ACTIVITY 12 (Lab-06)

Name: - Abhishek Srivastava
Reg No. : - 19BCE10071

Q. Stepwise investigate the implementations of logistic regression algorithm by considering any application and analyse the results in detail.

1.

```
> # Installing the package
> install.packages("dplyr")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/dplyr_1.0.8.zip'
Content type 'application/zip' length 1382943 bytes (1.3 MB)
downloaded 1.3 MB

package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\Rtmp4oKShU\downloaded_packages
```

2.

```
> # Loading package
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Warning message:
package 'dplyr' was built under R version 4.1.2
```

3.

```
> # Summary of dataset in package
> summary(mtcars)
      mpg             cyl             disp             hp       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
      drat             wt             qsec             vs        
 Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
 1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
 Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
 Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
 3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
 Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
       am              gear            carb      
 Min.   :0.0000   Min.   :3.000   Min.   :1.000  
 1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
 Median :0.0000   Median :4.000   Median :2.000  
 Mean   :0.4062   Mean   :3.688   Mean   :2.812  
 3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

4.

```
> install.packages("caTools")     # For Logistic regression
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/caTools_1.18.2.zip'
Content type 'application/zip' length 316490 bytes (309 KB)
downloaded 309 KB

package 'caTools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\Rtmp4oKShU\downloaded_packages
> install.packages("ROCR")        # For ROC curve to evaluate model
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
also installing the dependency 'gplots'

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/gplots_3.1.1.zip'
Content type 'application/zip' length 603342 bytes (589 KB)
downloaded 589 KB

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/ROCR_1.0-11.zip'
Content type 'application/zip' length 458418 bytes (447 KB)
downloaded 447 KB
```

5.

```
> # Loading package
> library(caTools)
Warning message:
package 'caTools' was built under R version 4.1.2
> library(ROCR)
Warning message:
package 'ROCR' was built under R version 4.1.2
```

6.

```
>    # Splitting dataset
> split <- sample.split(mtcars, SplitRatio = 0.8)
> split
 [1]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
```

7.

```
> train_reg <- subset(mtcars, split == "TRUE")
> test_reg <- subset(mtcars, split == "FALSE")
> # Training model
> logistic_model <- glm(vs ~ wt + disp,
+                       data = train_reg,
+                       family = "binomial")
> logistic_model

Call:  glm(formula = vs ~ wt + disp, family = "binomial", data = train_reg)

Coefficients:
(Intercept)           wt         disp
    1.05302      2.94567     -0.05736

Degrees of Freedom: 23 Total (i.e. Null);  21 Residual
Null Deviance:      32.6
Residual Deviance: 14.76        AIC: 20.76
```

8.

```
> # Summary
> summary(logistic_model)

Call:
glm(formula = vs ~ wt + disp, family = "binomial", data = train_reg)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.57373  -0.21664  -0.01282   0.51691   1.91521

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.05302    2.69352   0.391   0.6958
wt           2.94567    2.06932   1.423   0.1546
disp        -0.05736    0.02918  -1.966   0.0493 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 32.601  on 23  degrees of freedom
Residual deviance: 14.756  on 21  degrees of freedom
AIC: 20.756

Number of Fisher Scoring iterations: 7
```

9.

```
> # Predict test data based on model
> predict_reg <- predict(logistic_model,
+                        test_reg, type = "response")
> predict_reg
    Hornet 4 Drive          Duster 360          Merc 280C Cadillac Fleetwood
      1.372999e-02        1.140355e-04        8.282618e-01        2.608907e-05
          Fiat 128   Dodge Challenger          Fiat X1-9     Ford Pantera L
      9.534528e-01        1.093538e-03        9.021914e-01        5.882079e-05
```

10.

```
> # Changing probabilities
> predict_reg <- ifelse(predict_reg >0.5, 1, 0)
>
> # Evaluating model accuracy
> # using confusion matrix
> table(test_reg$vs, predict_reg)
   predict_reg
     0 1
   0 4 0
   1 1 3
```

11.

```
> missing_classerr <- mean(predict_reg != test_reg$vs)
> print(paste('Accuracy =', 1 - missing_classerr))
[1] "Accuracy = 0.875"
```

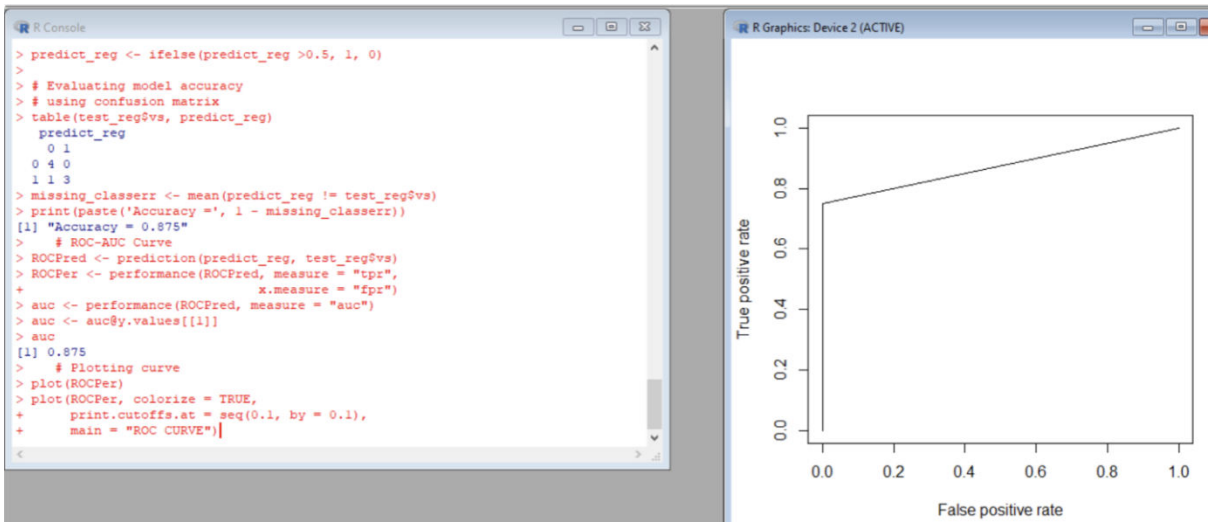
12.

```
>    # ROC-AUC Curve
> ROCPred <- prediction(predict_reg, test_reg$vs)
> ROCPer <- performance(ROCPred, measure = "tpr",
+                                x.measure = "fpr")
```

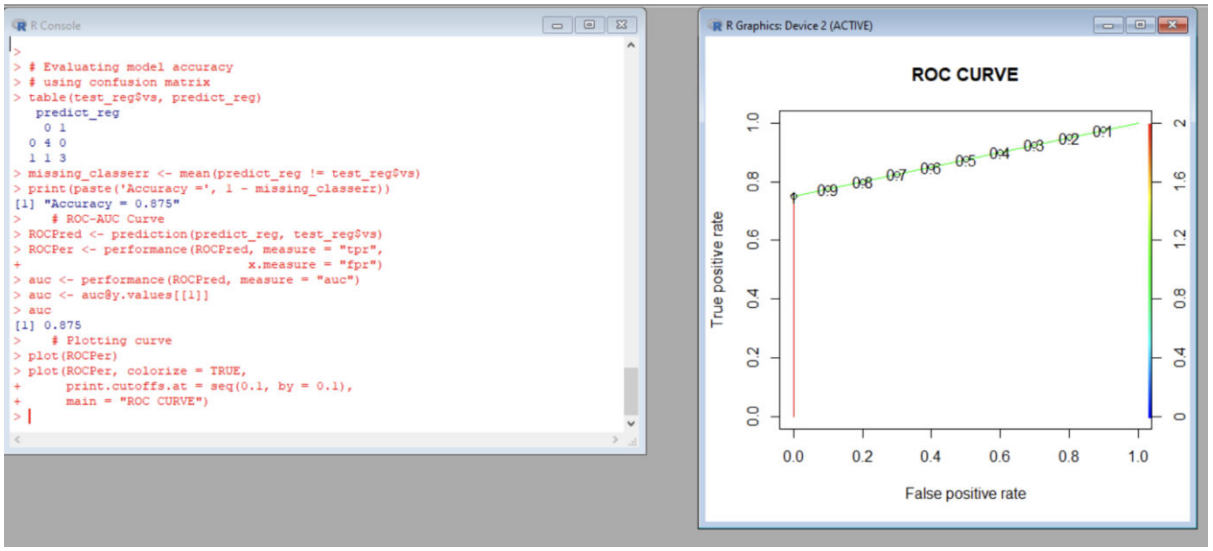
13.

```
> auc <- performance(ROCPred, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.875
```

14.


R Console
> predict_reg <- ifelse(predict_reg >0.5, 1, 0)
>
> # Evaluating model accuracy
> # using confusion matrix
> table(test_reg$vs, predict_reg)
predict_reg
0 1
0 4 0
1 1 3
> missing_classerr <- mean(predict_reg != test_reg$vs)
> print(paste('Accuracy =', 1 - missing_classerr))
[1] "Accuracy = 0.875"
> # ROC-AUC Curve
> ROCPred <- prediction(predict_reg, test_reg$vs)
> ROCPer <- performance(ROCPred, measure = "tpr",
+ x.measure = "fpr")
> auc <- performance(ROCPred, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.875
> # Plotting curve
> plot(ROCPer)
> plot(ROCPer, colorize = TRUE,
+ print.cutoffs.at = seq(0.1, by = 0.1),
+ main = "ROC CURVE")
R Graphics: Device 2 (ACTIVE)
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

15.


R Console
> # Evaluating model accuracy
> # using confusion matrix
> table(test_reg$vs, predict_reg)
predict_reg
0 1
0 4 0
1 1 3
> missing_classerr <- mean(predict_reg != test_reg$vs)
> print(paste('Accuracy =', 1 - missing_classerr))
[1] "Accuracy = 0.875"
> # ROC-AUC Curve
> ROCPred <- prediction(predict_reg, test_reg$vs)
> ROCPer <- performance(ROCPred, measure = "tpr",
+ x.measure = "fpr")
> auc <- performance(ROCPred, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.875
> # Plotting curve
> plot(ROCPer)
> plot(ROCPer, colorize = TRUE,
+ print.cutoffs.at = seq(0.1, by = 0.1),
+ main = "ROC CURVE")
R Graphics: Device 2 (ACTIVE)
ROC CURVE
True positive rate
False positive rate
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
1.0
0
0.4
0.8
1.2
1.6
2

ACTIVITY 10(Lab-05)

Name:- Abhishek Srivastava
Registration Number:- 19BCE10071
Course:- NAS2001- Advanced Data Analytics
Slot:- B21+B22+B23
Faculty:- Dr. Nilamadhab Mishra

Q. Investigate any classification problem for a dataset and try to implement those three algorithms i.e. Decision Tree, Naïve Bayes, K-Nearest Neighbours, and analyse the classification accuracy and other performance factors of each type of algorithm.

1. Decision Tree
    a.

```
> # Load the party package. It will automatically load other
> # dependent packages.
> install.packages("party")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies 'TH.data', 'libcoin', 'matrixStats', 'multcomp$

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/TH.data_1.1-0.zip'
Content type 'application/zip' length 8807519 bytes (8.4 MB)
downloaded 8.4 MB

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/libcoin_1.0-9.zip'
Content type 'application/zip' length 1005142 bytes (981 KB)
downloaded 981 KB

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/matrixStats_0.61.0$
Content type 'application/zip' length 594409 bytes (580 KB)
downloaded 580 KB

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/multcomp_1.4-18.zi$
Content type 'application/zip' length 735393 bytes (718 KB)
downloaded 718 KB
```

    b.

```
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich
Warning messages:
1: package 'party' was built under R version 4.1.2
2: package 'strucchange' was built under R version 4.1.2
3: package 'zoo' was built under R version 4.1.2
4: package 'sandwich' was built under R version 4.1.2
```

c.

