Mathematical Computations Using R

II B.Sc Statistics

Unit-I

1. History of R programming

- R was created by Ross Ihaka and Robert Gentleman at the University
- of Auckland, New Zealand, which is currently developed by the R Development Core Team.
- R made its first appearance in 1993.
- This programming language was named R, based on the first letter of first name of the two R authors (Robert Gentleman and Ross Ihaka), and partly a play on the name of the Bell Labs Language S.
- A large group of individuals has contributed to R by sending code and bug reports.
- Since mid-1997 there has been a core group (the "R Core Team") who can modify the R source code archive.

1.2 R Commands

help() Obtain documentation for a given R command
c(), scan() Enter data manually to a vector in R
seq() Make arithmetic progression vector
rep() Make vector of repeated values
data() Load (often into a data.frame) built-in dataset
View() View dataset in a spreadsheet-type format
str() Display internal structure of an R object read.csv(),
read.table() Load into a data.frame an existing data file
library(), require() Make available an R add-on package
dim() See dimensions (# of rows/cols) of data.frame
length() Give length of a vector
ls() Lists memory contents
rm() Removes an item from memory
names() Lists names of variables in a data.frame
hist() Command for producing a histogram
histogram() Lattice command for producing a histogram
stem() Make a stem plot
table() List all values of a variable with frequencies
xtabs() Cross-tabulation tables using formulas
mosaicplot() Make a mosaic plot
cut() Groups values of a variable into larger bins
mean(), median() Identify “center” of distribution
by() apply function to a column split by factors
summary() Display 5-number summary and mean
var(), sd() Find variance, sd of values in vector
sum() Add up all values in a vector
quantile() Find the position of a quantile in a dataset
plot() Produces a scatterplot
barplot() Produces a bar graph
barchart() Lattice command for producing bar graphs

boxplot() Produces a boxplot
bwplot() Lattice command for producing boxplots
xyplot() Lattice command for producing a scatterplot
lm() Determine the least-squares regression line
anova() Analysis of variance (can use on results of
predict() Obtain predicted values from linear model
nls() estimate parameters of a nonlinear model
residuals() gives (observed - predicted) for a model fit to data
sample() take a sample from a vector of data
replicate() repeat some process a set number of times
cumsum() produce running total of values for input vector
ecdf() builds empirical cumulative distribution function
dbinom(), etc. tools for binomial distributions
dpois(), etc. tools for Poisson distributions
pnorm(), etc. tools for normal distributions
qt(), etc. tools for student t distributions
pchisq(), etc. tools for chi-square distributions
binom.test() hypothesis test and confidence interval for 1 proportion
prop.test() inference for 1 proportion using normal approx.
chisq.test() carries out a chi-square test
fisher.test() Fisher test for contingency table
t.test() t test for inference on population mean
qqnorm(), qqline() tools for checking normality
addmargins() adds marginal sums to an existing table
prop.table() compute proportions from a contingency table
par() query and edit graphical settings
power.t.test() power calculations for 1- and 2-sample t
anova() compute analysis of variance table for fitted model

1.3 Random Numbers Generation

A sequence of random numbers R1, R2, …, must have two important statistical properties:
Uniformity
Independence.
Random Number, Ri, must be independently drawn from a uniform distribution
As we know, random numbers are described by a distribution.
That is, some function which specifies the probability that a random number is in some range.
For example $P(a < X \leq b)$. Often this is given by a probability density (in the continuous case) or by a function $P(X=k) = f(k)$ in the discrete case. R will give numbers drawn from lots of different distributions. In order to use them, you only need familiarize yourselves with the parameters that are given to the functions such as a mean, or a rate. Here are examples of the most common ones. For each, a histogram is given for a random sample of size 100, and density (using the ``d"
functions) is superimposed as appropriate.

Uniform

Uniform numbers are ones that are "equally likely" to be in the specified range. Often these numbers are in [0,1] for computers, but in practice can be between [a,b] where a,b depend upon the problem. An example might be the time you wait at a traffic light. This might be uniform on [0,2].

```
>runif(1,0,2) # time at light
   1.490857 # also runif(1,min=0,max=2)
>runif(5,0,2) # time at 5 lights
   0.07076444 0.01870595 0.50100158 0.61309213 0.77972391
>runif(5) # 5 random numbers in [0,1]
       0.1705696 0.8001335 0.9218580 0.1200221 0.1836119
```

The general form is runif(n,min=0,max=1) which allows you to decide how many uniform random numbers you want (n), and the range they are chosen from ([min,max])
To see the distribution with min=0 and max=1 (the default) we have

```
> x=runif(100) # get the random numbers
>hist(x,probability=TRUE,col=gray(.9),main="uniform on [0,1]")
>curve(runif(x,0,1),add=T)
```

Normal

Normal numbers are the backbone of classical statistical theory due to the central limit theorem. The normal distribution has two parameters a mean μ and a standard deviation s. These are the location and spread parameters. For example, IQs may be normally distributed with mean 100 and standard deviation 16, Human gestation may be normal with mean 280 and standard deviation about 10 (approximately). The family of normals can be standardized to normal with mean 0 (centered) and variance 1. This is achieved by "standardizing" the numbers, i.e. Z=(X-μ)/s.

Here are some examples

```
>rnorm(1,100,16) # an IQ score
   94.1719
>rnorm(1,mean=280,sd=10)
   270.4325 # how long for a baby (10 days early)
```

Here the function is called as rnorm(n,mean=0,sd=1) where one specifies the mean and the standard deviation.

```
> x=rnorm(100)
>hist(x,probability=TRUE,col=gray(.9),main="normal mu=0,sigma=1")
>curve(dnorm(x),add=T)
## also for IQs using rnorm(100,mean=100,sd=16)
```

Binomial

The binomial random numbers are discrete random numbers. They have the distribution of the number of successes in n independent Bernoulli trials where a Bernoulli trial results in success or failure, success with probability p.

A single Bernoulli trial is given with n=1 in the binomial

```
> n=1, p=.5 # set the probability
>rbinom(1,n,p) # different each time
  1
>rbinom(10,n,p) # 10 different such numbers
    0 1 1 0 1 0 1 0 1 0
```

A binomially distributed number is the same as the number of1's in n such Bernoulli numbers. For the last example, this would be There are then two parameters n (the number of Bernoulli trials) and p (the success probability). To generate binomial numbers, we simply change the value of n from 1 to the desired number of trials. For example, with 10 trials:

```
> n = 10; p=.5
>rbinom(1,n,p) # 6 successes in 10 trials
   6
>rbinom(5,n,p) # 5 binomial number
    6 6 4 5 4
```

The number of successes is of course discrete, but as n gets large, the number starts to look quite normal. This is a case of the central limit theorem which states in general that (X- μ)/s is normal in the limit (note this is standardized as above) and in our specific case that the graphs show 100 binomially distributed random numbers for 3 values of n and for p=.25. Notice in the graph, as n increases the shape becomes more and more bell-shaped. These graphs were made with the commands

```
> n=5;p=.25 # change as appropriate
> x=rbinom(100,n,p) # 100 random numbers
>hist(x,probability=TRUE,)
## use points, not curve as dbinom wants integers only for x
>xvals=0:n;points(xvals,dbinom(xvals,n,p),type="h",lwd=3)
>points(xvals,dbinom(xvals,n,p),type="p",lwd=3)
... repeat with n=15, n=50
```

Exponential

The exponential distribution is important for theoretical work. It is used to describe lifetimes of electrical components (to first order). For example, if the mean life of a light bulb is 2500 hours one may think its lifetime is random with exponential distribution having mean 2500. The one parameter is the rate = 1/mean. We specify it as follows rexp(n,rate=1). Here is an example with the rate being 1/2500.

```
> x=rexp(100,1/2500)
>hist(x,probability=TRUE,col=gray(.9),main="exponential ,mean=2500")
>curve(dexp(x,1/2500),add=T)
```

1.4 Data Types

In contrast to other programming languages like C and java in R, the variables are not declared as some data type. The variables are assigned with R-Objects and the data type of the R-object becomes the data type of the variable. There are many types of R-objects. The frequently used ones are

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frames

Vectors

When you want to create vector with more than one element, you should use c() function which means to combine the elements into a vector.

```
# Create a vector.
apple<- c('red','green',"yellow")
print(apple)
# Get the class of the vector.
print(class(apple))
```

Lists

A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

```
# Create a list.
list1 <- list(c(2,5,3),21.3,sin)
# Print the list.
print(list1)
```

Matrices

A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
# Create a matrix.
M = matrix( c('a','a','b','c','b','a'), nrow = 2, ncol = 3, byrow = TRUE)
print(M)
```

Arrays

While matrices are confined to two dimensions, arrays can be of any number of dimensions. The array function takes a dim attribute which creates the required number of dimension. In the below example we create an array with two elements which are 3x3 matrices each.

```
# Create an array.
a <- array(c('green','yellow'),dim = c(3,3,2))
print(a)
```

Factors

Factors are the r-objects which are created using a vector. It stores the vector along with the distinct values of the elements in the vector as labels. The labels are always character irrespective of whether it is numeric or character or Boolean etc. in the input vector. They are useful in statistical modeling. Factors are created using the factor() function. Then levels functions gives the count of levels.

```
# Create a vector.
apple_colors<- c('green','green','yellow','red','red','red','green')
# Create a factor object.
factor_apple<- factor(apple_colors)
```

```
# Print the factor.
print(factor_apple)
print(nlevels(factor_apple))
```

Data Frames:

Data frames are tabular data objects. Unlike a matrix in data frame each column can contain different modes of data. The first column can be numeric while the second column can be character and third column can be logical. It is a list of vectors of equal length.

Data Frames are created using the data.frame() function.

```
# Create the data frame.
BMI <- data.frame(
gender = c("Male", "Male","Female"),
height = c(152, 171.5, 165),
weight = c(81,93, 78),
Age = c(42,38,26)
)
print(BMI)
```

Objects

Objects are assigned values using <- , an arrow formed out of < and -. (An equal sign, =, can also be used.) For example, the following command assigns the value 5 to the object x.

```
x <- 5
```

After this assignment, the object x ‘contains’ the value 5. Another assignment to the same object will change the content.

```
x <- 107
```

we can check the content of an object by simply entering the name of the object on an interactive command line. Try that throughout these examples to see what the results are of the different operations and functions illustrated.

1.5 Basic data and Computations

R is case sensitive programming, it treats data as completely different objects. Statistics is the study of data. After learning how to start R, the first thing we need to be able to do is learn how to enter data into R and how to manipulate the data once there.

Example

```
help() #give help regarding a command, e.g. help(hist)
c() #concatenate objects,e.g.x = c(3,5,8,9)ory=
c(”Jack”,”Queen”,”King”)
1:19 #create a sequence of integers from 1 to 19
(…) #give arguments to a function, e.g. sum(x), or help(hist)
[…] #select elements from a vector or list, e.g. x[2] gives 5, x[c(2,4)]
gives 5 9 for x as above
matrix() #fill in (by row) the values from y in a matrix of 4 rows and 3
columns by giving #m = matrix(y,4,3,byrow=T)
dim() #gives the number of rows and the number of columns of a
matrix, or a data frame
head() #gives the first 6 rows of a large matrix, or data frame
tail() #gives the last 6 rows of a large matrix, or data frame
m[ ,3] #gives the 3rd column of the matrix m
m[2, ] #gives the 2nd row of the matrix m
= or <- #assign something to a variable, e.g. x = c(”a”,”b”,”b”,”e”)
== #ask whether two things are equal, e.g. x = c(3,5,6,3) and then
x == 3
< #ask whether x is smaller than y,
> #ask whether x is larger than y
& #logical „and‟
```

```
| #logical „or‟
sum() #get the sum of the values in x by sum(x)
mean() #get the mean of the values in x by mean(x)
median() #get the median of the values in x by median(x)
sd() #get the standard deviation of the values in x
var() #get the variance of the values in x
IQR() #get the IQR of the values in x
summary() #get the summary statistics of a single variable, or of all
variables in a data frame
round() #round values in x to 3 decimal places by round(x,3)
sort() #sort the values in x by giving sort(x)
unique() #get the non-duplicate values from a list,
e.g. x = c(3,5,7,2,3,5,9,3) and then
unique(x) #gives 3 5 7 2 9
length(x) #gives the length of the vector x, which is 8
hist() #create a histogram of the values in x by hist(x)
stem() #create a stem and leaf plot of the values in x by stem(x)
boxplot() #create a boxplot of the values in x by boxplot(x)
plot() #scatterplot of x vs. y by plot(x,y); for more parameters see
help(plot.default)
cor() #gives the linear correlation coefficient
lm() #fit a least squares regression of y (response) on x (predictor) by
fit = lm(y~x)
names() #get or set the names of elements in a R object. E.g. names(fit) will give the names of the R #object
named “fit”, or #get or set the names of variables in a data frame.
fit$coef #gives the least squares coefficients from the fit above, i.e. intercept and slope
fit$fitted #gives the fitted values for the regression fitted above
fit$residuals #gives the residuals for the regression fitted above
lines() #add a (regression) line to a plot by lines(x,fit$fitted)
abline() #add a straight line to a scatterplot
points() #add additional points (different plotting character) to a plot
scan() #read data for one variable from a text file,
e.g. y = scan(”ping.dat”)
read.table() #read spreadsheet data (i.e. more than one variable)
from a text file
table() #frequency counts of entries, ideally the entries
are factors
write() #write the values of a variable y in a file data.txt by
write(y,file=”data.txt”)
log() #natural logarithm (i.e. base e)
log10() #logarithm to base 10
seq() #create a sequence of integers from 2 to 11 by increment 3 with
seq(2,11,by=3)
rep() #repeat n times the value x, e.g. rep(2,5) gives 2 2 2 2 2
getwd() #get the current working directory.
setwd() #change the directory to.
E.g. setwd("c:/RESEARCH/GENE.project/Chunks/")
dir() #list files in the current working directory
search() #searching through reachable datasets and packages
library() #link to a downloaded R package to the current R session.
E.g. library(Biostrings) link to the
#R package #called “Biostrings” which you had downloaded earlier onto your laptop
```

1.6 Input and Display

