E-CONTENT PREPARED BY

Mr. Sanghamitra Sanyal

SACT of Department of Conservation Biology

Durgapur Government College, Durgapur, West Bengal

(*Affiliated to Kazi Nazrul University, Asansol, West Bengal*)

NAAC Accredited "A" Grade College

(Recognized under Section 2(f) and 12(B) of UGC Act 1956)

E-Content prepared for students of

M.Sc.(Semester-IV) in Conservation Biology

Name of Course:
Bioinformatics, Biostatistics and Computer Applications

Topic of the E-Content: Data Analysis using R

Table of Contents

R-PROGRAMMING: - AN INTRODUCTORY NOTE

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues.

R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible and finally, R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and macOS.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes an effective data handling and storage facility,

- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

CONSTRUCTION OF A PIE-CHART USING 'R'

Aim- To draw a pie-chart in R using the provided data.

Principle- R Programming language has numerous libraries to create charts and graphs. A pie-chart is a representation of values as slices of a circle with different colours. The slices are labeled and the numbers corresponding to each slice is also represented in the chart.

In R the pie chart is created using the pie() function which takes positive numbers as a vector input. The additional parameters are used to control labels, colour, title etc.

Syntax

The basic syntax for creating a pie-chart using the R is –

pie(x, labels, radius, main, col, clockwise)

Following is the description of the parameters used −

x is a vector containing the numeric values used in the pie chart.

labels are used to describe the slices.

radius indicates the radius of the circle of the pie chart. (value between −1 and +1).

main indicates the title of the chart.

col indicates the colour palette.

clockwise is a logical value indicating if the slices are drawn clockwise or anti-clockwise.

Procedure:

I: Inserting Data

```
#Define car vector with 7 values
cars=c(1,3,6,4,9,5,10)
length(cars)
> #Define car vector with 7 values
> cars=c(1,3,6,4,9,5,10)
> length(cars)
[1] 7
```

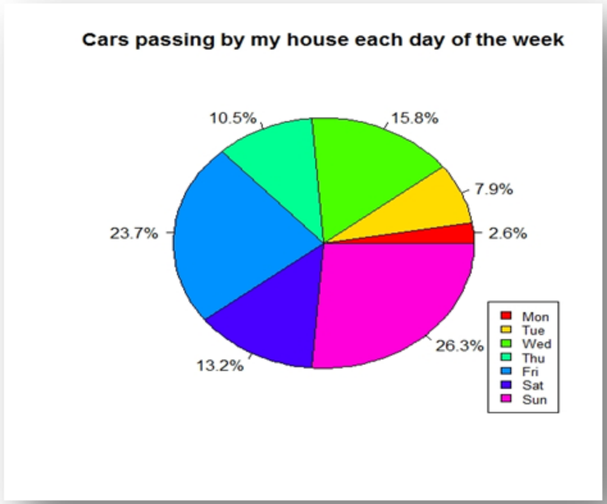
II: Calculating the Percentage Value

```
#Calculate the percentage for each day, rounded to one decimal place
car_labels=round(cars/sum(cars)*100,1)
car_labels
#Concatenate a '%' char after each value
car_labels=paste(car_labels,"%",sep="")
car_labels
> #Calculate the percentage for each day, rounded to one decimal place
> car_labels=round(cars/sum(cars)*100,1)
> car_labels
[1]  2.6  7.9 15.8 10.5 23.7 13.2 26.3
> #Concatenate a '%' char after each value
> car_labels=paste(car_labels,"%",sep="")
> car_labels
[1] "2.6%"  "7.9%"  "15.8%" "10.5%" "23.7%" "13.2%" "26.3%"
```

III: Draw a Pie-Chart

```
#Create a pie chart with defined heading and custom colours and labels
pie(cars,
main="Cars passing by my house each day of the week",
col=rainbow(length(cars)),
labels=car_labels,)
#Create a legend at the right
legend("bottomright", c("Mon","Tue","Wed","Thu","Fri","Sat","Sun"), cex=0.8,
fill=rainbow(length(cars)))
> #Create a pie chart with defined heading and custom colours and labels
>pie(cars,
+ main="Cars passing by my house each day of the week",
+ col=rainbow(length(cars)),
+ labels=car_labels,)
> #Create a legend at the right
>legend("bottomright", c("Mon","Tue","Wed","Thu","Fri","Sat","Sun"), cex=0.8,
+ fill=rainbow(length(cars)))
```

<u>Observation:</u>



<u>Conclusion:</u> The percentage of value of cars which are passed on Sunday is higher than all other days.

CONSTRUCTION OF A BARPLOT USING 'R'

Aim- To draw a bar plot in R using provided data.

Principle- A bar chart represents data in rectangular bars with a length of the bar proportional to the value of the variable. R uses the function **barplot()** to create bar charts. R can draw both vertical and horizontal bars in the bar chart. In the bar chart, each of the bars can be given different colours.

Procedure:

I: Inserting Data

```
#Enter Data

Atasi = c(42,41,40,45,41,47,43)
Chandrani =c(42,45,42,44,44,46,43)
Fortune = c(41,42,39,43,41,46,41)
Mou = c(40,44,39,44,43,46,42)
Sujata = c(40,43,35,45,43,46,41)

papers = c("MSCCONBC301","MSCCONBC302","MSCCONBC303","MSCCONBC304","Major-Theory",
"Major-Practical", "MSCCONBMIE301")

# convert to dataframe

sem3.marks = data.frame(Atasi,Chandrani, Fortune, Mou, Sujata)
str(sem3.marks)

#make room for legend
par(xpd=T, mar = par()$mar + c(0,0,0,3))
```

```
> #Enter Data
>
> Atasi = c(42,41,40,45,41,47,43)
> Chandrani =c(42,45,42,44,44,46,43)
> Fortune = c(41,42,39,43,41,46,41)
> Mou = c(40,44,39,44,43,46,42)
> Sujata = c(40,43,35,45,43,46,41)
>
> papers = c("MSCCONBC301","MSCCONBC302","MSCCONBC303","MSCCONBC304","Major-Theory",
"Major-Practical", "MSCCONBMIE301")
>
> # convert to dataframe
>
> sem3.marks = data.frame(Atasi,Chandrani, Fortune, Mou, Sujata)
>str(sem3.marks)
```

```
'data.frame':   7 obs. of  5 variables:
 $ Atasi    : num  42 41 40 45 41 47 43
 $ Chandrani: num  42 45 42 44 44 46 43
 $ Fortune  : num  41 42 39 43 41 46 41
 $ Mou      : num  40 44 39 44 43 46 42
 $ Sujata   : num  40 43 35 45 43 46 41
>
> #make room for legend
>par(xpd=T, mar = par()$mar + c(0,0,0,3))
>
```

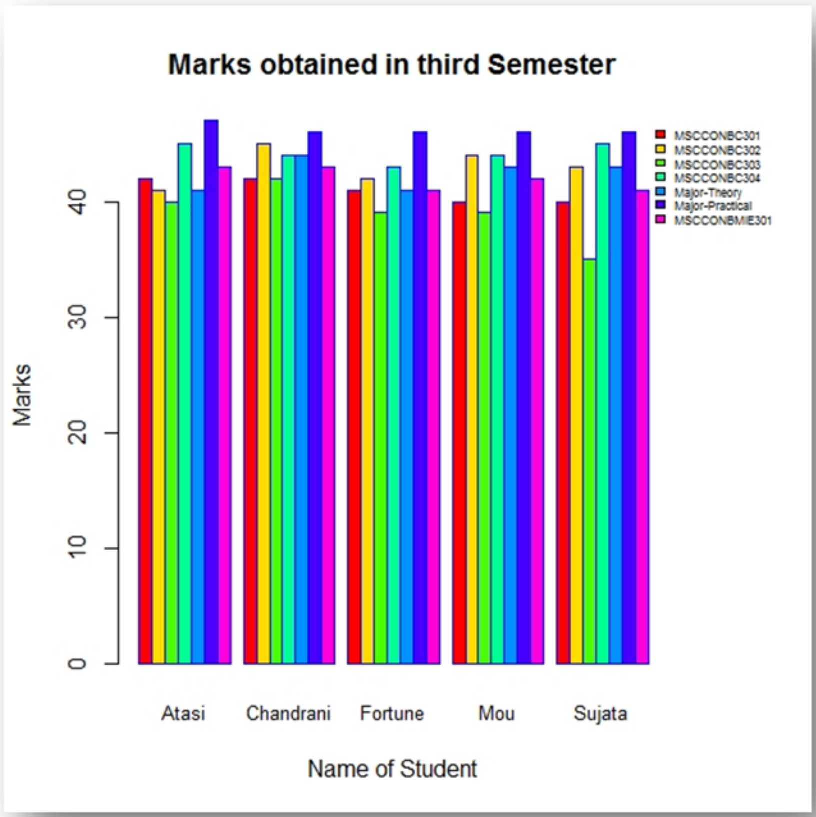
II: Draw a Barplot

```
# Draw a Bar Graph

barplot(as.matrix(sem3.marks),
        main = "Marks obtained in third Semester" ,
        ylab = "Marks" ,
        xlab = "Name of Student",
        beside = TRUE,
        bty = 'L',
cex.names= 0.8,
        border = "blue",
        col = rainbow(7)
        )
# Place a Legend at the topleft corner
legend("topright", papers, inset=c(-0.2,0), cex=0.5, bty = "n", fill = rainbow(7))
```

```
> # Draw a Bar Graph
>
> barplot(as.matrix(sem3.marks),
+         main = "Marks obtained in third Semester" ,
+         ylab = "Marks" ,
+         xlab = "Name of Student",
+         beside = TRUE,
+         bty = 'L',
+         cex.names= 0.8,
+         border = "blue",
+         col = rainbow(7)
+         )
```

```
> # Place a Legend at the topleft corner
>legend("topright", papers, inset=c(-0.2,0), cex=0.5, bty = "n", fill = rainbow(7))
```

Observation:

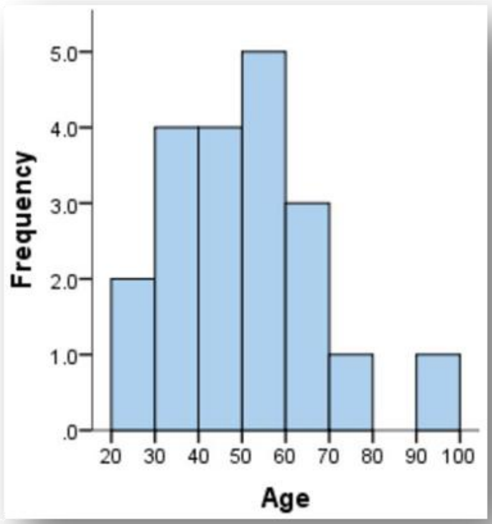


Conclusion: From this barplot it can be concluded that the overall marks of Chandrani is highest in M.Sc. third semester examination.

CONSTRUCTION OF A HISTOGRAM USING 'R'

Aim- To draw a bar plot in R using provided data.

Principle-A histogram is a plot that lets you discover and show the underlying frequency distribution (shape) of a set of continuous univariate data. This allows the inspection of the data for its underlying distribution (e.g. normal distribution), outliers, skewness, etc. An example of a histogram and the raw data it was constructed from is shown below:



To construct a histogram from a continuous variable we first need to split the data into intervals called bins. In the example above, age has been split into bins, with each bin representing a 10 year period starting at 20 years. Each bin contains the number of occurrences of scores in the data set that are contained within that bin. For the above data set, the frequencies in each bin have been tabulated along with the scores that contributed to the frequency in each bin:

Bin	Frequency	Scores IncludedinBin
20-30	2	25,22
30-40	4	36,38,36,38
40-50	4	46,45,48,46
50-60	5	55,55,52,58,55
60-70	3	68,67,61
70-80	1	72
80-90	0	-
90-100	1	91

Notice that, unlike a bar chart, there are no “gaps” between the bars (although some bars might be “absent” reflecting no frequencies). This is because a histogram represents a continuous data set and as such, there are no gaps in the data (although we will have to decide whether we round up or round down scores on the boundaries of bins).

In a histogram, it is the area of the bar that indicates the frequency of occurrences for each bin. This means that the height of the bar does not necessarily indicate how many occurrences of scores thee wee within each individual bin. It is the product of height multiplied by the width of the bin that indicates the frequency of occurrences within that bin. One of the reasons that the height of the bars is often incorrectly assessed as indicating frequency and not the area of the bar is due to the fact that a lot of histograms often have equally spaced bars (bins) and under these circumstances, the height the bin does not reflect the frequency.

Procedure:

I: Inserting Data

```
#Enter the data
chandrani=c(4,12,11,20,23,13,22,26,30,26,32,24,31,28,40,42,36,43,36,39,42,40,41,53,51,43,48,50,44,45,46,5
7,57,54,54,62,59,63,53,60,69,70,68,79,85,78,81,76,77,83,83,
88,97)
> #Enter the data
> chandrani=c(4,12,11,20,23,13,22,26,30,26,32,24,31,28,40,42,36,43,36,39,42,40,41,53,
+ 51,43,48,50,44,45,46,57,57,54,54,62,59,63,53,60,69,70,68,79,85,78,81,76,77,83,83,
+ 88,97)
>
```

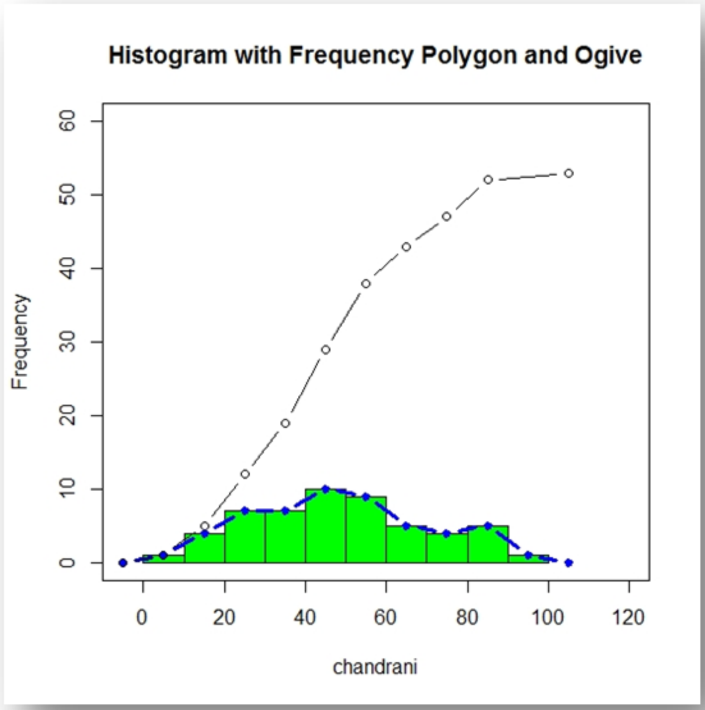
II: Draw a Histogram

```
h= hist(chandrani,
col="green",
xlim=c(-5,120),
ylim=c(0,60),
main= "Histogram of Chandrani with Frequency Polygon and Ogive",
freq=T
)
#Plot a histogram
h
#For frquency polygon
mp = c(min(h$mids) - (h$mids[2] - h$mids[1]), h$mids, max(h$mids) +
(h$mids[2] - h$mids[1]))
freq = c(0, h$counts, 0)

#Plot for frequency polygon
lines(mp,freq, type = "b", pch = 20, col = "blue", lwd =3)
#For Ogive
h.cumsum= cumsum(h$count)
```

```
ogive= c(0, h.cumsum)
ogive
mp2= mp[-11]
#Plot the Ogive
lines(mp2,ogive, type = "b", pch = 1, col = "black", lwd =1)
#Add a box around it
box()
> h= hist(chandrani,
+ col="green",
+ xlim=c(-5,120),
+ ylim=c(0,60),
+ main= "Histogram of Chandrani with Frequency Polygon and Ogive",
+ freq=T
+ )
> #Plot a histogram
> h
$breaks
 [1]   0  10  20  30  40  50  60  70  80  90 100

$counts
 [1]  1  4  7  7 10  9  5  4  5  1

$density
 [1] 0.001886792 0.007547170 0.013207547 0.013207547 0.018867925 0.016981132
 [7] 0.009433962 0.007547170 0.009433962 0.001886792

$mids
 [1]  5 15 25 35 45 55 65 75 85 95

$xname
[1] "chandrani"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
>
> #For frquency polygon
> mp = c(min(h$mids) - (h$mids[2] - h$mids[1]), h$mids, max(h$mids) +
+ (h$mids[2] - h$mids[1]))
> freq = c(0, h$counts, 0)
>
> #Plot for frequency polygon
>lines(mp,freq, type = "b", pch = 20, col = "blue", lwd =3)
>
> #For Ogive
>h.cumsum= cumsum(h$count)
> ogive= c(0, h.cumsum)
> ogive
```

```
 [1]  0  1  5 12 19 29 38 43 47 52 53
> mp2= mp[-11]
>
> #Plot the Ogive
>lines(mp2,ogive, type = "b", pch = 1, col = "black", lwd =1)
>
> #Add a box around it
>box()
>
```

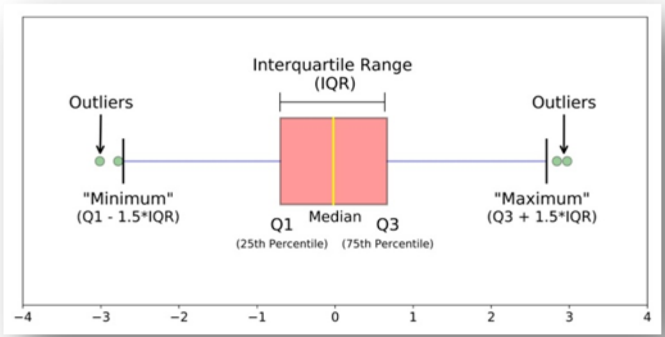
Observation:



CONSTRUCTION OF A BOXPLOT USING 'R'

Aim- To draw a boxplot in R using the provided data.

Principle- Boxplots are a measure of how well distributed is the data in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and the third quartile in the data set. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.



Boxplots are created in R by using the **boxplot()** function.

Syntax:-

The basic syntax to create a boxplot in R is −

boxplot(x, data, notch, varwidth, names, main)

Following is the description of the parameters used −

- **x** is a vector or a formula.
- **data** is the data frame.
- **a notch** is a logical value. Set as TRUE to draw a notch.
- **varwidth** is a logical value. Set as true to draw width of the box proportionate to the sample size.
- **names** are the group labels which will be printed under each boxplot.
- **main** is used to give a title to the graph.