```
> # Print some records from data set readingSkills.
> print(head(readingSkills))
  nativeSpeaker age shoeSize    score
1           yes   5 24.83189 32.29385
2           yes   6 25.95238 36.63105
3            no  11 30.42170 49.60593
4           yes   7 28.66450 40.28456
5           yes  11 31.88207 55.46085
6           yes  10 30.07843 52.83124
```

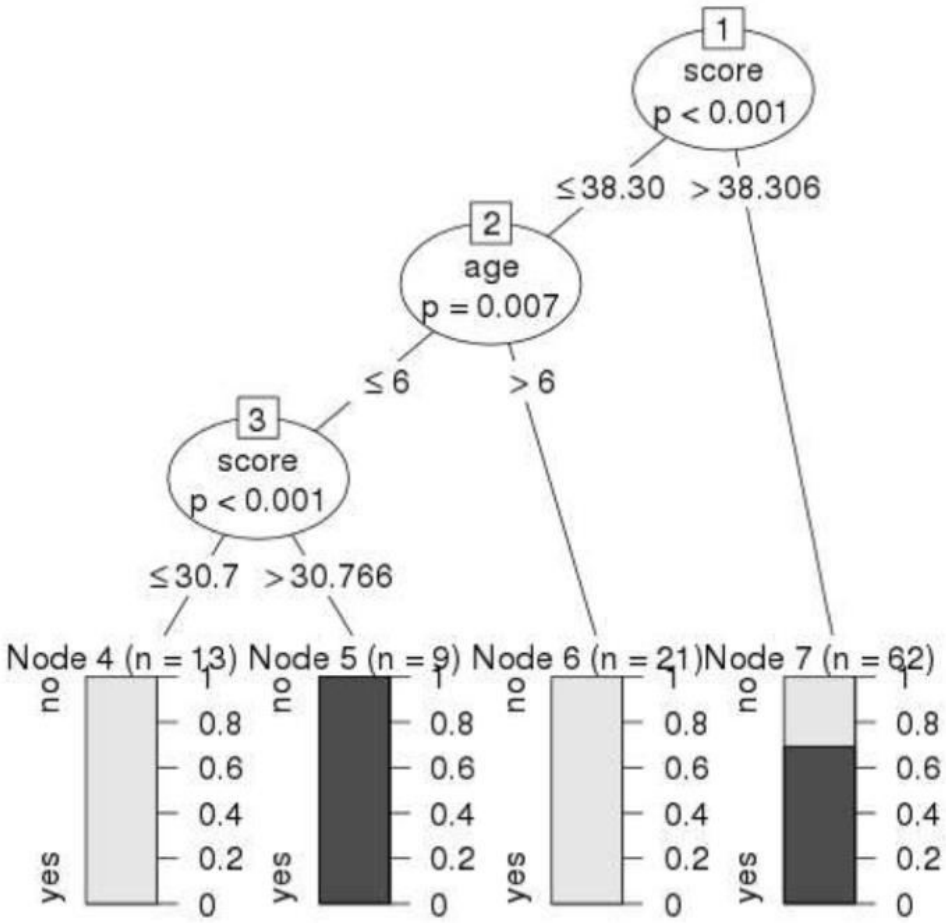
d.

```
> # Create the input data frame.
> input.dat <- readingSkills[c(1:105),]
> print(input.dat)
   nativeSpeaker age shoeSize    score
1            yes   5 24.83189 32.29385
2            yes   6 25.95238 36.63105
3             no  11 30.42170 49.60593
4            yes   7 28.66450 40.28456
5            yes  11 31.88207 55.46085
6            yes  10 30.07843 52.83124
7             no   7 27.25963 34.40229
8            yes  11 30.72398 55.52747
9            yes   5 25.64411 32.49935
10            no   7 26.69835 33.93269
11           yes  11 31.86645 55.46876
12           yes  10 29.15575 51.34140
13            no   9 29.13156 41.77098
14            no   6 26.86513 30.03304
15            no   5 24.23420 25.62268
16           yes   6 25.67538 35.30042
17            no   5 24.86357 25.62843
18            no   6 26.15357 30.76591
19            no   9 27.82057 41.93846
20           yes   5 24.86766 31.69986
21            no   6 25.21054 30.37086
```

e.

```
> # Give the chart file a name.
> png(file = "decision_tree.png")
>
> # Create the tree.
>  output.tree <- ctree(
+   nativeSpeaker ~ age + shoeSize + score,
+   data = input.dat)
>
> # Plot the tree.
> plot(output.tree)
>
```

f.



2. Naïve Bayes

a.

```
> # Loading data
> data(iris)
>
> # Structure
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1$
```

b.

```
> install.packages("e1071")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/e1071_1.7-9.zip'
Content type 'application/zip' length 1023666 bytes (999 KB)
downloaded 999 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\RtmpOilBRk\downloaded_packages
> install.packages("caTools")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/caTools_1.18.2.zip'
Content type 'application/zip' length 316473 bytes (309 KB)
downloaded 309 KB

package 'caTools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\RtmpOilBRk\downloaded_packages
> install.packages("caret")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
```

c.

```
> # Loading package
> library(e1071)
Warning message:
package 'e1071' was built under R version 4.1.2
> library(caTools)
Warning message:
package 'caTools' was built under R version 4.1.2
> library(caret)
Loading required package: ggplot2
Loading required package: lattice
Warning messages:
1: package 'caret' was built under R version 4.1.2
2: package 'ggplot2' was built under R version 4.1.2
>
```

d.

```
> # Splitting data into train
> # and test data
> split <- sample.split(iris, SplitRatio = 0.7)
> train_cl <- subset(iris, split == "TRUE")
> test_cl <- subset(iris, split == "FALSE")
>
> # Feature Scaling
> train_scale <- scale(train_cl[, 1:4])
> test_scale <- scale(test_cl[, 1:4])
```

e.

```
>
> # Fitting Naive Bayes Model
> # to training dataset
> set.seed(120)  # Setting Seed
> classifier_cl <- naiveBayes(Species ~ ., data = train_cl)
> classifier_cl

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Conditional probabilities:
            Sepal.Length
Y                [,1]      [,2]
  setosa     4.973333 0.3084257
  versicolor 5.966667 0.4929386
  virginica  6.520000 0.6764002

            Sepal.Width
```

f.

```
> # Predicting on test data'
> y_pred <- predict(classifier_cl, newdata = test_cl)
>
> # Confusion Matrix
> cm <- table(test_cl$Species, y_pred)
> cm
             y_pred
              setosa versicolor virginica
  setosa          20          0         0
  versicolor       0         19         1
  virginica        0          1        19
```

g.

```
> # Model Evaluation
> confusionMatrix(cm)
Confusion Matrix and Statistics

            y_pred
             setosa versicolor virginica
  setosa         20          0         0
  versicolor      0         19         1
  virginica       0          1        19

Overall Statistics

               Accuracy : 0.9667
                 95% CI : (0.8847, 0.9959)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.95

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9500           0.9500
```

3. K-Nearest Neighbours

a.

```
> df <- data(iris) ##load data
>  head(iris) ## see the studcture
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

b.

```
> ##Generate a random number that is 90% of the total number of rows in dataset.
>  ran <- sample(1:nrow(iris), 0.9 * nrow(iris))
>
> ##the normalization function is created
>  nor <-function(x) { (x -min(x))/(max(x)-min(x))   }
>
> ##Run nomalization on first 4 coulumns of dataset because they are the predic$
>  iris_norm <- as.data.frame(lapply(iris[,c(1,2,3,4)], nor))
```

```
> ##extract training set
> iris_train <- iris_norm[ran,]
> ##extract testing set
>  iris_test <- iris_norm[-ran,]
>  ##extract 5th column of train dataset because it will be used as 'cl' argume$
>  iris_target_category <- iris[ran,5]
```

c.

```
> ##extract 5th column if test dataset to measure the accuracy
>  iris_test_category <- iris[-ran,5]
> ##load the package class
>  library(class)
>
> ##run knn function
>  pr <- knn(iris_train,iris_test,cl=iris_target_category,k=13)
>
>  ##create confusion matrix
>  tab <- table(pr,iris_test_category)
>
> ##this function divides the correct predictions by total number of prediction$
>
>  accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
>  accuracy(tab)
[1] 100
```

d.

```
> ##because diamonds dataset is in ggplot2 package
>  library(ggplot2)
Warning message:
package 'ggplot2' was built under R version 4.1.2
> ##load data
>  data(diamonds)
```

e.

```
> ##store it as data frame
>  dia <- data.frame(diamonds)
>
> ##create a random number equal 90% of total number of rows
>  ran <- sample(1:nrow(dia),0.9 * nrow(dia))
>
> ##the normalization function is created
>  nor <-function(x) { (x -min(x))/(max(x)-min(x))   }
>
> ##normalization function is created
>  dia_nor <- as.data.frame(lapply(dia[,c(1,5,6,7,8,9,10)], nor))
```

f.

```
> ##training dataset extracted
>  dia_train <- dia_nor[ran,]
>
> ##test dataset extracted
>  dia_test <- dia_nor[-ran,]
>
> ##the 2nd column of training dataset because that is what we need to predict $
>  ##also convert ordered factor to normal factor
>  dia_target <- as.factor(dia[ran,2])
```

g.

```
> ##the actual values of 2nd couln of testing dataset to compaire it with value$
>  ##also convert ordered factor to normal factor
>  test_target <- as.factor(dia[-ran,2])
>
> ##run knn function
>  library(class)
>  pr <- knn(dia_train,dia_test,cl=dia_target,k=20)
>
> ##create the confucion matrix
>  tb <- table(pr,test_target)
```

h.

```
> ##check the accuracy
>  accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
>  accuracy(tb)
[1] 71.33852
```

TASK A2

Name- Abhishek Srivastava

Reg no- 19BCE1071

1. Investigate the Attribute or dimensions or features or variables with a suitable scenario and prepare your critical report?
   Nominal
   Binary
   ordinal

   Numeric: quantitative.

ANS-

1. **Nominal Attributes – related to names:** The names of things, or symbols, are the values of a Nominal attribute. Nominal attribute values reflect a category or state, which is why nominal attributes are sometimes known as categorical attributes, and there is no order (rank, position) among nominal attribute values.

Attributes	Values
Colours	Blue,Red,Green,Orange
Categorical Data	Lecturer, professor, assistant professor

2. **Binary Attributes:** Binary data has only 2 values/states. For Example yes or no, affected or unaffected, true or false.

- **Symmetric:** Both values are equally important (Gender).
- **Asymmetric:** Both values are not equally important (Result).

Attribute	Values

Gender	Male, female

Attribute	Values
Cancer Detected	Yes, No
Result	Pass, Fail

3. **Ordinal Attributes:** The Ordinal Attributes comprises values that have a logical sequence or ranking(order) between them, but the magnitude between them is unknown; the order of values indicates what is important but not how important it is.

Attribute	Value
Grade	A,B,C,D,E,F
Pay Scale	17,18,19

4. Numeric: Because it is a measurable quantity represented in integer or real values, a numeric quality is quantitative. There are two sorts of numerical attributes: **interval** and **ratio**.

- The numerical characteristics do not have the correct reference point, or what we can term zero points, while interval-scaled attributes contain values with interpretable differences. On an interval scale, data can be added and subtracted but not multiplied or divided. Consider the temperature in degrees Celsius as an example. We cannot state that one day is twice as hot as another if the temperature of one day is twice that of the other.
- **A numeric attribute** having a fixed zero-point is known as a ratio-scaled attribute. We can claim that a value is a multiple (or ratio) of another value if a measurement is ratio-scaled. The numbers are arranged, and we may compute the difference between them, as well as calculate the mean, median, mode, Quantile-range, and Five-number summary.

Assignment 3

Q1] a) Mean of City A's temp = 94.57.
Mean of City B's temp = 86.14

b) Subtracting the mean from each value:

City A = [0.4, -1.5, 0.4, -0.5, 1.4, -0.5, 0.4]

City B = [3.8, -5.1, 8.8, 4.8, -0.1, -4.1, -8.1]

c) Squaring each value.

City A = [0.16, 2.25, 0.16, 0.25, 1.96, 0.25, 0.16]

City B = [14.4, 26.01, 77.44, 23.04, 0.01, 16.81, 65.61]

d) Average of each squared elements. Sunday 09

City A = [0.74] = VarianceA

City B = [31.89] = VarianceB.

e) standard dev:

City A = 0.86 degrees

City B = 5.64 degrees.

Looking at the obtained results, we can say that City A's forecasts are more reliable than City B.

Q2] a) Mean = 85.2.

b) Subtracting each element →

Score = [-0.2, 0.8, 14.8, -9.2, -4.2, 7.8, -1.2, 13.8, -14.2, -16.2, 7.8, -0.2, -4.2, 1.8, 3.8]

c) Squares →

Score = [0.04, 0.64, 219.04, 84.64, 17.64, 60.84, 1.44, 190.44, 201.64, 262.44, 60.84, 0.04, 17.64, 3.24, 14.44]

d) Mean of squares (Variance)

Variance = 75.6

e) Standard deviation = 8.7.