```
load("c:/RData/pennstate1.RData") #load a R data frame
read.csv(filename="c:/stat251/ui.csv",header=T) #read .csv file with labels in first row
```

```
x=c(1,2,4,8,16) #create a data vector with specified elements
y=c(1:10) #create a data vector with elements 1-10
vect=c(x,y) #combine them into one vector of length 2n
mat=cbind(x,y) #combine them into a n x 2 matrix
mat[4,2] #display the 4th row and the 2nd column
mat[3,] #display the 3rd row
mat[,2] #display the 2nd column
```

Data Manipulation Examples

```
x.df=data.frame(x1,x2,x3 ...)
#combine different kinds of data into a data frame
scale() #converts a data frame to standardized scores
round(x,n) #rounds the values of x to n decimal places
ceiling(x) #vector x of smallest integers > x
floor(x) #vector x of largest integer< x
as.integer(x) #truncates real x to integers (compare to round(x,0)
as.integer(x <cutpoint)
#vector x of 0 if less than cutpoint, 1 if greater than cutpoint)
factor(ifelse(a <cutpoint, "Neg", "Pos"))
#is another way to dichotomize and to make a factor for analysis
transform(data.df,variable names = some operation)
#can be part of a set up for a data set
```

Statistical Tests

```
binom.test()
prop.test() #perform test with proportion(s)
t.test() #perform t test
chisq.test() #perform Chi-square test
pairwise.t.test()
power.anova.test()
power.t.test()
aov()
anova()
TukeyHSD()
kruskal.test()
```

Distributions

```
sample(x, size, replace = FALSE, prob = NULL) # take a simple random
sample of size n from the
# population x with or without replacement
rbinom(n,size,p)
pbinom()
qbinom()
dbinom()
rnorm(n,mean,sd) #randomly generate n numbers from a Normal
distribution with the specific mean and sd
pnorm() #find probability (area under curve) of a Normal(10,3^2)
distribution to the left
qnorm() #find quantity or value x such that area under
Normal(10,3^2)
```

1.7 Data Input

Unlike SAS, which has DATA and PROC steps, R has data structures (vectors, matrices, arrays, data frames) that you can operate on through functions that perform statistical analyses and create graphs. This section describes how to enter or import data into R, and how to prepare it for use in statistical analyses. Topics include R data structures, importing data (from Excel, SPSS, SAS, Stata, and ASCII Text Files), entering data from the keyboard, creating an interface with a database management system, exporting data (to Excel, SPSS, SAS, Stata, and Tab Delimited Text Files), annotating data (with variable labels and value labels), and listing data. In addition, methods for handling missing values and date values are presented.

1.8 Data Frames

A data frame is used for storing data tables. It is a list of vectors of equal length. For example, the following variable df is a data frame containing three vectors n, s, b.

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc")
> b = c(TRUE, FALSE, TRUE)
>df = data.frame(n, s, b) # df is a data frame
```

1.8 Graphics

This provides the most basic information to get started producing plots in R. This section provides an introduction to R graphics by way of a series of charts, graphs and visualization. R has also been used to produce figures that help to visualize important concepts or teaching points. The organization of R graphics this section briefly describes how R's graphics functions are organized so that the user knows where to start looking for a particular function. The R graphics system can be broken into four distinct levels: graphics packages; graphics systems; a graphics engine, including standard graphics devices; and graphics device packages

To visualize data:

- ggplot2 - R's famous package for making beautiful graphics.ggplot2 lets you use the grammar of graphics to build layered, customizable plots.
- ggvis - Interactive, web based graphics built with the grammar of graphics.
- rgl - Interactive 3D visualizations with R
- Colors : The package colorspace provides a set of functions for transforming between color spaces and mixcolor() for mixing colors within a color space.
- htmlwidgets - A fast way to build interactive (javascript based) visualizations with R. Packages that implement htmlwidgets include:
- leaflet (maps)
- dygraphs (time series)
- DT (tables)
- diagrammeR (diagrams)
- network3D (network graphs)
- threeJS (3D scatterplots and globes).

Graphics formats that R supports and the functions that open an appropriate R Programming language has numerous libraries to create charts and graphs.R provides the usual range of standard statistical plots, including scatterplots, boxplots, and histograms, bar plots, pie charts, and basic3Dplots

Types of charts

- scatterplots,
- boxplots
- histograms
- bar plots
- pie charts
- basic3Dplots

1.9 Table

A table is an arrangement of information in rows and columns that make comparing and contrasting information easier. As you can see in the following example, the data are much easier to read than they would be in a list containing thread.table() #read spreadsheet data (i.e. more than one variable) from a text file table() #frequency counts of entries, ideally the entries are factors(although#it works with integers or even reals)at same data.

Example

```
smoke <-matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=TRUE)
colnames(o) <-c("High","Low","Middle")
rownames(o) <-c("current","former","never")
smoke<-as.table(smoke)
```

smoke	High	Low	Middle
current	51	43	22
former	92	28	21
never	68	22	9

Unit II

Basics of Diagrammatic Presentation

Concept of Diagrammatic Presentation

- Diagrammatic presentation is a technique of presenting numeric data through Pictograms, Cartograms, Bar Diagrams & Pie Diagrams etc. It is the most attractive and appealing way to represent statistical data. Diagrams help in visual comparison and have a bird's eye view.
- Under Pictograms, we use pictures to present data. For example, if we have to show the production of cars, we can draw cars. Suppose, production of cars is 40,000. We can show it by a picture having four cars, where 1 Car represents 10,000 units.
- Under Cartograms, we make use of maps to show the geographical allocation of certain things.
- Bar Diagrams are rectangular in shape placed on the same base. Their height represents the magnitude/value of the variable. Width of all the bars and gap between the two bars is kept the same.
- Pie Diagram is a Circle which is sub-divided or partitioned to show the proportion of various components of the data.
- Out of the above, only One Dimensional Bar Diagrams and Pie Diagrams are in our scope.

General Guidelines

- **Title** – Every diagram must be given a suitable 'Title' which should be small and self-explanatory.
- **Size** – Size of the diagram should be appropriate neither too small nor too big.
- **Paper used** – Diagrams are generally prepared on blank paper.
- **Scale** – Under one-dimensional diagrams especially 'Bar Diagrams' generally Y-axis is more important from the point of view of the decision of scale because we represent magnitude along this axis.
- **Index** – When two or more variables are presented and different types of line/shading patterns are used to distinguish, then an index must be given to show their details.
- **Selection of Proper Type of Diagram** – It's very important to select the correct type of diagram to represent data effectively.

Advantages of Diagrammatic Presentation

(1) Diagrams Are Attractive and Impressive:

- Data presented in the form of diagrams are able to attract the attention of even a common man.

(2) Easy to Remember

- Diagrams have a great memorizing effect.
- The picture created in the mind by diagrams last much longer than those created by figures presented through the tabular form.

(3) Diagrams Save Time

- It presents complex mass data in a simplified manner.
- Data presented in the form of diagrams can be understood by the user very quickly.

(4) Diagrams Simplify Data

- Diagrams are used to represent a huge mass of complex data in a simplified and intelligible form, which is easy to understand.

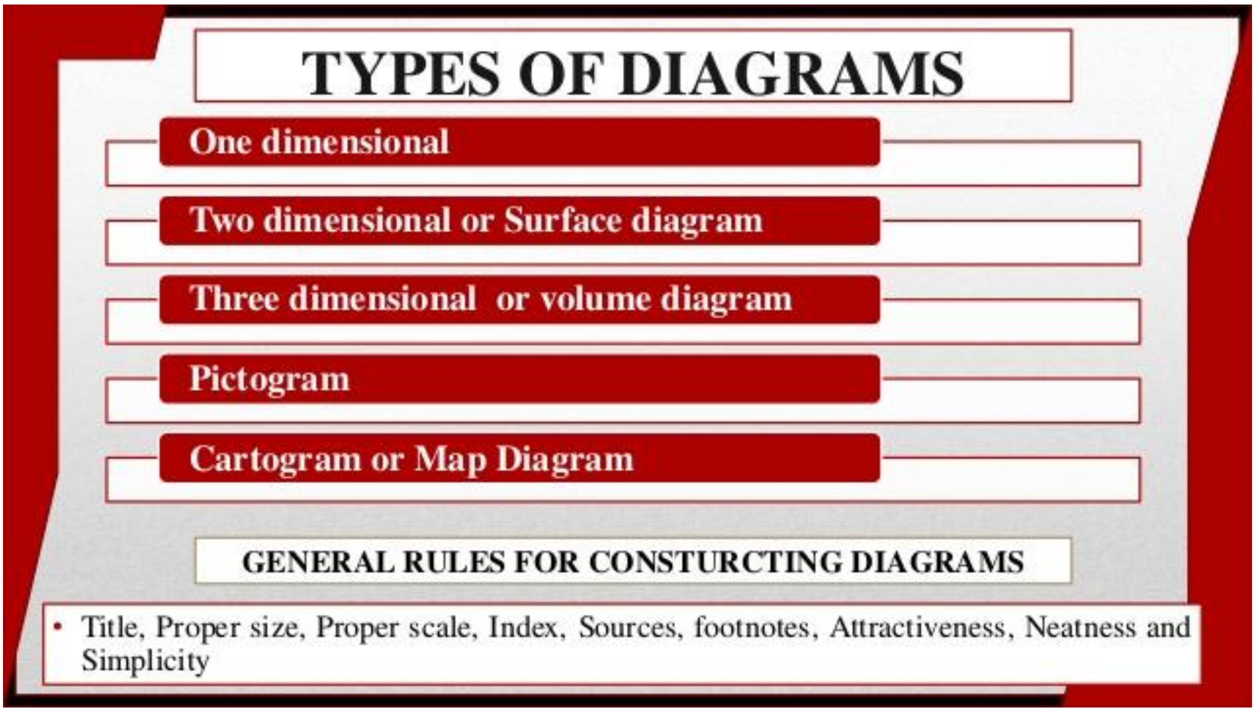
(5) Diagrams Are Useful in Making Comparisons

- It becomes easier to compare two sets of data visually by presenting them through diagrams.

(6) More Informative

- Diagrams not only depict the characteristics of data but also bring out other hidden facts and relations which are not possible from the classified and tabulated data.

Diagrammatic presentation is a technique of presenting numeric **data** through Pictograms, Cartograms, Bar Diagrams & Pie Diagrams etc. It is the most attractive and appealing way to represent statistical **data**. ... Under Pictograms, we use pictures to present **data**.



TYPES OF DIAGRAMS

- One dimensional
- Two dimensional or Surface diagram
- Three dimensional or volume diagram
- Pictogram
- Cartogram or Map Diagram

GENERAL RULES FOR CONSTURCTING DIAGRAMS

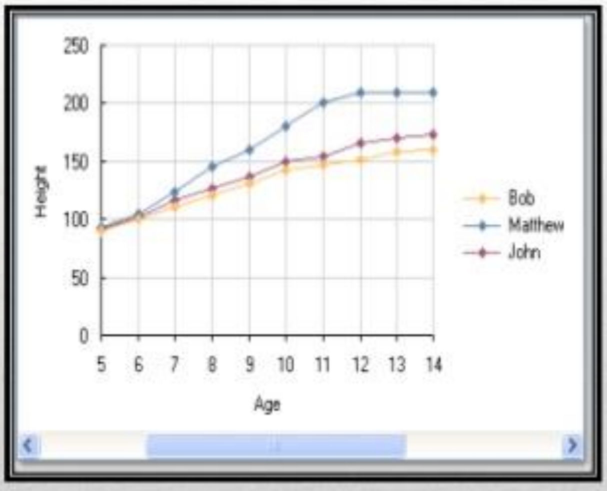
- Title, Proper size, Proper scale, Index, Sources, footnotes, Attractiveness, Neatness and Simplicity

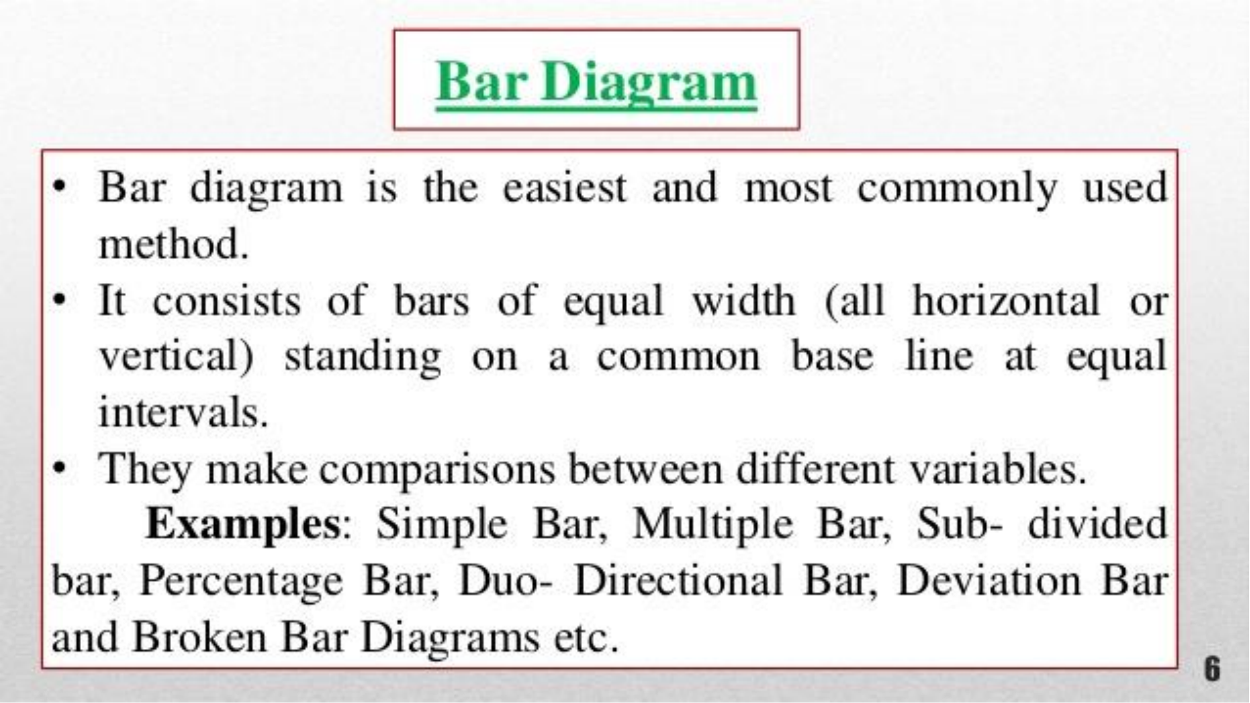
1. One Dimensional Diagram

- One dimensional diagram are such diagrams where only one dimensional measurement i.e. height is used.
- These diagram may be in the form of lines or bars.
- There is no importance of width or thickness in these diagrams.
- The heights of these lines or bars are taken on the basis of values.

