# GRAPHS AND CHARTS IN R

**M. A. Iquebal**
**I.A.S.R.I., Library Avenue, New Delhi – 110012**
**asif@iasri.res.in**

R includes two different packages for plotting data: *graphics* and *lattice*. The graphics package contains a wide variety of functions for plotting data. It is very easy to customize or modify charts with graphics package, or to interact with plots on the screen. The lattice package contains an alternative set of functions for plotting data. Lattice graphics are well suited for splitting data by a conditioning variable. This chapter gives an overview of the graphics package.

## AN OVERVIEW OF R GRAPHICS

Diagrammatic representation can make strong impact on a reader because of its visual effect. However, these plots and charts should be constructed carefully because visual effect can be easily misread or misused. Some of the types of diagrams that are commonly used are described below. R includes tools for drawing most common type of charts, including bar charts, pie charts, line charts and scatter plots. Additionally, R can also draw some less familiar charts like quantile-quantile (Q-Q) plots, mosaic plots and contour plots. Each graphical function has a large number of options making the production of graphics very flexible.

## 1. PLOT

### Description

Generic function for plotting of **R** objects. For simple scatter plots, plot.default will be used. However, there are plot methods for many **R** objects, including functions, data.frames, density objects, etc. Use methods(plot) and the documentation for these.

### Usage

plot(x, y, ...)

### Arguments

| | |
|---|---|
| x | the coordinates of points in the plot. Alternatively, a single plotting structure, function or *any **R** object with a plot method* can be provided. |
| y | the y coordinates of points in the plot, *optional* if x is an appropriate structure. |
| | type<br>what type of plot should be drawn. Possible types are<br>• "p" for **p**oints,<br>• "l" for **l**ines,<br>• "b" for **b**oth,<br>• "c" for the lines part alone of "b",<br>• "o" for both '**o**verplotted',<br>• "h" for '**h**istogram' like (or 'high-density') vertical lines,<br>• "s" for stair **s**teps,<br>• "S" for other **s**teps, see 'Details' below, |

| | |
|---|---|
| | • "n" for no plotting.<br><br>main :an overall title for the plot.<br>sub :a sub title for the plot.<br>xlab:a title for the x axis.<br>ylab:a title for the y axis. |

## 2. SCATTER PLOTS

plot.default {graphics}

**The default scatterplot function**

### Description

Draw a scatter plot with decorations such as axes and titles in the active graphics window.

### Usage

plot(x, y = NULL, type = "p",  xlim = NULL, ylim = NULL,log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL,ann = par("ann"), axes = TRUE, frame.plot = axes,panel.first = NULL, panel.last = NULL, asp = NA, ...)

### Arguments

| | |
|---|---|
| x, y | thex and y arguments provide the x and y coordinates for the plot. Any reasonable way of defining the coordinates is acceptable. See the function xy.coords for details. If supplied separately, they must be of the same length. |
| type | 1-character string giving the type of plot desired. The following values are possible, for details, "p" for points, "l" for lines, "o" for overplotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines. |
| xlim | the x limits (x1, x2) of the plot. Note that x1 > x2 is allowed and leads to a ‘reversed axis’.<br>The default value, NULL, indicates that the range of the finite values to be plotted should be used. |
| ylim | the y limits of the plot. |
| log | a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic. |
| main | a main title for the plot, see also title. |
| sub | a sub title for the plot. |
| xlab | a label for the x axis, defaults to a description of x. |
| ylab | a label for the y axis, defaults to a description of y. |
| ann | a logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot. |
| axes | a logical value indicating whether both axes should be drawn on the plot. Use |

| | |
|---|---|
| | graphical parameter"xaxt" or "yaxt" to suppress just one of the axes. |
| frame.plot | a logical indicating whether a box should be drawn around the plot. |
| panel.first | an ‘expression’ to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids or scatterplot smooths. Note that this works by lazy evaluation: passing this argument from other plot methods may well not work since it may be evaluated too early. |
| panel.last | an expression to be evaluated after plotting has taken place but before the axes, title and box are added. See the comments about panel.first. |
| asp | the *y/x* aspect ratio |

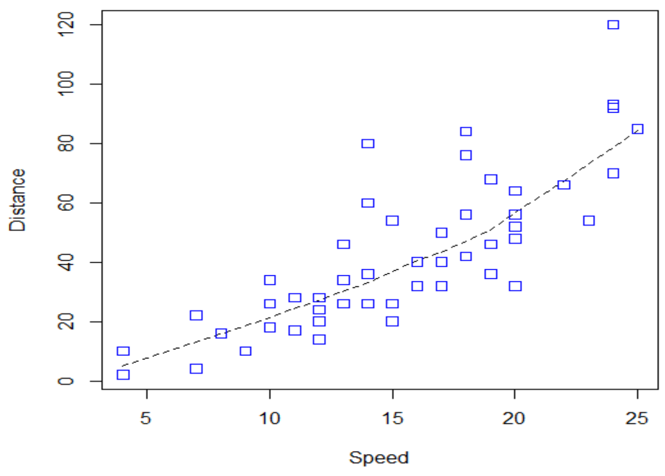
## Details

Commonly used graphical parameters are:

- col: The colors for lines and points. Multiple colors can be specified so that each point can be given its own color. If there are fewer colors than points they are recycled in the standard fashion. Lines will all be plotted in the first colour specified.
- bg: a vector of background colors for open plot symbols.
- pch: a vector of plotting characters or symbols.
- cex: a numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default. This works as a multiple of par("cex"). NULL and NA are equivalent to 1.0. Note that this does not affect annotation.
- lty: the line type.
- lwd: the line width.

## Examples

```
Speed <- cars$speed
Distance <- cars$dist
plot(Speed, Distance, panel.first = grid(8,8),pch = 0, cex = 1.2, col = "blue")
plot(Speed, Distance,panel.first = lines(stats::lowess(Speed, Distance), lty = "dashed"),pch
= 0, cex = 1.2, col = "blue")
```



## 3. BAR CHARTS

A bar chart is constructed to show frequencies of different categories of a categorical variable in the given data. The horizontal axis represents the different categories and verticle axis is used to show the number of cases (frequencies) of the characteristic in which we are interested. We explain the construction of simple bar chart using R-commands with example. To draw bar (or column) charts in R, use the *barplot* function.