Procedure:

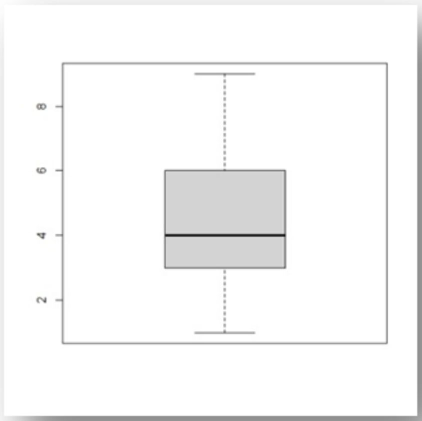
Example 1: I: Inserting Data

```
# Define cars vector with 5 values
cars = c(1, 3, 6, 4, 9)
```

II: Draw a Boxplot

```
# Create a box plot for cars
boxplot(cars)
```

Observation:



Conclusion: From the boxplot, the data distribution is slightly skewed. There are no outliers in this distribution.

Example 2: I: Inserting Data

```
Sites = c("TAMLA","PD-1","PD-2","MUCHIPARA")
acidity = c(20,30,20,20)
chloride = c(35.45,35.45,35.45,28.36)
total.hardness = c(120,80,40,60)
calcium.hardness = c(120,60,40,60)
COD = c(128,NA,64,96)
pH =c(7.4,8.29,8.1,8.2)
conductivity = c(619,420,410,336)
```

```
> Sites = c("TAMLA","PD-1","PD-2","MUCHIPARA")
> acidity = c(20,30,20,20)
> chloride = c(35.45,35.45,35.45,28.36)
>total.hardness = c(120,80,40,60)
>calcium.hardness = c(120,60,40,60)
> COD = c(128,NA,64,96)
> pH =c(7.4,8.29,8.1,8.2)
> conductivity = c(619,420,410,336)
```

II: Creating Data Frame

```
chandrani = data.frame(acidity,chloride,total.hardness,calcium.hardness,COD,pH,conductivity)
row.names(chandrani)= Sites
chandrani
```

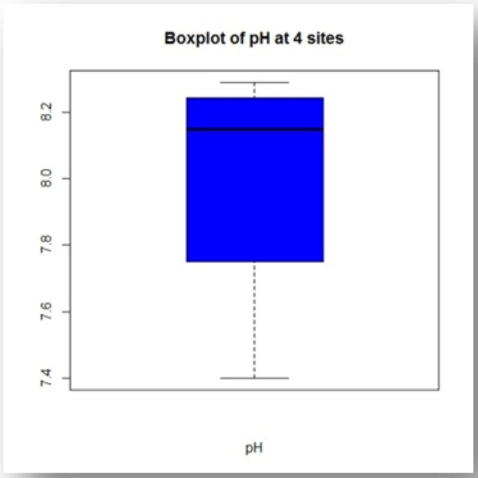
```
> chandrani = data.frame(acidity,chloride,total.hardness,calcium.hardness,COD,pH,conductivity)
>row.names(chandrani)= Sites
> chandrani
acidity  chloride  total.hardness  calcium.hardness  COD   pH
TAMLA    20    35.45          120                  120               128  7.40
PD-1       30    35.45           80                    60                 NA   8.29
PD-2          20     35.45           40                   40                  64    8.10
MUCHIPARA   20  28.36         60                    60                  96    8.20
                  conductivity
TAMLA            619
PD-1                420
PD-2                 410
MUCHIPARA    336
```

III: Draw a Boxplot

```
boxplot(chandrani$pH,
main="Boxplot of pH at 4 sites",
col="blue",
xlab="pH")
```

```
>boxplot(chandrani$pH,
+ main="Boxplot of pH at 4 sites",
+ col="blue",
+ xlab="pH")
```

<u>**Observation:**</u>



Conclusion: From the boxplot, it can be concluded that the pH data is mostly skewed. There are no outliers in thisdistribution.

Example 3: I: Inserting data

```
Tamla = c(10,20,30,40,50,64)
PD1 = c(12,43,54,23,34,34)
PD2 = c(43,34,23,12,54,65)
Muchipara = c(12,12,23,45,67,87)
```

```
> Tamla = c(10,20,30,40,50,64)
> PD1 = c(12,43,54,23,34,34)
> PD2 = c(43,34,23,12,54,65)
> Muchipara = c(12,12,23,45,67,87)
```

II: Creating Data Frame

```
dat = data.frame(Tamla,PD1,PD2, Muchipara)
dat
```

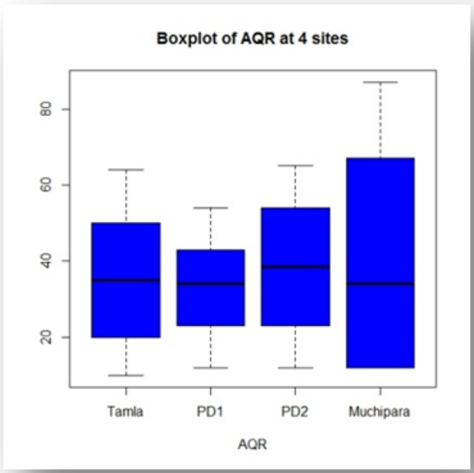
```
> dat = data.frame(Tamla,PD1,PD2, Muchipara)
> dat
  Tamla  PD1  PD2   Muchipara
1   10    12   43       12
2   20    43   34       12
3   30    54   23       23
4   40    23   12       45
5   50    34   54       67
6   64    34   65       87
```

III: Draw a Boxplot

```
boxplot(dat,
main="Boxplot of AQR at 4 sites",
col="blue",
xlab="AQR")
```

```
>  boxplot(dat,
+ main="Boxplot of AQR at 4 sites",
+ col="blue",
+ xlab="AQR")
```

Observation:



Conclusion: From the boxplot, it can be concluded that the AQR data of all four sites are widely distributed except Muchipara. The AQR data of Muchipara is slightly skewed. There are no outliers in any of these four distributions.

Example 4: I: Inserting Data

```
Tamla = c(10,20,30,40,50,64)
PD1 = c(12,43,54,23,34,34)
PD2 = c(43,34,23,12,54,65)
Muchipara = c(12,12,23,45,67,87)
```

```
> Tamla = c(10,20,30,40,50,64)
> PD1 = c(12,43,54,23,34,34)
> PD2 = c(43,34,23,12,54,65)
> Muchipara = c(12,12,23,45,67,87)
```

II: Creating Data Frame

```
dat = data.frame(Tamla,PD1,PD2,Muchipara)
dat
```

```
> dat = data.frame(Tamla,PD1,PD2,Muchipara)
> dat
  Tamla   PD1   PD2   Muchipara
1   10     12    43      12
2   20     43    34      12
```

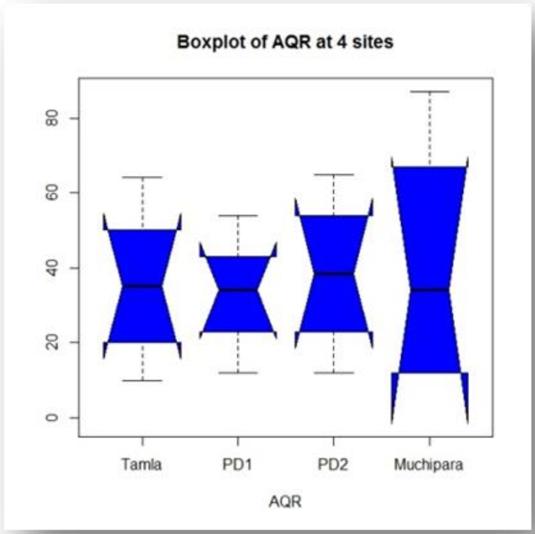
```
3   30      54      23      23
4   40      23      12      45
5   50      34      54      67
6   64      34      65      87
```

III: Draw a Boxplot

```
boxplot(dat,
main="Boxplot of AQR at 4 sites",
col="blue",
xlab="AQR",
notch=TRUE)
```

```
>  boxplot(dat,
+ main="Boxplot of AQR at 4 sites",
+ col="blue",
+ xlab="AQR",
+ notch=TRUE)
Warning message:
In bxp(list(stats = c(10, 20, 35, 50, 64, 12, 23, 34, 43, 54, 12,  :
  some notches went outside hinges ('box'): maybe set notch=FALSE
```

Observation:



Conclusion: From the boxplot, it can be concluded that the AQR data of all four sites are widely distributed except Muchipara. The AQR data of Muchipara is slightly skewed. There are no outliers in any of these three distributions.

CONSTRUCTION OF A SCATTERPLOT USING 'R'

Aim- To draw a scatterplot in R using the provided data.

Principle- Scatterplots show many points plotted in the Cartesian plane. Each point represents the values of two variables. One variable is chosen in the horizontal axis and another in the vertical axis.

The simple scatterplot is created using the plot() function.

Syntax

The basic syntax for creating scatterplot in R is −

plot(x, y, main, xlab, ylab, xlim, ylim, axes)

Following is the description of the parameters used −

- x is the data set whose values are the horizontal coordinates.
- y is the data set whose values are the vertical coordinates.
- main is the tile of the graph.
- xlab is the label in the horizontal axis.
- ylab is the label in the vertical axis.
- xlim is the limits of the values of x used for plotting.
- ylim is the limits of the values of y used for plotting.
- axes indicate whether both axes should be drawn on the plot.

Procedure:

I: Inserting Data and make a Data Frame

```
#Draw ScatterPlot for a Nitrate Estimation Calibration
#Enter data
nitrate.conc= c(0,1,2,3,4)
nitrate.OD= c(0,0.471,1.009,1.5,1.963)
#make dataframe
cal.dat= data.frame(x=nitrate.conc, y=nitrate.OD)
cal.dat
```

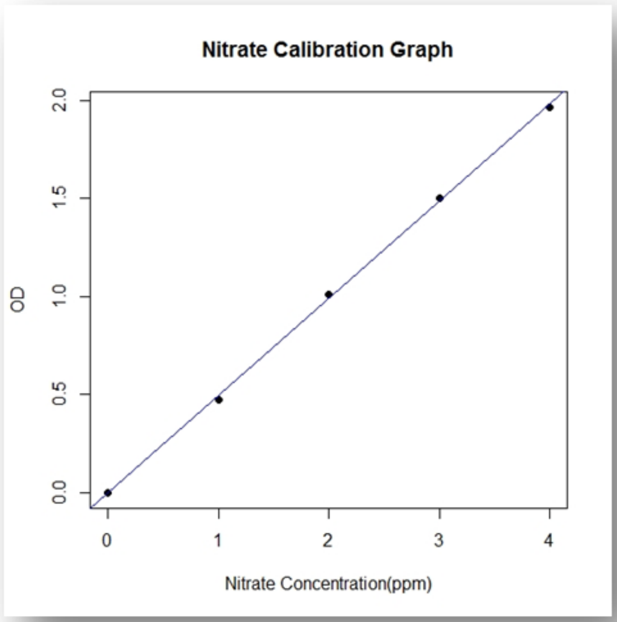
```
> #Draw ScatterPlot for a Nitrate Estimation Calibration
> #Enter data
>nitrate.conc= c(0,1,2,3,4)
>nitrate.OD= c(0,0.471,1.009,1.5,1.963)
> #make dataframe
> cal.dat= data.frame(x=nitrate.conc, y=nitrate.OD)
> cal.dat
  x     y
1 0 0.000
2 1 0.471
3 2 1.009
4 3 1.500
5 4 1.963
```

II. Draw a Scatterplot

```
#Plot with main and axis titles
plot(cal.dat$x,cal.dat$y,
main="Nitrate Calibration Graph",
xlab="Nitrate Concentration(ppm)",
ylab="OD",
pch=19,
frame=TRUE)
abline(lm(y~x,data=cal.dat),
col="blue")
fit=lm(y~x,data=cal.dat)
summary(fit)
> #Plot with main and axis titles
>plot(cal.dat$x,cal.dat$y,
+ main="Nitrate Calibration Graph",
+ xlab="Nitrate Concentration(ppm)",
+ ylab="OD",
+ pch=19,
+ frame=TRUE)
> abline(lm(y~x,data=cal.dat),
+ col="blue")
> fit=lm(y~x,data=cal.dat)
> summary(fit)
Call:
lm(formula = y ~ x, data = cal.dat)
Residuals:
     1       2       3       4       5
0.0024 -0.0221  0.0204  0.0159 -0.0166
Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002400   0.016963  -0.141    0.896
x           0.495500   0.006925  71.552 6.02e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.0219 on 3 degrees of freedom
```

Multiple R-squared: 0.9994,   Adjusted R-squared: 0.9992
F-statistic: 5120 on 1 and 3 DF, p-value: 6.016e-06

<u>Observation</u>:



<u>Conclusion</u>: Here we can see that the nitrogen concentration (ppm) values of five samples are placed in a straight line. There is a positive relation between nitrogen concentration (ppm) values and OD values.

PERFORMANCE OF t-TEST USING 'R'

Aim- To perform t-test in R using provided data.

Principle- A t-test can tell whether two groups have the same mean. A t-test is also called a Student Test. A t-test can be estimated for:

1. A single vector (i.e., one-sample t-test)
2. Two vectors from the same sample group (i.e., paired t-test).

You assume that both vectors are randomly sampled, independent and come from a normally distributed population with unknown but equal variances.

The basic idea behind a t-test is to use statistic to evaluate two contrary hypotheses:

H0: Null hypothesis: The average is the same as the sample used.

HA: Alternative hypothesis: The average is different from the sample used.

The t-test is commonly used with small sample sizes. To perform a t-test, you need to assume the normality of the data.

The basic syntax for t.test() is:

```
t.test(x, y = NULL,
mu = 0, var.equal = FALSE)
arguments:
- x: A vector to compute the one-sample t-test
- y: A second vector to compute the two-sample t-test
- mu: Mean of the population- var.equal: Specify if the variance of the two vectors is equal. By default, set to `FALSE`
```

ONE SAMPLE t-TEST

The t-test, or student's test, compares the mean of a vector against a theoretical mean. The formula used to compute the t-test is:

t = (xbar - mu)/(sigma/sqrt(n))

xbar = sample mean

mu = theoretical mean

sigma = sample standard deviation

n = sample population

To evaluate the statistical significance of the t-test, you need to compute the p-value. The p-value ranges from 0 to 1 and is interpreted as follows:

A p-value lower than 0.05 means you are strongly confident to reject the null hypothesis, thus alternative hypothesis is accepted.

A p-value higher than 0.05 indicates that you don't have enough pieces of evidence to reject the null hypothesis.

You can construct the p-value by looking at the corresponding absolute value of the t-test in the Student distribution with a degree of freedom equals to df = n – 1

For instance, if you have 5 observations, you need to compare our t-value with the t-value in the Student distribution with 4 degrees of freedom and at a 95 per cent confidence interval. To reject the null hypothesis, the t-value should be higher than 2.77.

<u>Problem:</u>

Glucose level (in mg/100ml) of blood sample from a human population is measured as 84, 96, 105, 96, 98, 106, 103, 106, 116 and 89. The normal blood glucose level for average male adult human is 98. Does the glucose level of the group significantly differ from the average population?

<u>Solution:</u>

You can use a one-sample t-test to check whether the level of glucose of the group differ from the population. You can draw a hypothesis test:

H_0: The average blood glucose level of the group does not significantly differ from population average.

H_A: The average blood glucose level of the group differs from thepopulation average.

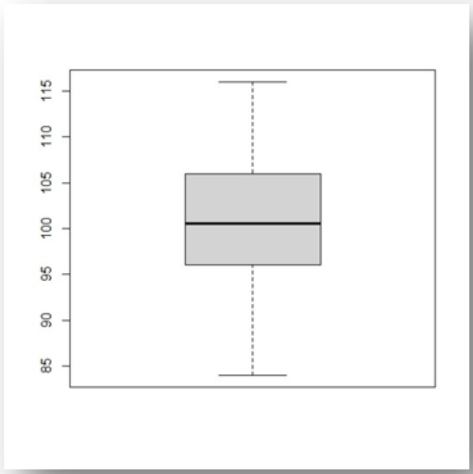
You use a significance level of 0.05.

```
#enter data
bg=c(84,96,105,96,98,106,103,106,116,89)
bg
str(bg)
mean(bg)
summary(bg)
```

```
[1]  84  96 105  96  98 106 103 106 116  89
```

```
num [1:10] 84 96 105 96 98 106 103 106 116 89

 [1] 99.9

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   84.0    96.0   100.5    99.9   105.8   116.0
```

```
#plot a boxplot
boxplot(bg)
```



To check for normality test we perform Shapiro-Wilk normality test. If the p value is >0.05 then accept the null hypothesis. That means data is normally distributed.

```
#Shapiro- Wilk test
shapiro.test(bg)

Shapiro-Wilk normality test
data:  bg
W = 0.96888, p-value = 0.8802
```

From the output, the p-value > 0.05 implying that the distribution of the data is not significantly different from a normal distribution. In other words, we can assume normality.

```
t.bg=t.test(bg,mu=98)
#print the result
t.bg
data:  bg
t = 0.64752, df = 9, p-value = 0.5335
alternative hypothesis: true mean is not equal to 98
95 percent confidence interval:
  93.2622 106.5378
sample estimates:
mean of x
    99.9
```

<u>Results:</u>There was not any significant effect of glucose level, t(9)= 0.65, p = 0.53.

<u>Conclusion:</u>

The p-value of the one-sample t-test is 0.5335 and above 0.05. You can be confident at 95% that the amount of blood glucose level is between 93.2622 and 106.5378 grams. We fail to reject the null hypothesis. The sample blood glucose level does not differ significantly from the population average.

<u>INDEPENDENT t-TEST</u>

The independent t-test also called the two-sample t-test, independent-samples t-test or student's t-test, is an inferential statistical test that determines whether there is a statistically significant difference between the means in two unrelated groups.

Null and alternative hypotheses for the independent t-test.

The null hypothesis for the independent t-test is that the population means from the two unrelated groups are equal:

H0: u1 = u2

In most cases, we are looking to see if we can show that we can reject the null hypothesis and accept the alternative hypothesis, which is that the population means are not equal:

HA: u1 ≠ u2

To do this, we need to set a significance level (also called alpha) that allows us to either reject or accept the alternative hypothesis. Most commonly, this value is set at 0.05.

What do you need to run an independent t-test?

To run an independent t-test, you need the following:

One independent, categorical variable that has two levels/groups.

One continuous dependent variable.