From the obtained standard deviation, it is known that all students are performing at the same level.

Q3] a) Mean = 8.4

b) differences:

[0.6, -1.4, 1.6, -0.4, 0.6, -1.4, -0.4, 0.6]

c) Squares: [0.36, 1.96, 2.56, 0.16, 0.36, 1.96, 0.16, 0.36]

d) Variance = 1.12

e) Standard deviation = 1.06.

From the standard deviation, ~~we know~~ the researcher knows that the result of the sample sizes are probably reliable.

Abhishek Srivastava

Slot- E21+E22+E23

19BCE10071

Activity- 11

Que. Explore a classification problem case by considering any realworld domain application, formulate a confusion matrix through scenario assumption for the classifier model and investigate the various parameters to measure the performance of the classifier model.

Classification is the process of categorizing a given set of data into classes. In Machine Learning(ML), we frame the problem, collect and clean the data, add some necessary feature variables(if any), train the model, measure its performance, improve it by using some cost function, and then it is ready to deploy.
A great way to measure the performance of a classifier is through the use of Confusion Matrix.

→What is a Confusion Matrix?

A confusion matrix is a summary of prediction results on a classification problem.

The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

The confusion matrix shows the ways in which your classification model is confused when it makes predictions.

It gives you insight not only into the errors being made by your classifier but more importantly the types of errors that are being made.

It is this breakdown that overcomes the limitation of using classification accuracy alone.

→How to Calculate a Confusion Matrix

Below is the process for calculating a confusion Matrix.

1. You need a test dataset or a validation dataset with expected outcome values.
2. Make a prediction for each row in your test dataset.
3. From the expected outcomes and predictions count:
    1. The number of correct predictions for each class.
    2. The number of incorrect predictions for each class, organized by the class that was predicted.

- **Expected down the side**: Each row of the matrix corresponds to a predicted class.
- **Predicted across the top**: Each column of the matrix corresponds to an actual class. The counts of correct and incorrect classification are then filled into the table.

The total number of correct predictions for a class go into the expected row for that class value and the predicted column for that class value.

In the same way, the total number of incorrect predictions for a class go into the expected row for that class value and the predicted column for that class value.

This matrix can be used for 2-class problems where it is very easy to understand, but can easily be applied to problems with 3 or more class values, by adding more rows and columns to the confusion matrix.

➔ Example of 2-Class Confusion Matrix:-

Let's pretend we have a two-class classification problem of predicting whether a photograph contains a man or a woman.

We have a test dataset of 10 records with expected outcomes and a set of predictions from our classification algorithm.

1 Expected, Predicted

2 man, woman

3 man, man

4 woman, woman

5 man, man

6 woman, man
woman

7 woman,

8 woman, woman

9 man, man

10 man, woman
woman

11 woman,

Let’s start off and calculate the classification accuracy for this set of predictions.

The algorithm made 7 of the 10 predictions correct with an accuracy of 70%. accuracy

= total correct predictions / total

1 predictions made * 100
2
accuracy = 7 / 10 * 100

But what type of errors were made?

Let’s turn our results into a confusion matrix.

First, we must calculate the number of correct predictions for each class.

1 men classified as men: 3

2 women classified as women: 4

Now, we can calculate the number of incorrect predictions for each class, organized by the predicted value.

1 men classified as women: 2

2 woman classified as men: 1

We can now arrange these values into the 2-class confusion matrix:

1	men	women
2	men	3 1

3 women 2 4

We can learn a lot from this table.

- The total actual men in the dataset is the sum of the values on the men column (3 + 2) □ The total actual women in the dataset is the sum of values in the women column (1 +4).
- The correct values are organized in a diagonal line from top left to bottom-right of the matrix (3 + 4).
- More errors were made by predicting men as women than predicting women as men.

Name: Abhishek Srivastava

Registration Number: 19BCE10071

ACTIVITY- 5

Investigate the Handling of Redundancy in Data Integration.

Explore the usage of correlation analysis and covariance analysis towards eliminating the redundant attributes along with relevant computations and scenario analysis.

Data Redundancy- Redundancy is another important issue in data integration. An attribute (such as annual revenue, for instance) may be redundant if it can be "derived" from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

- Some redundancies can be detected by correlation analysis. Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the χ 2 (chisquare) test.
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality
- For numeric attributes, we can use the correlation coefficient and covariance, both of which access how one attribute's values vary with those of another.

Correlation Analysis towards eliminating the redundant attributes-

χ 2 Correlation Test for Nominal Data

Example - Suppose that a group of 1,500 people was surveyed. The gender of each person was noted. Each person was polled as to whether their preferred type of reading material was fiction or nonfiction. Thus, we have two attributes, gender and preferred reading. The observed frequency (or count) of each possible joint event is summarized in the contingency table shown in Table below, where the numbers in parentheses are the expected frequencies.

	male	*female*	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

The expected frequencies are calculated based on the data distribution for both attributes using Equation

eij = count (A = ai) × count (B = bj)/n

Using Equation this, we can verify the expected frequencies for each cell. For example, the expected frequency for the cell (male, fiction) is

$$e_{11} = \frac{count(male) \times count(fiction)}{n} = \frac{300 \times 450}{1500} = 90,$$

Name: Abhishek Srivastava

Registration Number: 19BCE10071

and so on. Notice that in any row, the sum of the expected frequencies must equal the total observed frequency for that row, and the sum of the expected frequencies in any column must also equal the total observed frequency for that column. Using Equation

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}},$$

for χ 2 computation, we get

$$\begin{aligned}\chi^2 &= \frac{(250-90)^2}{90}+\frac{(50-210)^2}{210}+\frac{(200-360)^2}{360}+\frac{(1000-840)^2}{840}\\ &= 284.44+121.90+71.11+30.48=507.93.\end{aligned}$$

For this 2 × 2 table, the degrees of freedom are (2 − 1)(2 − 1) = 1. For 1 degree of freedom, the χ 2 value needed to reject the hypothesis at the 0.001 significance level is 10.828 (taken from the table of upper percentage points of the χ 2 distribution, typically available from any textbook on statistics). Since our computed value is above this, we can reject the hypothesis that gender and preferred reading are independent and conclude that the two attributes are (strongly) correlated for the given group of people.

Correlation Coefficient for Numeric Data

For numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient (also known as Pearson's product moment coefficient, named after its inventer, Karl Pearson). This is

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i-\overline{A})(b_i-\overline{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_ib_i)-n\overline{A}\,\overline{B}}{(n-1)\sigma_A\sigma_B}$$

where n is the number of tuples, and are the respective means of A and B, σA and σB are the respective standard deviation of A and B, and Σ(aibi) is the sum of the AB cross-product.

If rA,B > 0, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.

rA,B = 0: independent; rAB < 0: negatively correlated

Covariance of Numeric Data-

Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $$r_{A,B} = \frac{Cov(A,B)}{\sigma_A\sigma_B}$$

Name: Abhishek Srivastava

Registration Number: 19BCE10071

where n is the number of tuples, and are the respective mean or expected values of A and B, σA and σB are the respective standard deviation of A and B.

- Positive covariance: If CovA,B > 0, then A and B both tend to be larger than their expected values.
- Negative covariance: If CovA,B < 0 then if A is larger than its expected value, B is likely to be smaller than its expected value.
- Independence: CovA,B = 0 but the converse is not true.Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence.

Example of Covariance analysis of numeric attributes

Consider Table below, which presents a simplified example of stock prices observed at five time points for AllElectronics and HighTech, some high-tech company. If the stocks are affected by the same industry trends, will their prices rise or fall together?

Time point	*AllElectronics*	*HighTech*
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

Table 3.2: Stock prices for *AllElectronics* and *HighTech*.

$$E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4$$

and

$$E(HighTech) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = \$10.8.$$

Thus, using Equation 3.4, we compute

$$\begin{aligned} Cov(AllElectroncis, HighTech) &= \frac{6\times20+5\times10+4\times14+3\times5+2\times5}{5} - 4 \times 10.8 \\ &= 50.2 - 43.2 = 7. \end{aligned}$$

Therefore, given the positive covariance we can say that stock prices for both companies rise together.

ACTIVITY-08

Formulate the Hypothesis function for Linear Regression and Investigate the computational analysis of the linear regression model to estimate the coefficients for any real-world application.

Hypothesis function for Linear Regression:

$$h\theta = \theta 1 + \theta 2x$$

- Our hypothesis function is exactly the same as the equation of a line using the slope and y-intercept. $y=mx+b$
- θ1: intercept
- θ2: coefficient of x
- Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

Linear regression Applications:

- In linear regression in ML we assume that y and x are related with the following equation: $y = wx+\varepsilon$ Given an input x we would like to compute an output y, where w is a parameter and ε represents measurement error or other noise.
- There are different techniques to estimate the parameters of a model. One of the most popular is the **Ordinary Least Squares**.
- The premise of the Ordinary Least Squares method is to minimize the sum of the squares of the **residuals** of the model. Which is the difference between the predicted values and the actual values in the dataset.
- This way the model is calculating the best parameters, so that each point in the regression line is as *close* as possible to the dataset.

using a dataset from the National Basketball Association (NBA). This dataset includes salary information and points scored during the season for each player in the 2017–2018 season. We'll be investigating the relationship between points scored in a season and the salary of a player.

Player	PTS	Salary
James Harden	2,376	$ 28,299,399.00
Stephen Curry	2,375	$ 34,682,550.00
Tobias Harris	2,232	$ 16,000,000.00
Kevin Durant	2,029	$ 25,000,000.00
Joe Johnson	1,984	$ 10,254,905.00
Ish Smith	1,944	$ 6,000,000.00
LeBron James	1,920	$ 33,285,709.00
Damian Lillard	1,879	$ 26,153,057.00
Jeff Green	1,878	$ 2,116,955.00
Russell Westbrook	1,878	$ 28,530,608.00
Paul George	1,874	$ 19,508,958.00
DeMar DeRozan	1,830	$ 27,739,975.00
Isaiah Thomas	1,823	$ 6,261,395.00
Klay Thompson	1,771	$ 17,826,150.00

by running a simple regression model with *salary* as our dependent variable and *points* as our independent variable. The output of this regression model is below:

```
Call:
lm(formula = salary ~ points, data = nba)

Residuals:
      Min        1Q    Median        3Q       Max
-18777226  -4252904  -1077216   3985820  20171202

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1677561.9   623636.2    2.69   0.0075 **
points        10232.5      724.9   14.12   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Residual standard error: 6283000 on 334 degrees of freedom
Multiple R-squared:  0.3737,    Adjusted R-squared:  0.3718
F-statistic: 199.3 on 1 and 334 DF,  p-value: < 2.2e-16
```

The call section shows us the formula that R used to fit the regression model. *Salary* is our dependent variable and we are using *points* as a predictor (independent variable) from the NBA dataset.