Line Diagram

- In these diagrams, only line is drawn to represent one variable. These lines may be vertical or horizontal.
- Line diagram is used in case where there are many items and there is least difference between different value.
- The construction of this diagram is very simple. It makes comparison easy.
- It has no width and hence of very poor visual effect, so it is less attractive.





Bar Diagram

- Bar diagram is the easiest and most commonly used method.
- It consists of bars of equal width (all horizontal or vertical) standing on a common base line at equal intervals.
- They make comparisons between different variables.

Examples: Simple Bar, Multiple Bar, Sub- divided bar, Percentage Bar, Duo- Directional Bar, Deviation Bar and Broken Bar Diagrams etc.

6

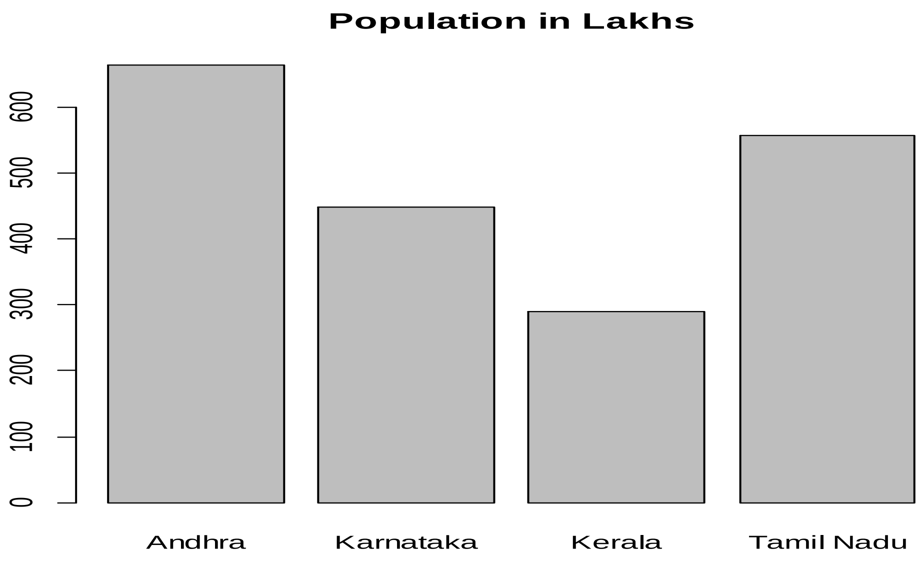
Simple Bar Diagram

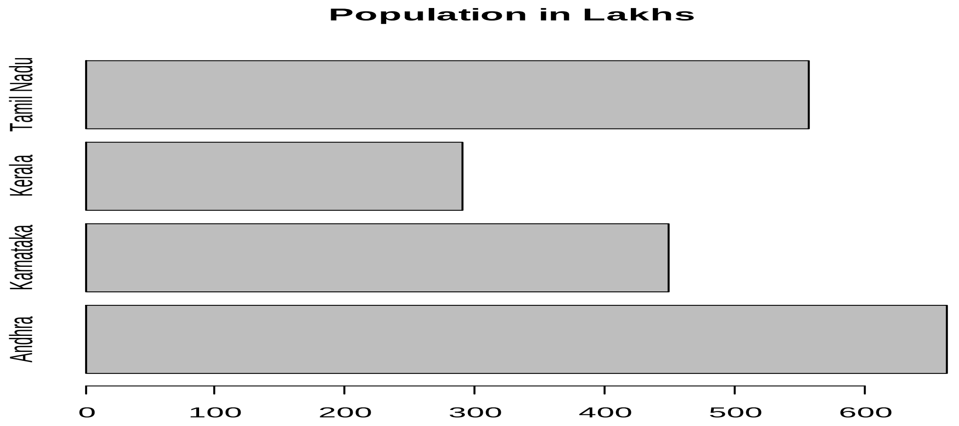
A **simple bar chart** is used to represent data involving only one variable classified on a spatial, quantitative or temporal basis. In a **simple bar chart**, we make **bars** of equal width but variable length, i.e. the magnitude of a quantity is represented by the height or length of the **bars**. Simple bar diagram is used for comparative study of two or more items or value of a single variable. These can also be drawn either vertically or horizontally. Distance between these bars should be equal.

Simple Bar Diagram

R-coding

```
> population<-c(663,448,290,556)

> state<-c("Andhra","Karnataka","Kerala","Tamil Nadu")

> barplot(population,names.arg=state,main="Population in Lakhs",horiz=TRUE)

> barplot(population,names.arg=state,main="Population in Lakhs",vertical=TRUE)
```





Construction of Multiple Bar Diagram

Multiple Bar Graphs

Sometimes there are more than two sets of data to be compared in a **bar graph**. In that case, a **multiple bar graph** can be used. A multiple bar graph compares as many sets of data you want. The process for creating a multiple bar graph is just like creating any other bar graph, only you will have more colors to represent different sets of data.

To create a multiple bar graph:

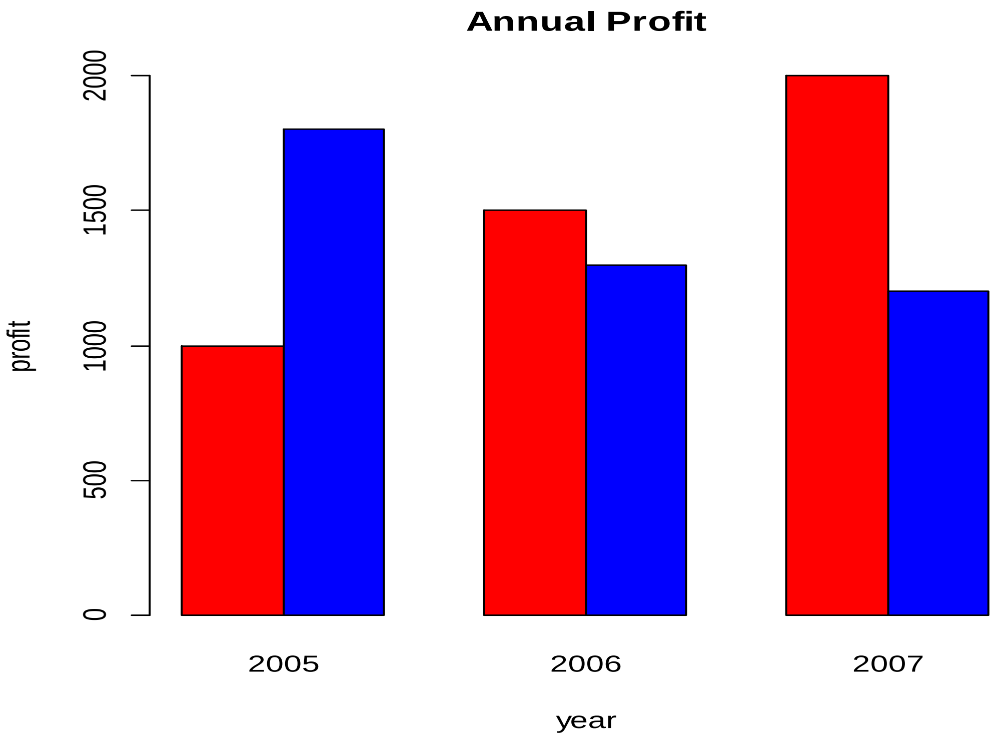
1. Draw the horizontal (x) and vertical (y) axis.
2. Give the graph a title.
3. Label the horizontal x axis.
4. Label the vertical y axis.
5. Look at the range in data and decide how the units on the vertical axis (y) should be labeled.

6. For each item on the horizontal (x) axis, draw a vertical column to the appropriate value however many times as you have sets of data. For example, if you are looking at 3 days worth of data, you will have 3 bars per item.
7. Choose three colors to represent each different data set. Make sure to label the data sets in a key alongside the graph.

Sometimes comparing data can also be done by comparing data sets across multiple different bar graphs. The difference is the data is split versus all being compared in one graph. Either method allows you to analyze and compare the data being displayed.

R-coding

```
> year<-c("2005","2006","2007")

> color<-c("red","blue")

> profit=matrix(c(1000,1500,2000,1800,1300,1200),nrow=2,ncol=3,byrow=T)

>barplot(profit,names.arg=year,xlab="year",ylab="profit",col=color,main="Annual Profit",beside=T)
```