<u>Unrelated groups</u>

Unrelated groups also called unpaired groups or independent groups, are grouped in which the cases (e.g., participants) in each group are different. Often we are investigating differences in individuals, which means that when comparing two groups, an individual in one group cannot also be a member of the other group and vice versa. An example would be gender - an individual would have to be classified as either male or female – not both.

Assumption of normality of the dependent variable

The independent t-test requires that the dependent variable is approximately normally distributed within each group.

Note: Technically, it is the residuals that need to be normally distributed, but for an independent t-test, both will give you the same result.

You can test for this using several different tests, but the Shapiro-Wilks test of normality or a graphical method, such as a Q-Q Plot, is very common. You can run these tests using SPSS Statistics, the procedure for which can be found in our Testing for Normality guide. However, the t-test is described as a robust test concerning the assumption of normality. This means that some deviation away from normality does not have a large influence on Type I error rates. The exception to this is if the ratio of the smallest to largest group size is greater than 1.5 (largest compared to smallest).

Assumption of homogeneity of variance

The independent t-test assumes the variances of the two groups you are measuring are equal in the population. If your variances are unequal, this can affect the Type I error rate. The assumption of homogeneity of variance can be tested using Levene's Test of Equality of Variances

levene test(response variable ~ group variable, data = data)

This test for homogeneity of variance provides an F-statistic and significance value (p-value). We are primarily concerned with the significance value – if it is greater than 0.05 (i.e., $p > .05$), our group variances can be treated as equal. However, if $p < 0.05$, we have unequal variances and we have violated the assumption of homogeneity of variances.

If the Levene's Test for Equality of Variances is statistically significant, which indicates that the group variances are unequal in the population, you can correct for this violation by not using the pooled estimate for the error term for the t-statistic.

- **UNPAIRED 2 SAMPLE t-TEST**

  The unpaired two-sample t-test is used to compare the mean of two independent groups.

  If the p-value is inferior or equal to the significance level 0.05, we can reject the null hypothesis and accept the alternative hypothesis. In other words we can conclude that the mean values of two different groups are significantly different.

  t.test(x,y,alternative= “two.sided”, var.equal=TRUE)

- x,y: numeric vectors
- alternative: the alternative hypothesis. Allowed value is one of “two.sided” (default), “greater” or “less”
- var.equal: a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch test is used

Problem:

Weights of a group of 9 men and 9 women are provided. Use a 2 sample t-test to verify whether the mean weight of men is equal to the mean weight of the women.

Women weight (kg): 38.9, 61.2, 73.3, 21.8, 63.4, 64.6, 48.4, 48.8, 48.5

Men weight (kg): 67.8, 60, 63.4, 76, 89.4, 73.3, 67.3, 61.3, 62.4

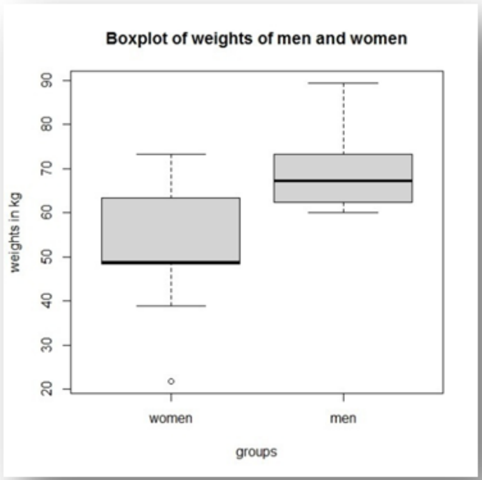
Solution:

Null Hypothesis : The mean weights of the women and men are equal.

Alternative Hypothesis: The mean weights of the women differ significantly from the mean weights of the men.

```
#Entering the data
women_weight=c(38.9,61.2,73.3,21.8,63.4,64.6,48.4,48.8,48.5)
men_weight=c(67.8,60,63.4,76,89.4,73.3,67.3,61.3,62.4)
#create a data frame
my_data=data.frame(group=rep(c("women","men"),each=9),weight=c(women_weight,men_weight))
#print all data
print(my_data)              #my_data
#check data
str(my_data)
   group  weight
1  women   38.9
2  women   61.2
3  women   73.3
4  women   21.8
5  women   63.4
6  women   64.6
7  women   48.4
8  women   48.8
9  women   48.5
10   men    67.8
11   men    60.0
12   men    63.4
13   men   76.0
14   men   89.4
15   men   73.3
```

```
16   men    67.3
17   men    61.3
18   men    62.4

'data.frame':   18 obs. of  2 variables:
 $ group: chr  "women" "women" "women" "women" ...
 $ weight: num  38.9 61.2 73.3 21.8 63.4 64.6 48.4 48.8 48.5 67.8 ...

#calculate mean and standard deviation of each group
mean(women_weight)            #find mean value
sd(women_weight)              #find standard deviation value
mean(men_weight)              #find mean value
sd(men_weight)                #find standard deviation value

> mean(women_weight)            #find mean value
[1] 52.1
> sd(women_weight)              #find standard deviation value
[1] 15.59671
> mean(men_weight)              #find mean value
[1] 68.98889
> sd(men_weight)                #find standard deviation value
[1] 9.375426

#check data
boxplot(women_weight,men_weight,main="Boxplot of weights of men and women",
xlab="groups",
ylab="weights in kg",
names=c("women","men")
)
```



```
#Shapiro-Wilk test for normalitry
shapiro.test(women_weight)

Shapiro-Wilk normality test
data:  women_weight
W = 0.94266, p-value = 0.6101
shapiro.test(men_weight)

  Shapiro-Wilk normality test
data:  men_weight
W = 0.86425, p-value = 0.1066

#Test for Homoscedaticicity
res.ftest=var.test(weight~group,data=my_data)
res.ftest


F test to compare two variances
data:  weight by group
F = 0.36134, num df = 8, denom df = 8, p-value = 0.1714
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.08150656 1.60191315
sample estimates:
ratio of variances
        0.3613398

#compute t-test(2 tailed)
t.res=t.test(women_weight,men_weight,var.equal=TRUE)
t.res

       Two Sample t-test

data:  women_weight and men_weight
t = -2.7842, df = 16, p-value = 0.01327
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -29.748019  -4.029759
sample estimates:
mean of x mean of y
52.10000  68.98889

#Compute t.test(1 tailed,less)
t1.res=t.test(women_weight,men_weight,var.equal=TRUE,alternative="less")
t1.res
  Two Sample t-test
data:  women_weight and men_weight
```

```
t = -2.7842, df = 16, p-value = 0.006633
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -6.298536
sample estimates:
mean of x mean of y
52.10000  68.98889
```

Results: Results of the independent sample t-tests indicated that there were women's weight is significantly less than men's weight, t(16)= -2.78, p< 0.001.

Conclusion:

Women's weight (mean= 52.1, std. dev= 15.6) is significantly different (alpha= 0.05) from men's weight (mean= 68.99, std. dev= 9.37) as per t.test (alpha= 0.05, df= 16, t= -2.78, p= 0.0137). One tailed t.test (alpha= 0.05, df= 16, t= -2.78, p= 0.0066) showed that women's weight is significantly less than men's weight.

- **PAIRED SAMPLE t-TEST**

The paired samples t-test is used to compare the means between two related groups of samples. In this case, you have two values (i.e. pair of values) for the same samples.

If the p-value is inferior or equal to 0.05, we can conclude that the difference between the two paired samples is significantly different.

To perform paired samples t-test comparing the means of two paired samples (x & y), the R function t.test() can be used follow:

t.test(x,y, paired= TRUE, alternative= "two.sided")

- x,y: numeric vectors
- paired: a logical value specifying that we want to compute a paired t-test
- alternative: the alternative hypothesis. Allowed value is one of "two.sided" (default), "greater" or "less"

Preliminary test to check paired t-test assumptions

Assumption 1: Are the two samples paired?

Yes, data has been collected from same sample twice.

Assumption 2: is this a large sample?

No, n < 30

Is the data normally distributed?

Use Shapiro-Wilk test to check for normality for the difference of data, a p-value greater than or equal to 0.05 will fail to reject the null hypothesis, i.e., we accept the null hypothesis, data is normally distributed.

The result of t.test() function is a list containing the following components:

- **statistic:** the value of the t test statistics
- **parameter:** the degrees of freedom for the t test statistics
- **p.value:** the p-value for the test
- **conf.int:** a confidence interval for the mean appropriate to the specified alternative hypothesis
- **estimate:** the means of the groups being compared 9in the case of independent t test) or difference in means (in the case of paired t test)

Problem:

The weight of 10 mice has been measured before and after the treatment during 3 months.

Solution:

```
#Data in two numeric vectors
#Weight of the mice before treatment
before=c(200.1,190.9,192.7,213,241.4,196.9,172.2,185.5,205.2,193.7)
#Weight of the mice after treatment
after=c(392.9,393.2,345.1,393,434,427.9,422,383.9,392.3,352.2)
#Create a data frame
my_data=data.frame(group=rep(c("before","after"),each=10),weight=c(before,after))
my_data
```

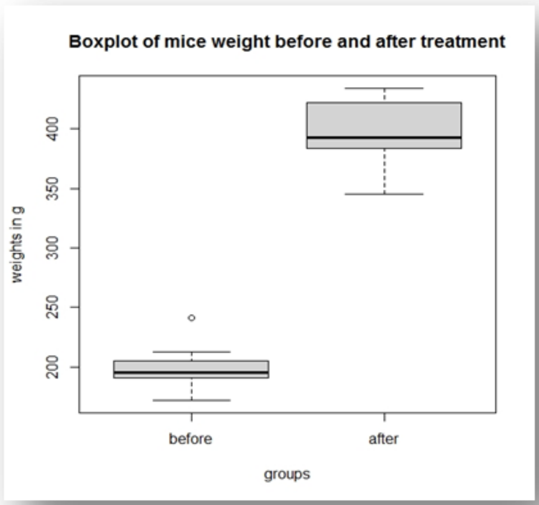
```
group  weight
1  before  200.1
2  before  190.9
3  before  192.7
4  before  213.0
5  before  241.4
6  before  196.9
7  before  172.2
8  before  185.5
9  before  205.2
10 before 193.7
11  after   392.9
12  after  393.2
13  after  345.1
14  after  393.0
15  after  434.0
16  after  427.9
17  after  422.0
18  after  383.9
```

```
19  after  392.3
20  after  352.2
```

```
print(paste("mean before=",mean(before)))
print(paste("mean after=",mean(after)))
print(paste("standard deviation before=",sd(before)))
print(paste("standard deviation after=",sd(after)))
```

```
>print(paste("mean before=",mean(before)))
[1] "mean before= 199.16"
>print(paste("mean after=",mean(after)))
[1] "mean after= 393.65"
>print(paste("standard deviation before=",sd(before)))
[1] "standard deviation before= 18.4735366282571"
>print(paste("standard deviation after=",sd(after)))
[1] "standard deviation after= 29.3980063571967"
```

```
#Draw boxplot
boxplot(before,after,main="Boxplot of mice weight before and after treatment",
xlab="groups",
ylab="weights in g",
names=c("before","after"))
```



```
#Check for normality
d=before-after
shapiro.test(d)
```

```
  Shapiro-Wilk normality test
data:  d
W = 0.94536, p-value = 0.6141
```

```
#Compute t-test
t.res=t.test(before,after,paired=TRUE,alternative="two.sided")
t.res
```

```
Paired t-test

data:  before and after

t = -20.883, df = 9, p-value = 6.2e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -215.5581 -173.4219
sample estimates:
mean of the differences
            -194.49
```

Results: Results of the dependent (paired) sample t-tests indicated that the average weight of mice before treatment is significantly different from the average weight aftertreatmentduring 3 months, t(9)= -20.88, p< 0.001.

Conclusion:
The t-value of the test is -20.88 (df= 9), which is less than the significance level at alpha= 0.05. We can the reject the null hypothesis and conclude that the average weight of the mice before (mean= 199.16, std. dev= 18.47) treatment is significantly different from the average weight after (mean= 393.65, std. dev= 20.40) treatment with p-value= $6.2e^{-09}$.

PERFORMANCE OF ANOVA USING 'R'

Aim- To perform one-way ANOVA in R using provided data

Principle- ANOVA -short for "analysis of variance"- is a statistical technique for testing if 3(+) population means are all equal.

The two simplest scenarios are:

1. one-way ANOVA:- for comparing 3(+) groups on 1 variable: do all children from school A, B and C have equal mean IQ scores?
2. repeated measures ANOVA:- for comparing 3(+) variables in 1 group: is the mean rating for movie A, B and C equal for all people?

Simple Example - One-Way ANOVA

A scientist wants to know if all children from schools A, B and C have equal mean IQ scores. Each school has 1,000 children. It takes too much time and money to test all 3,000 children. So a simple random sample of n = 10 children from each school are tested.

Descriptives Table

Right, so our data contain 3 samples of 10 children each with their IQ scores. Running a simple descriptives table immediately tells us the mean IQ scores for these samples. The result is shown below.

Our sample from school B has the highest mean IQ - roughly 113 points. The lowest mean IQ -some 93 points- is seen for school C.

Now, here's the problem: our mean IQ scores are only based on tiny samples of 10 children per school. So couldn't it be that all 1,000 children per school have the same mean IQ?

Perhaps we just happened to sample the smartest children from school B and the dumbest children from school C?* Is that realistic? We'll try and show that this statement -our null hypothesis- is not credible given our data.

ANOVA - Null Hypothesis

The null hypothesis for (any) ANOVA is that all population means are exactly equal.

If this holds, then our sample means will probably differ a bit. After all, samples always differ a bit from the populations they represent. However, the sample means probably shouldn't differ too much. Such an outcome would be unlikely under our null hypothesis of equal population means. So if we do find this, we'll probably no longer believe that our population means were equal.

ANOVA - Sums of Squares Between

So precisely how different are our 3 sample means? How far do these numbers lie apart? A number that tells us just that is the variance. So we'll compute the variance among our 3 sample means.

As you may (or may not) understand from the ANOVA formulas, this starts with the sum of the squared deviations between the 3 sample means and the overall mean. The outcome is known as the "sums of squares between" or SSbetween. So sums of squares between expresses the total amount of dispersion among the sample means.

Everything else equal, larger SSbetween indicates that the sample means differ more. And the more different our sample means, the more likely that our population means differ as well.

Degrees of Freedom and Mean Squares Between

When calculating a "normal" variance, we divide our sums of squares by its degrees of freedom (df). When comparing k means, the degrees of freedom (df) is (k - 1). Dividing SSbetween by (k - 1) results in mean squares between MSbetween. In short, mean squares between is the variance among sample means.

MSbetween thus indicates how far our sample means differ (or lie apart). The larger this variance between means, the more likely that our population means differ as well. So from SSbetween, we go to MSbetween by dividing by number of schools - 1 in this case, ie by 3-1 = 2

variance, sigma^2 = (x - xbar)^2 / (N-1)

ANOVA - Sums of Squares Within

If our population means are equal, then what difference between sample means -MSbetween- can we reasonably expect? Well, this depends on the variance within subpopulations. The figure below illustrates this for 3 scenarios.

The 3 leftmost histograms show population distributions for IQ in schools A, B and C. Their narrowness indicates a small variance within each school. If we'd sample n = 10 students from each school, should we expect very different sample means?

Probably not. Why? Well, due to the small variance within each school, the sample means will be close to the (equal) population means. These narrow histograms don't leave a lot of room for their sample means to fluctuate and -hence- differ. The 3 rightmost histograms show the opposite scenario: the histograms are wide, indicating a large variance within each school. If we'd sample n = 10 students from each school, the means in these samples may easily differ quite a lot. In short, larger variances within schools probably result in a larger variance between sample means per school. We estimate the within-groups population variances from the within-groups sample variances. Makes sense, right? The exact calculations are in the ANOVA formulas

In short: sums of squares within (SSwithin) indicates the total amount of dispersion within groups; degrees of freedom within (DFwithin) is (n - k) for n observations and k groups and mean squares within (MSwithin) -basically the variance within groups- is SSwithin / DFwithin. ANOVA Test Statistic – F So how likely are the population means to be equal? This depends on 3 pieces of information from our samples:

> the variance between sample means (MSbetween);

> the variance within our samples (MSwithin) and

> the sample sizes.

We combine all this information into a single number: our test statistic F. The diagram below shows how each piece of evidence impacts F.

<u>ANOVA - Assumptions</u>

The assumptions for ANOVA are:

1. Independent observations;
2. Normality: the outcome variable must follow a normal distribution in each subpopulation. Normality is only needed for small sample sizes, say n < 20 per group.
3. Homogeneity: the variances within all subpopulations must be equal. Homogeneity is only needed if sample sizes are very unequal. In this case, Levene's test indicates if it's met.

If these assumptions hold, then F follows an F-distribution with DFbetween and DFwithin degrees of freedom. In our example -3 groups of n = 10 each- that'll be F(2,27).

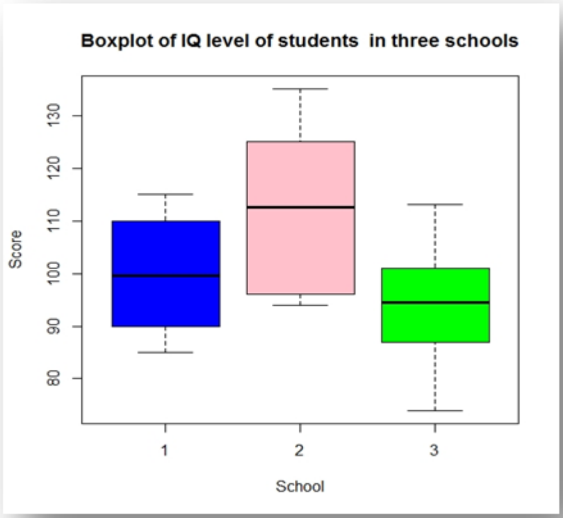
<u>Problem:</u>

A scientist wants to know if all children from schools A, B and C have equal mean IQ scores. Each school has 1,000 children. It takes too much time and money to test all 3,000 children. So a simple random sample of n = 10 children from each school are tested. Here A = 1, B = 2, C= 3 in Group, Score contains the 10 corresponding data.