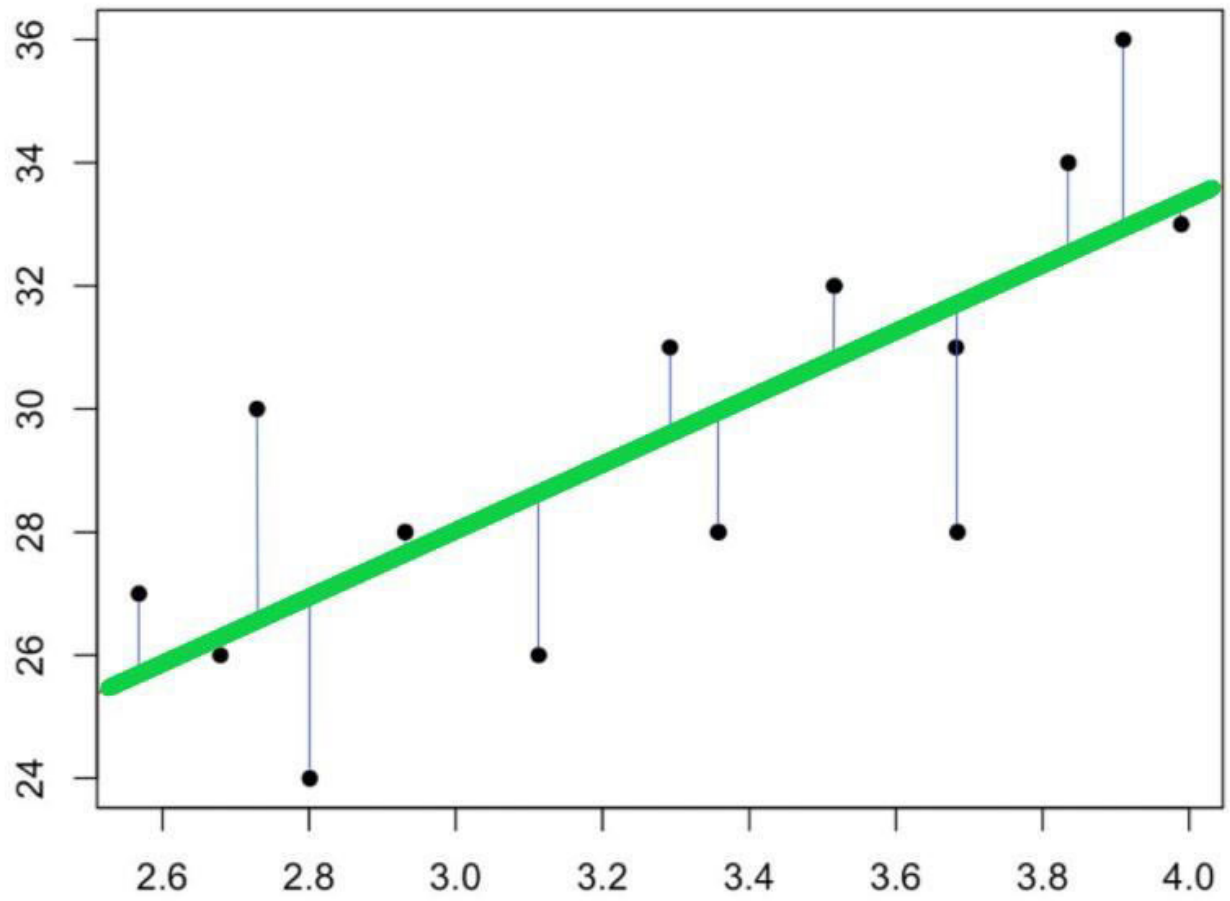
The residuals are the difference between the actual values and the predicted values. We can generate these same values by taking the actual values of *salary* and subtracting it from the predicted values of the model:

```
summary(nba$salary - model$fitted.values)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-18777226  -4252904  -1077216         0   3985820  20171202
```

To understand what the coefficients are, we need to go back to what we are actually trying to do when we build a linear model. We are looking to build a generalized model in the form of $y=mx+b$, where b is the intercept and m is the slope of the line. Because we often don't have enough information or data to know the exact equation that exists in the wild, we have to build this equation by generating estimates for

both the slope and the intercept. These estimates are most often generated through the ordinary least squares method.



It is from this line above that we obtain our coefficients. Where the line meets the y-axis is our intercept (*b*) and the slope of the line is our *m*. Using the understanding we've gained so far, and the estimates for the coefficients provided in the output above, we can now build out the equation for our model. We'll substitute *points* for *m* and *(Intercept)* for *b:*

y=$10,232.50(x) + $1,677,561.90

if an NBA player scored zero points during a season, that player would make $1,677,561.90 on average. Then, for each additional point they scored during the season, they would make $10,232.50.

Coefficients — Std. Error

The standard error of the coefficient is an estimate of the standard deviation of the coefficient. In effect, it is telling us how much uncertainty there is with our coefficient. The standard error is often used to create confidence intervals. For example we can make a 95% confidence interval around our slope, *points:*

$10,232.50 ± 1.96($724.90) =($8,811.70,$11,653.30)

Looking at the confidence interval, we can say we are 95% confident that the actual slope is between $8,811.70 and $11,653.30.

ACTIVITY-09(Lab-04)

Name:- Abhishek Srivastava
Reg. No:-19BCE10071
Slot:- A21+A22+A23

Q. Implement one-way and two-way ANOVA by considering any scenario and investigating the computational analysis of one-way and two-way ANOVA to estimate the P-value to take a decision.

Performing One Way ANOVA test in R

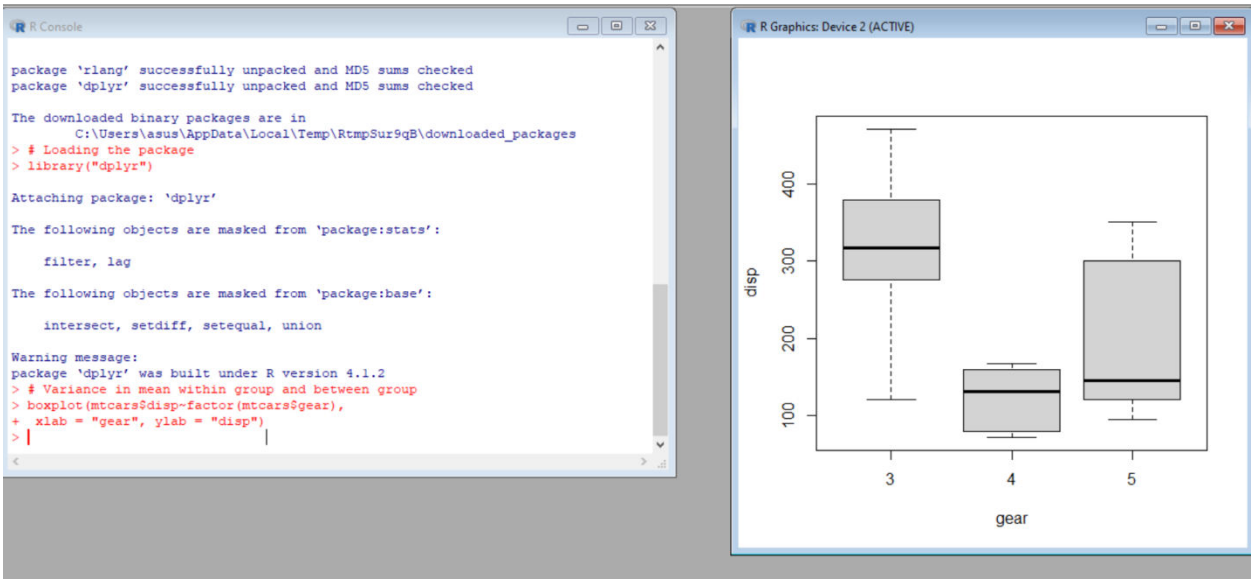
1.

```
> # Installing the package
> install.packages("dplyr")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
also installing the dependency 'rlang'

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/rlang_1.0.1.zip'
Content type 'application/zip' length 1611012 bytes (1.5 MB)
downloaded 1.5 MB

trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/dplyr_1.0.8.zip'
Content type 'application/zip' length 1380128 bytes (1.3 MB)
downloaded 1.3 MB

package 'rlang' successfully unpacked and MD5 sums checked
package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\RtmpSur9qB\downloaded_packages
```

2.

```
> # Loading the package
> library("dplyr")

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Warning message:
package 'dplyr' was built under R version 4.1.2
```

3.



4.

```
> # Step 1: Setup Null Hypothesis and Alternate Hypothesis
> # H0 = mu = mu01 = mu02(There is no difference
> # between average displacement for different gear)
> # H1 = Not all means are equal
```

5.

```
> # Step 2: Calculate test statistics using aov function
> mtcars_aov <- aov(mtcars$disp~factor(mtcars$gear))
> summary(mtcars_aov)
                    Df Sum Sq Mean Sq F value   Pr(>F)
factor(mtcars$gear)  2 280221  140110   20.73 2.56e-06 ***
Residuals           29 195964    6757
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.

```
> # Step 3: Calculate F-Critical Value
> # For 0.05 Significant value, critical value = alpha = 0.05
> # Step 4: Compare test statistics with F-Critical value
> # and conclude test p < alpha, Reject Null Hypothesis
```

Performing Two Way ANOVA test in R

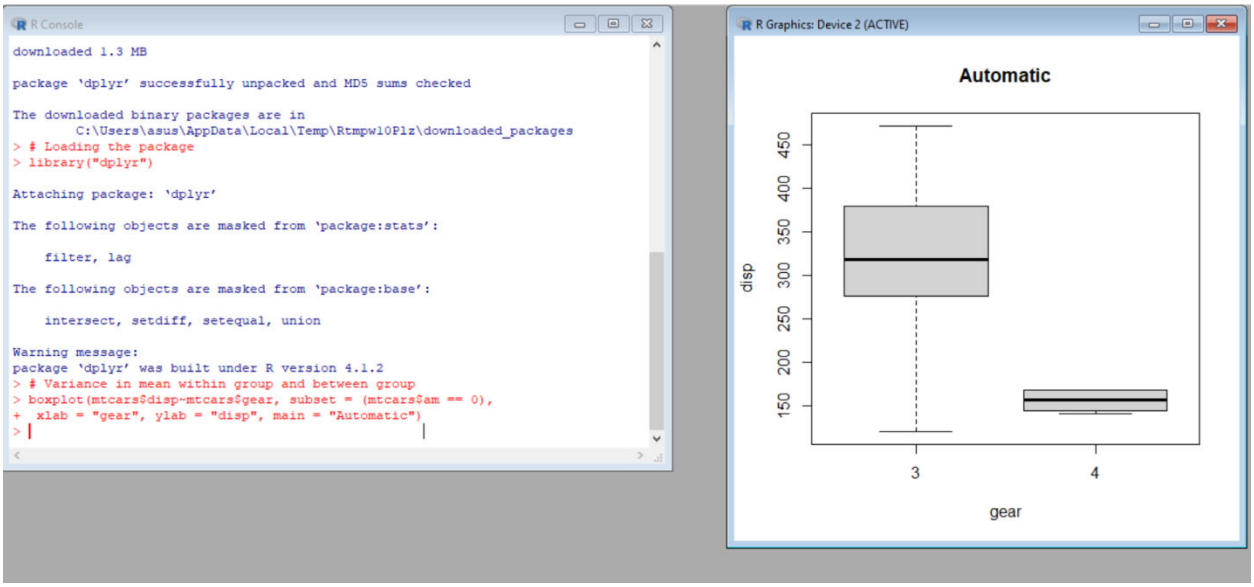
1.

```
> # Installing the package
> install.packages("dplyr")
Installing package into 'C:/Users/asus/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cran.seoul.go.kr/bin/windows/contrib/4.1/dplyr_1.0.8.zip'
Content type 'application/zip' length 1380128 bytes (1.3 MB)
downloaded 1.3 MB

package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\Rtmpw10Plz\downloaded_packages
```

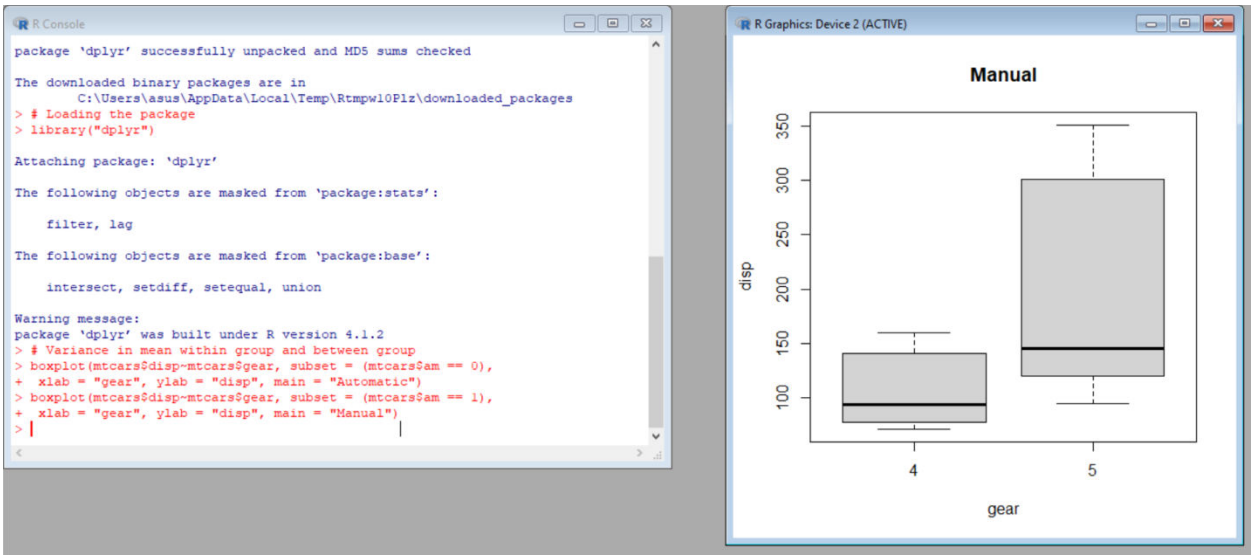
2.