**Description**
Creates a bar plot with vertical or horizontal bars.
**Usage**
barplot(height, ...)

## Default method:

```
barplot(height, width = 1, space = NULL,names.arg = NULL, legend.text = NULL,
        beside = FALSE,horiz = FALSE, density = NULL, angle = 45,col = NULL,
        border = par("fg"),main = NULL, sub = NULL, xlab = NULL, ylab = NULL,xlim
        = NULL, ylim = NULL, xpd = TRUE, log = "",axes = TRUE, axisnames =
        TRUE,cex.axis = par("cex.axis"), cex.names = par("cex.axis"),inside = TRUE,
        plot = TRUE, axis.lty = 0, offset = 0,add = FALSE, args.legend = NULL, ...)
```

**Arguments**

| | |
|---|---|
| height | either a vector or matrix of values describing the bars which make up the plot. If height is a vector, the plot consists of a sequence of rectangular bars with heights given by the values in the vector. If height is a matrix and beside is FALSE then each bar of the plot corresponds to a column of height, with the values in the column giving the heights of stacked sub-bars making up the bar. If height is a matrix and beside is TRUE, then the values in each column are juxtaposed rather than stacked. |
| width | optional vector of bar widths. Re-cycled to length the number of bars drawn. Specifying a single value will have no visible effect unless xlim is specified. |
| space | the amount of space (as a fraction of the average bar width) left before each bar. May be given as a single number or one number per bar. If height is a matrix and beside is TRUE, space may be specified by two numbers, where the first is the space between bars in the same group, and the second the space between the groups. If not given explicitly, it defaults to c(0,1) if height is a matrix and beside is TRUE, and to 0.2 otherwise. |
| names.arg | a vector of names to be plotted below each bar or group of bars. If this argument is omitted, then the names are taken from the names attribute of height if this is a vector, or the column names if it is a matrix. |
| legend.text | a vector of text used to construct a legend for the plot, or a logical indicating whether a legend should be included. This is only useful when height is a matrix. In that case given legend labels should correspond to the rows of height; if legend.text is true, the row names of height will be used as labels if they are non-null. |
| beside | a logical value. If FALSE, the columns of height are portrayed as stacked bars, and if TRUE the columns are portrayed as juxtaposed bars. |

| | |
|---|---|
| horiz | a logical value. If FALSE, the bars are drawn vertically with the first bar to the left. If TRUE, the bars are drawn horizontally with the first at the bottom. |
| density | a vector giving the density of shading lines, in lines per inch, for the bars or bar components. The default value of NULL means that no shading lines are drawn. Non-positive values of density also inhibit the drawing of shading lines. |
| angle | the slope of shading lines, given as an angle in degrees (counter-clockwise), for the bars or bar components. |
| col | a vector of colors for the bars or bar components. By default, grey is used if height is a vector, and a gamma-corrected grey palette if height is a matrix. |
| border | thecolor to be used for the border of the bars. Use border = NA to omit borders. If there are shading lines, border = TRUE means use the same colour for the border as for the shading lines. |
| main,sub | overall and sub title for the plot. |
| xlab | a label for the x axis. |
| ylab | a label for the y axis. |
| xlim | limits for the x axis. |
| ylim | limits for the y axis. |
| xpd | logical. Should bars be allowed to go outside region? |
| log | string specifying if axis scales should be logarithmic; see plot.default. |
| axes | logical. If TRUE, a vertical (or horizontal, if horiz is true) axis is drawn. |
| axisnames | logical. If TRUE, and if there are names.arg (see above), the other axis is drawn (with lty=0) and labeled. |
| cex.axis | expansion factor for numeric axis labels. |
| cex.names | expansion factor for axis names (bar labels). |
| inside | logical. If TRUE, the lines which divide adjacent (non-stacked!) bars will be drawn. Only applies when space = 0 (which it partly is when beside = TRUE). |

**Example 1:** Annual sales (in lakh of Rs.) of a pharmaceutical firm for six years (1995-2000) given below:

Table: Annual sales of pharmaceutical company

| Year | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 |
|---|---|---|---|---|---|---|
| Annual sales | 15.0 | 25.0 | 27.0 | 28.0 | 26.0 | 26.6 |

Represent the data by bar chart.
**Solution**: Following are the appropriate R commands:

```
c1<-1995:2000; #Vector of years
c2<-c(15, 25, 27, 28, 26, 26.6); #Vector of sales
sales.year<-data.frame(year=c1, sales=c2);
sales.year; #Print table
attach (sales.year);
barplot (sales, xlab="Year", ylab ="Sales", main="", col="white");
```

We have used barplot function to create bar plot.

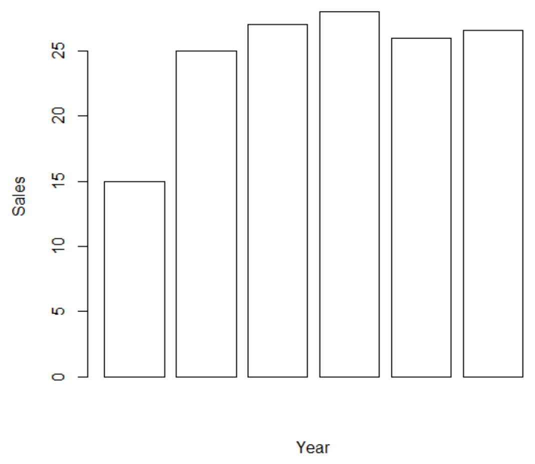


Figure: Bar diagram

**Example 2:** A group of 25 people was surveyed to find their soft-drink preference. The categories of soft-drink used in the survey were Limca, Coca-cola, Pepsi and Mangola. The data are:

3, 4, 1, 1, 3, 4, 3, 3, 1, 3, 2, 1, 2, 1, 2, 3, 2, 3, 1, 1, 1, 1, 4, 3, 1.

Represent the data by: (i) barplot of frequencies and (ii) barplot of proportions.

**Solution:** First we use the scan function to input data.

```
soft<-scan()
1:3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
Read 25 items
barplot(table(soft), xlab="Soft-drink", ylab="Frequency of preferences", main="", col="white");
```

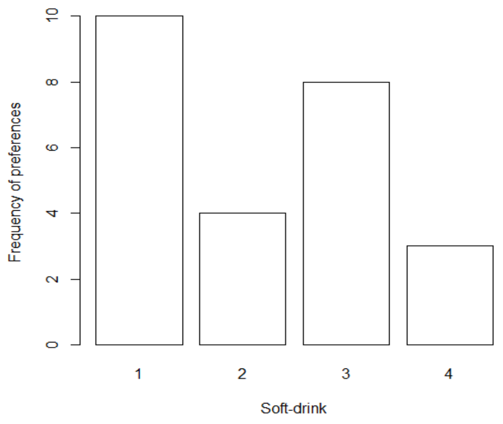


Figure: Bar diagram with frequencies

**Remark:** Observe the first argument of barplot. One has to call barplot with summarized data as its first argument; table(soft) command summarizes the data according to four categoies of soft drinks. Then this summarized categorical data is represented in form of a bar plot. Use the table function and see how you get frequencies of the four categories. Note that barplot(soft) is not the correct way to obtain the desired bar plot. We must use the following command to construct barplot of proportions:

```
barplot(table(soft)/length(soft), xlab="Soft-drink", ylab="Proportion of preferences",
main="", col="gray70");
```