<u>Solution:</u>

```
#enter data
Score= c(90, 87, 93, 115, 97, 85, 102, 110, 111, 102, 135, 125, 107, 96, 114, 125, 94, 123, 111, 96, 93,
101, 74, 87, 76, 87, 98, 108, 113, 96)
```

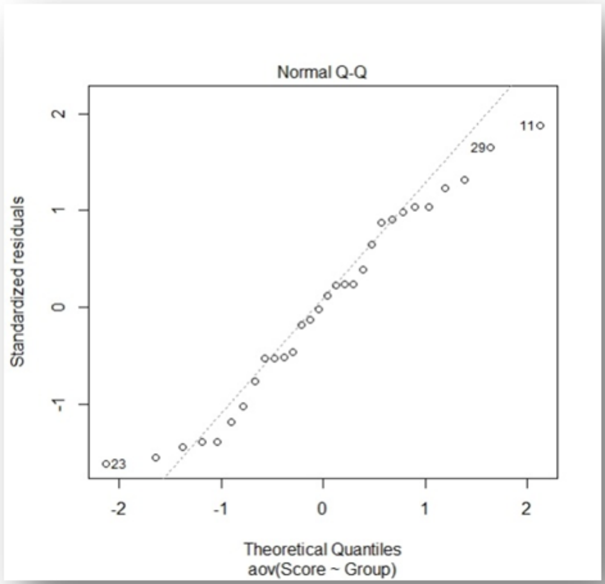
```
Group=c(rep(1,10),rep(2,10),rep(3,10))
my.dat=data.frame(Group,Score)
my.dat]

  Group Score
1     1    90
2     1    87
3     1    93
4     1   115
5     1    97
6     1    85
7     1   102
8     1   110
9     1   111
10    1   102
11    2   135
12    2   125
13    2   107
14    2    96
15    2   114
16    2   125
17    2    94
18    2   123
19    2   111
20    2    96
21    3    93
22    3   101
23    3    74
24    3    87
25    3    76
26    3    87
27    3    98
28    3   108
29    3   113
30    3    96
#check data structure
str(my.dat)

'data.frame':   30 obs. of  2 variables:
 $ Group: num  1 1 1 1 1 1 1 1 1 1 ...
 $ Score: num  90 87 93 115 97 85 102 110 111 102 ...

#convert group to factor from number
my.dat$Group=factor(my.dat$Group)
str(my.dat)

'data.frame':   30 obs. of  2 variables:
 $ Group: Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ Score: num  90 87 93 115 97 85 102 110 111 102 ...

#visualize thr data using Boxplot
boxplot(Score~Group,
data=my.dat,
col=c("BLUE","PINK","GREEN"),
order=c(1,2,3),
main="Boxplot of IQ level of students  in three schools",
ylab="Score",
xlab="School")
```



```
#perform one-way ANOVA
res.aov=aov(Score~Group,data=my.dat)
#observe the result
summary(res.aov)

            Df Sum Sq Mean Sq F value Pr(>F)
Group        2   1956   978.1    6.15 0.0063 **
Residuals   27   4294   159.0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

#Tukey multiple pairwise-comparisons for our data
TukeyHSD(res.aov)

Tukey multiple comparisons of means
   95% family-wise confidence level

Fit: aov(formula = Score ~ Group, data = my.dat)

$Group
     diff        lwr       upr     p adj
2-1  13.4  -0.5835966 27.383597 0.0622882
```

```
3-1  -5.9 -19.8835966  8.083597 0.5549304
3-2 -19.3 -33.2835966 -5.316403 0.0054803
```

```
#check ANOVA assumptions:test validity?
#1.Homogeneity of variances
plot(res.aov,1)
bartlett.test(Score~Group,data=my.dat)
Bartlett test of homogeneity of variance
data:  Score by Group
Bartlett's K-squared = 0.80105, df = 2, p-value = 0.67
library(car)
leveneTest(Score~Group,data=my.dat)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group 2  0.5349 0.5918
      27
#2.Normality
plot(res.aov,2)
```



<u>Results:</u>Results of the ANOVA indicated that there was a significant main effect between students and their IQ level, F(2,27)= 6.15, p> 0.001.

<u>Conclusion:</u>

The model summary first lists the independent variables being tested in the model (in this case we have only one, ‘Scores’) and the model residuals (‘Residual’). All of the variations that are not explained by the independent variables is called residual variance.

The Df column displays the degrees of freedom for the independent variable (the number of levels in the variable minus 1), and the degrees of freedom for the residuals (the total number of observations minus one and minus the number of levels in the independent variables).

The Sum Sq column displays the sum of squares (a.k.a. the total variation between the group means and the overall mean).

The Mean Sq column is the mean of the sum of squares, calculated by dividing the sum of squares by the degrees of freedom for each parameter.

The F-value column is the test statistic from the F test. This is the mean square of each independent variable divided by the mean square of the residuals. The larger the F value, the more likely it is that the variation caused by the independent variable is real and not due to chance.

The Pr(>F) column is the p-value of the F-statistic. This show how likely it is that the F-value calculated from the test would have occurred if the null hypothesis of no difference among group means were true.

As the p-value is less than the significance level 0.05, we can conclude that there are significant differences between the groups highlighted with “*” in the model summary.

Multiple pairwise-comparison between the means of groups

In one-way ANOVA test, a significant p-value indicates that some of the group means are different, but we don’t know which pairs of groups are different.

It’s possible to perform multiple pairwise-comparisons, to determine if the mean difference between specific pairs of group is statistically significant.

Tukey multiple pairwise-comparisons

As the ANOVA test is significant, we can compute Tukey HSD(Tukey Honest Significant Differences, R function: TukeyHSD ()) for performing multiple pairwise-comparison between the means of groups.

The function TukeyHD () takes the fitted ANOVA as an argument.

#Tukey multiple pairwise-comparisons for our data

1. diff: difference between means of the groups
2. lwr,upr: the lower and the upper end point of the confidence interval at 95%(default)

P adj: p-value after adjustment for the multiple comparisons.

Check the ANOVA assumptions: test validity?

Check the homogeneity of variance assumption

The residuals versus fits plot can be used to check the homogeneity of variances.

1. Homogeneity of variances

In the plot below, there is no evident relationships between residuals and fitted values (the mean of each groups), which is good. So, we can assume the homogeneity of variances.
Barlett test shows us that the variances are homogenous (i.e. a non-significant P value). The reason we may not use a Barlett's test all of the time is because it is highly sensitive to departures from normality (i.e. non-normal datasets). If we suspect our data is not-normal or is slightly not-normal and want to test homogeneity of variances anyways, we can use a Levene's Test to account for this.

2. Check the normality assumption

Normality plot of residuals. In the plot below, the quantiles of the residuals are plotted against the quantiles of the normal distribution. A 45 degree reference line is also plotted.
The normal probability plot of residuals is used to check the assumption that the residuals are normally distributed. It should approximately follow a straight line.

<u>**Two -Way ANOVA**</u>

A two-way ANOVA is used when you have two or more independent variables. A two-way ANOVA is the ANOVA you use when you have two or more independent variables with multiple conditions. A one-way ANOVA is used when you have one independent variable with multiple conditions.

<u>**Problem:**</u>

To determine the effects of different types of fertilizer and the frequency with which you water the tree on the number of fruits produced. Your two independent variables are 1) fertilizer type and 2) frequency tree is water. In the two-way ANOVA example, we are modelling crop yield as a function of the type of fertilizer and planting density. First, we use aov () to run the model, then we use summary () to print the summary of the model.

<u>**Solution;**</u>

```
# enter the data
density = c(rep(1:2,48))
 block = c(rep(1:4,24))
fertilizer = c(rep(1,32),rep(2,32),rep(3,32))
yield = c(177.228692278688, 177.550041265426, 176.408461852371, 177.703625478918,
177.125486343584, 176.778342481435, 176.746301895632, 177.061164221059, 176.274949297277,
177.967202929576, 176.601299834494, 177.030542802918, 177.479507160588, 176.874129801452,
176.114388315927, 176.008394511531, 176.108312589654, 178.35744091425, 177.262445084271,
176.918844938972, 176.239015775225, 176.573069754873, 176.039297939843, 176.817922165499,
176.160586502816, 177.226424131397, 175.938533030004, 177.16493668281, 175.360839598647,
177.276995681026, 175.945443795425, 175.882779618556, 176.479340919814, 176.0443421205,
177.412461749009, 177.360818239535, 177.385499180836, 176.975807696393, 177.379778689577,
177.99799506429, 176.434862565867, 176.933265092142, 175.983480169992, 177.034092659355,
176.436762368181, 176.06774497035, 177.121048632786, 177.197721367386, 176.60372408049,
177.208171431058, 177.148828595107, 176.819076695268, 176.999066949946, 178.134604582348,
176.429156002757, 176.668322935082, 176.895866859859, 177.779492860266, 176.414495002246,
176.878897743394, 177.580683078112, 176.957268920193, 175.747545582883, 177.352595079015,
177.104186398865, 178.079635168278, 176.903422145535, 177.540284161493, 177.03270969652,
178.28604192157, 176.405410230471, 176.430830126686, 177.396330635021, 176.92557577835,
177.055045778695, 177.344163946335, 177.128367531761, 177.168302204533, 176.353940642858,
179.060899036915, 176.300517050366, 177.59335237605, 177.115245241837, 177.794457435485,
177.004038102495, 178.036858366448, 177.701366283038, 177.632808263613, 177.652274608017,
177.100417857738, 177.187967031073, 177.405291855943, 178.14164435679, 177.710612540335,
177.687264356834, 177.118175977614)
crop.data
=data.frame(density=as.factor(density),block=as.factor(block),fertilizer=as.factor(fertilizer),yield)
summary(crop.data)
```

data.frame is making the vectors into a data frame, and, density=as.factor(density) this part is telling R that the data in density is to be considered as a factor and not numerical. Summary() gives the 5 point descriptive stats of the data.

```
density block fertilizer    yield
1       1     1         1 177.2287
2       2     2         1 177.5500
3       1     3         1 176.4085
4       2     4         1 177.7036
5       1     1         1 177.1255
6       2     2         1 176.7783
7       1     3         1 176.7463
8       2     4         1 177.0612
```

9	1	1	1	176.2749
10	2	2	1	177.9672
11	1	3	1	176.6013
12	2	4	1	177.0305
13	1	1	1	177.4795
14	2	2	1	176.8741
15	1	3	1	176.1144
16	2	4	1	176.0084
17	1	1	1	176.1083
18	2	2	1	178.3574
19	1	3	1	177.2624
20	2	4	1	176.9188
21	1	1	1	176.2390
22	2	2	1	176.5731
23	1	3	1	176.0393
24	2	4	1	176.8179
25	1	1	1	176.1606
26	2	2	1	177.2264
27	1	3	1	175.9385
28	2	4	1	177.1649
29	1	1	1	175.3608
30	2	2	1	177.2770
31	1	3	1	175.9454
32	2	4	1	175.8828
33	1	1	2	176.4793
34	2	2	2	176.0443
35	1	3	2	177.4125
36	2	4	2	177.3608
37	1	1	2	177.3855
38	2	2	2	176.9758
39	1	3	2	177.3798
40	2	4	2	177.9980

41	1	1	2 176.4349
42	2	2	2 176.9333
43	1	3	2 175.9835
44	2	4	2 177.0341
45	1	1	2 176.4368
46	2	2	2 176.0677
47	1	3	2 177.1210
48	2	4	2 177.1977
49	1	1	2 176.6037
50	2	2	2 177.2082
51	1	3	2 177.1488
52	2	4	2 176.8191
53	1	1	2 176.9991
54	2	2	2 178.1346
55	1	3	2 176.4292
56	2	4	2 176.6683
57	1	1	2 176.8959
58	2	2	2 177.7795
59	1	3	2 176.4145
60	2	4	2 176.8789
61	1	1	2 177.5807
62	2	2	2 176.9573
63	1	3	2 175.7475
64	2	4	2 177.3526
65	1	1	3 177.1042
66	2	2	3 178.0796
67	1	3	3 176.9034
68	2	4	3 177.5403
69	1	1	3 177.0327
70	2	2	3 178.2860
71	1	3	3 176.4054
72	2	4	3 176.4308

```
73     1    1       3 177.3963
74     2    2       3 176.9256
75     1    3       3 177.0550
76     2    4       3 177.3442
77     1    1       3 177.1284
78     2    2       3 177.1683
79     1    3       3 176.3539
80     2    4       3 179.0609
81     1    1       3 176.3005
82     2    2       3 177.5934
83     1    3       3 177.1152
84     2    4       3 177.7945
85     1    1       3 177.0040
86     2    2       3 178.0369
87     1    3       3 177.7014
88     2    4       3 177.6328
89     1    1       3 177.6523
90     2    2       3 177.1004
91     1    3       3 177.1880
92     2    4       3 177.4053
93     1    1       3 178.1416
94     2    2       3 177.7106
95     1    3       3 177.6873
96     2    4       3 177.1182

density   block  fertilizer    yield
   1:48     1:24   1:32       Min.   :175.4
    2:48     2:24   2:32      1st Qu.:176.5
            3:24    3:32      Median :177.1
            4:24               Mean   :177.0
                             3rd Qu.:177.4
                              Max.   :179.1
```

Now two way shows that crop yield varies both on type of fertilizer use and density of crops. Now the question arises is there a relation between fertilizer used and density that we are not seeing. Adding interactions between variables. Sometimes you have reason to think that two of your independent variables have an interaction effect rather than an additive effect. For example, in our crop yield experiment, planting density may affect the plants' ability to take up fertilizer. This might influence the effect of fertilizer type in a way that isn't accounted for in the two-way model.

```
one.way = aov(yield ~ fertilizer, data = crop.data)
summary(one.way)
Df Sum Sq Mean Sq F value Pr(>F)
fertilizer   2   6.07  3.0340   7.863  7e-04 *
Residuals   93  35.89  0.3859
---
Signif. codes:  0 '*' 0.001 '*' 0.01 '' 0.05 '.' 0.1 ' ' 1
two.way = aov(yield ~ fertilizer + density, data = crop.data)
summary(two.way)
            Df Sum Sq Mean Sq F value   Pr(>F)
fertilizer   2  6.068   3.034   9.073 0.000253 *
density      1  5.122   5.122  15.316 0.000174 *
Residuals   92 30.765   0.334
---
Signif. codes:  0 '*' 0.001 '*' 0.01 '' 0.05 '.' 0.1 ' ' 1
```

To test whether two variables have an interaction effect in ANOVA, simply use an asterisk instead of a plus-sign in the model:

```
interaction= aov(yield ~ fertilizer*density, data = crop.data)
summary(interaction)
                   Df Sum Sq Mean Sq F value   Pr(>F)
fertilizer          2  6.068   3.034   9.001 0.000273 ***
density             1  5.122   5.122  15.195 0.000186 ***
fertilizer:density  2  0.428   0.214   0.635 0.532500
Residuals          90 30.337   0.337
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
blocking = aov(yield ~ fertilizer + density + block, data = crop.data)
summary(blocking)
Df Sum Sq Mean Sq F value   Pr(>F)
fertilizer   2  6.068   3.034   9.018 0.000269 *
density      1  5.122   5.122  15.224 0.000184 *
block        2  0.486   0.243   0.723 0.488329
Residuals   90 30.278   0.336
---
Signif. codes:  0 ‘*’ 0.001 ‘*’ 0.01 ‘’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The ‘block’ variable has a low sum-of-squares value (0.486) and a high p-value (p = 0.48), so it’s probably not adding much information to the model. It also doesn’t change the sum of squares for the two independent variables, which means that it’s not affecting how much variation in the dependent variable they explain.

```
library(AICcmodavg)
model.set = list(one.way, two.way, interaction, blocking)
model.names = c("one.way", "two.way", "interaction", "blocking")
aictab(model.set, modnames = model.names)
Model selection based on AICc:
            K   AICc Delta_AICc AICcWt Cum.Wt     LL
two.way     5 173.86       0.00   0.71   0.71 -81.59
blocking    7 176.93       3.08   0.15   0.86 -80.83
interaction 7 177.12       3.26   0.14   1.00 -80.92
one.way     4 186.41      12.56   0.00   1.00 -88.99
```

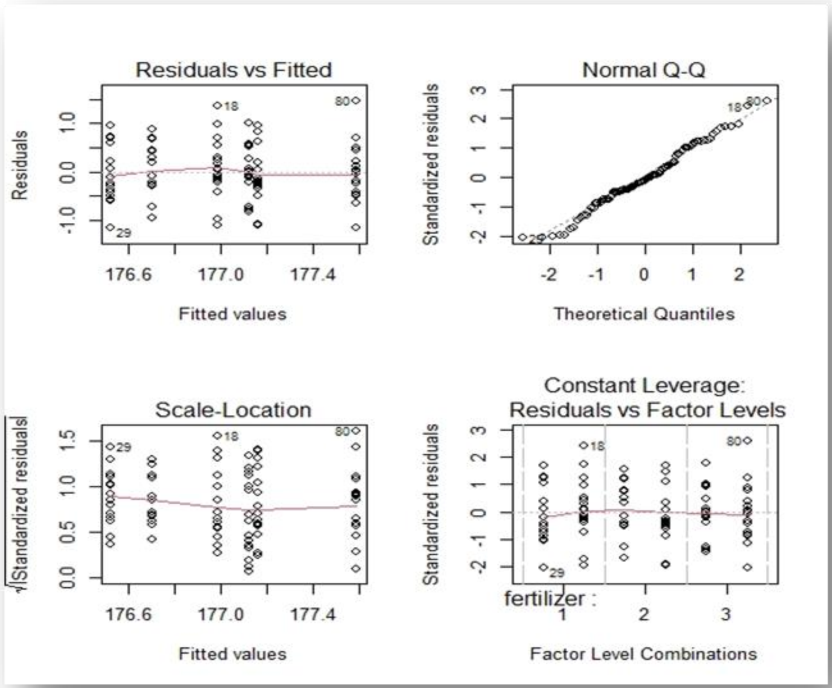
From these results, it appears that the two.way model is the best fit. The two-way model has the lowest AIC value, and 71% of the AIC weight, which means that it explains 71% of the total variation in the dependent variable that can be explained by the full set of models.

The model with blocking term contains an additional 15% of the AIC weight, but because it is more than 2 delta-AIC worse than the best model, it probably isn’t good enough to include in your results.

To check whether the model fits the assumption of homoscedasticity, look at the model diagnostic plots in R using the plot() function:

```
par(mfrow=c(2,2))
```

```
plot(two.way)
par(mfrow=c(1,1))

