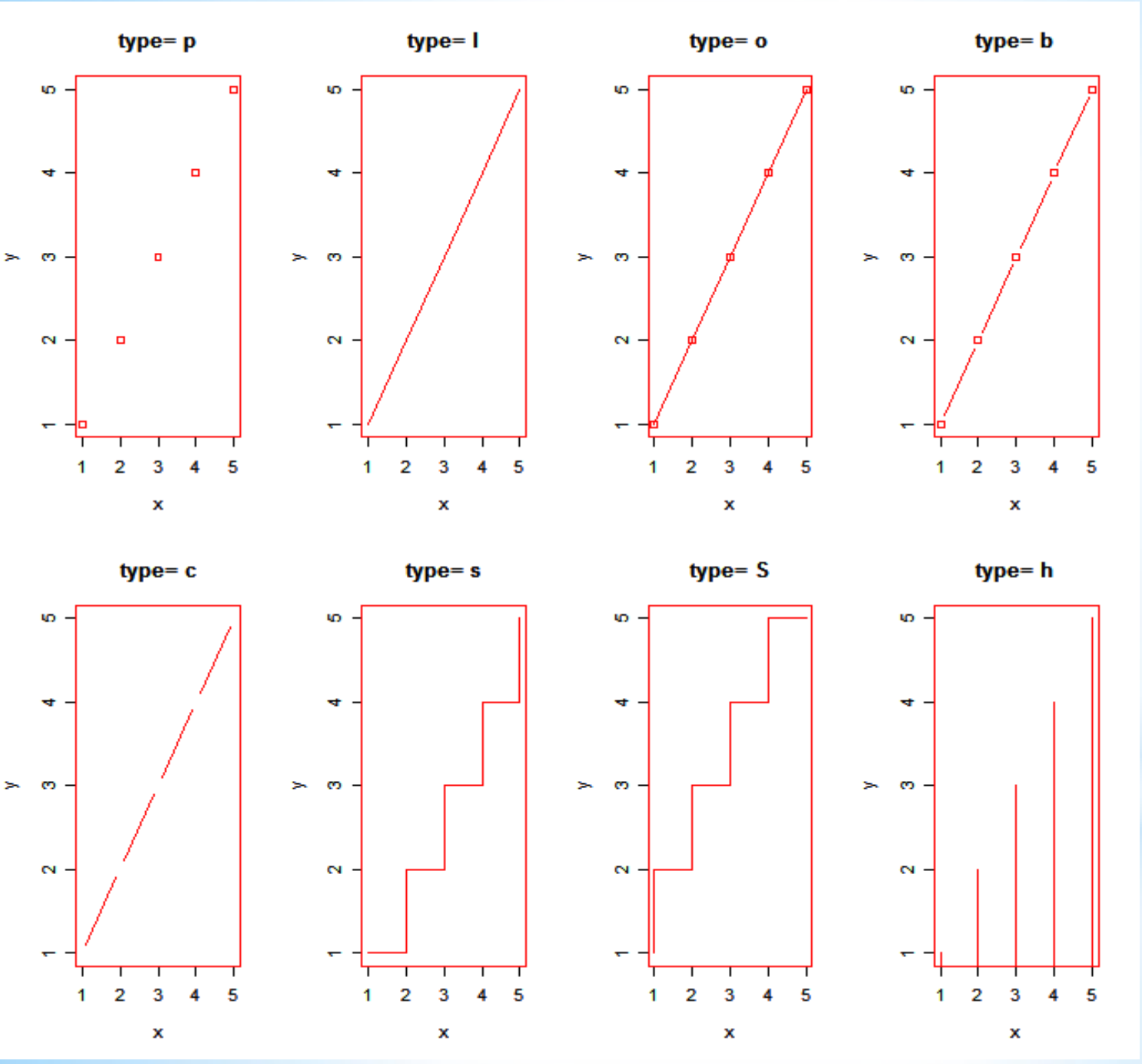
Create line plots and change line types

We can change the plot type with the argument type. It accepts the following strings and has the given effect.

- "p" - points
- "l" - lines
- "b" - both points and lines
- "c" - empty points joined by lines
- "o" - overplotted points and lines
- "s" and "S" - stair steps
- "h" - histogram-like vertical lines
- "n" - does not produce any points or lines

The argument lty is used to change the line type. A graph of the different line types is shown below:




type= p
type= l
type= o
type= b
type= c
type= s
type= S
type= h
x
y
1
2
3
4
5

Create a plot with multiple lines

Simple format of line type function:
plot(x,y,type ='l',lty=1). Create the main plot frame.
lines(x,y,type ='l',lty=1). Add lines onto the plot.