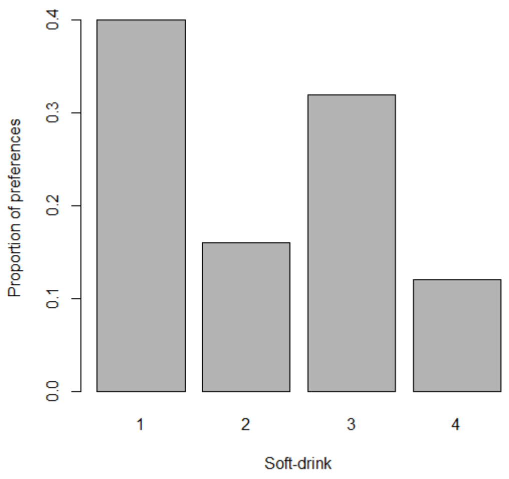


Figure: Bar diagram of proportions

**Remark:** Observe the first argument of barplot function. It is now table(soft)/length(soft). Here, length() function computes the number of observations in the data set. Thustable(soft)/length(soft) computes the category wise proportions of the four soft drinks.

## 4. SUBDIVIDED BAR DIAGRAM

Sometimes it is necessary to show break up of one variable into several components. A simple bar diagram is not useful in such a situation; a subdivided or component bar diagram may be used to represent data relating to such a variable over time or space. In case of simple bar diagram, bars (or rectangles) are drawn to represent total magnitude of the variable; each bar representing one category. In a subdivided bar plot, this bar is divided into segments, each segment representing a component of the corresponding category.

**Example:** Table shows the number of students in different faculties of a university.

| **Sr. No.** | **Year** | **Humanity** | **Science** | **Commerce** |
|---|---|---|---|---|
| 1 | 1996 | 2810 | 890 | 540 |
| 2 | 1997 | 3542 | 1363 | 471 |
| 3 | 1998 | 4301 | 1662 | 652 |
| 4 | 1999 | 5362 | 2071 | 895 |
| 5 | 2000 | 6593 | 2752 | 1113 |

(i) Represent the total number of students for different years by means of a simple bar diagram.
(ii) Represent the data as a subdivided barplot.

**Solution:**

```
year.stud<-
matrix(c(2810,890,540,3542,1363,471,4301,1663,652,5362,2071,895,6593,2752,1113),
byrow=T, ncol=3);
rownames(year.stud)<-c("1996","1997","1998","1999","2000");
colnames(year.stud)<-c("Humanity","Science","Commerce");
year.stud;
```

```
     Humanity Science Commerce
1996     2810     890      540
1997     3542    1363      471
1998     4301    1663      652
1999     5362    2071      895
2000     6593    2752     1113
```

total.stud<-margin.table(year.stud,1); # To prepare table of marginal totals we use margin.table function. The second argument 1 indicates that we are taking row totals (for column totals, the argument will be 2).
total.stud; #View the table of marginals.

```
1996  1997  1998  1999  2000
4240  5376  6616  8328 10458
```

```
barplot(total.stud, col="white", main="");
```

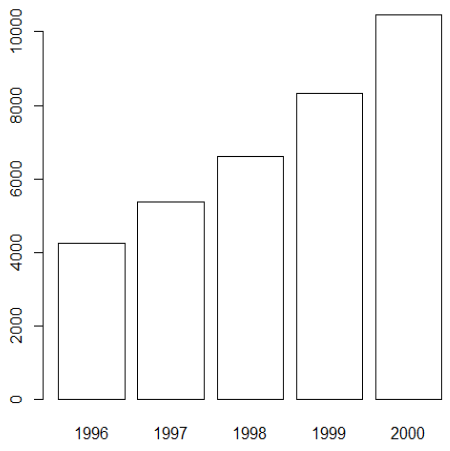


Figure: Bar diagram of marginals

```
barplot(t(year.stud), col=c("white", "gray80", "black"), main="", beside=FALSE);
```

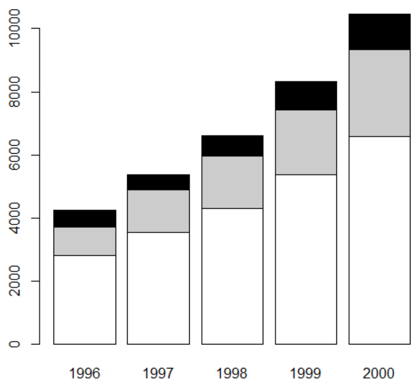


Figure: Subdivided bar plot

legend(locator(1), legend=c("hu","sc","com"), fill=c("white", "gray80", "black"), horiz=T);

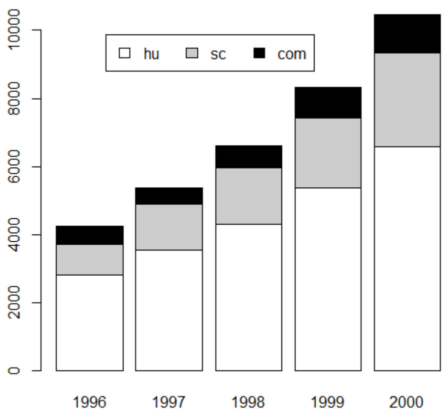


Figure: Subdivided bar plot

**Note:**

(i) Observe that we have used *matrix*() function to input data in a table form.

(ii) The function *barplot* accepts a matrix as an argument; in this case, it is the transpose of "year.stud". (t(year.stud) means transpose of "*year.stud*"). The default is a subdivided chart (that is why the argument "*beside*=FALSE").

(iii)The *locator* function will allow you to choose a location for legend. The argument 1 means you can locate one point. The argument *legend* in legend function is the text for legend and argument *fill* specifies the colours of the boxes next to legend text. The argument *horiz* causes legend to be placed horizontally when true. The default is vertical.

## 5. MULTIPLE BAR DIAGRAM

A multiple bar diagram is used for two or three-dimensional comparison. For comparison of magnitudes of one variable in two or three aspects or comparison of magnitudes of two or three variables, rectangles in a group are placed side by side. The R-commands are similar to those used for subdivided barplot. Only we change the default value of argument *beside* to true. We illustrate the construction of multiple barplot using the following data.