#Tukey's Honestly Significant Difference(TUkey's HSD) post-hoc test for pairwise comparisons:
tukey.two.way=TukeyHSD(two.way)
Tukey multiple comparisons of means  95% family-wise confidence level
Fit: aov(formula = yield ~ fertilizer + density, data = crop.data)
$fertilizer
        diff        lwr       upr     p adj
2-1 0.1761687 -0.16822506 0.5205625 0.4452958
3-1 0.5991256  0.25473179 0.9435194 0.0002219
3-2 0.4229569  0.07856306 0.7673506 0.0119381
$density
       diff      lwr       upr     p adj
2-1 0.461956 0.2275204 0.6963916 0.0001741
```



Results: Results of the two-way ANOVA indicated that there was a significant main effect in fertilizer type, F(2,92)= 9.07, p < 0.001 and planting density, F(1,92)= 15.32, p < 0.001 but there was no any significant interaction between fertilizer type and planting density i.e. there was no effect of fertilizer type on density of plants, F(2,90)= 0.64, p = .53.

Conclusion:

We found a statistically-significant difference in average crop yield by both fertilizer type (F(2)=9.018, p < 0.001) and by planting density (F(1)=15.316, p<0.001).

A Tukey post hoc test revealed that fertilizer mix 3 resulted in a higher yield on average than fertilizer mix 1, and a higher yield on average than fertilizer mix 2. Planting density was also significant, with planting density 2 resulting in a higher yield over planting density 1. A subsequent GroupWise comparison showed the strongest yield gains at planting density 2, fertilizer mix 3, suggesting that this mix of treatments was most advantageous for crop growth under our experimental conditions.

PERFORMANCE OF CORRELATION USING ‘R’

Aim- To perform correlation in R using provided data

Principle- What is correlation test?

Correlation test is used to evaluate the association between two or more variables.

For instance, if we are interested to know whether there is a relationship between the heights of fathers and sons, a correlation coefficient can be calculated to answer this question.

If there is no relationship between the two variables (father and son heights), the average height of son should be the same regardless of the height of the fathers and vice versa.

Here, we will describe the different correlation methods and we will provide practical examples using R software.

Install and load required R packages

We will use the ggpubr R packagefor easy ggplot-2 based data visualization

- Install from CRAN as follow:
  install. packages(“ggpubr”)
  install.packages(“ggplot2”)
- Load ggpubr as follow:
  library(“ggplot2”)
  library(“ggpubr”)

Methods for correlation analysis

There are different methods to perform correlation analysis:

- **Pearson correlation (r)**, which measures a linear dependence between two variables (x and y). It is also known as a **parametric correlation** test because it depends to the distribution of the data. It can be used only when x and y are from normal distribution. The plot of y= f(x) is named the **linear regression** curve.
- **Kendall tau** and **Spearman rho**, which are rank-based correlation coefficients (nonparametric).

The most commonly used method is a **Pearson correlation** method.

Correlation formula

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}}$$

In the formula below,

- x and y are two vectors of length n
- mx and my corresponds to the means of x and y, respectively.

Pearson Correlation Formula

The p-value (significance level) of the correlation can be determined:

1. by using the correlation coefficient table for the degrees of freedom: df=n-2df=n-2, where nn is the number of observation in x and y variables.
2. or by calculating the **t value** as follow:

$$t = \frac{r \times \sqrt{n-2}}{\sqrt{1-r^2}}$$

In the case (2) the corresponding p-value is determined using **t distribution table**for df=n-2df=n-2
If the p-value is <5%, then the correlation between x and y is significant.

Spearman Correlation Formula
The **Spearman correlation** method computes the correlation between the rank of x and the rank of y variables.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Where,

r = Pearson Correlation Coefficient

x_i = x variable samples $\quad$ y_i = y variable sample

$\bar{x}$ = mean of values in x variable $\quad$ $\bar{y}$ =mean of values in y variable

Kendall Correlation Formula

The **Kendall correlation** method measures the correspondence between the ranking of x and y variables. The total number of possible pairings of x and y observations is n(n-1)/2n(n-1)/2, in where n is a size of x and y.

The process is as follow:

- Begin by ordering the pairs by the x values. If x and y are correlated, then they would have the same relative rank orders.
- Now, for each yi, count the number of yj>yiyj>yi (**concordant pairs (c)**) and the number of yj<yiyj<yi (**discordant (d)**).

Kendall correlation distance is defined as follow:

$$\tau = \frac{2(n_c - n_d)}{n(n-1)}$$

Where,

- nc: total number of concordant pairs
- nd: total number of discordant pairs
- n: size of x and y

Compute correlation in R

R functions

Correlation coefficient can be computed paid using the functions**cor()** or **cor.test()**:

- **cor()** computes the **correlation coefficient**
- **cor.test()** test for association/ correlation between paired samples. It returns both **correlation coefficient** and the **significance level** (or p-value) or the correlation.
  The simplified formants are:

```
cor(x, y, method=c("pearson", "kendall", "spearman"))
cor.test(x, y, method=c("pearson", "kendall", "spearman"))
```

- **x, y**: numeric vectors with the same length
- **method**: correlation method

If your data contain missing values, use the following R code to handle missing values by case-wise deletion.

```
cor(x , y, method= “pearson”, use= “complete.obs”)
```

Pearson Correlation Test

```
res= cor.test(dat$x, dat$y, method= “pearson”)
res
```

n the result above:

- **t** is the **t-test statistic** value,
- **df** is the degrees of freedom,

- **p-value** is the significance level of the **t-test**.
- **conf.int** is the **confidence interval** of the correlation coefficient and 95%
- **sample estimates**is the correlation coefficient

Kendall Rank Correlation Test

The **Kendall rank correlation coefficient** or **Kendall's tau**statistic is used to estimate a rank-based measure of association which test may be used if the data do not necessarily come from a bivariate normal distribution.

```
res2= cor.test(dat$x, dat$y, method= “kendall”)
res2
```

Spearman Rank Correlation Coefficient

Spearman's rho statistic is also used to estimate a rank-based measure of association. This test may be used if the data do not come from a bivariate normal distribution.

```
res3= cor.test(dat$x, dat$y, method= “spearman”)
res3
```

Interpret correlation coefficient

Correlation coefficient is compromise between **-1** and **1:**

- **-1** indicates a strong **negativecorrelation:** this means that every time x **increases** and y **decreases**
- **0** means that there is no **association** between the two variables (x and y)
- **1** indicates a strong **positive correlation:** this means that y **increases** with **x**

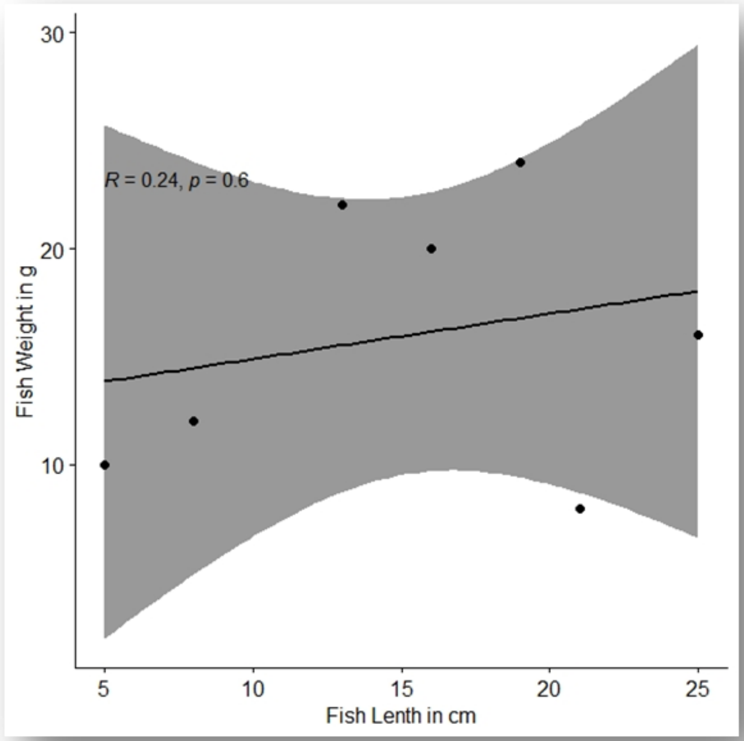
<u>Example</u>:

```
#Enter Data for correlation
fish.length= c(8,13,5,25,21,19,16)
fish.weight= c(12,22,10,16,8,24,20)
dat=data.frame(fish.length, fish.weight)
dat

# Perform Correlation Analysis
res1 = cor.test(dat$fish.length, dat$fish.weight, method= "pearson")
res2 = cor.test(dat$fish.length, dat$fish.weight, method= "kendall")
res3 = cor.test(dat$fish.length, dat$fish.weight, method= "spearman")
# Report Result
res1
res2
res3

library("ggpubr")
library("ggplot2")
ggscatter(dat,
x="fish.length",
y="fish.weight",
add="reg.line",
conf.int=TRUE,
cor.coef=TRUE,
xlab="Fish Lenth in cm",
ylab="Fish Weight in g")
```

<u>Observation</u>:

```
> #Enter Data for correlation
>fish.length= c(8,13,5,25,21,19,16)
>fish.weight= c(12,22,10,16,8,24,20)
> dat=data.frame(fish.length, fish.weight)
> dat
fish.length fish.weight
1         8         12
2        13          22
3         5         10
4        25          16
5        21           8
6        19          24
7        16          20
>
> # Perform Correlation Analysis
> res1 = cor.test(dat$fish.length, dat$fish.weight, method= "pearson")
> res2 = cor.test(dat$fish.length, dat$fish.weight, method= "kendall")
> res3 = cor.test(dat$fish.length, dat$fish.weight, method= "spearman")
> # Report Result
> res1

        Pearson's product-moment correlation

data:  dat$fish.length and dat$fish.weight
t = 0.55385, df = 5, p-value = 0.6035
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6259664  0.8411837
sample estimates:
      cor
0.2404258

> res2

        Kendall's rank correlation tau
```

```
data:  dat$fish.length and dat$fish.weight
T = 12, p-value = 0.7726
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.1428571

> res3
        Spearman's rank correlation rho

data:  dat$fish.length and dat$fish.weight
S = 50, p-value = 0.8397
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.1071429

>
> library("ggpubr")
> library("ggplot2")
>ggscatter(dat,
+ x="fish.length",
+ y="fish.weight",
+ add="reg.line",
+ conf.int=TRUE,
+ cor.coef=TRUE,
+ xlab="Fish Lenth in cm",
+ ylab="Fish Weight in g")
`geom_smooth()` using formula 'y ~ x'
>
>
```



<u>Conclusion</u>: Results of the Pearson correlation indicated that there was no significant positive association between Fish weight (grams) and Fish Length (cm), (r (5) = 0.24, p = 0.6035).

Results of the Kendall's rank correlation indicated that there was no significant positive association between Fish weight (grams) and Fish Length (cm), (T = 12, P = 0.7726, tau = 0.14285).

Results of the Spearman correlation indicated that there was no significant positive association between Fish weight (grams) and Fish Length (cm), (S = 50, p =0.8379, Rho = 0.1071429).

PERFORMANCE OF REGRESSION USING ‘R’

Aim- To perform regression in R using provided data

Principle- Regression analysis is a way to find trends in data. Regression analysis will provide us with an equation for a graph so that we can make predictions about our data. For example, if I have been putting on weight over the last few years, it can predict how much I will weigh in ten years time if I continue to put on weight at the same rate. It will also give me a slew of statistics (including a p-value and a correlation coefficient) to tell me how accurate my model is. Most elementary stats courses cover very basic techniques, like making scatter plots and performing linear regression. However, I may come across more advanced techniques like multiple regressions.

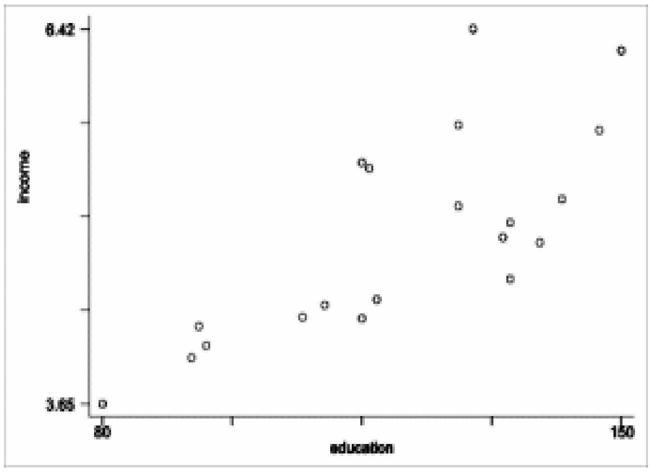
Multiple regression analysis is used to see if there is a statistically significant relationship between sets of variables. It is used to find trends in those sets of data.

Multiple regression analysis is almost the same as simple linear regression. The only difference between simple linear regression and multiple regression is in the number of predictors (“x” variables) used in the regression.

- **Simple regression analysis uses a single “x” variable for each dependent “Y” variable. For example: (x_1, Y_1).**
- **Multiple regressionsuse multiple “x” variables for each independent variable: $(x1)_1$, $(x2)_1$, $(x3)_1$, Y_1).**

In one-variable linear regression, I would input one dependent variable (i.e. “sales”) against an independent variable (i.e. “profit”). I could set my X_1 as one type of sales, my X_2 as another type of sales and so on.

Ordinary linear regression usually isn’t enough to take into account all of the real-life factors that have an effect on an outcome. For example, the following graph plots a single variable (number of doctors) against another variable (life-expectancy of women). From this graph it might appear there is a relationship between life-expectancy of women and the number of doctors in the population. In fact, that’s probably true and you could say it’s a simple fix: put more doctors into the population to increase life expectancy. But the reality is



you would have to look at other factors like the possibility that doctors in rural areas might have less education or experience. Or perhaps they have a lack of access to medical facilities like trauma centers. The addition of those extra factors would cause me to add additional dependent variables to my regression analysis and create a multiple regression analysis model.
Regression analysis is always performed in software, like Excel or SPSS. The output differs according to how many variables I have but it's essentially the same type of output I would find in a simple linear regression. There's just more of it:

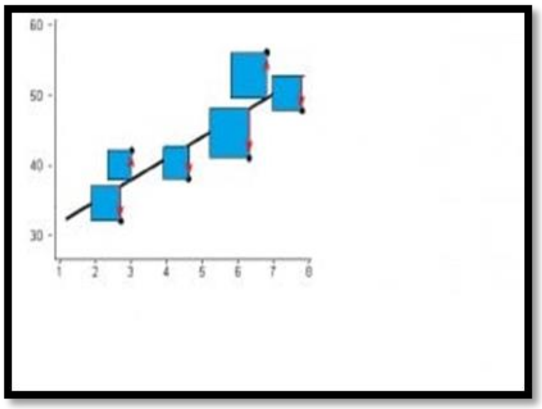
- **Simple regression: $Y = b_0 + b_1$ x.**
- **Multiple regression: $Y = b_0 + b_1$ x1 + $b_0 + b_1$ x2…b_0…b_1 xn.**
-

The output would include a summary, similar to a summary for simple linear regression that includes:

- **R (the multiple correlation coefficient),**
- **R squared (the coefficient of determination),**
- **adjusted R-squared,**
- **The standard error of the estimate.**

These statistics help me figure out how well a regression model fits the data. The ANOVA table in the output would give me the p-value and f-statistic.

If I am concerned with finding accurate values for squared multiple correlation coefficient, minimizing theshrinkage of the squared multiple correlation coefficient or have another specific goal.



Overfitting is where my model is too complex for your data — it happens when my sample size is too small. If I put enough predictor variables in my regression model, I will nearly always get a model that looks significant.
While an overfitted model may fit the idiosyncrasies of my data extremely well, it won't fit additional test samples or the overall population. The model'sp-values, R-Squared and regression coefficients can all be misleading. Basically, you're asking too much from a small set of data.

In linear modeling (including multiple regression), I should have at least 10-15 observations for each term I am trying to estimate. Any less than that, and I run the risk of overfitting my model.
"Terms" include:

- **Interaction Effects,**
- **Polynomial expressions (for modeling curved lines),**
- **Predictor variables.**

The easiest way to avoid overfitting is to increase my sample size by collecting more data. If I can't do that, the second option is to reduce the number of predictors in your model — either by combining or eliminating them. Factor Analysis is one method you can use to identify related predictors that might be candidates for combining.

1. Cross-Validation
Use cross validation to detect overfitting: this partitions your data, generalizes your model, and chooses the model which works best. One form of cross-validation is predicted R-squared. Most good statistical software will include this statistic, which is calculated by:

- **Removing one observation at a time from your data,**
- **Estimating the regression equation for each iteration,**
- **Using the regression equation to predict the removed observation.**

Cross validation isn't a magic cure for small data sets though, and sometimes a clear model isn't identified even with an adequate sample size.

2. Shrinkage & Resampling
Shrinkage and resampling techniques (like this R-module) can help you to find out how well my model might fit a new sample.

3. Automated Methods
Automated stepwise regression shouldn't be used as an overfitting solution for small data sets.

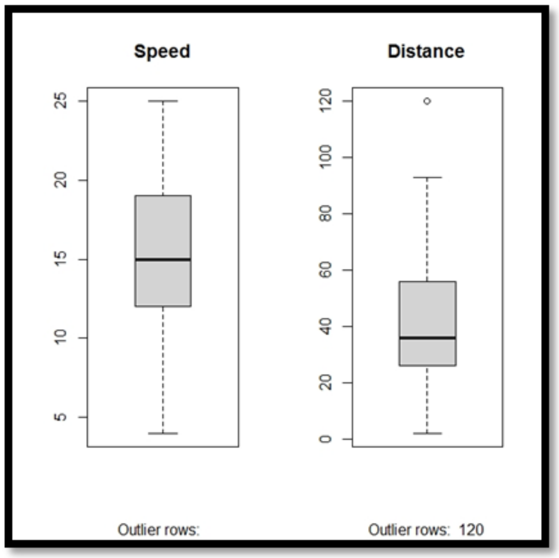
Example:

```
head(cars)
sctter.smooth(x=cars$speed, y=cars$dist, main="Dist~Speed")  #scatterplot

cor(cars$speed, cars$dist, method="pearson")

par(mfrow=c(1, 2))  # divide graph area in 2 columns
boxplot(cars$speed, main="Speed",
sub=paste("Outlier rows: ", boxplot.stats(cars$speed)$out))  # box plot for speed

boxplot(cars$dist, main="Distance",
sub=paste("Outlier rows: ", boxplot.stats(cars$dist)$out))  # box plot for distance
hist(cars$dist)
hist(cars$speed)

#build linear regression model on full data
linearMod= lm(dist~speed, data=cars)
summary(linearMod)