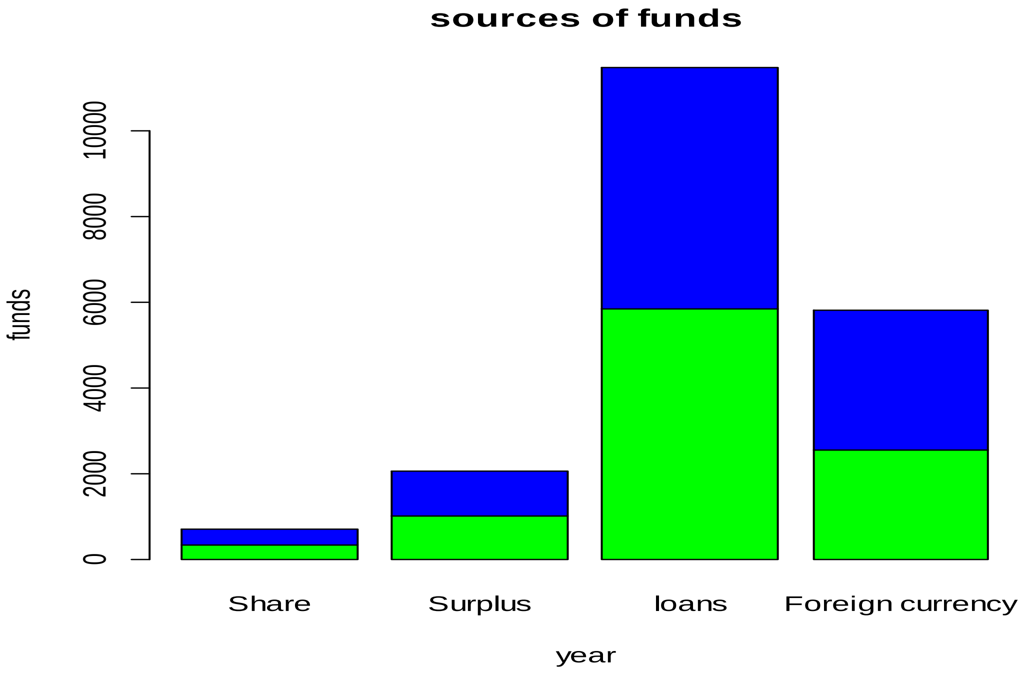


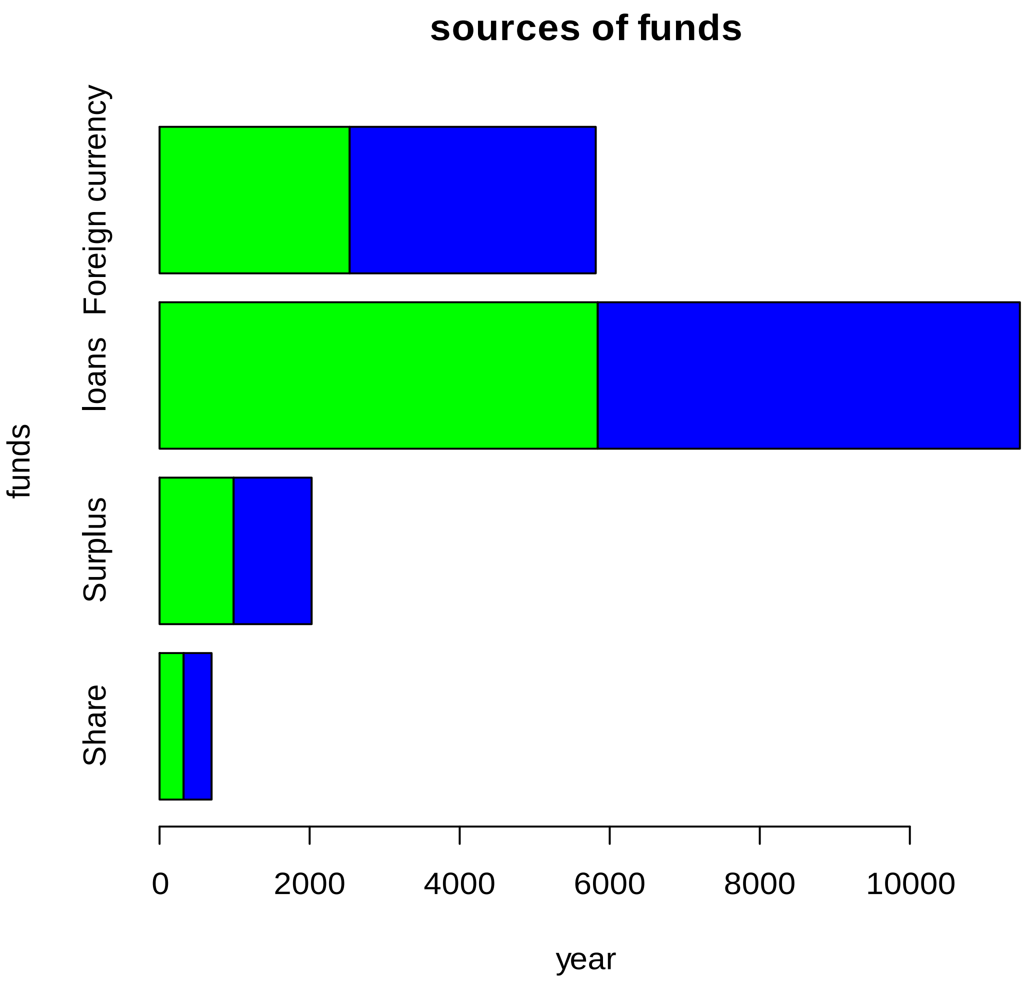
Construction of Sub divided Bar Diagram

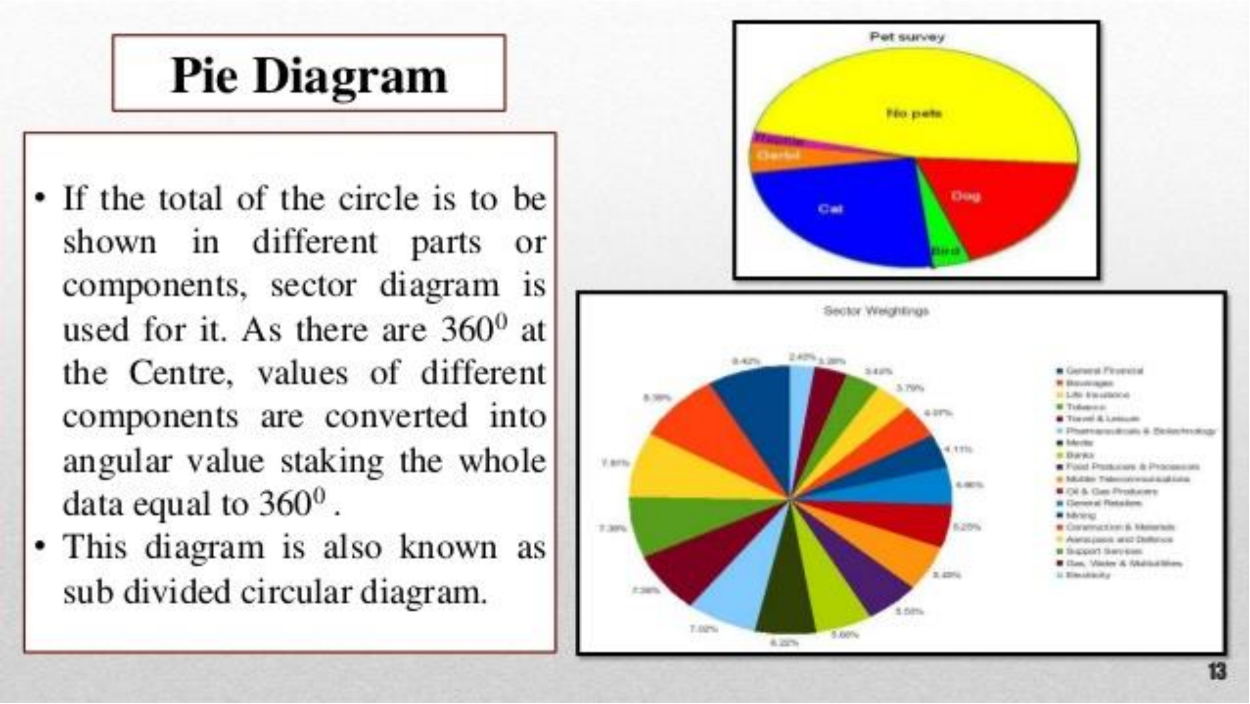
A **sub-divided** or component **bar chart** is used to represent data in which the total magnitude is **divided** into different or components. In this **diagram**, first we make simple **bars** for each class taking the total magnitude in that class and then **divide** these simple **bars** into parts in the ratio of various components.

R-code

```
> funds<-c("Share","Surplus","loans","Foreign currency")

> colors<-c("green","blue")

> values<-matrix(c(339,998,5843,2552,352,1043,5614,3262),nrow=2,ncol=4,byrow=TRUE)

> barplot(values,names.arg=funds,xlab="year",ylab="funds",main="sources of funds",col=colors)

> barplot(values,names.arg=funds,xlab="year",ylab="funds",main="sources of funds",col=colors)

> barplot(values,names.arg=funds,xlab="year",ylab="funds",main="sources of funds",col=colors,horiz=TRUE)
```







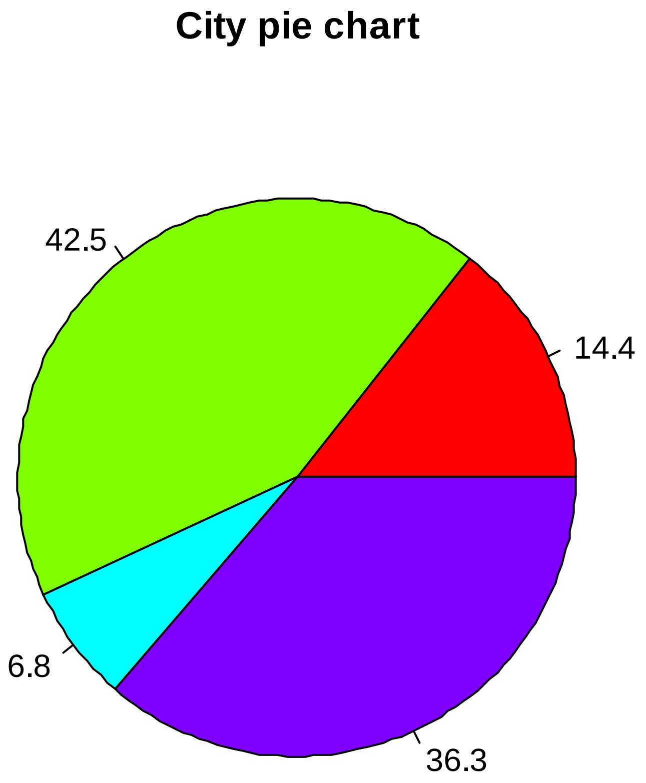
Pie Diagram

- If the total of the circle is to be shown in different parts or components, sector diagram is used for it. As there are 360^0 at the Centre, values of different components are converted into angular value staking the whole data equal to 360^0 .
- This diagram is also known as sub divided circular diagram.

Construction Pie Diagram

R code

```
> x <-  c(21, 62, 10,53)
> labels <-  c("London","New York","Singapore","Mumbai")
> piepercent<- round(100*x/sum(x), 1)
> pie(x, labels = piepercent, main = "City pie chart",col = rainbow(length(x)))
```



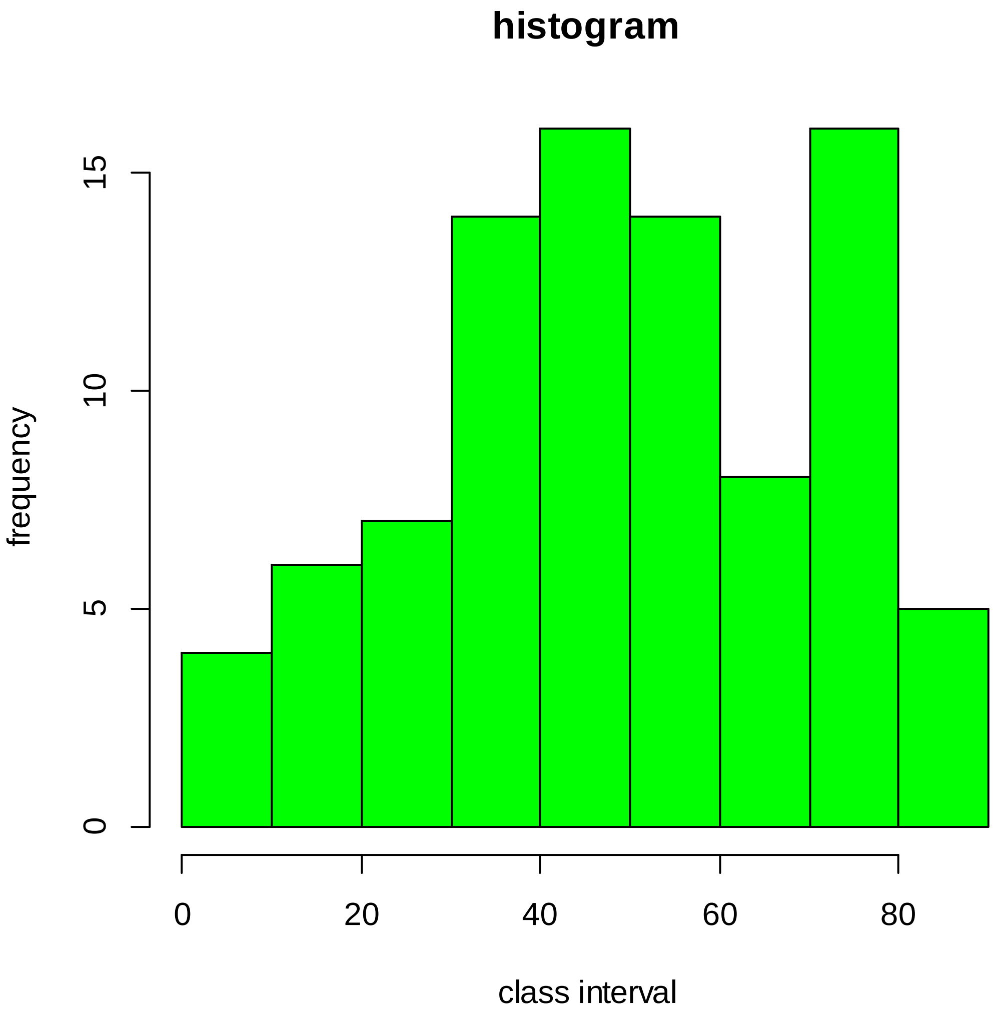
What Is a Histogram?

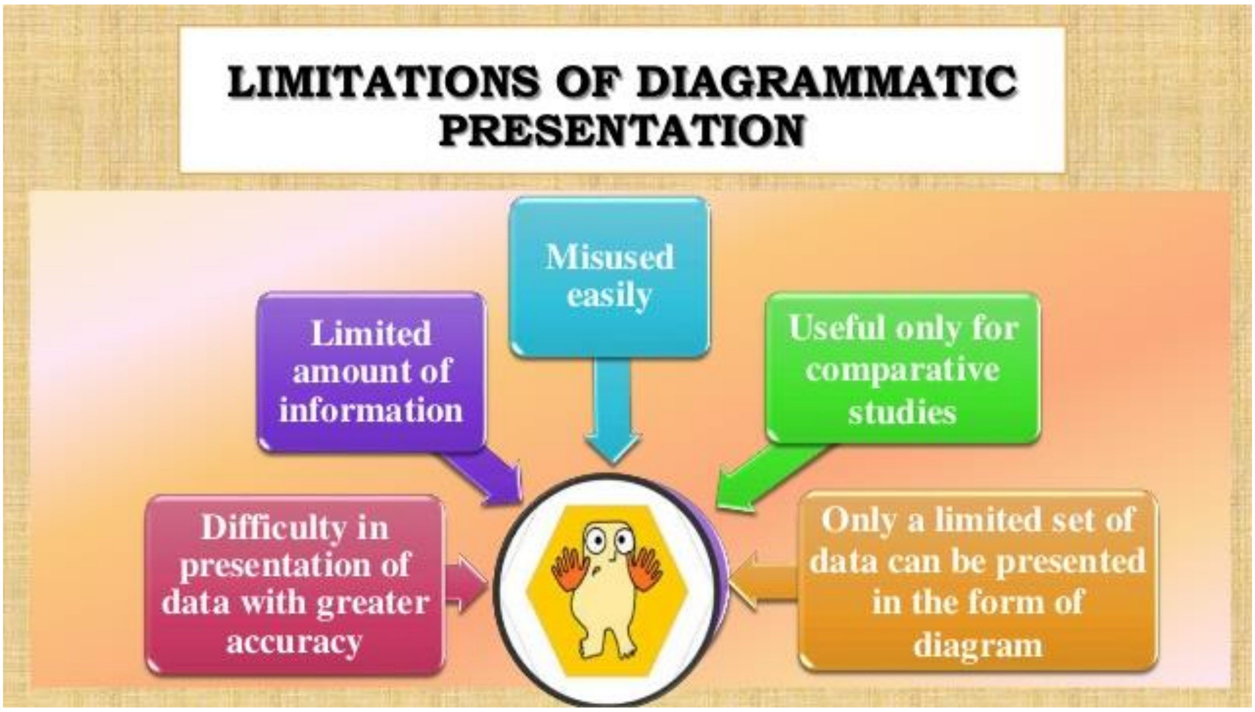
A histogram is a graphical representation that organizes a group of data points into user-specified ranges. It is similar in appearance to a bar graph. The histogram condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges or bins.

Construction of Histogram

R-code

```
> x<-c(5,15,25,35,45,55,65,75,85)
> f<-c(4,6,7,14,16,14,8,16,5)
> a<-rep(x,f)
> brk=seq(0,90,by=10)
> hist(a,brk,xlab="class       interval",ylab="frequency",col="green",main="histogram")
```


histogram
frequency
0
5
10
15
0
20
40
60
80
class interval



LIMITATIONS OF DIAGRAMMATIC PRESENTATION

Misused easily

Limited amount of information

Useful only for comparative studies

Difficulty in presentation of data with greater accuracy

Only a limited set of data can be presented in the form of diagram

Computation measures of Central Values

A measure of central tendency (also referred to as measures of centre or central location) is a summary measure that attempts to describe a whole set of data with a single value that represents the middle or centre of its distribution.

There are three main measures of central tendency:

The mode

The median

The mean.