**Example**: The number of blood donation in the years 1995 and 2000 in various blood groups are as given in the following table. Represent the data by multiple bar diagram.

| Year | Blood Groups | | | |
|---|---|---|---|---|
| | O | A | B | AB |
| 1995 | 1154 | 526 | 775 | 155 |
| 2000 | 1700 | 1125 | 1280 | 560 |

**Solution**:

```
donation<- matrix(c(1154, 1700, 526, 1125, 775, 1280, 155, 560), ncol=4);
colnames(donation) <-c("O", "A", "B", "AB");
rownames(donation) <-c("1995", "2000");
donation; #Print table
```

```
        O    A    B  AB
1995 1154  526  775 155
2000 1700 1125 1280 560
```

```
barplot(t(donation),col=c("white", "gray50", "gray70", "black"),main="",beside=T);
legend(locator(1), legend = c("O", "A", "B", "AB"), fill = c("white", "gray50", "gray70", "black"), cex = 0.5);
```

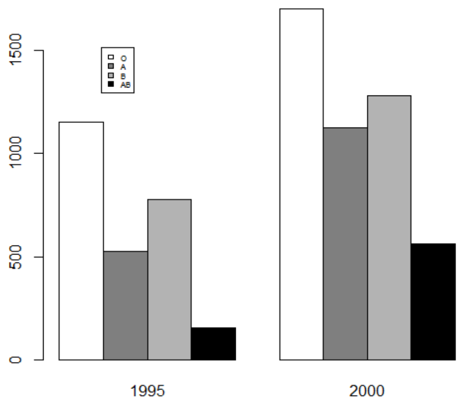


Figure: Multiple barplot

## 6. PIE CHARTS

The circle is used in a pie chart to represent the whole, and “slices” are used to represent the categories, one slice for each category. The size of a slice is proportional to relative frequency of the corresponding category.

**Example**: The tax revenue of India (in crores of Rs.), as provided in 1984085 budget, when broken into various sources are given below. Represent the data by a pie chart.

| Sources | Excise | Customs | Corporation tax | Income tax | Other |
|---|---|---|---|---|---|
| Tax revenue | 6526 | 7108 | 2568 | 560 | 763 |

**Solution**: R-commands for the pie chart are as follows:

```
pie.tax<- c(6526, 7108, 2568, 560, 763);
names(pie.tax) <- c("Excise", "Customs", "Corporation", "Income", "Other");
```

```
pie(pie.tax, main = "The Tax Revenue of India (1984-85)", col = c("white", "black",
"gray80", "gray50", "gray70"));
```

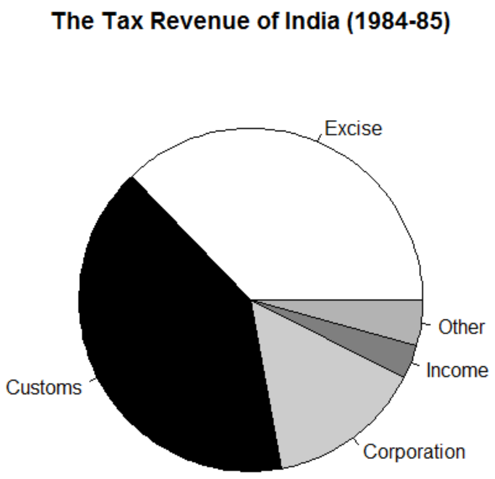


Figure: Pie plot

## 7. STEM-AND-LEAF PLOT

First, we split each number into two parts. The first part of the number serves as s “stem” and the other part that follows is a “leaf”. We divide the range into intervals of fixed length depending on the nature of data. These intervals are usually units of 0.5, 1 or 2 times a suitable power of 10. Further, the leaves belonging to each stem are ordered.

**Example**: Following are the numbers of rooms occupied in a hotel for 30 days. Draw stem-and-leaf diagram.

20 14 21 29 43 17 15 26 8 14 39 23 16 46 28 11 26 35 26 28 30 22 23 7 32 19 22 18 27 9

**Solution**:

```
x <-scan()
1: 20 14 21 29 43 17 15 26 8 14 39 23 16 46 28 11 26 35 26 28 30 22 23 7 32 19 22 18 27
9
31:
Read 30 items
>stem(x); #We use stem function to draw stem-and-leaf diagram. Output of the command
is:
```

```
  The decimal point is 1 digit(s) to the right of the |

  0 | 789
  1 | 144
  1 | 56789
  2 | 012233
  2 | 6667889
  3 | 02
  3 | 59
  4 | 3
  4 | 6
```

Interpretation of stem and leaf plot:

(i) Each number is a two-digit number. The first digit constitutes the stem and the second, a leaf. For instance, 2 is a stem and 0 is the leaf for first number 20. Altogether, there are five stems, namely 0, 1, 2, 3 and 4, as the data ranges from 07 to 46. The number 7 has stem 0 and its leaf part is 7.

(ii) In this case, the fixed length of interval is (0.5*10) = 5. The stems are written to the left of vertical line and corresponding leaves to the right.

(iii) From the diagram, reading horizontally along stem 1 we see that 11, 14 and 14 rooms are occupied on three days.

(iv) Given stem-and-leaf plot of two digit numbers we can reconstruct the given data.

(v) We can see that there is more concentration at the lower values.

**Transformation or re-expression of the data**: It is one of the most powerful tools available to the data analyst. If used effectively, it enables one to make clearer comparisons. The most common motive for the transformation is to arrive at symmetric distribution of the data. Let us see how stem-and-leaf plot displays the effect of a transformation with help of the following example.

Example: Below are 72 numbers, representing the counts of insects in agricultural experimental unit treated with insecticides.

10 11 00 03 03 11 07 07 01 05 05 09 20 21 07 12 03 15 14 11 02 06 05 22 14 16 03 04 03 15 12 14 01 03 06 16 10 17 02 05 01 13 23 17 01 05 03 10 17 19 03 05 02 26 20 21 00 05 06 26 14 07 01 02 01 24 13 13 04 04 04 13

(i) Prepare a stem-and-leaf plot.
(ii) Consider the square root transformation. For the transformed data, prepare a stem-and-leaf plot.
(iii)Comment on the result.