#Draw a graph
par(mfrow=c(1,1))
```

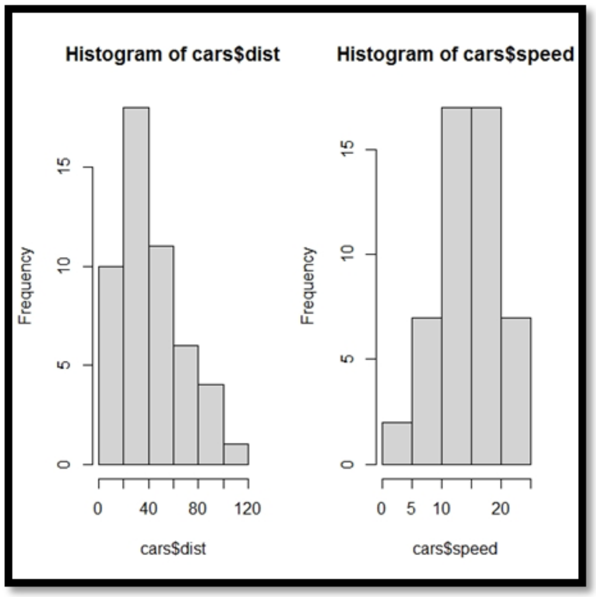
```
plot(dist~speed, data=cars,
main= "Plot of Distance travelled vs Speed of Car")
lines(linearMod$fitted.values~cars$speed,col="blue")

#predict the distance if speed is 80km/h
newdata=data.frame(speed=80)
predict(linearMod, newdata, interval="confidence")
```

Observation:

```
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
>sctter.smooth(x=cars$speed, y=cars$dist, main="Dist~Speed")  #scatterplot
Error in sctter.smooth(x = cars$speed, y = cars$dist, main = "Dist~Speed") :
  could not find function "sctter.smooth"
>
>cor(cars$speed, cars$dist, method="pearson")
[1] 0.8068949
>
>par(mfrow=c(1, 2))  # divide graph area in 2 columns
>boxplot(cars$speed, main="Speed",
+ sub=paste("Outlier rows: ", boxplot.stats(cars$speed)$out))  # box plot for speed
>
>boxplot(cars$dist, main="Distance",
+ sub=paste("Outlier rows: ", boxplot.stats(cars$dist)$out))  # box plot for distance
```

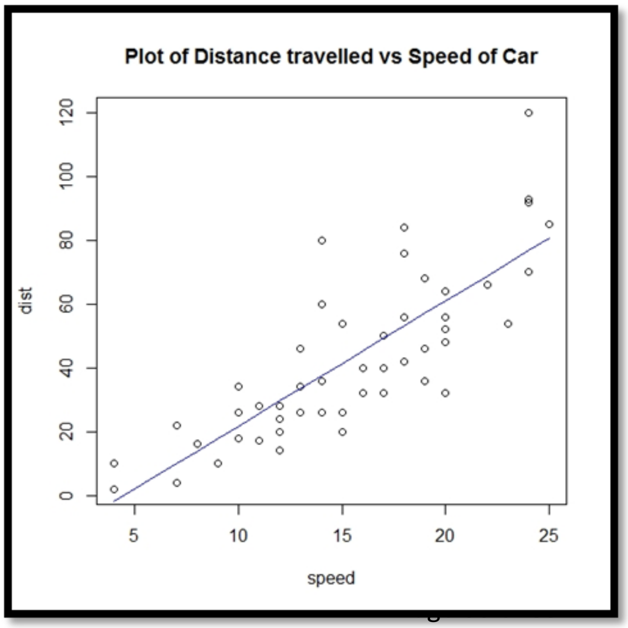


```
> hist(cars$dist)
> hist(cars$speed)
>
```



```
> #build linear regression model on full data
> linearMod= lm(dist~speed, data=cars)
> summary(linearMod)

Call:
lm(formula = dist ~ speed, data = cars)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12

>
> #Draw a graph
> par(mfrow=c(1,1))
>plot(dist~speed, data=cars,
+ main= "Plot of Distance travelled vs Speed of Car")
>lines(linearMod$fitted.values~cars$speed,col="blue")
>
```



```
> #predict the distance if speed is 80km/h
> newdata=data.frame(speed=80)
>predict(linearMod, newdata, interval="confidence")
     fit    lwr     upr
1 297.0136242.867 351.1602
```

<u>Results:</u> A simple linear regression was calculated to predict the covered distance by cars based on its speed. A significant regression equation was found (F(1,46)=89.57, $p < 0.001$) , with an R^2 of 0.6511. Cars predicted covered distance is equal to (-17.58) + 3.9324 x speed.Cars' covered distance increased 3.9324 km for each kilometer per hour of speed.

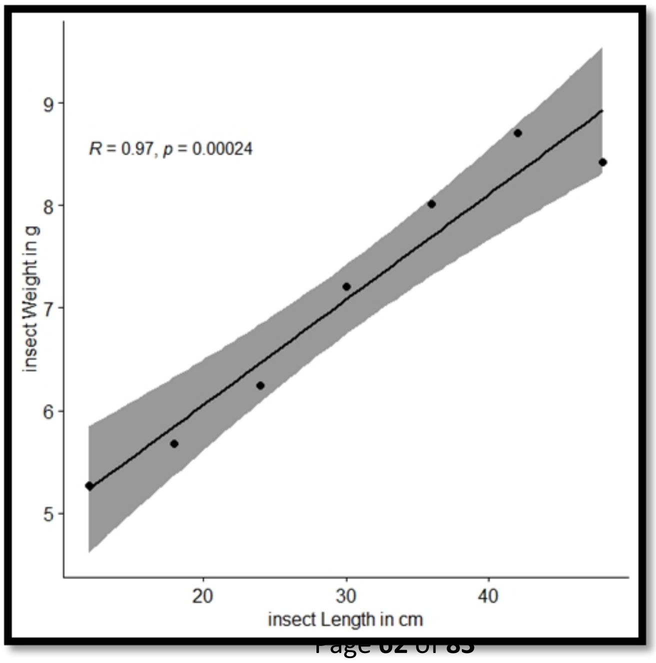
<u>Conclusion</u>: According to this model, the distance would be 297.01 km for a speed of 80 kmph. The 95% confidence interval of the distance for the speed of 80 kmph is between 242.867 km and 351.1602 km.

<u>Example</u>:

```
length= c(12,18,24,30,36,42,48)
weight= c(5.27,5.68,6.25,7.21,8.02,8.71,8.42)

insect= data.frame(length,weight)
summary(insect)

library("ggpubr")
ggscatter(insect,
          x = "length",
          y = "weight",
         add = "reg.line",
         conf.int = TRUE,
cor.coef = TRUE,
cor.method = "pearson",
         xlab = "insect Length in cm",
         ylab = "insect Weight in g")


fit = lm(weight~length, data= insect)
summary(fit)
newdata = data.frame(length=40)
predict(fit,newdata, interval ="confidence")
```

Observation:

```
> length= c(12,18,24,30,36,42,48)
> weight= c(5.27,5.68,6.25,7.21,8.02,8.71,8.42)
>
> insect= data.frame(length,weight)
> summary(insect)
    length      weight
 Min.   :12   Min.   :5.270
 1st Qu.:21   1st Qu.:5.965
 Median :30   Median :7.210
 Mean   :30   Mean   :7.080
 3rd Qu.:39   3rd Qu.:8.220
 Max.   :48   Max.   :8.710
>
> library("ggpubr")
Loading required package: ggplot2
>ggscatter(insect,
+          x = "length",
+          y = "weight",
+         add = "reg.line",
+         conf.int = TRUE,
+         cor.coef = TRUE,
+         cor.method = "pearson",
+         xlab = "insect Length in cm",
+         ylab = "insect Weight in g")
`geom_smooth()` using formula 'y ~ x'
>
```



```
>
> fit = lm(weight~length, data= insect)
> summary(fit)

Call:
lm(formula = weight ~ length, data = insect
Residuals:
      1        2        3        4        5        6        7
 0.04143 -0.16571 -0.21286  0.13000  0.32286  0.39571 -0.51143

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.99429    0.35656  11.202  9.9e-05 ***
length       0.10286    0.01104   9.321 0.000239 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3504 on 5 degrees of freedom
Multiple R-squared:  0.9456,    Adjusted R-squared:  0.9347
F-statistic: 86.87 on 1 and 5 DF,  p-value: 0.0002393