Add a legend to a base R chart

To add a legend to a plot made in base R, the legend( ) function can be used.
The main arguments are:

- legend : names to display
- col : symbol colour
- pch : symbol type
- cex : text size
- topright : legend position :bottomright, bottom, bottomleft, left, topleft, top, topright, right, center

Lets see an example

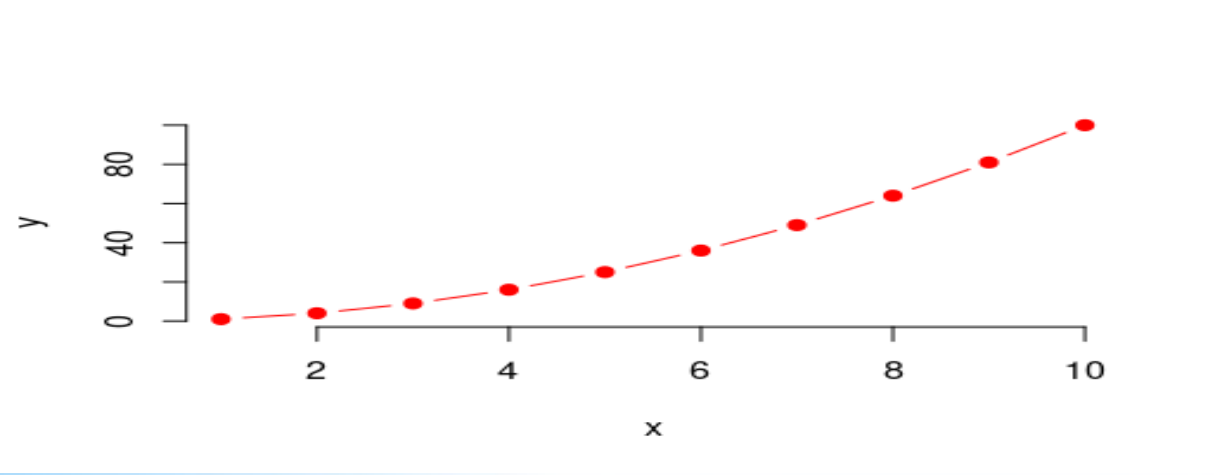
We start by plotting a first single line with a solid line type. Next we add a second line with a dashed line style.Finally we add a legend on the plot using the R base function legend(),which take the same col and lty arguments as the line function. The option cex is used to set the legend text size.

Create some variables

```
x<-c(1:10)
y1<-x*x
y2<-2*y1
```

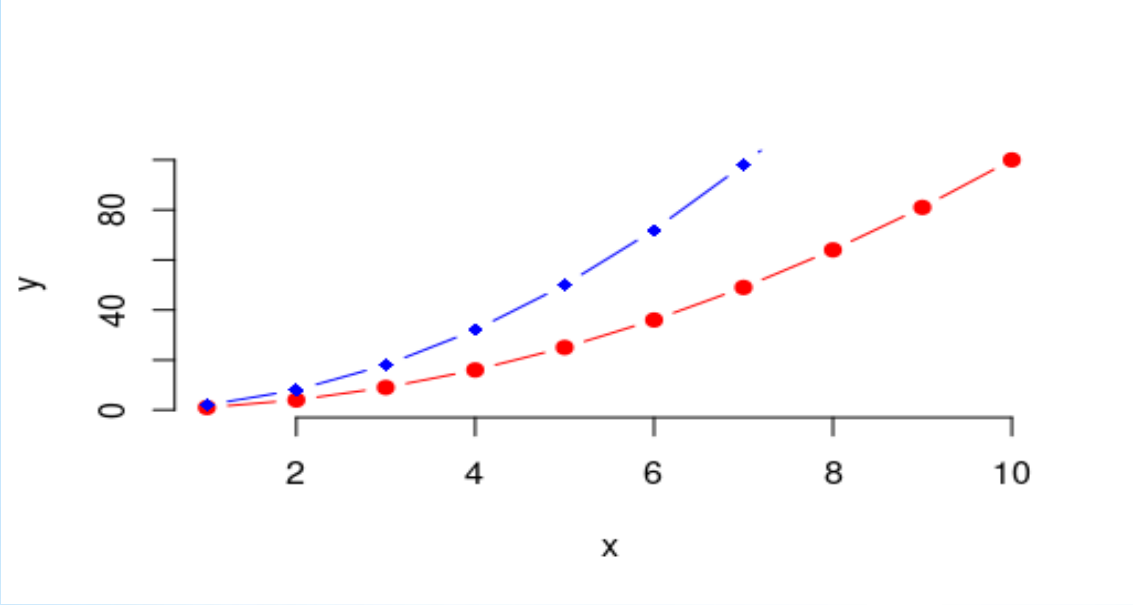
Plot a first line

```
plot(x,y1,type='b',pch=19,frame='FALSE',col='red',xlab='x',ylab='y',lty=1,lwd=1)
```