**Solution**: We enter data using the scan() function

```
x <- scan()
1: 10 11 00 03 03 11 07 07 01 05 05 09 20 21 07 12 03 15 14 11 02 06 05 22 14 16 03 04
03 15 12 14 01 03 06 16 10 17 02 05 01 13 23 17 01 05 03 10 17 19 03 05 02 26 20 21 00
05 06 26 14 07 01 02 01 24 13 13 04 04 04 13
73:
Read 72 items
>stem(x)

  The decimal point is 1 digit(s) to the right of the |

  0 | 0011111122222333333334444
  0 | 555555566677779
  1 | 0001112233334444
  1 | 55667779
  2 | 0011234
  2 | 66
```

The stem-and-leaf display of the data (x), output of the command stem(x), illustrates that there is more concentration at lower values. In other words, the data display a positive skew pattern. Now we give the command:

```
y=x^0.5; #Transform the data by using square root transformation.
stem(y); # Prepare stem-and-leaf for transformed data.
```

The output is:

```
The decimal point is at the |

  0 | 00
  1 | 0000004444777777777
  2 | 0000222222224446666
  3 | 02223335566667777799
  4 | 0011145566789
  5 | 11
```

We have considered square root transformation because it is commonly used for count data. The stem-and-leaf display of the transformed data (y), is nearly symmetric.

## 8. FREQUENCY DISTRIBUTION

One way to achieve summary of the data is by forming a frequency table or a frequency distribution. A frequency distribution is a list of all distinct data values along with the number of times they occur.

Preparation of frequency table: We illustrate the construction of frequency table with help of the following example.

Simulate 100 rolls of a die and prepare a frequency table.

```
roll<- sample(1:6, 100, replace = T);
roll; # Print the simulated sample.

  [1] 5 5 4 3 4 6 1 2 5 2 5 6 1 2 1 6 4 2 4 5 4 4 3 6 5 4 2 3 5 4 6 2 6 3 2 2 2
 [38] 4 4 3 5 3 2 4 1 5 2 5 3 1 6 3 2 1 2 5 2 3 1 2 2 4 3 4 3 1 6 2 6 5 3 6 3 5
 [75] 1 5 3 3 3 5 5 2 2 4 1 3 6 6 6 3 6 6 6 6 3 5 2 4 5 4

rolltab<- table(roll); #Prepare frequency table.
rolltab; #Print table.
```

Following is the formatted output:

```
roll
 1  2  3  4  5  6
10 20 19 16 18 17
```

Note: With the “sample” command, every time we will get a different sample and hence a different frequency table.

Preparation of frequency distribution: We illustrate the construction of frequency distribution with the help of following example.

Following are scores of eighty students in the examination:

62 73 85 42 68 54 38 27 32 63 68 69 75 59 52 58 36 85 88 72 52 52 63 68 29 73 29 76 29 57 46 43 28 32 09 66 72 68 42 76 38 38 39 28 19 12 78 72 92 82 72 33 92 69 28 39 85 59 68 52 85 59 76 80 72 74 54 48 29 36 10 82 58 88 68 58 46 37 29 35

```
score<- scan()

1: 62 73 85 42 68 54 38 27 32 63 68 69 75 59 52 58 36 85 88 72 52 52 63 68 29 73 29 76
29 57 46 43 28 32 09 66 72 68 42 76 38 38 39 28 19 12 78 72 92 82 72 33 92 69 28 39 85
59 68 52 85 59 76 80 72 74 54 48 29 36 10 82 58 88 68 58 46 37 29 35
81:
Read 80 items
n <- length(score);
k <-6; # Number of classes
w <- 15; # Width of class interval
i <-1:6;
x <-c(12.5+w*(i-1)); #12.5 is the midpoint of the first class, namely, 5-19
x; # Print x
[1] 12.5 27.5 42.5 57.5 72.5 87.5
f <-1:6;for(i in 1:6){f[i]<-length(score[(x[i]-w/2)<=score & score <=(x[i]+w/2)])};
f; #Print vector frequencies
[1]  4 13 15 16 23 11
fr.dist<-data.frame(midvalue = x, frequency = f);
fr.dist; # Print frequency distribution

  midvalue frequency
1     12.5         4
2     27.5        13
3     42.5        15
4     57.5        16
5     72.5        23
6     87.5        11
```

Example: Following are heights of 45 students. Heights are recorded in centimetres. Prepare a frequency distribution of the data. Use the rule: First k for which $2^k$>=n, to determine the number of k classes for n = 45 observations.

170 151 154 160 158 154 171 156 160 157 160 157 148 165 158 159 155 151 152 161 156 164 156 163 174 153 170 149 166 154 166 160 160 161 154 163 164 160 148 162 167 165 158 158 176

**Solution**: From the rule we get k = 6. The smallest observation is 148 and the largest observation is 176. The difference is 28. The class width is 28/6=6.67. This number is nearly equal to 5, which is more convenient to use. Height is a continuous variable. What needs to be done is to choose a set of proper class intervals or class boundaries. For example, with a 5 centimetres class interval, all heights between and including 148 and 152 will be placed in same class. Since these heights are measured to the nearest centimetre, this particular class interval would extend from 147.5 to 152.5. These limits are known as proper class limits or class boundaries. These class boundaries will not change the mid-value of the class. The mid-value of the first class is 150.

```
height<- scan()
1: 170 151 154 160 158 154 171 156 160 157 160 157 148 165 158 159 155 151 152 161
156 164 156 163 174 153 170 149 166 154 166 160 160 161 154 163 164 160 148 162
167 165 158 158 176
46:
Read 45 items
n <- length(height);
k <- 6;
w <-5;
i <-1:6;
x <-c(150+w*(i-1)); # Vector of mid points
x;
The output is:
[1] 150 155 160 165 170 175
```

Following loop will compute the frequencies of the classes

```
f <-1:6; for(i in 1:6){f[i] <- length(height[(x[i]/w/2) <= height & height <= (x[i]+w/2)])};
f;
[1]  6 17 31 40 43 45

fr.dist<- data.frame(midvalue = x, frequency = f);
fr.dist;
```

```
  midvalue frequency
1      150         6
2      155        17
3      160        31
4      165        40
5      170        43
6      175        45
```

**Graphical representation of frequency distribution**: Frequency distributions are easier to visualize if they are represented graphically. We illustrate the construction of the various graphs to represent frequency distribution using R-commands.

Rod or spike graph: The graph is used to represent the frequency table or ungrouped frequency distribution of a discrete variable.
Example: Four similar coins are tossed 100 times. The number of heads (x) in each of the 100 tosses was noticed. Following is the frequency table of number of heads in 100 tosses of the four coins.

| Number of heads (x) | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Frequency (f) | 6 | 28 | 36 | 25 | 5 |

Prepare a rod chart or spike chart.