Each of these measures describes a different indication of the typical or central value in the distribution.

Measures of Central Tendency

Arithmetic mean

Mean:

The mean is the sum of the value of each observation in a dataset divided by the number of observations. This is also known as the arithmetic average.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Looking at the retirement age distribution again:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

The mean is calculated by adding together all the values (54+54+54+55+56+57+57+58+58+60+60 = 623) and dividing by the number of observations (11) which equals 56.6 years.

Advantage of the mean:

The mean can be used for both continuous and discrete numeric data.

Limitations of the mean:

The mean cannot be calculated for categorical data, as the values cannot be summed. As the mean includes every value in the distribution the mean is influenced by outliers and skewed distributions.

R-code

```
> Family<-c("A","B","C","D","E","F","G","H","I","J")
> Expenditure<-c(30,70,10,75,500,8,42,250,40,36)
> mean(Expenditure)
output
mean= 106.1
```

R-code

```
> persons<-c(2,3,4,5,6)
> house<-c(10,25,30,25,10)
> fx=sum(persons*house)
> fx
[1] 400
> f=sum(house)
> f
[1] 100
> fxx=(fx/f)
> fxx
```

```
Output
Mean=  4
```

Harmonic mean

R-code

```
> har<-c(6,15,35,40,900,520,300,400,1800,2000)
> aa=(1/har)
> aa
 [1] 0.1666666667 0.0666666667 0.0285714286 0.0250000000 0.0011111111
 [6] 0.0019230769 0.0033333333 0.0025000000 0.0005555556 0.0005000000
```

```
> stt=data.frame(har,st)
> stt
    har       X_data
1     6 0.1666666667
2    15 0.0666666667
3    35 0.0285714286
4    40 0.0250000000
5   900 0.0011111111
6   520 0.0019230769
7   300 0.0033333333
```

```
8   400 0.0025000000
9  1800 0.0005555556
10 2000 0.0005000000
> n=length(har)
> n
[1] 10
> sttt=sum(st)
> sttt
[1] 0.2968278
> haa=(n/sttt)
> haa
```

output

```
[1] 33.68956
```

Geometric mean

In **statistics**, the **geometric mean** is calculated by raising the product of a series of numbers to the inverse of the total length of the series. The **geometric mean** is most useful when numbers in the series are not independent of each other or if numbers tend to make large fluctuations.

```
a = c(10, 2, 19, 24, 6, 23, 47, 24, 54, 77)

n = length(a) #now n is equal to the number of elements in a

prod(a)^(1/n) #compute the geometric mean
```

```
    geoMean<-function(values){
prod(values)^(1/length(values))
}
values<-c(2,4,6,8)
geoMean(values)
```

Harmonic Mean

Harmonic mean is a type of average that is calculated by dividing the number of values in a data series by the sum of the reciprocals (1/x_i) of each value in the data series. The **harmonic mean** is often used to calculate the average of the ratios or rates.

```
a = c(10, 2, 19, 24, 6, 23, 47, 24, 54, 77)

1/mean(1/a) #compute the harmonic mean
```

Mode

The mode is the most commonly occurring value in a distribution. Consider this dataset showing the retirement age of 11 people, in whole years.

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

This table shows a simple frequency distribution of the retirement age data.

Age Frequency

54	3
55	1
56	1
57	2
58	2
60	2

The most commonly occurring value is 54, therefore the mode of this distribution is 54 years.

Using R- code

Mode

```
Create the function.
getmode <- function(v) {
   ss <- unique(v)
   ss[which.max(tabulate(match(v, ss)))]
}

# Create the vector with numbers.
v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)

# Calculate the mode using the user function.
result <- getmode(v)
print(result)
```

Advantage of the mode:

The mode has an advantage over the median and the mean as it can be found for both numerical and categorical (non-numerical) data.

Limitations of the mode:

The are some limitations to using the mode. In some distributions, the mode may not reflect the centre of the distribution very well. When the distribution of retirement age is ordered from lowest to highest value, it is easy to see that the centre of the distribution is 57 years, but the mode is
lower, at 54 years.

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

It is also possible for there to be more than one mode for the same distribution of data, (bi-modal, or multi-modal). The presence of more than one mode can limit the ability of the mode in describing the centre or
125
typical value of the distribution because a single value to describe the centre cannot be identified.

In some cases, particularly where the data are continuous, the distribution may have no mode at all (i.e. if all values are different). In cases such as these, it may be better to consider using the median or mean, or group the data in to appropriate intervals, and find the modal class.

Median

The median is the middle value in distribution when the values are arranged in ascending or descending order. The median divides the distribution in half (there are 50% of observations on either side of the median value). In a distribution with an odd number of observations, the median value is the middle value. Looking at the retirement age distribution (which has 11 observations), the median is the middle value, which is 57 years:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

When the distribution has an even number of observations, the median value is the mean of the two middle values. In the following distribution, the two middle values are 56 and 57, therefore the median equals 56.5 years:

52, 54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

Advantage of the median:

The median is less affected by outliers and skewed data than the mean, and is usually the preferred measure of central tendency when the distribution is not symmetrical.

Limitation of the median:

The median cannot be identified for categorical nominal data, as it cannot be logically ordered.

Measures of Dispersion

- **The measure of dispersion shows how the data is spread or scattered around the mean.**

Such as range, variance, standard deviation, and coefficient of variation—can be calculated with standard functions in the native stats package. In addition, a function, here called summary.list, can be defined to output whichever statistics are of interest.

Range

- Simplest measure of dispersion
- Difference between the largest and the smallest values:

$$\text{Range} = X_{\text{largest}} - X_{\text{smallest}}$$

Standard Deviation

- Most commonly used measure of variation
- Shows variation about the mean
- Is the **square root of the variance**
- Has the same units as the original data
- Steps for Calculating Standard Deviation
- 1. Calculate the difference between each value and the mean.
- 2. Square each difference.
- 3. Add the squared differences.
- 4. Divide this total by n-1 to get the sample variance.
- 5. Take the square root of the sample variance to get the sample standard deviation.

- Sample standard deviation:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}}$$

Example

Sample Data (X_i) : 10 12 14 15 17 18 18 24

n=8 mean = $\overline{X}$ =16

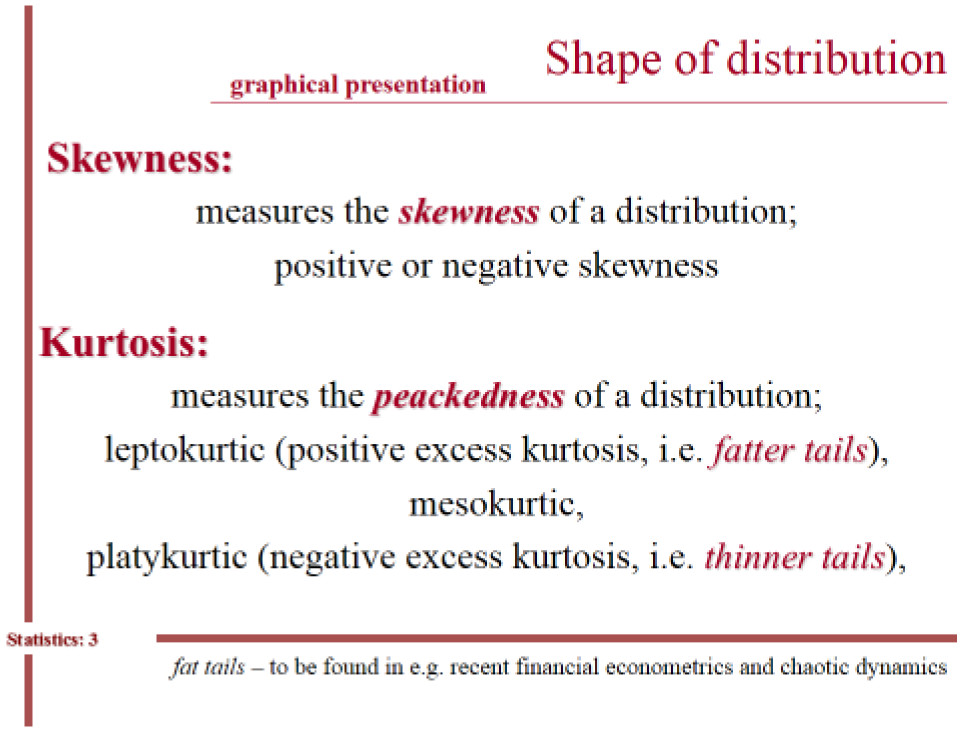
$$S=\sqrt{\frac{(10-\overline{X})^2+(12-\overline{X})^2+(14-\overline{X})^2+\cdots+(24-\overline{X})^2}{n-1}}$$

$$=\sqrt{\frac{(10-16)^2+(12-16)^2+(14-16)^2+\cdots+(24-16)^2}{8-1}}$$

$$=\sqrt{\frac{130}{7}}\quad=\quad 4.3095$$

Skewness
measures the ***skewness*** of a distribution;
positive or negative skewness

Kurtosis



Shape of distribution

graphical presentation

Skewness:

measures the ***skewness*** of a distribution;
positive or negative skewness

Kurtosis:

measures the ***peackedness*** of a distribution;
leptokurtic (positive excess kurtosis, i.e. *fatter tails*),
mesokurtic,
platykurtic (negative excess kurtosis, i.e. *thinner tails*),

Statistics: 3

fat tails – to be found in e.g. recent financial econometrics and chaotic dynamics

a bit of history

Skewness

Relationship between location measures:

mean – mode = 3(mean – median)

Coefficient of skewness:

independent of measurment units

$$s_k = \frac{\bar{x} - x^M}{\sigma}$$

Combining both:

$$s_k = \frac{3(\bar{x} - m)}{\sigma}$$

We will be using it

Karl Pearson (1857-1938)

x^M – mode, a value that occurs most frequently in the sample or population

formulas

Skewness

Skweness:

$$s_k = \frac{\sum_{i=1}^{T}(x_i - \bar{x})^3}{\sigma^3}$$

sum of deviation from mean value devided by the cubed standard deviation

***Excel* formula:**

$$s_k = \frac{T}{(T-1)(T-2)} \frac{\sum_{i=1}^{T}(x_i - \bar{x})^3}{\sigma^3}$$

adjusted Fisher-Pearson standardised *moment coefficient*

compare both formulas

Kurtosis

formulas

Kurtosis:

$$k = \frac{\sum_{i=1}^{T}\left(x_i - \bar{x}\right)^4}{\sigma^4}$$

sum of deviation from mean value divided by the standard deviation to the 4th power

Excel formula:

$$k = \frac{T(T+1)}{(T-1)(T-2)(T-3)} \frac{\sum_{i=1}^{T}\left(x_i - \bar{x}\right)^4}{\sigma^4} - \frac{3(T-1)^2}{(T-2)(T-3)}$$

population excess kurtosis in comparison to the normal distribution (bell-shaped distribution)

Kurtosis

interpretation

Positive and large:

leptokurtic distribution

(high-frequency financial data, abnormal rate or meturs, long time-series covering periods of crisises and expansions)

Negative and large:

platykurtic distribution

(large variability)

mesokurtik zero-excess kurtosis

Discrete Distributions

In this chapter we introduce discrete random variables, those who take values in a finite or countably infinite support set. We discuss probability mass functions and some special ex- pectations, namely, the mean, variance and standard deviation. Some of the more important discrete distributions are explored in detail, and the more general concept of expectation is defined, which paves the way for moment generating functions.

3.1Discrete Random Variables

3.1.1 Probability Mass Functions

Discrete random variables are characterized by their supports which take the form

$$S_X = \{u_1, u_2, \ldots, u_k\} \text{ or } S_X = \{u_1, u_2, u_3 \ldots\}. \qquad (3.1.1)$$

Probability Mass Function(PMF)

Every discrete random variable X has associated with it a probability mass function (PMF) $f_X : S_X \rightarrow [0, 1]$ defined by

$$f_X(x) = \mathrm{IP}(X = x), \quad x \in S_X. \qquad (3.1.2)$$

Since values of the PMF represent probabilities, we know from Chapter 4 that PMFs enjoy certain properties. In particular, all PMFs satisfy

1. $f_X(x) > 0$ for $x \in S$,
2. $f_X(x) = 1$, and
3. $\mathrm{IP}(X \in A) = \sum f_X(x)$, for any event $A \subset S$.

Example 3.1. Toss a coin 3 times. The sample space would be

$$S = \{HHH, HTH, THH, TTH, HHT, HTT, THT, TTT\} .$$

Now let X be the number of Heads observed. Then X has support $S_X = \{0, 1, 2, 3\}$. Assuming that the coin is fair and was tossed in exactly the same way each time, it is not unreasonable to suppose that the outcomes in the sample space are all equally likely. What is the PMF of

X? Notice that X is zero exactly when the outcome TTT occurs, and this event has probability 1/8. Therefore, $f_X(0) = 1/8$, and the same reasoning shows that $f_X(3) = 1/8$. Exactly three outcomes result in $X = 1$, thus, $f_X(1) = 3/8$ and $f_X(3)$ holds the remaining 3/8 probability (the total is 1). We can represent the PMF with a table:

$x \in S_X$	0	1	2	3	Total
$f_X(x) = \mathrm{IP}(X = x)$	1/8	3/8	3/8	1/8	1

3.1.1 Mean, Variance, and Standard Deviation

There are numbers associated with PMFs. One important example is the mean μ,

$$\mu = \mathrm{IE}\, X = \sum_{x\in S} x\, f_X(x), \qquad (3.1.3)$$

provided the (potentially infinite) series $\sum |x|\, f_X(x)$ is convergent. Another important number is the variance:

$$\sigma^2 = \mathrm{IE}(X-\mu)^2 = \sum_{x\in S}(x-\mu)^2, \qquad (3.1.4)$$

which can be computed with the alternate formula $\sigma^2 = \mathrm{IE}\, X^2 - (\mathrm{IE}\, X)^2$.