> newdata = data.frame(length=40)
>predict(fit,newdata, interval ="confidence")
  fit      lwr      upr
1 8.108571 7.665459 8.551684
>
```

Result: A simple linear regression was calculated to predict the weight of an insect based on its length. A significant regression equation was found ($F(1,5)=86.87$, $p < 0.001$) , with an R^2 of 0.9456. Insects predicted weight is equal to 3.99 + 0.10286 x length grams when the length is measured in centimeter. Insects' weight increased 0.10286 grams for each centimeter of length.

Conclusion: According to this model, the weight would be 8.11 g for a length of 40 cm. The 95% confidence interval of the weight for the length of 40 cm is between 7.665 g and 8.552 g.

ESTIMATION OF DIVERSITY INDICES VALUES OF GIVEN SPECIES USING PAST4.03 SOFTWARE

Principle- A Diversity Index (Plural: Indices) is a quantitative measurement that reflects how many different type (such as species) are there in a given dataset, and simultaneously takes into account how evenly the basic entities (such as individuals) are distributed among these types. In PAST 4.03 software, there are 13 diversity indices, including the important ones like Shannon's Index, Simpson's index, Margalef Index, etc.

Procedure-

Step 1: PAST 4.03 icon on the desktop screen was clicked. This led to opening of the software user interface.

Step 2: Once the interface appeared on the screen, the 2 boxes on the left top corner of the interface, ascribed with **Row attributes** and **Column attributes** were checked.

Step 3: This opened up a pink-coloured horizontal band, which has, written on it – A, B, C, D, E and so on. A similar pink-coloured vertical band also appeared with numbers written on it, with a heading called Name.

Step 4: The variables entered on the horizontal band were the Column attributes, whereas, the variables entered on the vertical band form the Row attributes.

Step 5: After the variables were entered, the previously mentioned boxes were unchecked; this led to closure of the attribute rows and columns. The entered variable then appeared on the rows and columns.

Step 6: The concerned dataset was then entered against the respective variables.

Step 7: The entire dataset was then selected.

Step 8: Then, the option for Diversity was selected from the topmost taskbar (8th from extreme left corner). A menu dropped from it. The option Diversity Indices (topmost option) was selected.

Step 9: Then, it took few seconds to complete the calculation.

Step 10: A table appeared with all the values of the 13 Diversity Indices. The table has 2 tabs namely Number and Plot. The table of values appeared against the Number tab.

Step 11: One can also obtain the graphical representation of all the 13 Indices, by clicking on the Plot tab. All the 3 boxes on the right–hand side of the graph, namely **Error Box**, **Connecting line** and **Flip axes** should be checked. On the right-hand upper side, there is a drop-down menu, where one can select the diversity index type, for which he/she want the graph.

Step 12: One can save the graph, by clicking on the **Graph Setting** option, then clicking the **Save As** option under Export Tab, then entering the desired directory, file name and Jpg format. This gives the desired table and graphs.

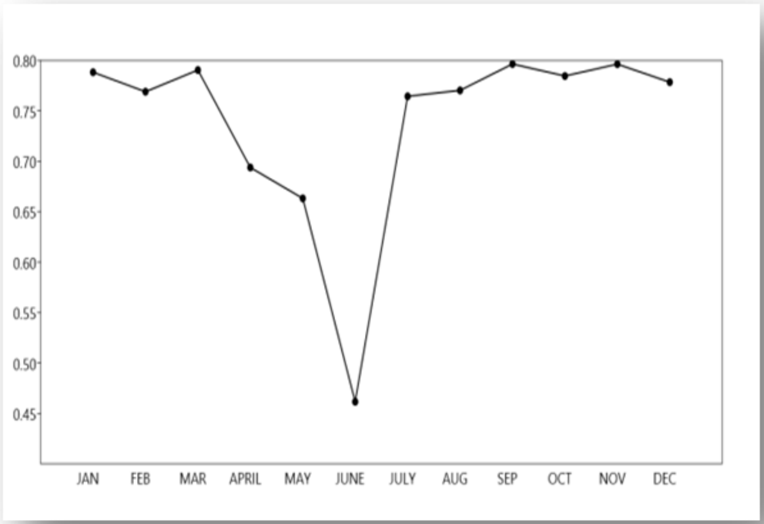
	JAN	FEB	MAR	APRIL	MAY	JUNE	JULY	AUG	SEP	OCT	NOV	DEC
Taxa_S	5	5	5	4	4	3	5	5	5	5	5	5
Individuals	276	183	49	7	14	13	155	214	384	31	107	278
Dominance_D	0.2117	0.231	0.2095	0.3061	0.3367	0.5385	0.2355	0.2297	0.2035	0.2154	0.2036	0.2215
Simpson_1-D	0.7883	0.769	0.7905	0.6939	0.6633	0.4615	0.7645	0.7703	0.7965	0.7846	0.7964	0.7785
Shannon_H	1.579	1.525	1.584	1.277	1.233	0.7903	1.507	1.517	1.601	1.571	1.601	1.552
Evenness_e^H/S	0.97	0.9193	0.9746	0.8965	0.8576	0.7347	0.9031	0.9113	0.9917	0.9627	0.9912	0.9437
Brillouin	1.54	1.471	1.433	0.8629	0.9634	0.6122	1.447	1.469	1.571	1.362	1.516	1.513
Menhinick	0.301	0.3696	0.7143	1.512	1.069	0.8321	0.4016	0.3418	0.2552	0.898	0.4834	0.2999
Margalef	0.7117	0.7678	1.028	1.542	1.137	0.7797	0.7931	0.7454	0.6722	1.165	0.856	0.7108
Equitability_J	0.9811	0.9477	0.984	0.9212	0.8892	0.7193	0.9367	0.9423	0.9949	0.9764	0.9945	0.964
Fisher_alpha	0.8671	0.9494	1.394	3.878	1.871	1.223	0.9877	0.9161	0.8115	1.687	1.087	0.8658
Berger-Parker	0.2609	0.306	0.2449	0.4286	0.5	0.6923	0.2903	0.271	0.25	0.2903	0.243	0.2842
Chao-1	5	5	5	4.5	4	3	5	5	5	5	5	5

Results-

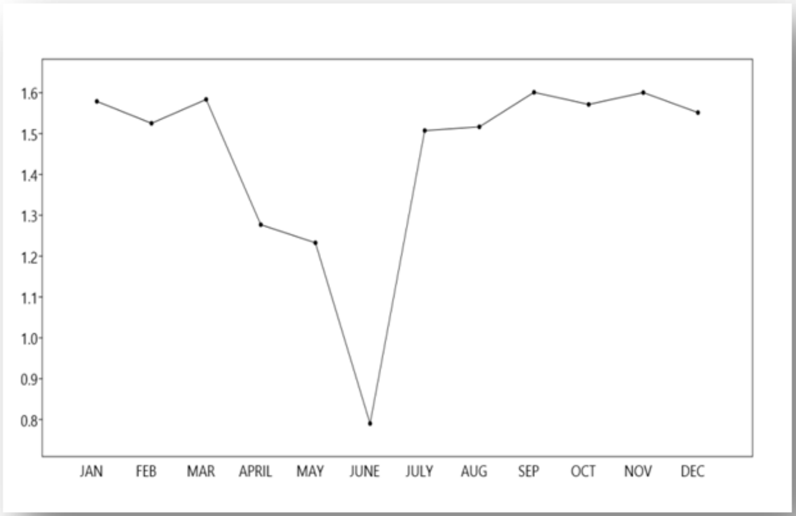
Table: Diversity Indices results for the given dataset

Comments-

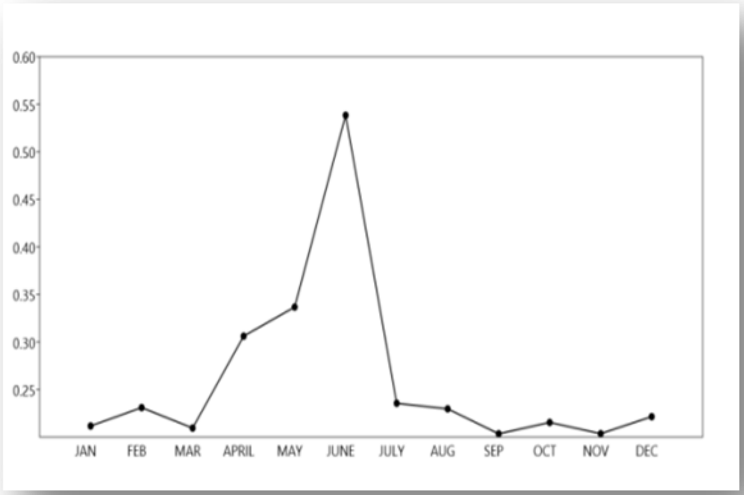
1. **Simpson's Index:** According to Simpson's index, the highest value is shown in the month of September i.e. 0.7965 followed by the month November, March, January, October (i.e. 0.7964, 0.7905, 0.7883, 0.7846 respectively) whereas the lowest diversity index value is shown in the month June i.e. 0.4615.



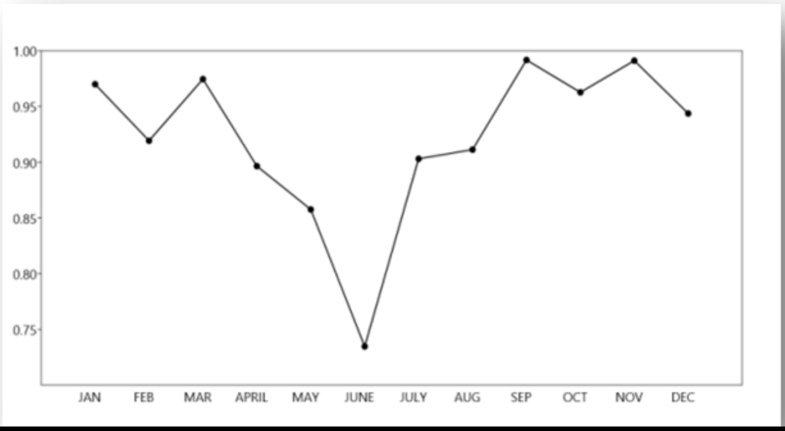
2. **Shannon H Index:** According to Shannon H index, we can conclude that the month September and November show the equal and highest value i.e. 1.601 followed by the months March, January, October whereas the month June has the lowest value i.e. 0.7903.
If we look carefully, Simpson's index follows a similar trend as that of Shannon H index as clear from the table as well as the plot i.e. in both the indices September has the highest diversity and June has the lowest diversity.



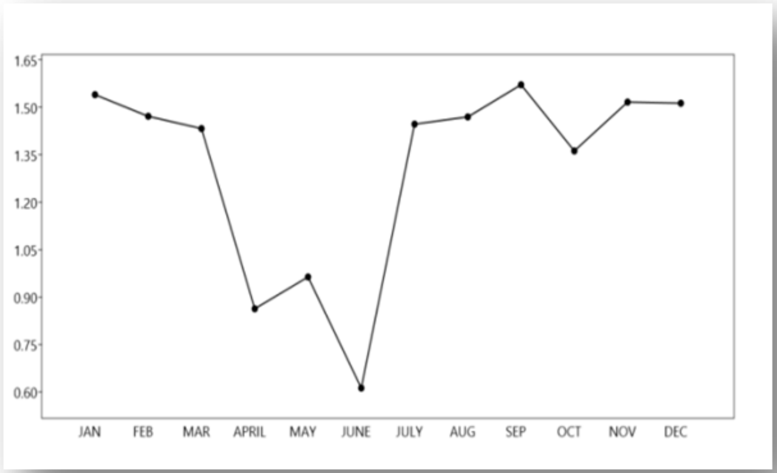
3. **Dominance D:** We all know that dominance index show an inverse relation to both Shannon and Simpson's index. As we have seen the month September has highest diversity followed by whereas the month June has more dominance i.e. 05385. So, it can be concluded that any one particular species in our data is more abundant in the month June which makes it a dominant zone.



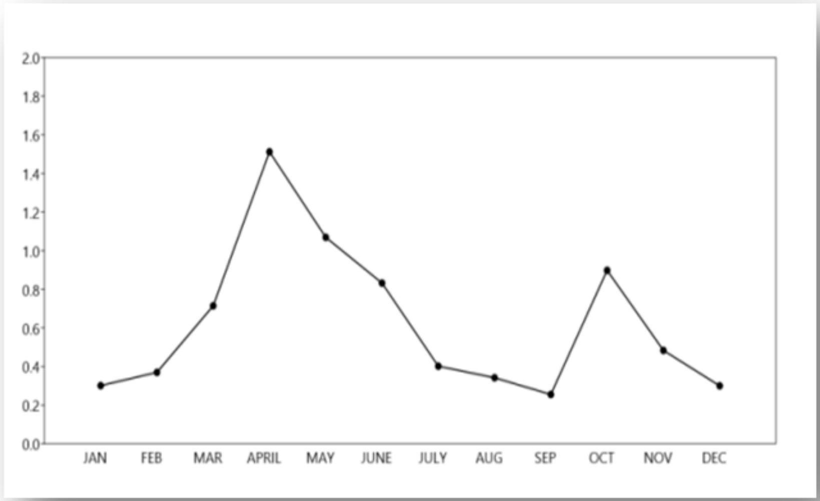
4. **Evenness:**According to this index, the month September is more even than the other months, which means all species more or less evenly distributed in the month September.



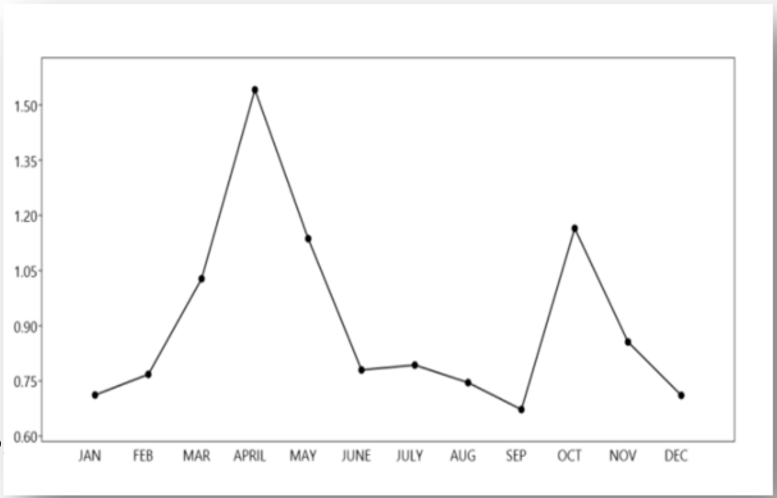
5. **Brillouin Index:** According to this index, the diversity of species is high in the month September i.e. 1.571 followed by the month January, November, December whereas in the month June the diversity is very low i.e. 0.6122.



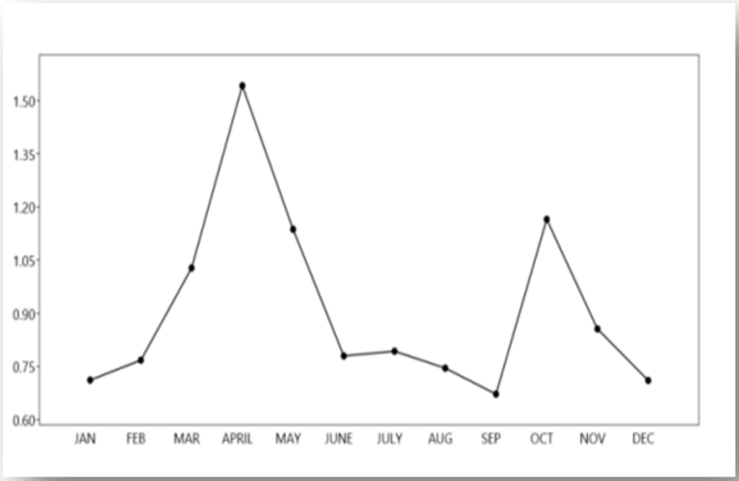
6. **Menhinick Index:** According to this index, we can conclude from our data as well as plot that the month April shows the highest value i.e. 1.512 than the other months and the month September shows the lowest value i.e. 0.2552.



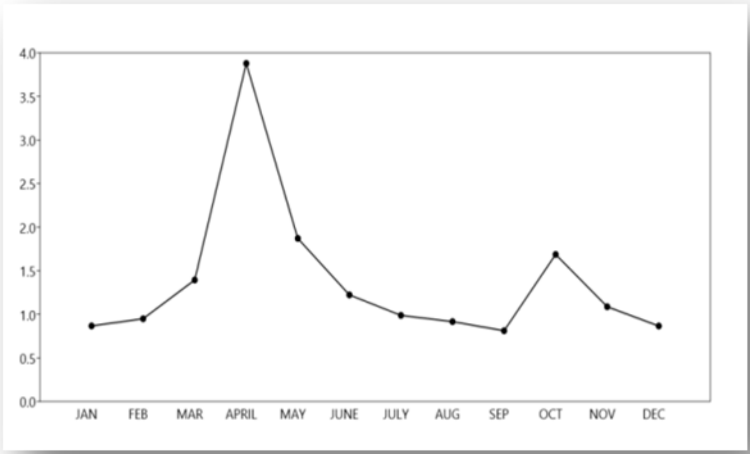
7. **Margalef Richness Index:** According to this index, we can see clearly from our data as well as plot that the value of richness index is maximum in the month April i.e. 1.542 whereas the value of richness index is minimum in the month December i.e. 0.7108.



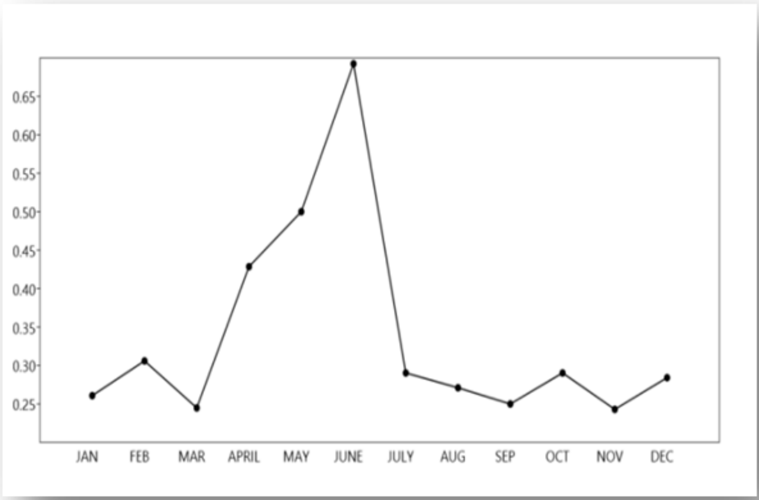
8. **Equitability J:** Equitability J is also measure of evenness. So, it is same as that of Pielou's evenness index. The month September has the highest valuei.e. 0.9949, indicating even distribution of species whereas the month June has the lowest value i.e. 0.7193, indicating uneven distribution of species.



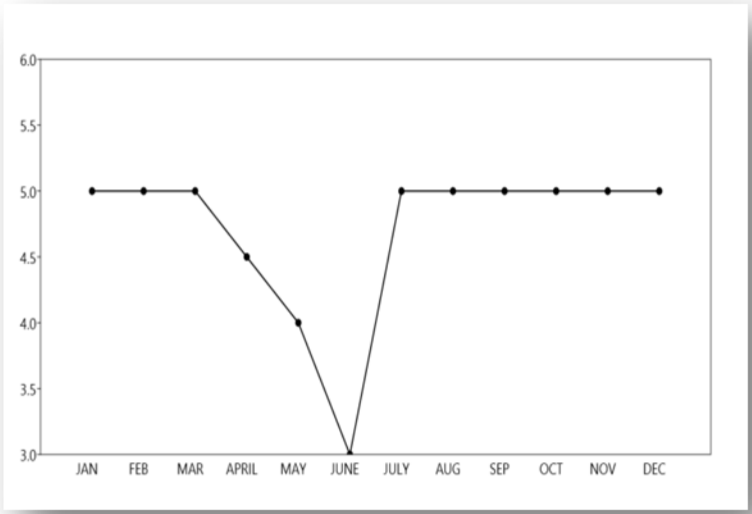
9. **Fisher_alpha:**From the data as well as plot, it can be concluded that the month April shows highest Fisher_alpha value i.e. 3.878 whereas the month September shows lowest Fisher_alpha value i.e. 0.8115.



10. **Berger-Parker:**According to Berger-Parker index, the abundant species is found in the month June. The dominance value is high in June. So, it can be concluded that any one particular species is more abundant in the month June.



11. **Chao-1:** According to Chao estimator, we can see that all the months have same species richness i.e. 5 except the two months April and May.



<u>Concluding Remarks</u>- Out of all months September with high values of both Simpson and Shannon Index show that it is a diversified month and the individuals are more or less equally distributed making the Equitability and Evenness value higher than other months. Thus it can be concluded that the month September has higher number of species whereas the month June has dominant species.

FASTA -AN INTRODUCTORY NOTE

The FASTA algorithm is a heuristic method for string comparison. It was developed by Lipman and Pearson in 1985 and further improved in 1988. FASTA stands for fast-all" or "FastA".

FASTA compares a query string against a single text string. When searching the whole database for matches to a given query, we compare the query using the FASTA algorithm to every string in the database.

When looking for an alignment, we might expect to find a few segments in which there will be absolute identity between the two compared strings. The algorithm is using this property and focuses on these identical regions.

The current FASTA package contains programs for protein-protein, DNA-DNA, protein-translated DNA (with frameshifts), and ordered or unordered peptide searches.

In addition to rapid heuristic search methods, the FASTA package provides SSEARCH, an implementation of the optimal Smith–Waterman algorithm.

A major focus of the package is the calculation of accurate similarity statistics, so that biologists can judge whether an alignment is likely to have occurred by chance, or whether it can be used to infer homology. The FASTA package is available from the University of Virginia and the European Bioinformatics Institute.

The FASTA file format used as input for this software is now largely used by other sequence database search tools (such as BLAST) and sequence alignment programs (Clustal, T-Coffee, etc.).

RETRIEVAL OF THE MITOCHONDRIAL DNA SEQUENCE OF HUMAN (*Homo sapiens*) FROM NCBI DATABASE AND DOWNLOADING IT IN FASTA FORMAT

Aim- To access NCBI database and download a file in FASTA format.

Theory- An important step in providing sequence database access was the development of pages that allow queries to make for retrieval of major sequence of nucleotide from such databases (e.g., GenBank, EMBI, etc). NCBI stands for 'National Centre for Biotechnology Information' and is a part of United States of National Library of Medicine (NLM), a branch of NIH – National Institutes of Health. It is directed by David Lipman one of the original author of GENINFO & BLAST.

Requirement-A computer with Internet connection.

Procedure-

STEP-1: NCBI website was accused **(https://www.ncbi.nlm.nih.gov/)**

STEP-2: On the left side upper portion of the HOME Screen of NCBI, there lies a dropdown menu from which, the option "**Nucleotide**" is selected.

STEP-3: Then the keyword "amylase" was entered in the Search Tab.

STEP-4: The Option "**mitochondrial DNA [*Homo sapiens*]**" with a unique AccessionNumber **AH001266.2** was selected and clicked.

STEP-5: One will find option for Download formats on the upper part of the screen. For downloading it in FASTA format, click on FASTA take written in blue colour.

STEP-6: The sequence thus obtained were copied and pasted in Notepad.

STEP-7: Thus file is saved.

Thus the desired nucleotide sequence is obtained for ***Homo sapiens* (Human) mitochondrial DNA**sequence.

Observation-

```
>AH001266.2 Homo sapiens Human (!Kung 9,10) mitochondrial DNA sequences, 5' end