```
> # Loading the package
> library("dplyr")

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Warning message:
package 'dplyr' was built under R version 4.1.2
```

3.



4.



5.

```
> # Step 1: Setup Null Hypothesis and Alternate Hypothesis
> # H0 = mu0 = mu01 = mu02(There is no difference between
> # average displacement for different gear)
> # H1 = Not all means are equal
```

6.

```
> # Step 2: Calculate test statistics using aov function
> mtcars_aov2 <- aov(mtcars$disp~factor(mtcars$gear) *
+   factor(mtcars$am))
> summary(mtcars_aov2)
                     Df Sum Sq Mean Sq F value   Pr(>F)
factor(mtcars$gear)   2 280221  140110  20.695 3.03e-06 ***
factor(mtcars$am)     1   6399    6399   0.945    0.339
Residuals            28 189565    6770
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.

```
> # Step 3: Calculate F-Critical Value
> # For 0.05 Significant value, critical value = alpha = 0.05
> # Step 4: Compare test statistics with F-Critical value
> # and conclude test p < alpha, Reject Null Hypothesis
```

```
bcdedit/set hypervisorlaunchtype off
```

Activity-17

NAME- Abhishek Srivastava
Regno.- 19BCE10071
Slot- E21+E22+E23
Faculty:- Dr. Nilamdhab Sir

Q.
1. Using K-means clustering, cluster the following data into two clusters and show each step.
{3, 5, 10, 13, 4, 21, 31, 12, 26}.
Give your step-by-step computational analysis.
2. Formulate any four cluster scenarios to Calculate purity to measure the quality of each cluster.

3. Investigate the computational processes of the K-medoid algorithm with a suitable scenario.
1.
Solution: (Method-1)
Given: {3, 5, 10, 13, 4, 21, 31, 12, 26}

Step 1: Assign alternate value to each cluster randomly.

Step 2: k1= {3, 10, 4, 31, 26} Mean value= 14.8

k2= {5, 13, 21, 12} Mean value = 12.75

Step 3: Again, assign the values,

k1 = {21, 31, 26} Mean value = 26

k2= {3, 5, 10, 13, 4, 12} Mean value = 7.83

Step 4: Again, assign the values,

k1 = {21, 31, 26} Mean value = 26

k2= {3, 5, 10, 13, 4, 12} Mean value = 7.83
Step-2:
K1 cluster having cluster centroid C1 = 14.8

K2 cluster having Cluster centroid C2 = 12.75

Computation to move from Step-2 to Step-3.

In step-2, The clusters centroid values are as follows:

C1= 14.8 of K1 cluster
C2 = 12.75 of K2 cluster
Now Consider each data point from K1 and k2 clusters,

compute the distance from C1 and C2,
consider the minimum distance, and assign the respective data point to the cluster

k1 or k2.
Ex: for data point '2': Min (| 2-14.8 | , | 2-12.75 |) = 10.75

so, data point '2' assigns to cluster K2 having centroid C2[K2/C2].
(Method-2)
{3, 5, 10, 13, 4, 21, 31, 12, 26}
Step 1: Randomly assign the means: m1 = 4, m2 = 5

Step 2: Group the numbers close to mean m1 = 4 are grouped into cluster

k1 and m2 = 5 are grouped into cluster k2
Step 3: k1 = {3, 4}, k2 = {5, 10, 13, 21, 31, 12, 26}, m1= 3.5, m2 = 16.8

Step 4: k1 = {3, 4, 5}, k2 = {10, 13, 21, 31, 12, 26}, m1= 4 m2 = 18.8

Step 5: k1 = {3, 4, 5, 10}, k2= {13, 21, 31, 12, 26}, m1= 5.5, m2 = 20.6

Step 6: k1 = {3, 4, 5, 10, 12, 13}, k2 = {21, 31, 26}, m1 = 7.8, m2 = 26

Step 7: k1 = {3, 4, 5, 10, 12, 13}, k2 = {21, 31, 26}, m1 = 7.8, m2 = 26

Step 8: Stop. The clusters in step 6 and 7 are same.

Final answer: k1 = {3, 4, 5, 10, 12, 13} and k2 = {21, 31, 26}
2. Solution:
Assume that we cluster three category of data items (those colored with red, blue and green) into three clusters.
Calculate purity to measure the quality of each cluster.
Cluster I: 5 red, 1 blue, 0 green

Cluster II: 1 green, 4 blue, 1 red

Cluster III: 2 red, 0 blue, 3 green

Cluster IV: 3 red, 2 blue, 1 green

Cluster I: Purity = 1/6 (max (5, 1, 0)) = 5/6 = 83%

Cluster II: Purity = 1/6 (max (1, 4, 1)) = 4/6 = 67%

Cluster III: Purity = 1/5 (max (2, 0, 3)) = 3/5 = 60%

Cluster IV: Purity = 1/6(max (3,2,1)) = 3/6= 50%

3.

K-Medoids (also called as Partitioning Around Medoid) algorithm:

A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.

The dissimilarity of the medoid (Ci) and object (Pi) is calculated by using E = |Pi - Ci|

The cost in K-Medoids algorithm is given as

.

Algorithm:

1. Initialize: select k random points out of the n data points as the medoids.

2. Associate each data point to the closest medoid by using any common distance metric methods.

3. While the cost decreases:

For each medoid m, for each data o point which is not a medoid:

1. Swap m and o, associate each data point to the closest medoid, recompute the cost.

$$c = \sum_{Ci} \sum_{Pi \in Ci} |Pi - Ci|$$

2. If the total cost is more than that in the previous step, undo the swap.

Let’s consider the following example:

If a graph is drawn using the above data points, we obtain the following:

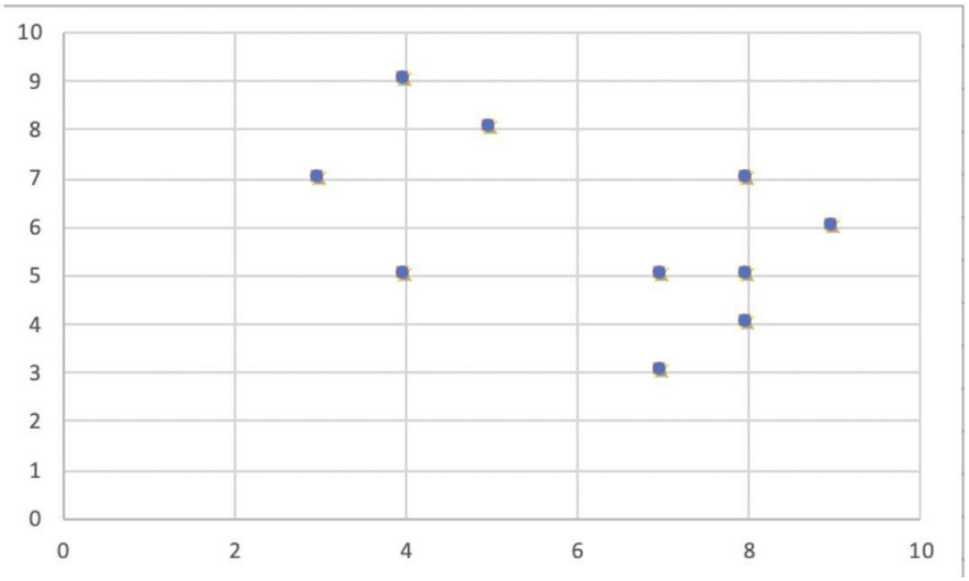
Step 1:

Let the randomly selected 2 medoids, so select k = 2 and let C1 -(4, 5) and **C2 -(8, 5)** are the two medoids.

Step 2: Calculating cost.

The dissimilarity of each non-medoid point with the medoids is calculated and

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5



tabulated:

Each point is assigned to the cluster of that medoid whose dissimilarity is less.

The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

Step 3: randomly select one non-medoid point and recalculate the cost.

Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.

Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22

Swap Cost = New Cost – Previous Cost = 22 – 20 and **2 >0**

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	8	4	-	-
8	7	5	3	2
9	4	5	-	-

As the swap cost is not less than zero, we undo the swap. Hence (3, 4) and (7, 4) are the final medoids. The clustering would be in the following way

The **time complexity** is

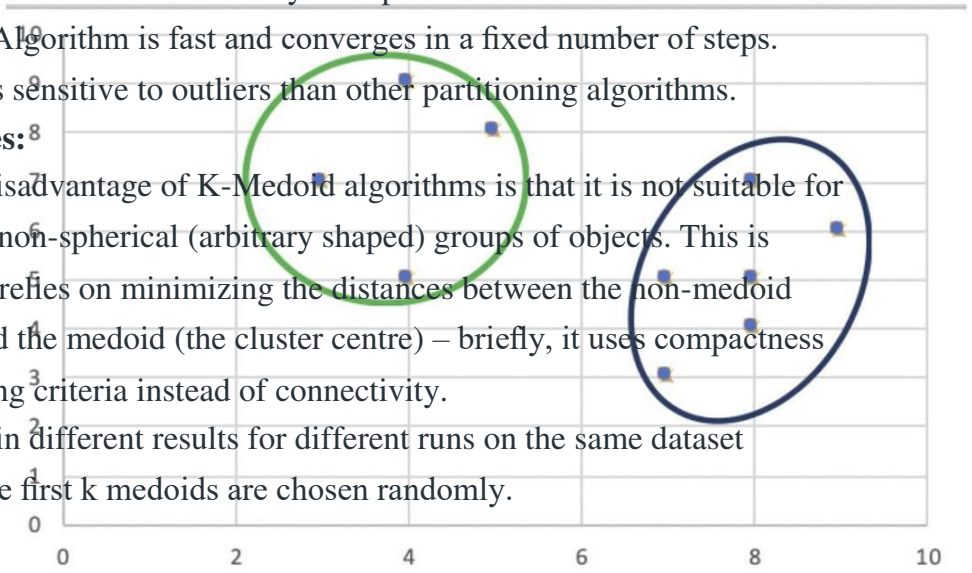
.

Advantages:

1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. PAM is less sensitive to outliers than other partitioning algorithms.

Disadvantages:

4. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.
5. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.



$$O(k * (n-k)^2)$$

Investigate the R implements of mean, median, standard deviation, variance, correlation, and covariance.

Mean

It is calculated by taking the sum of the values and dividing with the number of values in a data series.

The function mean() is used to calculate this in R.

Syntax

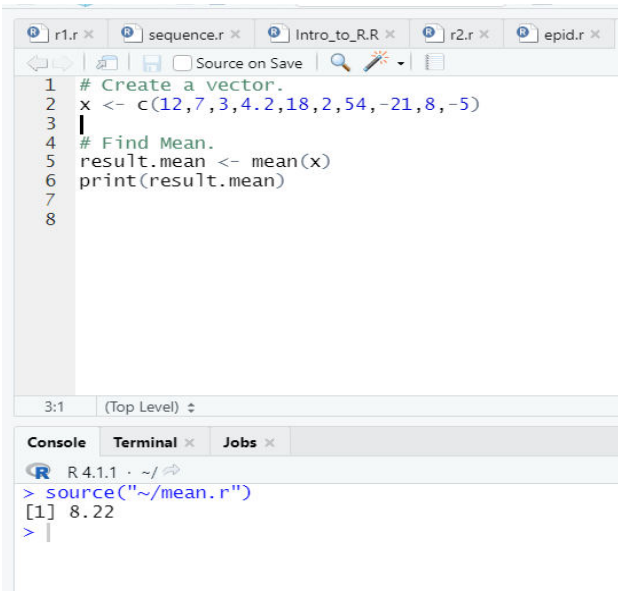
The basic syntax for calculating mean in R is −

mean(x, trim = 0, na.rm = FALSE, ...)

Following is the description of the parameters used −

- x is the input vector.
- trim is used to drop some observations from both end of the sorted vector.
- na.rm is used to remove the missing values from the input vector.

Example :



Median

The middle most value in a data series is called the median. The median() function is used in R to calculate this value.