**Solution**: R-commands for spike chart are as follows.

```
X <- c(0, 1, 2, 3, 4);
f <- c(6, 28, 36, 25, 5);
plot(x, f, "h", xlab="Number of success", ylab="Frequency");
points(x, f, pch=16); # The argument pch gives the type of the point
```

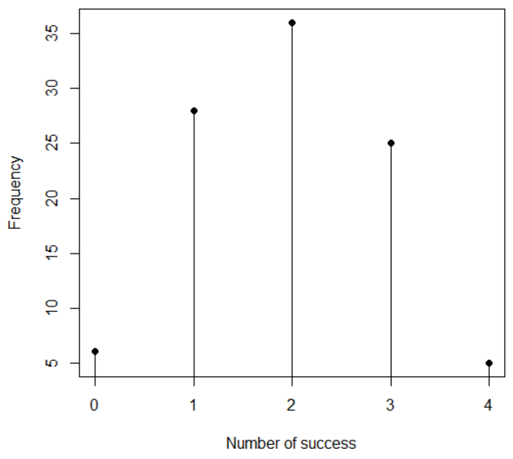


Figure: Spike chart

## 9. HISTOGRAM

Histogram is commonly used for continuous variables. It may also be used for a grouped frequency distribution of discrete variable. It is drawn as described below.

(a) The class boundaries are marked along the x-axis and are taken as bases of the rectangles.
(b) On each base (class), a rectangle is drawn with the area proportional to the frequency or relative frequency (that is, proportion) of that class.

**Illustration**: Suppose we are given 25 observations as follows:

29.6 28.2 19.6 13.7 13.0 7.8 3.4 2.0 1.9 1.0 0.7 0.4 0.4 0.3 0.3 0.3 0.3 0.3 0.2 0.2 0.1 0.1 0.1 0.1 0.1

First we use the scan function to input the data and then draw some histograms.

```
x <- scan()
1: 29.6 28.2 19.6 13.7 13.0 7.8 3.4 2.0 1.9 1.0 0.7 0.4 0.4 0.3 0.3 0.3 0.3 0.3 0.2 0.2 0.1 0.1
0.1 0.1 0.1
26:
```

Read 25 items
hist(x, xlab ="", ylab="Frequency", main="", col=gray(0.9));

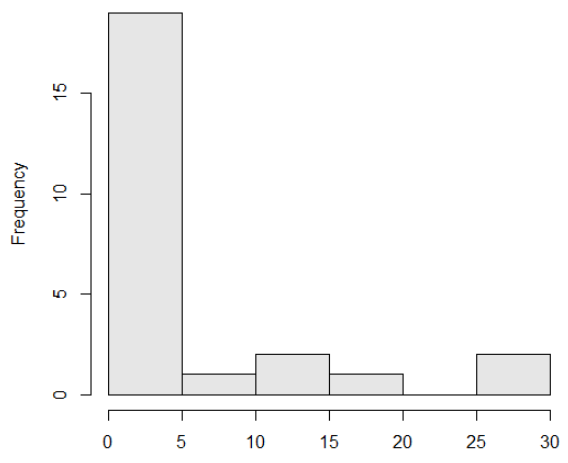


Figure: Histogram of frequencies

hist(x, prob=T, xlab="", ylab="Relative Frequency", main="", col=gray(0.9)); #Histogram of proportions

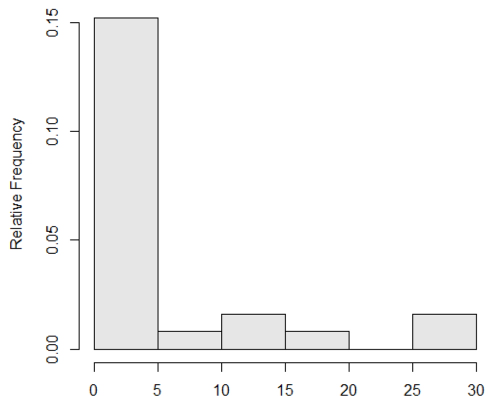


Figure: Histogram of proportions

hist(x, breaks=c(0, 1, 2, 3, 4, 5, 10, 20, max(x))); #Specify break point i.e. class limits. 9 break points are specified.

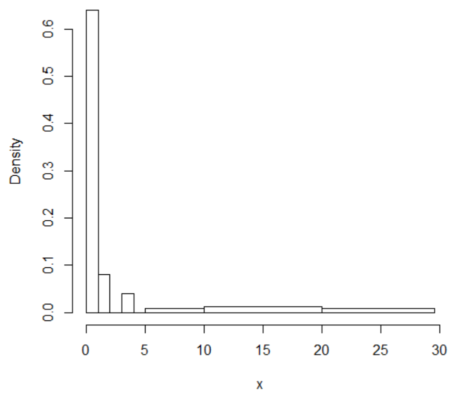


Figure: Histogram of proportions with eight break points

```
hist(x, breaks=c(0, 5, 10, 20,max(x))); #Only 5 break points are specified.
```

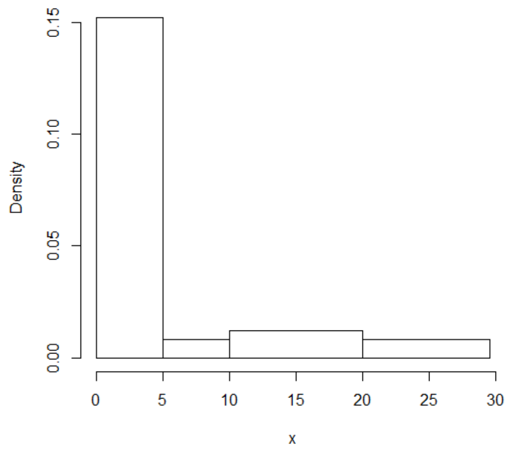


Figure: Histogram of proportions with eight break points

**Histogram for grouped frequency distribution**: The histogram drawn so far used the raw data as their argument. It is possible that we are given data in the form of a grouped frequency distribution. The following example explains the method of drawing histogram for such a data.

Illustration: Example: Following are heights of 45 students. Heights are recorded in centimetres.