Directly defined from the variance is the standard deviation $\sigma = \sqrt{\sigma^2}$

Example 3.2. We will calculate the mean of X in Example 3.1.

$$\mu=\Sigma f_X(x) = 3.5$$

We interpret μ = 3.5 by reasoning that if we were to repeat the random experiment many times, independently each time, observe many corresponding outcomes of the random variable X, and take the sample mean of the observations, then the calculated value would fall close to 3.5. The

Remark 3.3. Note that although we say X is 3.5 on the average, we must keep in mind that our X never actually equals 3.5 (in fact, it is impossible for X to equal 3.5).

Related to the probability mass function $f_X(x) = \mathrm{IP}(X = x)$ is another important function called the cumulative distribution function (CDF), F_X. It is defined by the formula

$$F_X(t) = \mathrm{IP}(X \leq t), \quad -\infty < t < \infty. \qquad (5.1.5)$$

We know that all PMFs satisfy certain properties, and a similar statement may be made for CDFs. In particular, any CDF F_X satisfies

- F_X is nondecreasing ($t_1 \leq t_2$ implies $F_X(t_1) \leq F_X(t_2)$).
- F_X is right-continuous ($\lim_{t \to a^+} F_X(t) = F_X(a)$ for all $a \in \mathrm{R}$).
- $\lim_{t \to -\infty} F_X(t) = 0$ and $\lim_{t \to \infty} F_X(t) = 1$.

We say that X has the distribution F_X and we write $X \sim F_X$. In an abuse of notation we will also write $X \sim f_X$ and for the named distributions the PMF or CDF will be identified by the family name instead of the defining formula.

3.1.2 How to do it with R

The mean and variance of a discrete random variable is easy to compute at the console. Let's return to Example 3.2. We will start by defining a vector x containing the support of X, and a vector **f** to contain the values of f_X at the respective outcomes in x:

```
> x <- c(0,1,2,3)
> f <- c(1/8, 3/8, 3/8, 1/8)
```

To calculate the mean μ, we need to multiply the corresponding values of x and **f** and add them. This is easily accomplished in R since operations on vectors are performed *element-wise*

```
> mu <- su*m(x      f)
> mu

   1.5
```

To compute the variance σ^2, we subtract the value of mu from each entry in x, square the answers, multiply by f, and sum. The standard deviation σ is simply the square root of σ^2.

```
> sigma2 <- sum((x-mu*)^2      f)
> sigma2

  0.75

> sigma <- sqrt(sigma2)
> sigma

    0.8660254
```

Finally, we may find the values of the CDF F_X on the support by accumulating the proba- bilities in f_X with the cumsum function.

```
> F = cumsum(f)
> F
[1] 0.125 0.500 0.875 1.000
```

As easy as this is, it is even easier to do with the distrEx package [74]. We define a random variable X as an object, then compute things from the object such as mean, variance, and standard deviation with the functions E, var, and sd:

```
> library(distrEx)
> X <- DiscreteDistribution(supp = 0:3, prob = c(1,3,3,1)/8)
> E(X); var(X); sd(X)

[1] 1.5
[1] 0.75
[1] 0.8660254
```

Distributions In The Stats Package

Density, cumulative distribution function, quantile function and random variate generation for many standard probability distributions are available in the stats package.

Keywords

distribution

Details

The functions for the density/mass function, cumulative distribution function, quantile function and random variate generation are named in the form `dxxx`, `pxxx`, `qxxx` and `rxxx` respectively.

For the beta distribution see `dbeta`.

For the binomial (including Bernoulli) distribution see `dbinom`.

For the Cauchy distribution see `dcauchy`.

For the chi-squared distribution see `dchisq`.

For the exponential distribution see `dexp`.

For the F distribution see `df`.

For the gamma distribution see `dgamma`.

For the geometric distribution see `dgeom`. (This is also a special case of the negative binomial.)

For the hypergeometric distribution see `dhyper`.

For the log-normal distribution see `dlnorm`.

For the multinomial distribution see `dmultinom`.

For the negative binomial distribution see `dnbinom`.

For the normal distribution see `dnorm`.

For the Poisson distribution see `dpois`.

For the Student's t distribution see `dt`.

For the uniform distribution see `dunif`.

For the Weibull distribution see `dweibull`.

The Bernoulli Distribution

Density, distribution function, quantile function and random generation for the Bernoulli distribution with parameter `prob`

Usage

```
dbern(x, prob, log = FALSE)
pbern(q, prob, lower.tail = TRUE, log.p = FALSE)
qbern(p, prob, lower.tail = TRUE, log.p = FALSE)
rbern(n, prob)
```

Arguments

x, q vector of quantiles.
P vector of probabilities.
N number of observations. If `length(n) > 1`, the length is taken to be the number required.
Prob probability of success on each trial.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X<=x],="" otherwise,="" p[x="">x].

Details

The Bernoulli distribution with `prob` =p has probability mass function

$$P(x) = p^x (1-p)^{1-x} \qquad , \textit{for } x = 0 \textit{ or } 1$$

If an element of `x` is not `0` or `1`, the result of `dbern` is zero, without a warning. p(x) is computed using Loader's algorithm, see the reference below.

The quantile is defined as the smallest value x such that F(x)≥p, where F is the distribution function.

Value

`dbern` gives the density, `pbern` gives the distribution function, `qbern` gives the quantile function and `rbern` generates random deviates.

The Binomial Distribution

Density, distribution function, quantile function and random generation for the binomial distribution with parameters `size` and `prob`.

This is conventionally interpreted as the number of ‘successes’ in `size` trials.

Usage

```
dbinom(x, size, prob, log = FALSE)
pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
```

Arguments

x, q vector of quantiles.
p vector of probabilities.
n number of observations. If `length(n) > 1`, the length is taken to be the number required.
Size number of trials (zero or more).
Prob probability of success on each trial.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X≤x], otherwise, P[X>x].

Details

The binomial distribution with `size` =n and `prob` =p has density

$$p(x) = \binom{n}{x} p^{x} q^{n-x} \qquad ,x=0,1,2,\ldots,n$$

If an element of `x` is not integer, the result of `dbinom` is zero, with a warning.

p(x) is computed using Loader's algorithm, see the reference below.

The quantile is defined as the smallest value x such that F(x)≥p, where F is the distribution function.

Value

`dbinom` gives the density, `pbinom` gives the distribution function, `qbinom` gives the quantile function and `rbinom` generates random deviates.

If `size` is not an integer, `NaN` is returned.

The length of the result is determined by `n` for `rbinom`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

The Poisson Distribution

Density, distribution function, quantile function and random generation for the Poisson distribution with parameter `lambda`.

Usage

```
dpois(x, lambda, log = FALSE)
ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)
rpois(n, lambda)
```

Arguments

x vector of (non-negative integer) quantiles.
q vector of quantiles.
p vector of probabilities.
n number of random values to return.
Lambda vector of (non-negative) means.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X≤x], otherwise, P[X>x].

Details

The Poisson distribution has density $p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$ for x=0,1,2,… . The mean and variance are E(X)=Var(X)=λ.

Note that λ=0 is really a limit case (setting 0^0=1) resulting in a point mass at 0, see also the example.

If an element of `x` is not integer, the result of `dpois` is zero, with a warning. p(x) is computed using Loader's algorithm, see the reference in `dbinom`.

The quantile is right continuous: `qpois(p, lambda)` is the smallest integer x such that P(X≤x)≥p.

Setting `lower.tail = FALSE` allows to get much more precise results when the default, `lower.tail = TRUE` would return 1, see the example below.

Value

`dpois` gives the (log) density, `ppois` gives the (log) distribution function, `qpois` gives the quantile function, and `rpois` generates random deviates.

Invalid `lambda` will result in return value `NaN`, with a warning.

The Geometric Distribution

Density, distribution function, quantile function and random generation for the geometric distribution with parameter `prob`

Usage

```
dgeom(x, prob, log = FALSE)
pgeom(q, prob, lower.tail = TRUE, log.p = FALSE)
```

```
qgeom(p, prob, lower.tail = TRUE, log.p = FALSE)
rgeom(n, prob)
```

Arguments

x, q vector of quantiles representing the number of failures in a sequence of Bernoulli trials before success occurs.

P vector of probabilities.

n number of observations. If `length(n) > 1`, the length is taken to be the number required.

Prob probability of success in each trial. `0 < prob <= 1`.

log, log.p logical; if TRUE, probabilities p are given as log(p).

lower.tail logical; if TRUE (default), probabilities are P[X≤x], otherwise, P[X>x].

Details

The geometric distribution with `prob` =p has density $p(x)=p(1-p)^x$ for $x=0,1,2,\ldots$, $0<p\leq 1$.

If an element of `x` is not integer, the result of `dgeom` is zero, with a warning.

The quantile is defined as the smallest value x such that $F(x)\geq p$, where F is the distribution function.

Value

`dgeom` gives the density, `pgeom` gives the distribution function, `qgeom` gives the quantile function, and `rgeom` generates random deviates.

Invalid `prob` will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rgeom`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

Unit –IV

The Normal Distribution

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

x, q vector of quantiles.

P vector of probabilities.

n number of observations. If `length(n) > 1`, the length is taken to be the number required.

Mean vector of means.

Sd vector of standard deviations.

log, log.p logical; if TRUE, probabilities p are given as log(p).

lower.tail logical; if TRUE (default), probabilities are P[X≤x] otherwise, P[X>x].

Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has densityf(x)=12πσe−(x−μ)2/2σ2where μ is the mean of the distribution and σ the standard deviation.

Value

`dnorm` gives the density, `pnorm` gives the distribution function, `qnorm` gives the quantile function, and `rnorm` generates random deviates.

The length of the result is determined by `n` for `rnorm`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

For `sd = 0` this gives the limit as `sd` decreases to 0, a point mass at `mu`. `sd < 0` is an error and returns `NaN`.

The Uniform Distribution

These functions provide information about the uniform distribution on the interval from `min` to `max`. `dunif` gives the density, `punif` gives the distribution function `qunif` gives the quantile function and `runif` generates random deviates.

Keywords

distribution

Usage

```
dunif(x, min = 0, max = 1, log = FALSE)
punif(q, min = 0, max = 1, lower.tail = TRUE, log.p = FALSE)
qunif(p, min = 0, max = 1, lower.tail = TRUE, log.p = FALSE)
runif(n, min = 0, max = 1)
```

Arguments

x, q vector of quantiles.
p vector of probabilities.
n number of observations. If `length(n) > 1`, the length is taken to be the number required.
min, max lower and upper limits of the distribution. Must be finite.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X≤x], otherwise, P[X>x].

Details

If `min` or `max` are not specified they assume the default values of `0` and `1` respectively.

The uniform distribution has densityf(x)=1max−minfor min≤x≤max.

For the case of u:=min==max, the limit case of X≡u is assumed, although there is no density in that case and `dunif` will return `NaN` (the error condition).

`runif` will not generate either of the extreme values unless `max = min` or `max-min` is small compared to `min`, and in particular not for the default arguments.

Value

`dunif` gives the density, `punif` gives the distribution function, `qunif` gives the quantile function, and `runif` generates random deviates.

The length of the result is determined by `n` for `runif`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

The Exponential Distribution

Density, distribution function, quantile function and random generation for the exponential distribution with rate `rate` (i.e., mean `1/rate`).

Keywords

distribution

Usage

```
dexp(x, rate = 1, log = FALSE)
pexp(q, rate = 1, lower.tail = TRUE, log.p = FALSE)
qexp(p, rate = 1, lower.tail = TRUE, log.p = FALSE)
rexp(n, rate = 1)
```

Arguments

x, q vector of quantiles.
p vector of probabilities.
n number of observations. If `length(n) > 1`, the length is taken to be the number required.

Rate vector of rates.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X≤x], otherwise, P[X>x].

Details

If `rate` is not specified, it assumes the default value of `1`.

The exponential distribution with rate λ has densityf(x)=λe−λxfor x≥0.

Value

`dexp` gives the density, `pexp` gives the distribution function, `qexp` gives the quantile function, and `rexp` generates random deviates.

The length of the result is determined by `n` for `rexp`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

The Gamma Distribution

Density, distribution function, quantile function and random generation for the Gamma distribution with parameters `shape` and `scale`.

Keywords

distribution

Usage

```
dgamma(x, shape, rate = 1, scale = 1/rate, log = FALSE)
pgamma(q, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
       log.p = FALSE)
qgamma(p, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
       log.p = FALSE)
rgamma(n, shape, rate = 1, scale = 1/rate)
```

Arguments

x, q vector of quantiles.
P vector of probabilities.
n number of observations. If `length(n) > 1`, the length is taken to be the number required.
Rate an alternative way to specify the scale.
shape, scale shape and scale parameters. Must be positive, `scale` strictly.
log, log.p logical; if `TRUE`, probabilities/densities p are returned as log(p).
lower.tail logical; if TRUE (default), probabilities are P[X≤x], otherwise, P[X>x].

Details

If `scale` is omitted, it assumes the default value of `1`.

The Gamma distribution with parameters `shape` =α and `scale` =σ has densityf(x)=1σαΓ(α)xα−1e−x/σfor x≥0, α>0 and σ>0. (Here Γ(α) is the function implemented by R's `gamma()` and defined in its help. Note that a=0 corresponds to the trivial distribution with all mass at point 0.)

The mean and variance are $E(X)=\alpha\sigma$ and $Var(X)=\alpha\sigma2$.

The cumulative hazard $H(t)=-\log(1-F(t))$ is

```
-pgamma(t, ..., lower = FALSE, log = TRUE)
```

Note that for smallish values of `shape` (and moderate `scale`) a large parts of the mass of the Gamma distribution is on values of x so near zero that they will be represented as zero in computer arithmetic. So `rgamma` may well return values which will be represented as zero. (This will also happen for very large values of `scale` since the actual generation is done for `scale = 1`.)

Value

`dgamma` gives the density, `pgamma` gives the distribution function, `qgamma` gives the quantile function, and `rgamma` generates random deviates.