Syntax

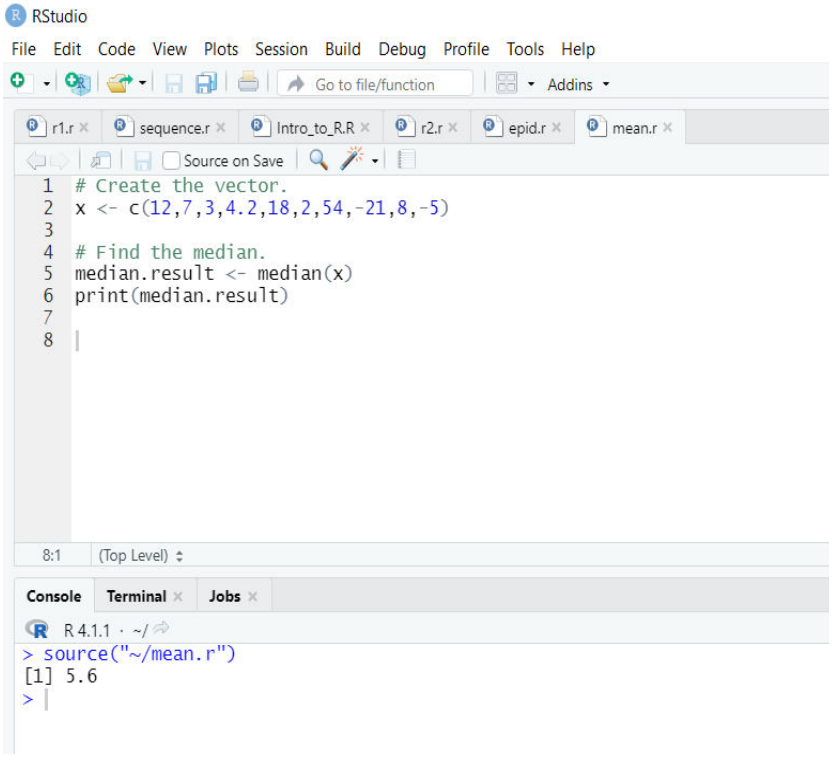
The basic syntax for calculating median in R is −

median(x, na.rm = FALSE)

Following is the description of the parameters used −

- x is the input vector.
- na.rm is used to remove the missing values from the input vector.

EXAMPLE

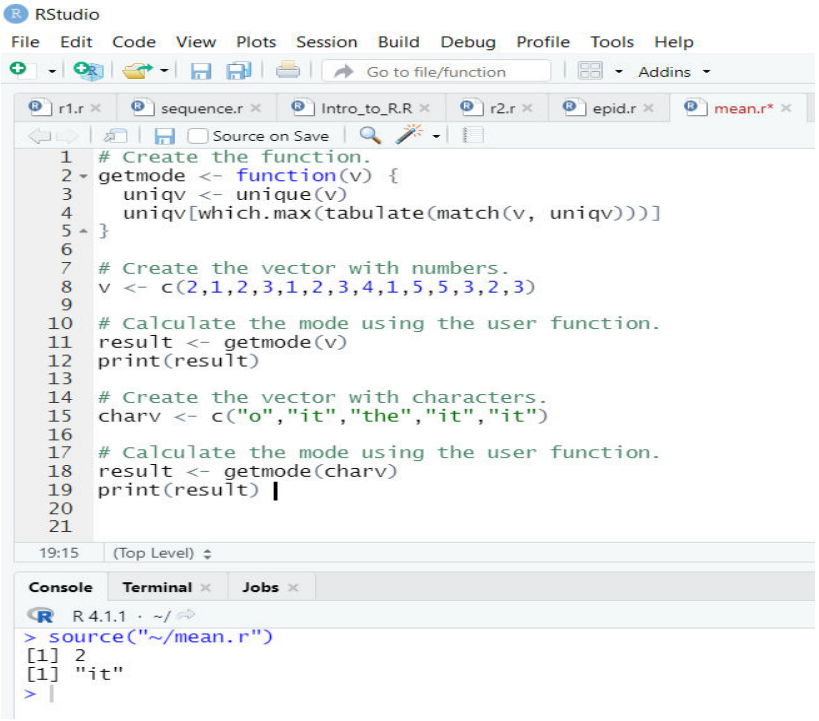


Mode

The mode is the value that has highest number of occurrences in a set of data. Unike mean and median, mode can have both numeric and character data.

R does not have a standard in-built function to calculate mode. So we create a user function to calculate mode of a data set in R. This function takes the vector as input and gives the mode value as output.

EXAMPLE:



Standard Deviation

sd() Function

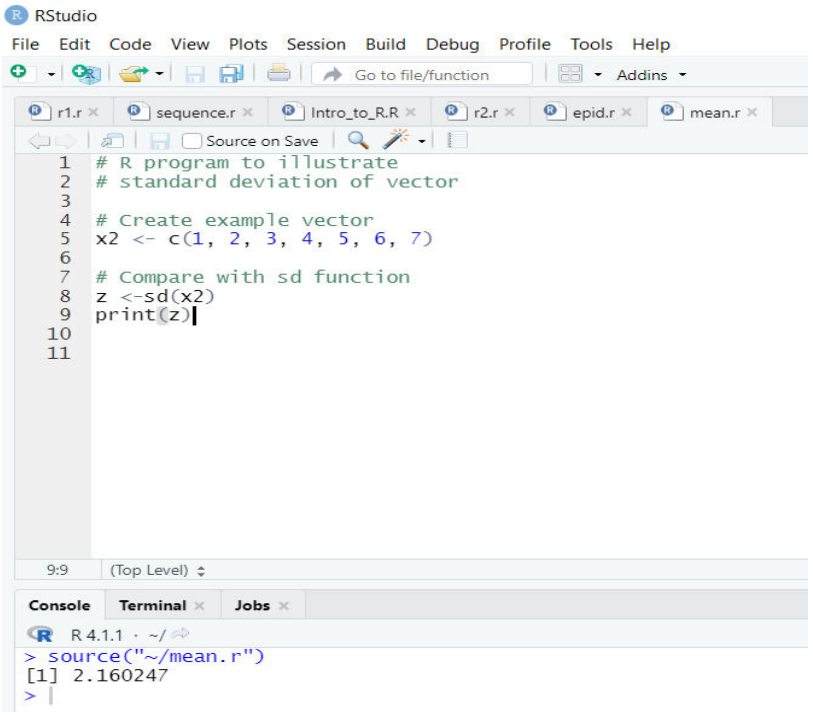
sd() function is used to compute the standard deviation of given values in R. It is the square root of its variance.

Syntax: sd(x)

Parameters:

x: numeric vector

EXAMPLE



VARIANCE

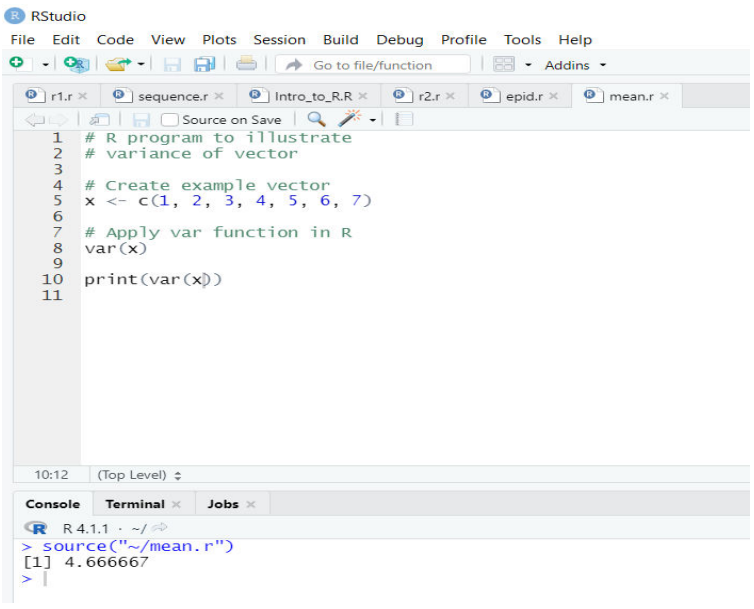
var() function in R Language computes the sample variance of a vector. It is the measure of how much value is away from the mean value.

Syntax: var(x)

Parameters:

x : numeric vector

EXAMPLE:



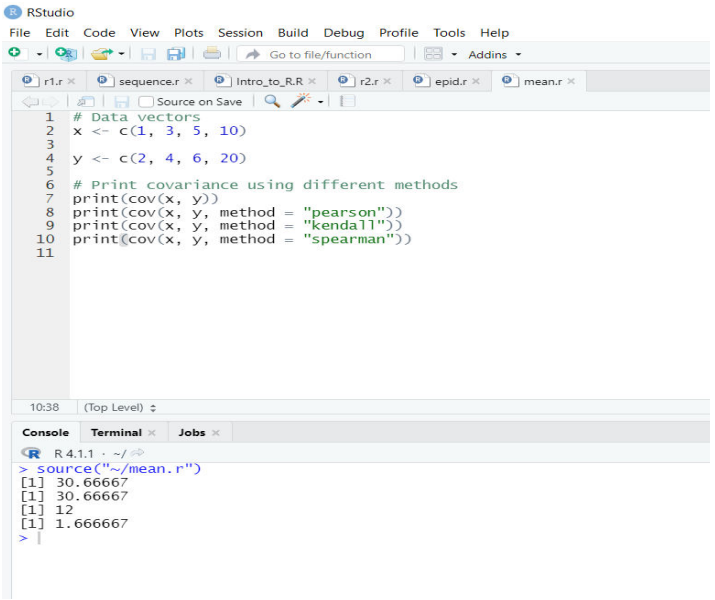
Covariance

In R programming, covariance can be measured using cov() function. Covariance is a statistical term used to measures the direction of the linear relationship between the data vectors.

Syntax: cov(x, y, method)

where,

- x and y represents the data vectors
- method defines the type of method to be used to compute covariance. Default is "pearson".

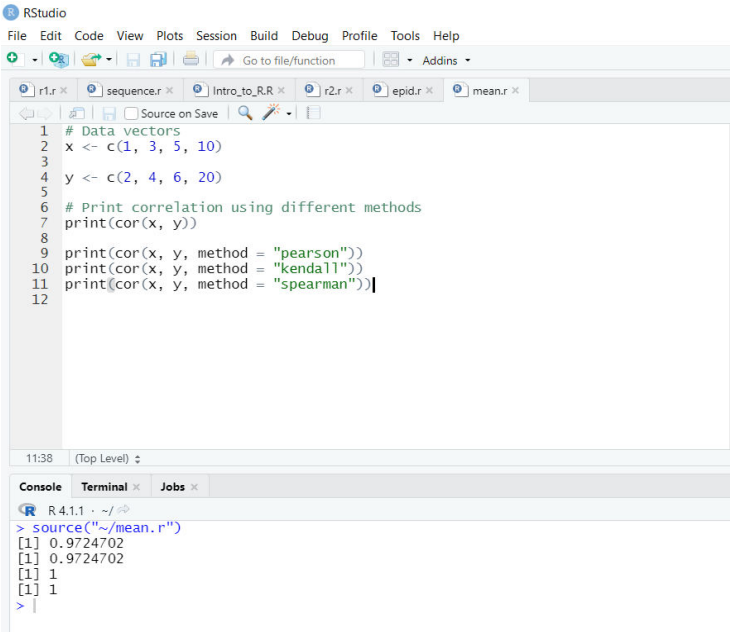


Correlation

Syntax: cor(x, y, method)

where,

- x and y represents the data vectors
- method defines the type of method to be used to compute covariance. Default is "pearson".


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
r1.r
sequence.r
Intro_to_R.R
r2.r
epid.r
mean.r
Source on Save
1 # Data vectors
2 x <- c(1, 3, 5, 10)
3
4 y <- c(2, 4, 6, 20)
5
6 # Print correlation using different methods
7 print(cor(x, y))
8
9 print(cor(x, y, method = "pearson"))
10 print(cor(x, y, method = "kendall"))
11 print(cor(x, y, method = "spearman"))
12
11:38
(Top Level)
Console
Terminal
Jobs
R 4.1.1 · ~/
> source("~/mean.r")
[1] 0.9724702
[1] 0.9724702
[1] 1
[1] 1
>

Activity 1

Advance Data Analytics, NAS2001, E21

NAME ⇒ Abhishek Srivastava

REGISTRATION NO. ⇒ 19BCE10071

Data mining and Analysis

Data Mining is the process of extracting important patterns from large datasets. Simply, it is a process that is used to turn raw data into meaningful data. It is otherwise also called Knowledge Discovery in Databases (KDD). The improvement in computing prowess has allowed data mining to become streamlined and mainstream. It helps the organizations build more innovative strategies, increase sales, generate revenue, and grow a business by cost reduction.

Eg - The best example of a data mining application is in the E-commerce sector, where websites display options for those who purchased and viewed the specific product.