TTCTTTCATGGGGAAGCAGATTTGGGTACCACCCAAGTATTGACTCACCCATCAACAACCGCTATGTATT
TCGTACATTACTGCCAGCCACCATGAATATTGTACAGTACCATAAATACTTGACCACCTGTAGTACATAA
AAACCCAATCCACATCAAAACCCTCCCCCCATGCTTACAAGCAAGTACAGCAATCAACCTTCAACTGTCA
CAATCAACCGCAACTCCAAAGCCACCCCTCACCCACTAGGATACCAACAAACCTACCCACCCTTAACAGT
ACATAGCACATAAAGCCATTTACCGTACATAGCACATTACAGTCAAATCCCTTCTCGTCCCCATGGATGA
```

CCCCCCTCAGATAGNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTATTTTCGTCTGGGGGGTGTGCACG
C
GATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCAGTATCTGTCTTTGATTCCTGCCCCATC
CCATTATTTATCGCACCTACGTTCAATATTACAGGCGAACATACCTATTAAAGTGTGTTAATTAATTAAT
GCTTGTAGGACATAATAATAACAATTAAATGTCTGCACAGCCACTTTCCACACAGACATCATAACAAAAA
ATTTCCACCAAACCCCCCCCTCCCCCCGCTTCTGGCCACAGCACTTAAACACATCTCTGCCAAACCCCAA
AAACAAAGAACCCTAACACCAGCCTAACCAGATTTCAAATTTTAT

Conclusion- The sequence we have used in our FASTA search is the mitochondrial DNA nucleotide of ***Homo sapiens*** **(Human)** under the GenBank accession number **AH001266.2**. This was sequenced by Vigilant, L., Pennington, R., Harpending, H., Kocher, T.D. and Wilson, A.C. in their publication "Mitochondrial DNA sequences in single hairs from a southern African Population", This FASTA sequence can be used for various works including BLAST search, MSA preparation and phylogenetic tree construction.

BLAST - AN INTRODUCTORY NOTE

BLAST, is an acronym for Basic Local Alignment Search Tool and refers to a suite of programs most commonly used to generate alignments between a nucleotide or protein sequence, referred to as a “query” and nucleotide or protein sequences within a database, referred to as “subject” sequences. The original BLAST program used a protein “query” sequence to scan a protein sequence database. Specialized variants of BLAST allow fast searches of nucleotide databases with very large query sequences.

BLAST is available from the National Centre for Biotechnology Information (www.ncbi.nlm.nih.gov).

The BLAST algorithm works by finding a short, or local, region of high similarity between two sequences, and then extending this match out from this starting point to both the left and the right.

A score is assigned to the match. The score will increase as more residues are found to match, and will decrease if there are gaps in the alignment.

Alignments with a score that exceed a certain threshold are reported in the output.

APPLICATION OF BLAST TO FIND 10 SIMILAR SEQUENCE OF MITOCHONDRIAL DNA OF HUMAN (*Homo sapiens*) NUCLEOTIDE SEQUENCE FROM NCBI DATABASE

Aim- To perform BLAST search in NCBI website.

Theory- Multiple Sequence Alignment (MSA) is a sequence alignment of 3 or more biological sequence – generally, DNA or RNA or protein. In many cases, the input – set of query sequence are assumed have an evolutionary relationship by which their sharing of a lineage, can be interpreted for understanding their descendent from a common ancestry. From the resulting MSA, sequence homology can be inferred and phylogenetic analysis can be conduct to access the sequence – whether they share evolutionary origins or not. MSA is often used to access sequence conservations of protein, domains, tertiary and secondary structure of proteins and even individuals amino acids.

Requirement- A computer with Internet connection.

Procedure-

STEP-1: NCBI website was accused.(https://www.ncbi.nlm.nih.gov/)

STEP-2: On the left side upper portion of the HOME Screen of NCBI there lies a dropdown menu from which the option "**Nucleoide**" is selected.

STEP-3: Then the keyword "**human mitochondrial DNA**" was entered in the Search Tab.

A list of the proteins was obtained.

STEP-4: The Option "**mitochondrial DNA [*Homo sapiens*]**" with a unique Accession Number "**AH001266.2**" was selected and clicked.

STEP-5: The accession number was copied using **Ctrl+C** option.

STEP-6: Then the NCBI-BLAST Home page was accused by clicking BLAST, present in right centre part of the lower portion of the page or simply entering the url https://blast.ncbi.nlm.nih.gov/, subsequently, the **nucleotide** BLAST tab was selected. This was also termed **blastn**.

STEP-7: There was an option Enter Accession Number(s), gi(s) or FASTA sequence(s) in the **Enter Query Sequence** section on the upper left part of page, with a large white rectangular white box below it. The accession number was pasted here.

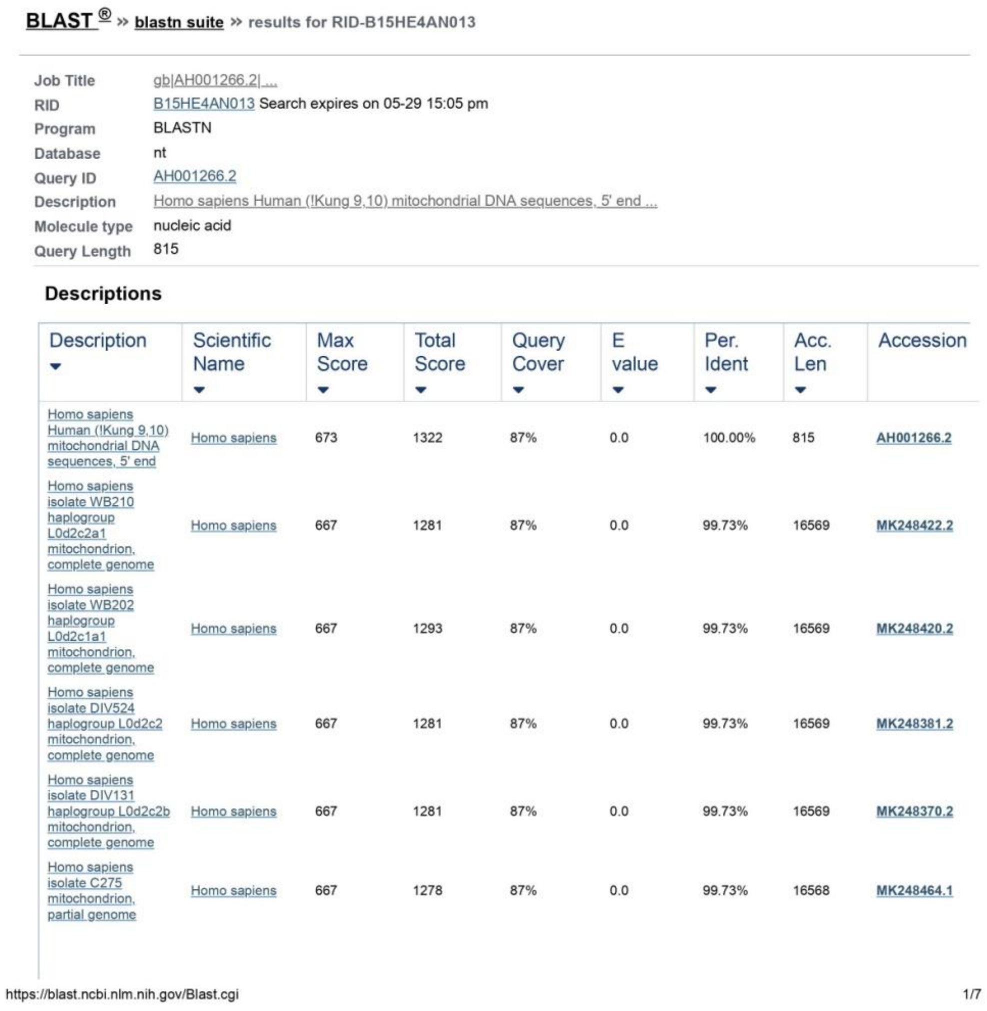
STEP-8: Then, in the**Choose Search Set** section, there was dropdown menu against Database option. Selected **Nucleotide collection (nr/nt)** option.

STEP-9: In the Program selection section, selected **Highly similar sequences (megablast)** option.

STEP-10: Then clicked the blue button, inscribed with BLAST.

It would take around 30 seconds time, and then, a chart, similar to that on the left hand page will appear. This is the desired MSA of mitochondrial DNA (***Homo sapiens***).The page is printed in FASTA format.

Observation-



BLAST ® » blastn suite » results for RID-B15HE4AN013

Job Title	gb|AH001266.2| ...
RID	B15HE4AN013 Search expires on 05-29 15:05 pm
Program	BLASTN
Database	nt
Query ID	AH001266.2
Description	Homo sapiens Human (!Kung 9,10) mitochondrial DNA sequences, 5' end ...
Molecule type	nucleic acid
Query Length	815

Descriptions

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Homo sapiens Human (!Kung 9,10) mitochondrial DNA sequences, 5' end	Homo sapiens	673	1322	87%	0.0	100.00%	815	AH001266.2
Homo sapiens isolate WB210 haplogroup L0d2c2a1 mitochondrion, complete genome	Homo sapiens	667	1281	87%	0.0	99.73%	16569	MK248422.2
Homo sapiens isolate WB202 haplogroup L0d2c1a1 mitochondrion, complete genome	Homo sapiens	667	1293	87%	0.0	99.73%	16569	MK248420.2
Homo sapiens isolate DIV524 haplogroup L0d2c2 mitochondrion, complete genome	Homo sapiens	667	1281	87%	0.0	99.73%	16569	MK248381.2
Homo sapiens isolate DIV131 haplogroup L0d2c2b mitochondrion, complete genome	Homo sapiens	667	1281	87%	0.0	99.73%	16569	MK248370.2
Homo sapiens isolate C275 mitochondrion, partial genome	Homo sapiens	667	1278	87%	0.0	99.73%	16568	MK248464.1

https://blast.ncbi.nlm.nih.gov/Blast.cgi 1/7

Conclusion- The given accession number represent a protein sequence. From the BLAST search result we can conclude that amylase (*Tetradon nigroviridis*) is 100% matched whereas amylase 1 protein (*Tetradon nigroviridis*) is 95.32%, amylase 2 protein (*Tetradon nigroviridis*) and amylase 3 protein (*Tetradon nigroviridis*) are 94.35% and 92.59% matched with the given protein sequence respectively.

MULTIPLE SEQUENCE ALIGNMENT (MSA): - AN INTRODUCTORY NOTE

A Multiple Sequence Alignment (MSA) is an alignment of more than two sequences. We could align several DNA or protein sequences. From the output, homology can be inferred and the evolutionary relationships between the sequences studied.

The multiple sequence alignment assumes that the sequences are homologous; they descend from a common ancestor. The algorithms will try to align homologous positions or regions with the same structure or function. Hence computational algorithms are used to produce and analyze these alignments.

Most MSA algorithms use dynamic programming and heuristic methods. Some of the most usual uses of the multiple alignments are:

- phylogenetic analysis
- conserved domains
- protein structure comparison and prediction.

DERIVATION OF MULTIPLE SEQUENCE ALIGNMENT (MSA) USING MULTIPLE SEQUENCE VIEWER, VERSION 1.20.0 TO RETRIEVE SEQUENCE OF A UNKNOWN PROTEIN SEQUENCE

Aim-To download selected files from BLAST result and perform multiple sequence alignment using MUSCLE in NCBI.

Theory- Multiple Sequence Alignment of 3 or more biological sequence generally, DNA or RNA or protein. In way cases, the input-set of query sequences are assumed to have an evolutionary relationship by which, their sharing of a lineage, can be interpreted for understanding their descendent from a common ancestry. From the resulting MSA, sequence homology can be inferred and phylogenetic analysis can be conducted to assess the sequences- whether they share evolutionary origins or not. MSA is often proteins, domains, tertiary and secondary structure of proteins and even individuals amino acids.

Requirement- A computer with Internet connection.

Procedure-

STEP-1: NCBI website was accused **(https://www.ncbi.nlm.nih.gov/)**

STEP-2: Then the NCBI-BLAST Home page was accused by clicking BLAST , present in right centre part of the lower portion of the page or simply entering the url https://blast.ncbi.nlm.nih.gov/, subsequently, the protein BLAST tab was selected. This was also termed as **blastp**.

STEP-3: There was an option Enter Accession Number(s), gi(s) or FASTA sequence(s) in the Enter Query Sequence section on the upper left part of page, with a large white rectangular white box below it. Paste a unknown FASTA sequence.

STEP-4: Then keeping the parameters in Choose Search Set section, as default **(Non- redundant protein sequence(nr)).** The algorithm option - **blastp(),** in the Program Selection section was selected.

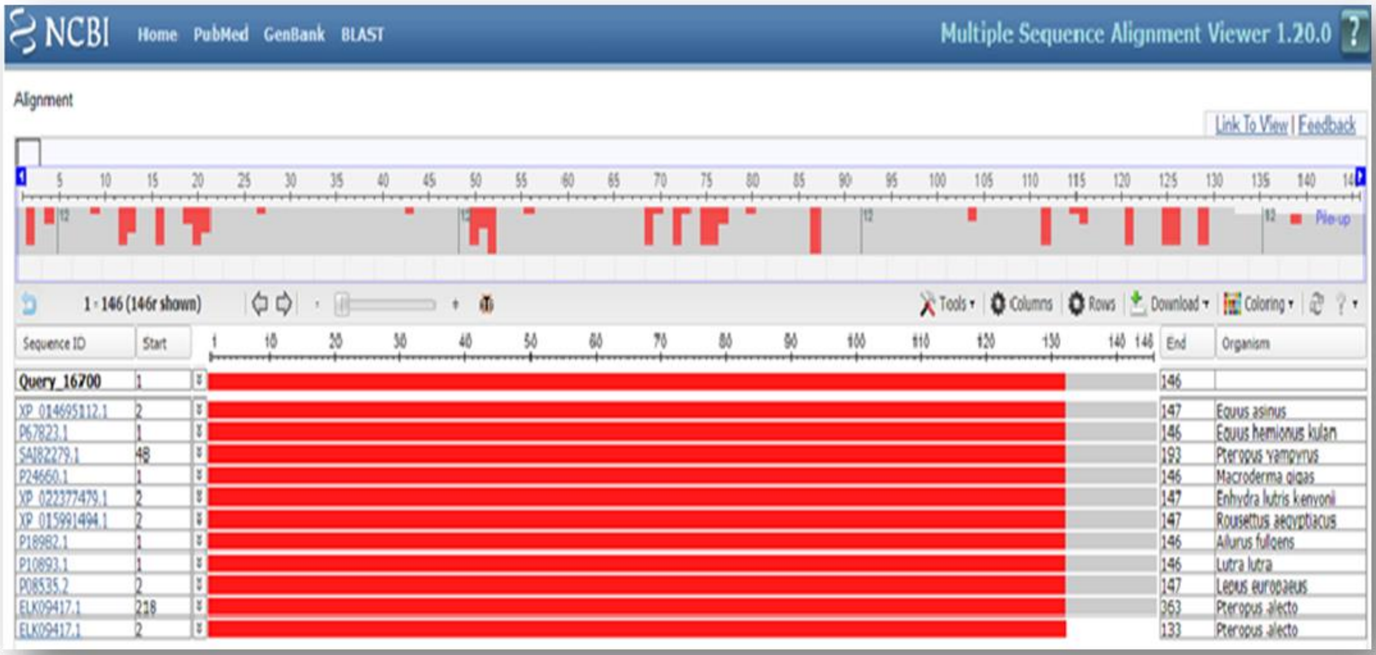
STEP-5: Then clicked the blue button, inscribed with BLAST. It would take around 5 seconds, and then a distribution chart for top 100 BLAST Horton 100 subject sequence, will appear.

STEP-6: There would be a menu just below the chart, with the heading Accession Description with a list of similar sequence and square shaped dialogue boxes on their side 10 of such sequences were selected.

STEP-7: After selecting 10 sequences of different organisms clicked on to the Multiple Sequence Alignment Viewer option present in the top most right corner.

STEP-8: We have to wait for few seconds and will finally get the multiple alignment sequence of 10 selected sequence of different organism.

<u>Observation</u>-



Sequence ID	Start	End	Organism
Query_16700	1	146	
XP_014695112.1	2	147	Equus asinus
P67823.1	1	146	Equus hemionus kulan
SAI82279.1	48	193	Pteropus vampyrus
P24660.1	1	146	Macroderma gigas
XP_022377479.1	2	147	Enhydra lutris kenyoni
XP_015991494.1	2	147	Rousettus aegyptiacus
P18982.1	1	146	Ailurus fulgens
P10893.1	1	146	Lutra lutra
P08535.2	2	147	Lepus europaeus
ELK09417.1	218	363	Pteropus alecto
ELK09417.1	2	133	Pteropus alecto

<u>Conclusion</u>- From our observations it can be concluded that in multiple sequence alignment (MSA) 10 sequences of different organisms were selected and all have 100% similar chain *Equus asinus* hemoglobin protein which are shown by red bands.

PHYLOGENETIC TREE: - AN INTRODUCTORY NOTE

Phylogenetic tree is a diagram that represents evolutionary relationships among organisms. Phylogenetic trees are hypotheses, not definitive facts. From the time of Charles Darwin, it has been the dream of many biologists to reconstruct the evolutionary history of all organisms on Earth and express it in the form of a phylogenetic tree. Phylogeny uses evolutionary distance, or evolutionary relationship, as a way of classifying organisms (taxonomy). The pattern of branching in a phylogenetic tree reflects how species or other groups evolved from a series of common ancestors. Phylogenetic relationship between organisms is given by the degree and kind of evolutionary distance. To understand this concept better, let us define taxonomy. Taxonomy is the science of naming, classifying and describing organisms. Taxonomists arrange the different organisms in taxa (groups). These are then further grouped together depending on biological similarities. This grouping of taxa reflects the degree of biological similarity. Phylogenic relationships have been traditionally studied based on morphological data. Scientists used to examine different traits or characteristics and tried to establish the degree of relatedness between organisms. Then scientists realized that not all shared characteristics are useful in studying relationships between organisms. This discovery led to a study of systematics called cladistics. Cladistics is the study of phylogenetic relationships based on shared, derived characteristics. There are two types of characteristics, primitive traits and derived traits.

CONSTRUCTION OF A PHYLOGENETIC TREE FROM THE OUTPUT OF MULTIPLE SEQUENCE ALIGNMENT (MSA) TO RETRIEVE 10 SEQUENCE OF A UNKNOWN PROTEIN SEQUENCE USING NCBI

Aim-To draw a phylogenetic tree online using NCBI website.

Theory- A Phylogenetic tree or evolutionary tree is a branching diagram or " tree"-shaped schematic representation, that shows the inferred evolutionary relationship among various biological species or other entities based upon similarities and differences, in their physical and/or genetic characteristics. The taxa joined together in the tree are implied to have descended from a common ancestor.

Requirement- A computer with Internet

Procedure-

STEP-1: NCBI website was accused **(https://www.ncbi.nlm.nih.gov/)**

STEP-2: Then the NCBI-BLAST Home page was accused by clicking BLAST , present in right centre part of the lower portion of the page or simply entering the URL https://blast.ncbi.nlm.nih.gov/, subsequently, the protein BLAST tab was selected. This was also termed as **blastp**.

STEP-3: There was an option Enter Accession Number(s), gi(s) or FASTA sequence(s) in the Enter Query Sequence section on the upper left part of page, with a large white rectangular white box below it. Paste a unknown FASTA sequence.

STEP-4: Then keeping the parameters in Choose Search Set section, as default **(Non- redundant protein sequence(nr)).** The algorithm option - **blastp(),** in the Program Selection section was selected.

STEP-5: Then clicked the blue button, inscribed with BLAST. It would take around 5 seconds, and then a distribution chart for top 100 BLAST Horton 100 subject sequence, will appear.

STEP-6: There would be a menu just below the chart, with the heading Accession Description with a list of similar sequence and square shaped dialogue boxes on their side 10 of such sequences were selected.

STEP-7: After selecting 10 sequences of different organisms clicked on to the Multiple Alignment option present in the top most right corner.

STEP-8: We have to wait for few seconds and will get the multiple alignment results (Cobalt RID) of 10 selected sequences of different organisms.

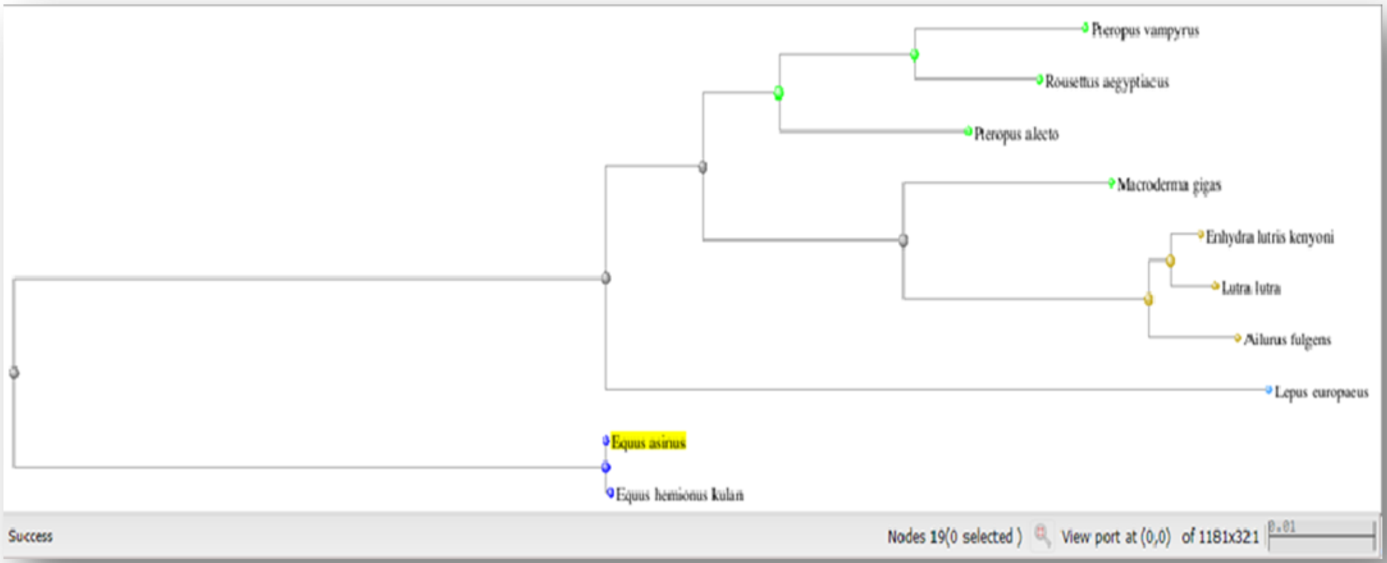
STEP-9: Now clicked on the Phylogenetic Tree option top most left side corner.

STEP-10: Now a new page open named with Phylogenetic Tree View where the phylogenetic tree analysis appeared of the selected 10 different organisms.

STEP-11: Next clicked on the dropdown button in the Sequence Lebel tab. Now clicked on the Taxonomic Name.

This procedure of the Phylogenetic tree for Multiple Sequence Alignment of **An UNKNOWN PROTEIN SEQUENCE**.

<u>Observation</u>-



<u>Conclusion</u>-According to the cladogram, *Equus asinus* originated from a common ancestral group. *Lepus europaeus*is the most closely related species to *Equus asinus*. If we look into the cladogram deeply, we could see that *Equus asinus and Equus hemionus kulan* are the most remote species as it originated early from the remote common unknown ancestor.