Invalid arguments will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rgamma`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

Unit V

LINEAR CORRELATION

The term correlation is used by a common man without knowing that he is making use of the term correlation. For example when parents advice their children to work hard so that they may get good marks, they are correlating good marks with hard work. The study related to the characteristics of only variable such as height, weight, ages, marks, wages, etc., is known as univariate analysis. The statistical Analysis related to the study of the relationship between two variables is known as Bi-Variate Analysis. Sometimes the variables may be inter-related. In health sciences we study the relationship between blood pressure and age, consumption level of some nutrient and weight gain, total income and medical expenditure, etc. The nature and strength of relationship may be examined by correlation and Regression analysis. Thus Correlation refers to the relationship of two variables or more. Correlation is statistical Analysis which measures and analyses the degree or extent to which the two variables fluctuate with reference to each other. The word relationship is important. It indicates that there is some connection between the variables. It measures the closeness of the relationship. Correlation does not indicate cause and effect relationship. Price and supply, income and expenditure are correlated.

Meaning of Correlation:

In a bivariate distribution we may interested to find out if there is any correlation or covariation between the two variables under study. If the change in one variable affects a change in the other variable, the variables are said to be correlated. If the two variables deviate in the same direction, i.e., if the increase in one results in a corresponding increase in theother, correlation is said to be direct or positive.

Example

- The heights or weights of a group ofpersons
- The income and expenditure is positive and correlation between

- Price and demand of a commodity
- The volume and pressure of a perfect gas; is negative

Correlation is said to be perfect if the deviation one variable is followed by a corresponding and proportional deviation in theother.

Definitions:

Ya-Kun-Chou:

Correlation Analysis attempts to determine the degree of relationship between variables.

A.M. Tuttle:

Correlation is an analysis of the covariation between two or more variables. Correlation expresses the inter-dependence of two sets of variables upon each other. One variable may be called as (subject) independent and the other relative variable (dependent). Relative variable is measured in terms of subject.

Uses of correlation:

1. It is used in physical and social sciences.

2. It is useful for economists to study the relationship between variables like price, quantity.

4. Businessmen estimates costs, sales, price etc. using correlation.

4. It is helpful in measuring the degree of relationship between the variables like income and expenditure, price and supply, supply and demand etc.

5. Sampling error can be calculated.

6. It is the basis for the concept of regression.

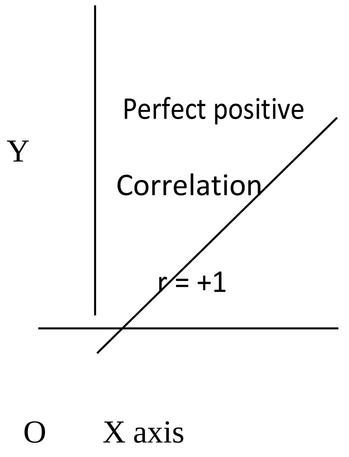
SCATTER DIAGRAM

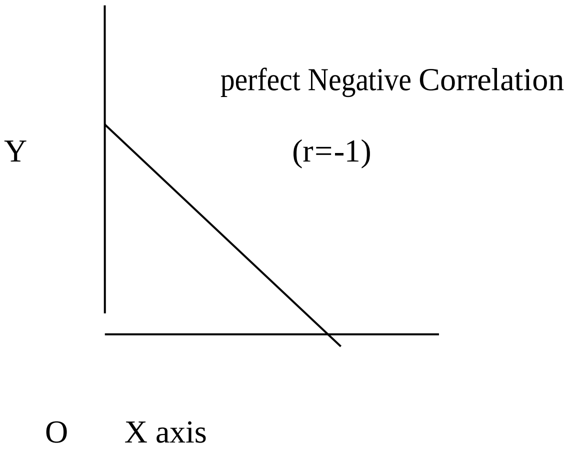
Scatter diagram pertaining independent variables, it is easily verifiable that if any line is drawn through the plotted points, not more than two points will be lying on the line most of the other points will be at a considerable distance from this line. Scatter diagram that the two variables are linearly related, the problem arises on deciding which of the many possible lines the best fitted line is. The lease square method is the most widely accepted method of fitting a straight line and is discussed here adequately.

Use of Scatter Diagram:

- When you have paired numerical data
- When trying to identify potential root causes of problems.
- After brain storming causes and effects using a bishbone diagram, to determine objectively whether a particular cause and effect are related.
- When determining whether two effects that appear to be related both occur with the same cause

It is the simplest method of studying the relationship between two variables diagrammatically. One variable is represented along the horizontal axis and the second variable along the vertical axis. For each pair of observations of two variables, we put a dot in the plane. There are as many dots in the plane as the number of paired observations of two variables. The direction of dots shows the scatter or concentration of various points. This will show the type of correlation.

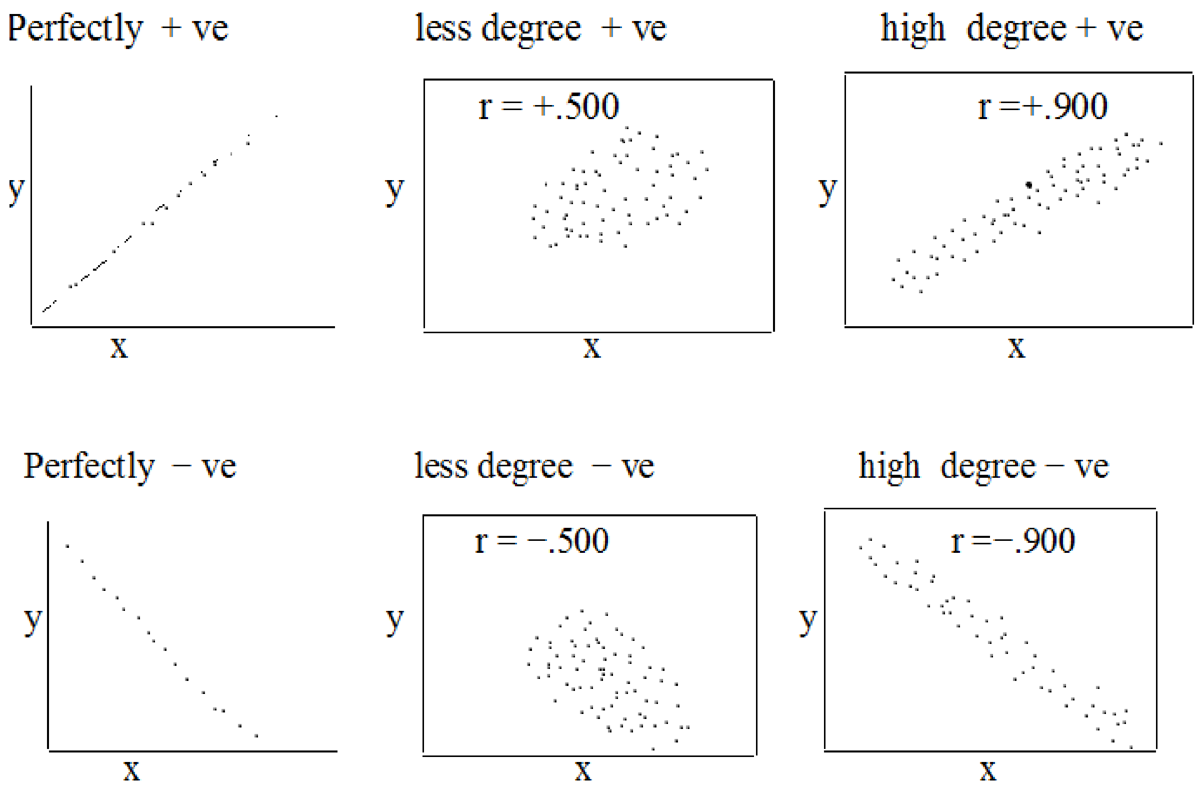




1. If all the plotted dots lie on a straight line falling from upper left hand corner to lower right hand corner, there is a perfect negative correlation between the two variables. In this case the coefficient of correlation takes the value r = -1.

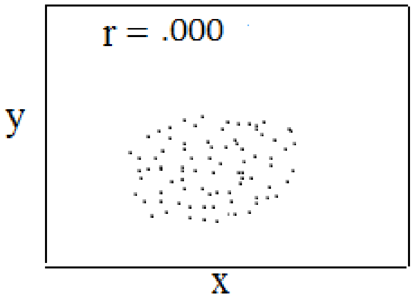
2. If all the plotted points form a straight line from lower left hand corner to the upper right hand corner then there is Perfect positive correlation. We denote this as r = +1

3. If the plotted points in the plane form a band and they show a rising trend from the lower left hand corner to the upper right hand corner the two variables are highly positively correlated. Highly Positive Highly Negative



1. If the points fall in a narrow band from the upper left hand corner to the lower right hand corner, there will be a high degree of negative correlation.

2. If the plotted points in the plane are spread all over the diagram there is no correlation between the two variables.



Merits:

1. It is a simplest and attractive method of finding the nature of correlation between the two variables.

2. It is a non-mathematical method of studying correlation. It is easy to understand.

3. It is not affected by extreme items.

4. It is the first step in finding out the relation between the two variables.

5. We can have a rough idea at a glance whether it is a positive correlation or negative correlation.

Demerits:

By this method we cannot get the exact degree or correlation between the two variables.

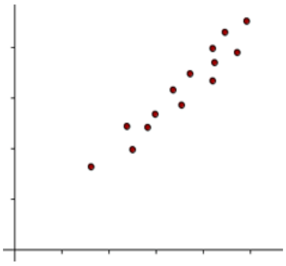
Types of Correlation:

Correlation is classified into various types. The most important ones are

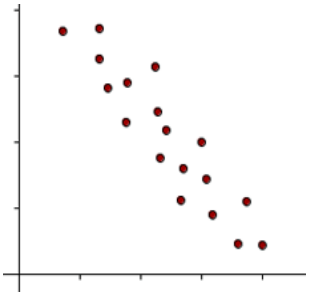
- Positive and negative.
- Linear and non-linear.
- Partial and total.
- Simple and Multiple.

Positive and Negative Correlation

It depends upon the direction of change of the variables. If the two variables tend to move together in the same direction (i. e) an increase in the value of one variable is accompanied by an increase in the value of the other, (or) a decrease in the value of one variable is accompanied by a decrease in the value of other, then the correlation is called positive or direct correlation. Price and supply, height and weight, yield and rainfall, are some examples of positive correlation.



If the two variables tend to move together in opposite directions so that increase (or) decrease in the value of one variable is accompanied by a decrease or increase in the value of the other variable, then the correlation is called negative (or) inverse correlation. Price and demand, yield of crop and price, are examples of negative correlation.



Linear and Non-linear correlation:

If the ratio of change between the two variables is a constant then there will be linear correlation between them.

Example

Consider the variables with the following values.

X	10	20	30	40	50
Y	20	40	60	80	100

Here the ratio of change between the two variables is the same. If we plot these points on a graph we get a straight line.

If the amount of change in one variable does not bear a constant ratio of the amount of change in the other. Then the relation is called Curve- linear (or) non-linear correlation. The graph will be a curve.

Example

Consider the variables with the following values

X	10	20	30	40	50
Y	10	30	70	90	120

Here there is a non linear relationship between the variables. The ratio between them is not fixed for all points. Also if we plot them on the graph, the points will not be in a straight line. It will be a curve.

Simple and Multiple correlation:

When we study only two variables, the relationship is simple correlation. For example, quantity of money and price level, demand and price. But in a multiple correlation we study more than two variables simultaneously. The relationship of price, demand and supply of a commodity are an example for multiple correlations.

Example:

Calculate coefficient of correlation from the following data.

x	1	2	3	4	5	6	7	8	9
y	9	8	10	12	11	13	14	16	15

Solution:

x	Y	x^2	y^2	xy
1	9	1	81	9
2	8	4	64	16
3	10	9	100	30
4	12	16	144	48
5	11	25	121	55
6	13	36	169	78
7	14	49	196	98
8	16	64	256	128
9	15	81	225	135
45	108	285	1356	597

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

$$r = \frac{9 \times 597 - 45 \times 108}{\sqrt{(9 \times 285 - (45)^2).(9 \times 1356 - (108)^2)}}$$