Data Analysis is the process of analyzing and organizing raw data to determine useful information and decisions. It is a superset of Data Mining, which includes removing, cleaning, changing, demonstrating the data to reveal significant and valuable insights.

Eg. - The best example of data analysis is the study of the census.
Additionally, Data Mining and Data Analytics are different words but they indeed are very similar. Data mining is a step in the process of data analytics. Data Analytics is the umbrella that deals with every step in the pipeline of any data-driven model.
Before we jump into the Core Tasks, Applications & Algorithms of both the terms, let's look

into some identities through which we can find some differences between both terms.

- Data mining is catering the data collection and deriving crude but essential insights. Data analytics then uses the data and crude hypothesis to build upon

that and create a model based on the data.

- Data mining shines its brightest when the data in question is well structured. Meanwhile, data analysis can be performed on any data; it would still be able to derive meaningful insights that could help in propelling the corporation to even greater heights.

- Data mining is tasked to accomplish the main job to make the data that is being used more usable. Whereas, data analysis is used to hypothesize and, in the end, culminate itself in providing valuable information to help in business decisions.

- 4. Data mining does not need any bias or any notions which are instilled before tackling the data. Whereas, data analysis is majorly used for hypothesis testing.

- 5. Data mining uses scientific and mathematical models and methods to identify patterns or trends in the data that is being mined. On the other hand, data analysis is employed to task with business analytics problems and derive analytical models.

- 6. Data mining usually does not need any visualizations, bar charts, graphs, GIPs,
  etc., whereas these visualizations /are the bread and butter of data analysis. Without a good representation of the data in question, all the efforts which are put into the analysis of the data would not come to fruition.

Core Tasks

Data Mining: There are several data mining tasks such as classification, prediction, time-series analysis, association rule learning, clustering, summarization, etc. All these tasks are either predictive data mining tasks or descriptive data mining tasks. A data mining system can execute one or more of the above-specified tasks as part of data mining.

Data Analytics: The tasks in Data analytics depend on the data-driven decision-making practices adopted by the organization. Some of the general tasks are mentioned below:

1. Designing and maintaining data systems and databases

2. Mining data from primary and secondary sources
3. Using statistical tools to interpret data sets, paying particular attention to trends and patterns could be valuable for diagnostic and predictive analytics efforts.

Generally, there exist 2 methods through which tasks can be performed: Qualitative research and Quantitative research.

Application

Data Mining: There exist many applications of data mining, but some major ones are mentioned below:

1. Financial Analysis
2. Telecommunication Industry
3. Intrusion Detection
4. Retail Industry
5. Higher Education
6. Energy Industry
7. Spatial Data Mining
8. Biological Data Analysis
9. Other Scientific Applications
10. Manufacturing Engineering
11. Criminal Investigation
12. Counter-Terrorism

Data Analytics: Some of the major applications are mentioned below:

1. Transportation - strategies to plan alternative routes, reduce congestions and traffics, optimize the buyer's experience in the travels.
2. Logistics and Delivery - best shipping routes, approximate delivery times, real-time status of goods.
3. Web Search or Internet Web Results - The searched data is considered as a keyword and all the related pieces of information are presented in a sorted manner that one can easily understand.
4. Manufacturing - prediction analysis, regression analysis, budgeting, etc.
5. Security - Security Analytics, identify danger before it gets an

opportunity to affect your framework.
6. Education - adaptive learning, innovations, adaptive content, etc.
7. Healthcare - channel enormous measures of information in seconds to discover treatment choices or answers for various illnesses.
8. Military - augmented reality and psychological science

Algorithms

Data Mining: There are many algorithms but we'll mention a few.

1. C4.5 Algorithm
2. K-mean Algorithm
3. Support Vector Machines
4. Apriori Algorithm
5. Expectation-Maximization Algorithm
6. PageRank Algorithm
7. Adaboost Algorithm
8. KNN Algorithm
9. Naive Bayes Algorithm
10. CART Algorithm

Data Analytics: For data analytics, first we'll see the types of analytics -

1. Descriptive analytics examines what happened in the past: Monthly revenue, quarterly sales, yearly website traffic, and so on. These types of findings allow an organization to spot trends.
2. Diagnostic analytics considers why something happened by comparing descriptive data sets to identify dependencies and patterns. This helps an organization
determine the cause of a positive or negative outcome.
3. Predictive analytics seeks to determine likely outcomes by detecting tendencies in descriptive and diagnostic analyses. This allows an organization to take proactive action—like reaching out to a customer who is unlikely to renew a contract, for example.
4. Prescriptive analytics attempts to identify what business action to take. While this type of analysis brings significant value in the ability to address potential problems or stay ahead of industry trends, it often

requires the use of complex algorithms and advanced technology such as machine learning.

As data analytics is a superset of data mining, hence to perform the type of analysis on data, we use the same types of algorithms as in data mining. The best 5 algorithms in analytics for big data are Linear Regression, Logistics Regression, CART, KNN, and K-Means.

References

- Top 10 Most Common Data Mining Algorithms You Should Know | upgrade a blog
- Top 8 Applications Of Data Analytics to look out for in 2021 (jigsawacademy.com)
- 5 Advanced Analytics Algorithms for Your Big Data Initiatives | Transforming Data with Intelligence (tdwi.org)
- Data Mining Tasks | Data Mining tutorial by Wideskills
- 13 Really Cool Quotes About Data | The TIBCO Blog

Matters of Discussion

Brief Evolution of DSA

DW-OLAP

KDD

Data mining& analytics

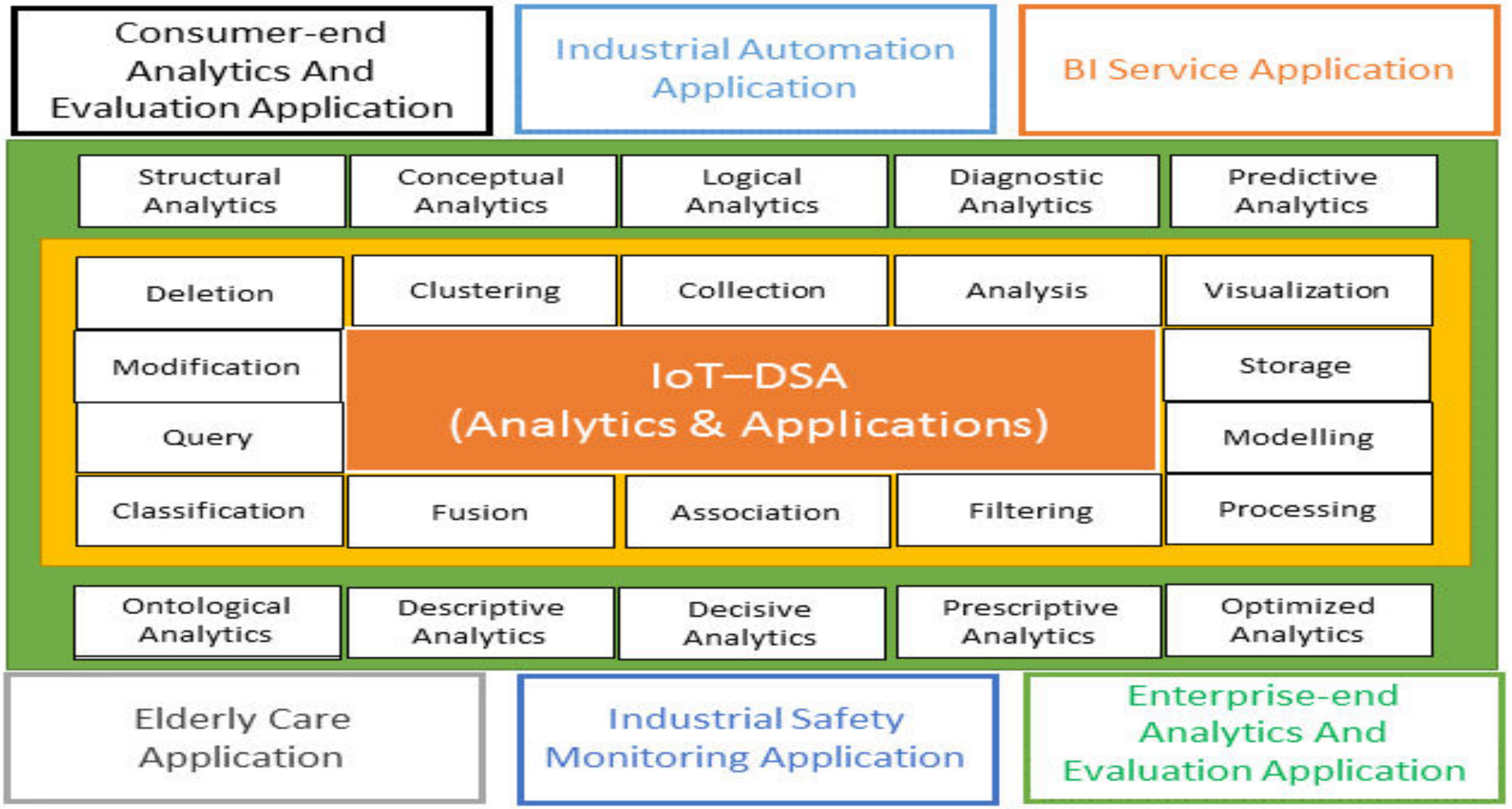
Core Tasks, Apps & Algorithms

Compiled By: Dr. Nilamadhab Mishra [(PhD- CSIE) Taiwan]

Brief Evolution of DSA

Year/Duration	Features Included in DSA
1960	Data science as a substitute of CS
1974	DS as data processing methods
1977	Exploratory data analysis
1989-1996	Data classification, mining, and knowledge discovery
1997-2001	Statistical computing, KDD
2005	Analytics and fact based decision
2010-11	Statistics & machine learning
2012 to till date	IoTA, Cognitive learning, Big data analytics.

DSA For Analytics and Applications



Information Framework for IoT-DSA.

TECHNOLOGIES & SUPPORTED TOOLS/ PROCESSES/ FRAMEOWRKS/ TASKS/ APPLICATIONS

DW-OLAP

- **Like SQL in DBMS**
- OLAP is the dynamic synthesis, and analysis of large volumes of multi-dimensional data.
- OLAP uses multi-dimensional view of aggregate data to make forecasting.
- **OLAP finds- what is happening?**

Multi-dimensional data

City	Time	Total Revenue
Glasgow	Q1	29726
Glasgow	Q2	30443
Glasgow	Q3	30582
Glasgow	Q4	31390
London	Q1	43555
London	Q2	48244
London	Q3	56222
London	Q4	45632
Aberdeen	Q1	53210
Aberdeen	Q2	34567
Aberdeen	Q3	45677
Aberdeen	Q4	50056
.....	.....	.....
.....	.....	.....