170 151 154 160 158 154 171 156 160 157 160 157 148 165 158 159 155 151 152 161
156 164 156 163 174 153 170 149 166 154 166 160 160 161 154 163 164 160 148 162
167 165 158 158 176

A. Suppose we are given raw data:
B. Suppose we given grouped frequency distribution constructed from the raw data of part A

```
Midvalue Frequency
                         1    150     6
                         2    155    17
                         3    160    31
                         4    165    40
                         5    170    43
                         6    175    45
```

R-commands:

```
midvalue<- c(150, 155, 160, 165, 170, 175)
frequency<- c(6, 11, 14, 9, 3, 2);
y <- rep(midvalue, frequency);
brk<- seq(147.5, 177.5, 5);
brk;
[1] 147.5 152.5 157.5 162.5 167.5 172.5 177.5
```

```
Par(mfrow=c(1, 2));
hist(x, breaks=brk, col="white");
hist(y, breaks=brk, col="white");
par(mfrow=c(1, 1));
```

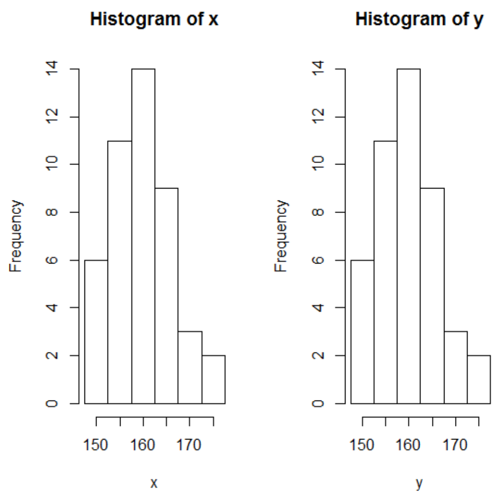


Figure: Histograms for Raw (x) and grouped data

**Note**: Observe that vector x contains raw data i.e. 45 observations. The vector y also contains 45 observations generated from given grouped frequency distribution under the assumption that the frequency of each class is concentrated at the mid point of the class interval. It may be noted that the breakpoints for histogram of raw data are taken as class limits of the grouped data. We have prepared histogram for x as well as y for comparison. Thus, histogram for frequency distribution can be constructed by using the same class limits of the frequency distribution as the break points and making usual assumption that the frequency of the class is concentrated at mid points of the class intervals.

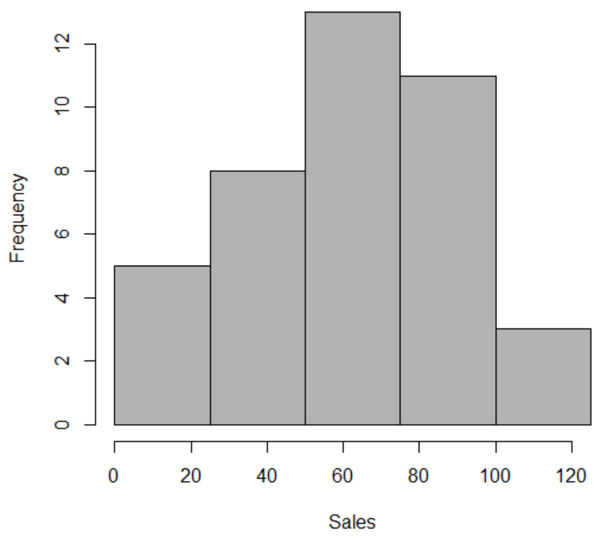
**Example**: The following table shows the frequency distribution for Book Store Sales Data. Construct the histogram for this data set.

Table: Frequency distribution of Daily sales

| Class Interval (Daily sales | Frequency |
|---|---|
| 0-25 | 5 |
| 25-50 | 8 |
| 50-75 | 13 |
| 75-100 | 11 |
| 100-125 | 3 |

**Solution**: The following figure displays the histogram using the following R-commands.

```
midx<- seq(12.5, 112.5, 25);
frequency<- c(5, 8, 13, 11, 3);
y <-rep(midx, frequency);
brk<- seq(0, 125, 25);
hist(y, breaks = brk, xlab = "Sales", main = "", col = "gray70");
```



Histogram of Sales of Books

**Remarks**: The rule of equal class intervals is inconvenient when data are spread over wide range but are highly concentrated in a small part of the range with relatively few numbers elsewhere. Using smaller intervals where the data are highly concentrated and larger intervals where the data are sparse helps to reduce the loss of information due to grouping. Tabulations of income, population and some such characteristics in official reports are often made with unequal class intervals. We consider one such example.

**Example**: The following table shows the projected population (in millions) of a country for the year 2005. The projections are broken down by age groups where grouping follow natural areas of interest such as preschool (below 5 years), education group ( divided into 3 intervals, 5-13, 14-17 and 18-24), adult group (covering 25-64 years with 4 intervals of equal widths) and finally senior citizens' group (65 and above). Construct a histogram for the data.

Table: Projected population

| Age Group | Projected Population |
|---|---|
| Below 5 | 18 |
| 5-14 | 35 |
| 14-18 | 16 |
| 18-25 | 25 |
| 25-35 | 34 |
| 35-45 | 41 |
| 45-55 | 36 |
| 55-65 | 22 |
| 65 and above | 32 |

**Solution**: We assume arbitrarily large upper bound, say 100, for the last class, which is open end class.

```
midx<- seq(12.5, 112.5, 25);
frequency<- c(5, 8, 13, 11, 3);
```

```
y <-rep(midx, frequency);
brk<- seq(0, 125, 25);
hist(y, breaks = brk, xlab = "Sales", main = "", col = "gray70");
midx<-c(2.5, 9.5, 16, 25, 30, 40, 50, 60, 82.5);
frequency<-c(18, 35, 16, 25, 34, 41, 36, 22, 32);
brk<-c(0.5, 14, 18, 25, 35, 45, 55, 65, 100);
y <-rep(midx, frequency);
hist(y, breaks=brk, xlab="Age Group", ylab="Agewise Projected Population",
col="gray70");
```

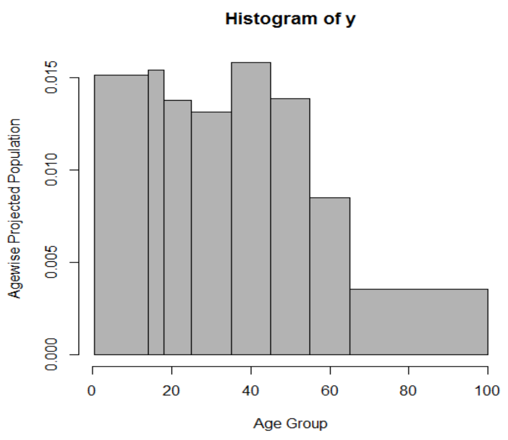


Figure: Histogram of frequency distribution with open-end-class

## 10. FREQUENCY POLYGON

One graphical method to clarify and amplify the content of the frequency table if the frequency polygon. It is obtained by joining the points $(x_i, f_i)$, where $x_i$ is the midpoint of the ith class interval and $f_i$ is the corresponding frequency. In order to close the frequency polygon, two additional points $(x_0, 0)$ and $(x_{k+1}, 0)$ are added. Either (i) $x_0$ is the lower limit of the lowest class interval and $x_{k+1}$ is the upper limit of the highest class interval or (ii) $x_0$ is the midpoint of the class interval previous to the lowest class interval and $x_{k+1}$ is the midpoint of the class interval next to the highest class interval. We explain R-commands for plotting frequency polygon with examples.