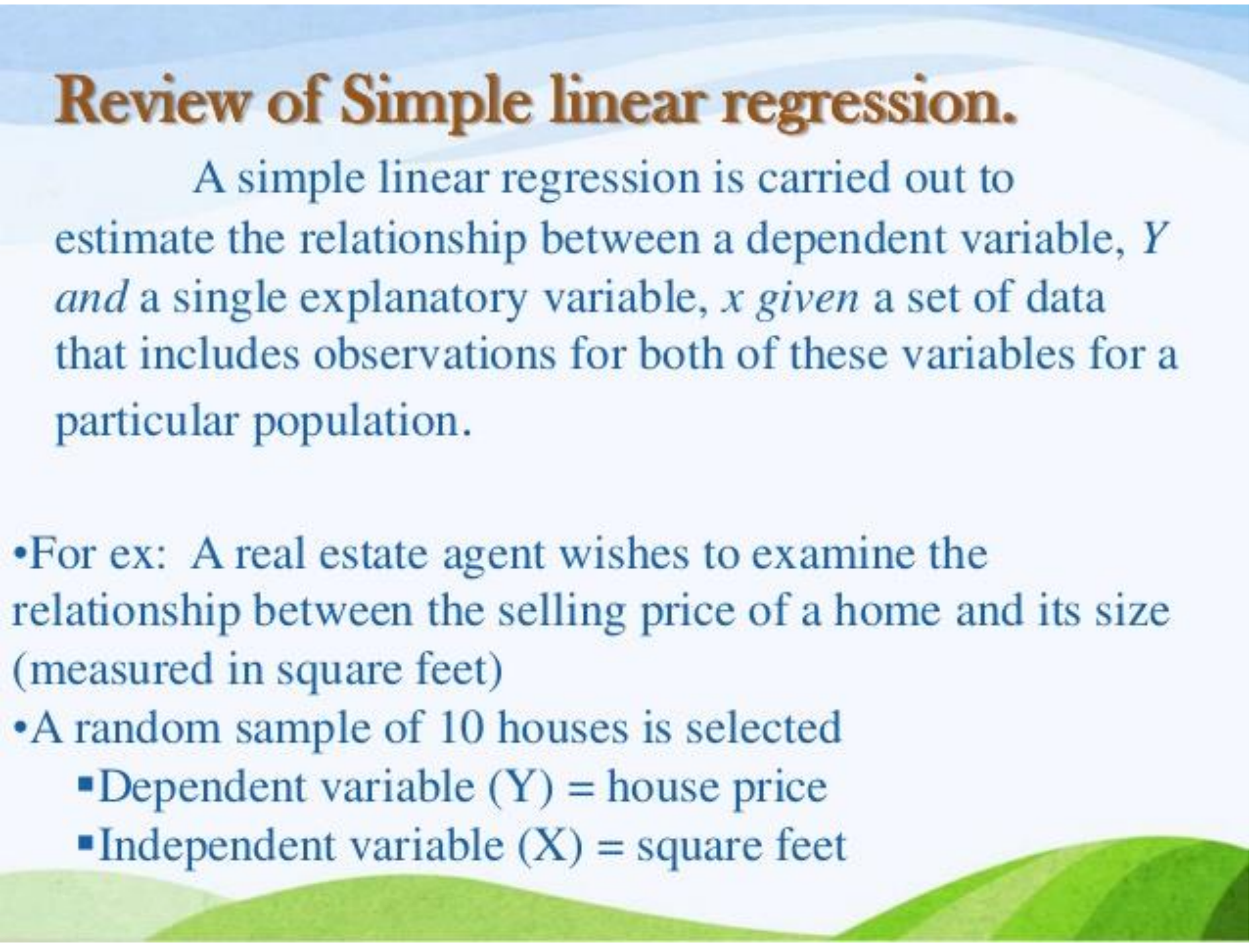
$$r = \frac{5373 - 4860}{\sqrt{(2565 - 2025).(12204 - 11664)}}$$

$= 0.95$

$r = 0.95$

Regression

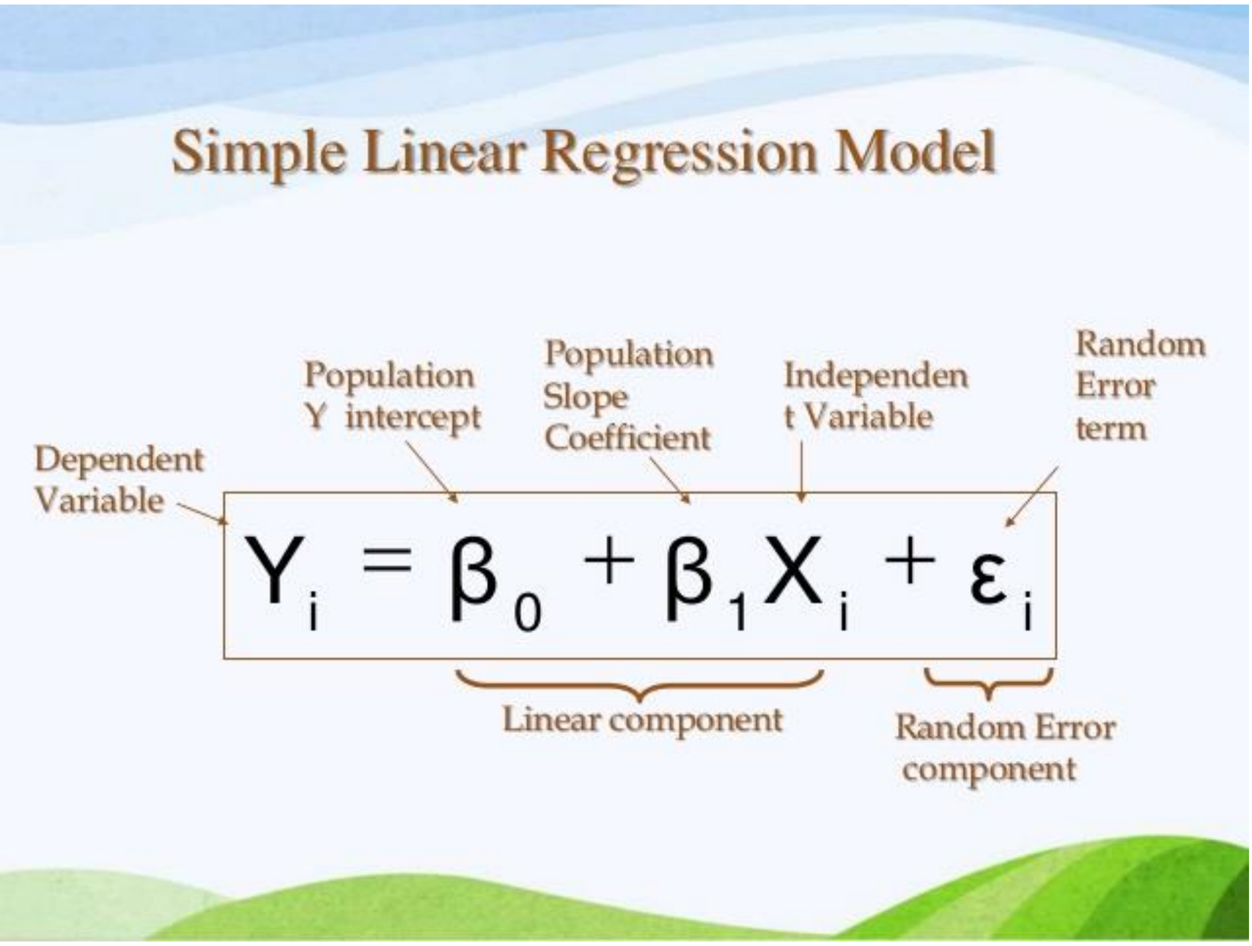
MEANING OF REGRESSION: The dictionary meaning of the word Regression is ‘Stepping back’ or ‘Going back’. Regression is the measures of the average relationship between two or more variables in terms of the original units of the data. And it is also attempts to establish the nature of the relationship between variables that is to study the functional relationship between the variables and thereby provide a mechanism for prediction, or forecasting.



Review of Simple linear regression.

A simple linear regression is carried out to estimate the relationship between a dependent variable, *Y and* a single explanatory variable, *x given* a set of data that includes observations for both of these variables for a particular population.

- For ex: A real estate agent wishes to examine the relationship between the selling price of a home and its size (measured in square feet)
- A random sample of 10 houses is selected
  - Dependent variable (Y) = house price
  - Independent variable (X) = square feet



Simple Linear Regression Model

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Example 9.9

Calculate the regression coefficient and obtain the lines of regression for the following data

X	1	2	3	4	5	6	7
Y	9	8	10	12	11	13	14

Solution:

X	Y	X^2	Y^2	X^Y
1	9	1	81	9
2	8	4	64	16
3	10	9	100	30
4	12	16	144	48
5	11	25	121	55
6	13	36	169	78
7	14	49	196	98
$\sum X = 28$	$\sum Y = 77$	$\sum X^2 = 140$	$\sum Y^2 = 875$	$\sum XY = 334$

Table 9.7

$$\overline{X} = \frac{\Sigma X}{N} = \frac{28}{7} = 4\ ,$$

$$\overline{Y} = \frac{\Sigma Y}{N} = \frac{77}{7} = 11$$

Regression coefficient of *X* on *Y*

$$b_{xy} = \frac{N\Sigma XY - (\Sigma X)(\Sigma Y)}{N\ \Sigma Y^2 - (\Sigma Y)^2}$$

$$= \frac{7(334) - (28)(77)}{7(875) - (77)^2}$$

$$= \frac{2338 - 2156}{6125 - 5929}$$

$$= \frac{182}{196}$$

$$b_{xy} = 0.929$$

(i) Regression equation of *X* on *Y*

$$X - \overline{X} = b_{xy}(\mathrm{Y} - \overline{Y})$$

$$X - 4 = 0.929(Y - 11)$$

$$X - 4 = 0.929Y - 10.219$$

$\therefore$ The regression equation X on Y is $X = 0.929Y - 6.219$

(ii) Regression coefficient *of Y* on *X*

$$b_{yx} = \frac{N\Sigma XY-(\Sigma X)(\Sigma Y)}{N\,\Sigma X^2-(\Sigma X)^2}$$

$$= \frac{7(334)-(28)(77)}{7(140)-(28)^2}$$

$$= \frac{2338-2156}{980-784}$$

$$= \frac{182}{196}$$

$$\therefore \quad b_{yx} = 0.929$$

(iii) Regression equation of *Y* on *X*

$$Y-\overline{Y} = b_{yx}(X-\overline{X})$$

$$Y-11 = 0.929\ (X-4)$$

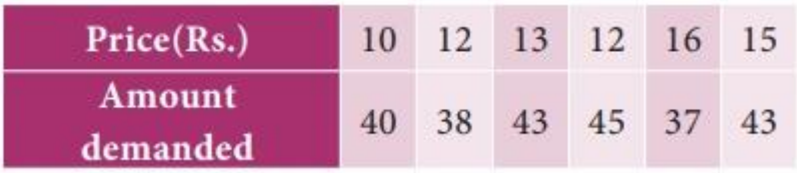
$Y = 0.929X–3.716+11$

$= 0.929X+7.284$

The regression equation of *Y* on *X* is *Y*= 0.929*X* + 7.284

Example 9.10

Calculate the two regression equations of *X* on *Y* and *Y* on *X* from the data given below, taking deviations from a actual means of *X* and *Y*.

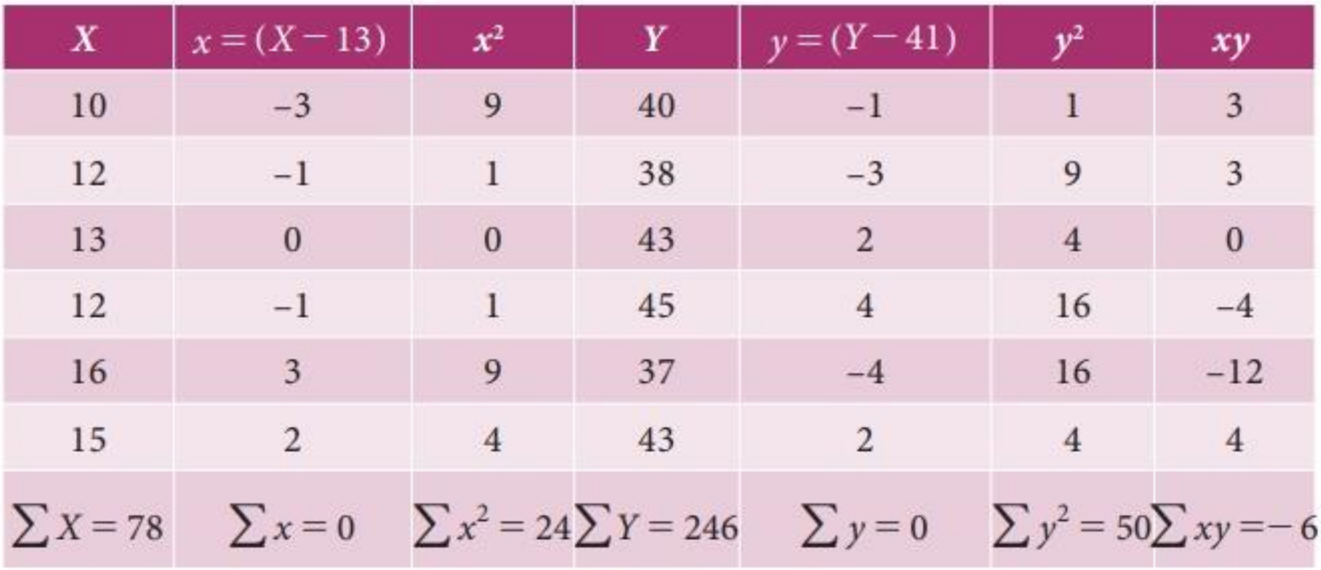


Price(Rs.)	10	12	13	12	16	15
Amount demanded	40	38	43	45	37	43

Estimate the likely demand when the price is Rs.20.

Solution:

Calculation of Regression equation



X	$x=(X-13)$	x^2	Y	$y=(Y-41)$	y^2	xy
10	–3	9	40	–1	1	3
12	–1	1	38	–3	9	3
13	0	0	43	2	4	0
12	–1	1	45	4	16	–4
16	3	9	37	–4	16	–12
15	2	4	43	2	4	4
$\sum X=78$	$\sum x=0$	$\sum x^2=24$	$\sum Y=246$	$\sum y=0$	$\sum y^2=50$	$\sum xy=-6$

Table 9.8

(i) Regression equation of *X* on *Y*

$$X-\overline{X}=r\frac{\sigma_x}{\sigma_y}(Y-\overline{Y})$$

$$\overline{X}=\frac{78}{6}=13\ ,\ \overline{Y}=\frac{246}{6}=41$$

$$b_{xy}=r\frac{\sigma_x}{\sigma_y}=\frac{\Sigma xy}{\Sigma y^2}=\frac{-6}{50}=-0.12$$

$$X-13=-0.12\ (Y-41\ )$$

$$X-13=-0.12Y+4.92$$

$$X=-0.12Y+17.92$$

(ii) Regression Equation of Y on X

$$Y-\overline{Y}=r\frac{\sigma_y}{\sigma_x}(X-\overline{X}\ )$$

$$b_{yx}=r\frac{\sigma_y}{\sigma_x}=\frac{\Sigma xy}{\Sigma x^2}=-\frac{6}{24}=-0.25$$

$$Y-41=-0.25\ (X-13\ )$$

$$Y-41=-0.25\ X+3.25$$

$$Y=-0.25\ X+44.25$$

When *X* is 20, *Y* will be

$$= -0.25\ (20)+44.25$$

$$= -5+44.25$$

= 39.25 (when the price is Rs. 20, the likely demand is 39.25)