Add a second line

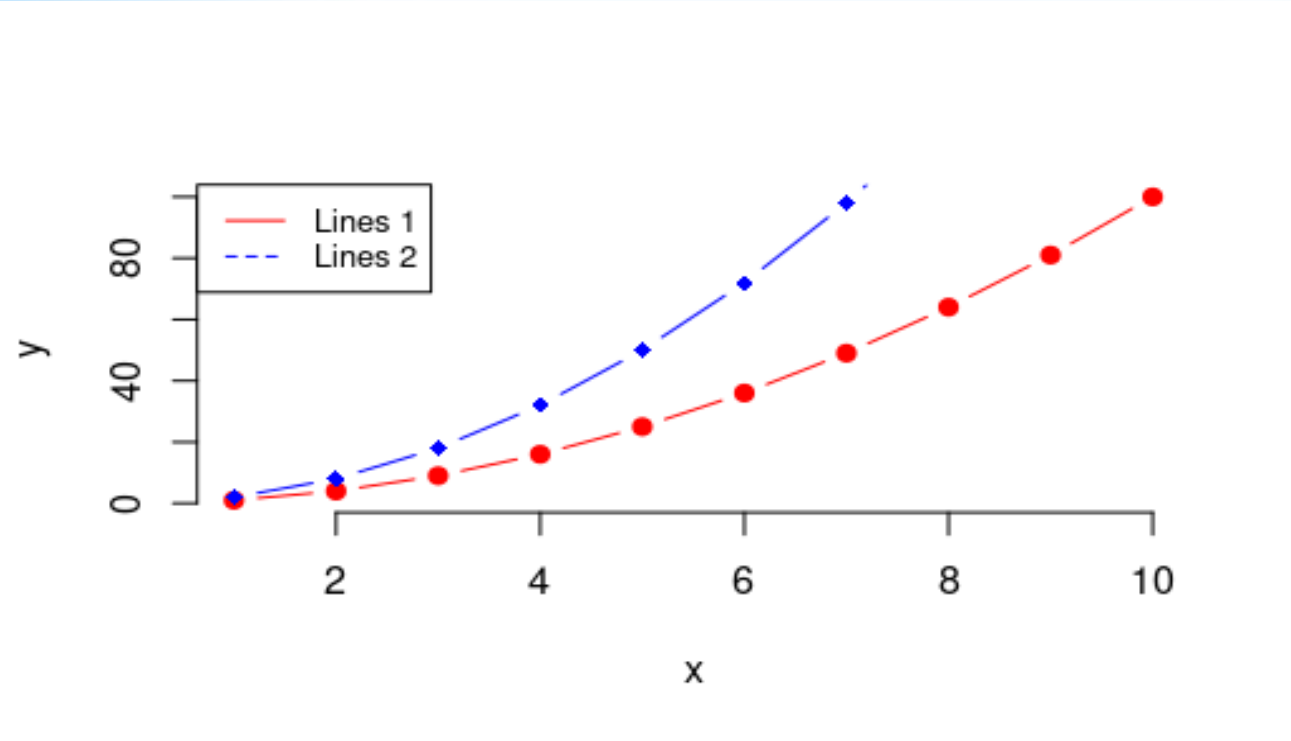
```
lines(x,y2,type='b',pch=18,col='blue',xlab='x',ylab='y',ity=2,lwd=1)
```



Add a legend to the plot

```
legend("topleft",legend=c('Lines1','Lines2'),col=c('red','blue'),lty=1:2,cex=0.8)
```