Example: Following table shows the frequency distribution of college student according to their pocket money (daily). Draw histogram and frequency polygon on the same graph paper.

Frequency Distribution of Pocket Money

| Pocket Money (In Rs) | 20-29 | 30-39 | 40-49 | 50-59 | 60-69 | 70-79 | 80-89 |
|---|---|---|---|---|---|---|---|
| No. of Students | 10 | 24 | 18 | 12 | 8 | 5 | 3 |

**Solution**: Figure shows the graph drawn using following R-commands.

```
midx<- seq(25, 85, 10);
frequency<-c(10, 24, 18, 12, 8, 5, 3);
```

```
x<- rep(midx, frequency);
brk<-seq(20, 90, 10);
hist(x, breaks=brk, main="", xlab="Pocket Money", ylab="No. of Students");
temp<-hist(x, xlab="Pocket Money", ylab="No. of Students", main="", col=gray(0.8));
lines(c(min(temp$breaks), temp$mids, max(temp$breaks)), lwd = 2, c(0, temp$counts, 0),
type = "l");
```

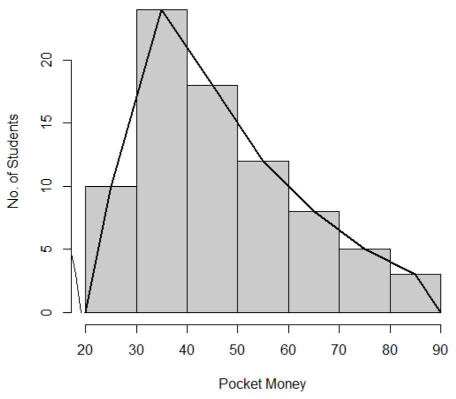


Figure: Histogram and frequency polygon

**Example**: Following table gives the distribution of deaths from scarlet fever classified according to age. Plot frequency polygon.

Frequency Distribution of Deaths

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Age (mid value) | 2.5 | 7.5 | 12.5 | 17.5 | 22.5 | 27.5 | 32.5 |
| No. of Deaths | 322 | 213 | 70 | 27 | 26 | 17 | 16 |
| Age (mid value) | 37.5 | 42.5 | 47.5 | 52.5 | 57.5 | 62.5 | |
| No. of Deaths | 11 | 10 | 7 | 6 | 5 | 1 | |

Solution: R-commands:

```
midx<- seq(-2.5, 67.5, 5);
frequency<-c(0, 322, 213, 70, 27, 26, 17, 16, 11, 10, 7, 6, 5, 1, 0);
plot(midx, frequency, type="o", xlab = "Age", ylab = "Deaths", main="", lwd = 2);
```

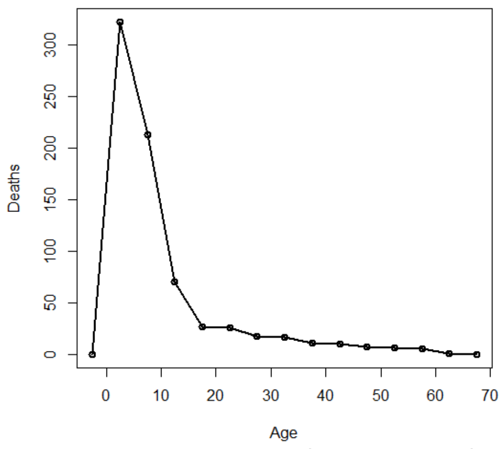


Figure: Age-wise deaths from scarlet fever

**Note**: We have used the plot() function to draw frequency polygon. The first two arguments to this function are x and y coordinates respectively. The third argument specifies the type of plotting parameter. The default for this is “p” (point. You can use “l” (line), or “b” (line and point both) to see the difference between the outputs. We have used “o” (overlap of point and line).

**Graphs showing cumulative frequencies**: “Less than” type cumulative frequency of a class is the number of observations less than or equal to the upper limit (upper class boundary) of the class. Similarly, “more than” type cumulative frequency of a class is the number of observations bigger than lower limit (lower class boundary) of the class. The cumulative frequency distribution is represented by “ogive” or cumulative frequency curve. We illustrate the construction of these two types of curves.

**Example**: The following frequency distribution relates to lives of 400 light bulbs. Draw “less than” and “more than” type ogive curves on the small graph paper.

Table: Life of Bulbs

| Life of bulb (in hrs) | 600-699 | 700-799 | 800-899 | 900-999 | 1000-1099 |
|---|---|---|---|---|---|
| No. of bulbs | 85 | 77 | 124 | 78 | 36 |

**Solution**: The following figure is obtained by using following R-commands:

```
f <-c(0, 85, 77, 124, 78, 36, 0);
lc<-cumsum(f);
lc;
[1]   0  85 162 286 364 400 400
uc<-1:7;for(i in 7:1){uc[i]<-sum(f[7:i])};uc
[1] 400 400 315 238 114  36   0
lbx<-seq(499.5, 1099.5, 100); #The vector of lower class boundaries.
lbx;
[1]  499.5  599.5  699.5  799.5  899.5  999.5 1099.5
ubx<-seq(599.5, 1199.5, 100); #The vector of upper class boundaries.
ubx:
[1]  599.5  699.5  799.5  899.5  999.5 1099.5 1199.5
plot(ubx, lc, type = "l", xlim=c(499.5, 1199.5), xlab = "Class interval", ylab="Cumulative Frequency", main= "", lwd = 2);
lines(lbx, uc, lty = 2, lwd = 2);
```

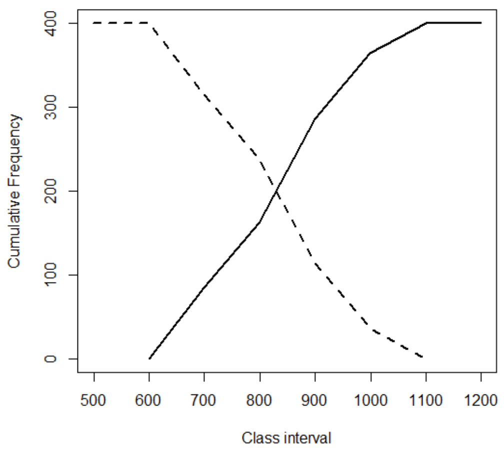


Figure: Ogive

**Note**: We have used one more argument xlim (limits of x) to plot function. It is necessary because we want to include lower as well as upper class boundaries.

## REFERENCES

Murrell, P. (2005) *R Graphics*. Chapman & Hall, CRC Press.

Purohit, G. S., Gore S. D. and Deshmukh, S. R. (2010). *Statistics using R*. Narosa Publishing House Pvt Ltd., New Delhi.