OUTPUT



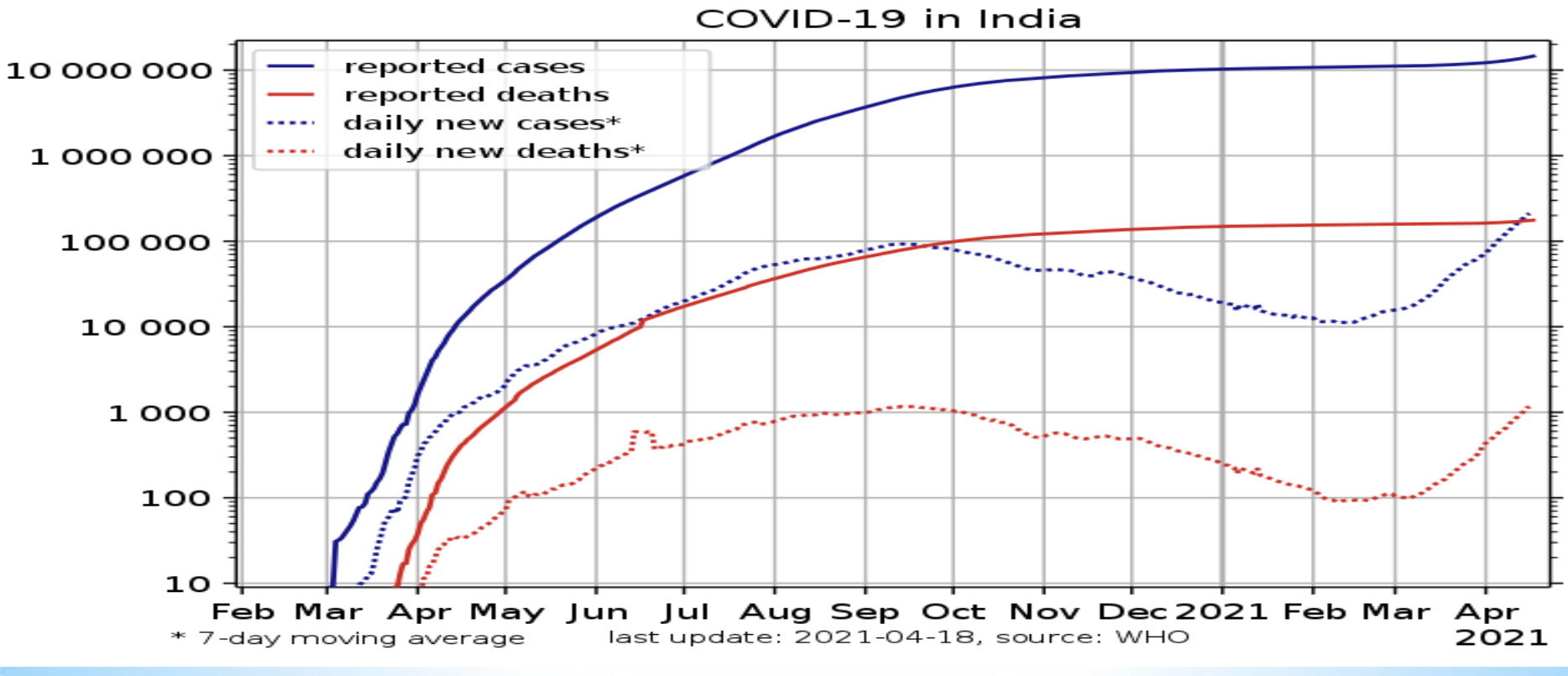
USES OF LINE CHART

- **Line charts** are **used** to track changes over short and long periods of time. When smaller changes exist, **line charts** are better to **use** than bar **graphs.**
- **Line charts** can also be **used** to compare changes over the same period of time for more than one group.
- **Line charts** can give a quick analysis of data.
- You're able to quickly tell the range, minimum/maximum, as well as if there are any gaps or clusters.
- Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.
- They can also be used to display several dependent variables against one independent variable.
- When comparing data sets, line graphs are only useful if the axes follow the same scales.

Real Life Use Of Line Chart

Line Chart can be very useful in Real life. One instance of this is the line chart representing the Situation of Covid 19

Interpreting data from such line graphs helps us to know how well our own community is doing in fighting the coronavirus pandemic



THANKYOU

BY
RIYA (MAT/19/30)
PAYAL (MAT/19/68)