(a)

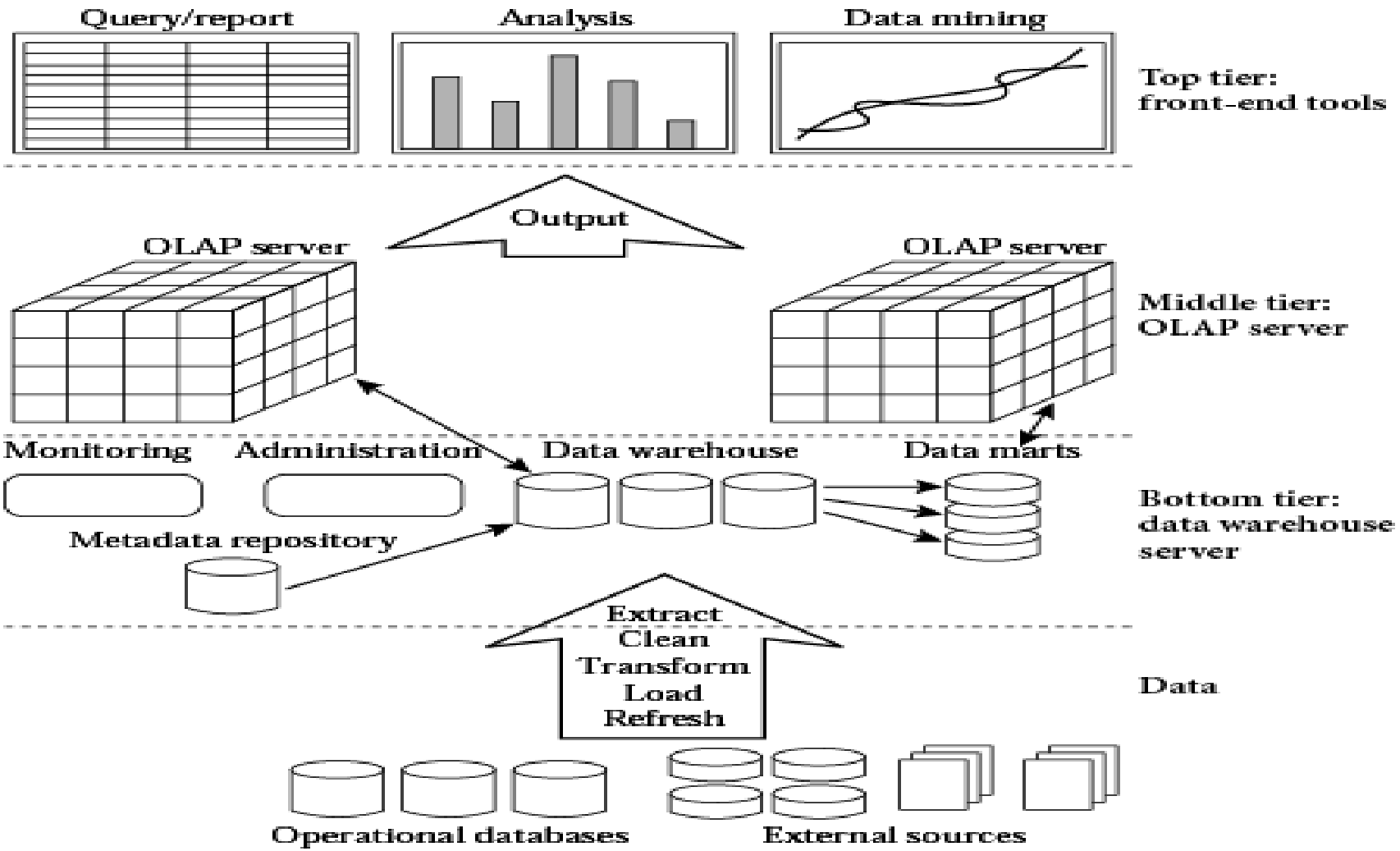
City

Time \ City	Glasgow	London	Aberdeen	.....
Q1	29726	43555	53210	.....
Q2	30443	48244	34567	.....
Q3	30582	56222	45677	.....
Q4	31390	45632	50056	.....

(b)

One-dimensional **Two-dimensional**

OLAP Architecture



OLAP Applications

1. Finance: Budgeting, activity-based costing, financial performance analysis, and financial modeling.
2. Sales: Sales analysis and sales forecasting.
3. Marketing: Market research analysis, sales forecasting, promotions analysis, customer analysis, and market/customer segmentation.
4. Manufacturing: Production planning and defect analysis.

OLAP Limitations

- Limitation:-
  - can not predict :
  - what will happen in future?
  - Why happens?
- **How to overcome this limitation--**KDD

KDD process

- knowledge discovery from database[KDD].
- KDD- find useful information or knowledge & pattern from data.
- Data mining uses algorithms to extract information & pattern derived by KDD process.

Cont..

- **ANN/ machine learning**:- transform database into a knowledge base system. **Part of data mining technique.**
- **Data mining is a part of KDD.**
- KDD process- **selection**(obtain data from source), **preprocessing**(data cleaning), **transformation**(into desired data format), **data mining**(obtain desired result), **interpretation**(present result to user meaningfully).

- Data Mining:- computational process of discovering patterns in large data sets.
- Integration of artificial intelligence, machine learning, statistics, and database systems.
- Knowledge Discovery in Databases (KDD) process:-

1. Data Selection
2. Pre-processing (attribute extraction & Normalization)
3. Transformation- transform data into desired format.
4. Data Mining-- discovering patterns.
5. Interpretation/Evaluation

Data mining Core Tasks, Apps & Algorithms

1. Classification task :- Identifying to which category an object belongs to.

Applications: e-mail Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest.

2. Regression task :- Predicting a attribute value associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression.

3. Clustering task:- Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering.

4. Dimensionality reduction task:- How to choose a good set of attributes.

Applications: Visualization, Increased efficiency

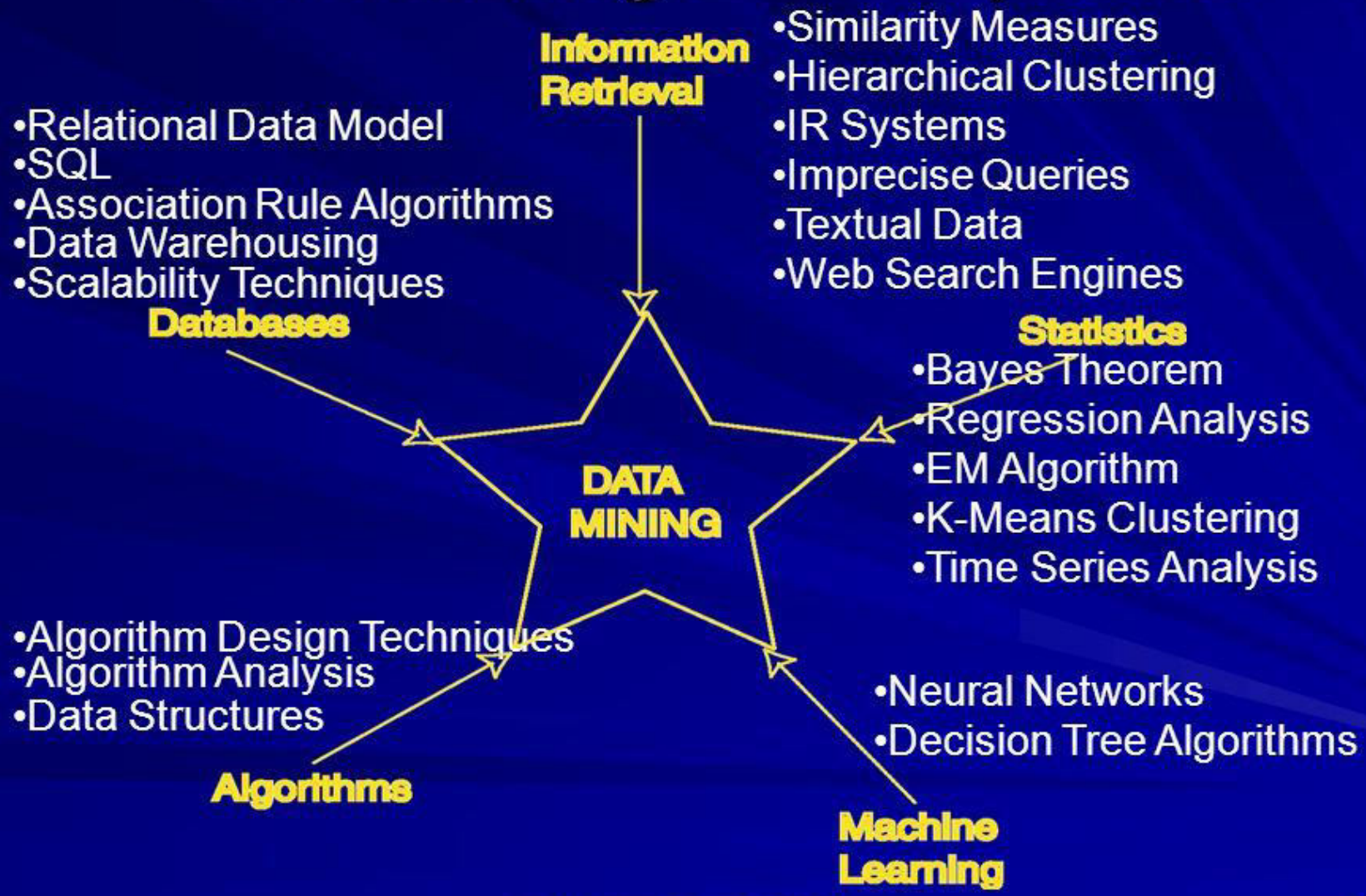
Algorithms: PCA, feature selection.

5. **Anomaly detection** (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.

6. **Association rule learning** (Dependency modeling) – Searches for relationships between variables.

For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

Data Mining Development



pcai.com

- Knowledge Based Systems, AI Languages, Neural Networks, Machine Learning, Genetic Algorithms, Evolutionary Software, Expert Systems, Fuzzy Logic, Data Mining, Intelligent Agents, Business Rules, Case-Based Reasoning, Common Sense, Data Visualization, Inferencing, Forecasting, Pattern Matching, Speech, Rule-Based Systems, Text Mining, Vision, Robotics.

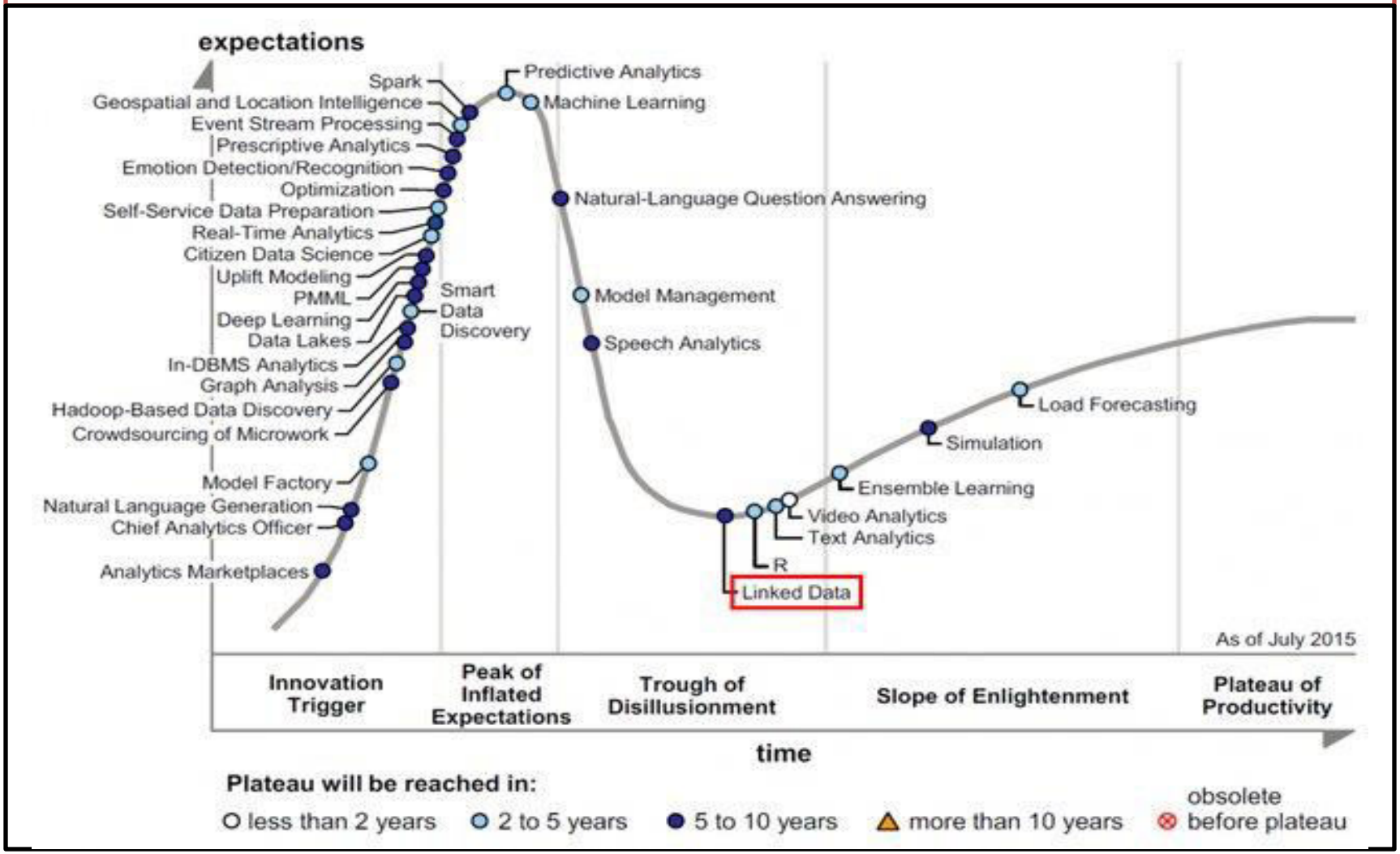
Analytic is a never ending process.

Analytic is the Major part of Data science,

Analytic is a never ending process because of progressive technological change requirements as well as the business change requirements.

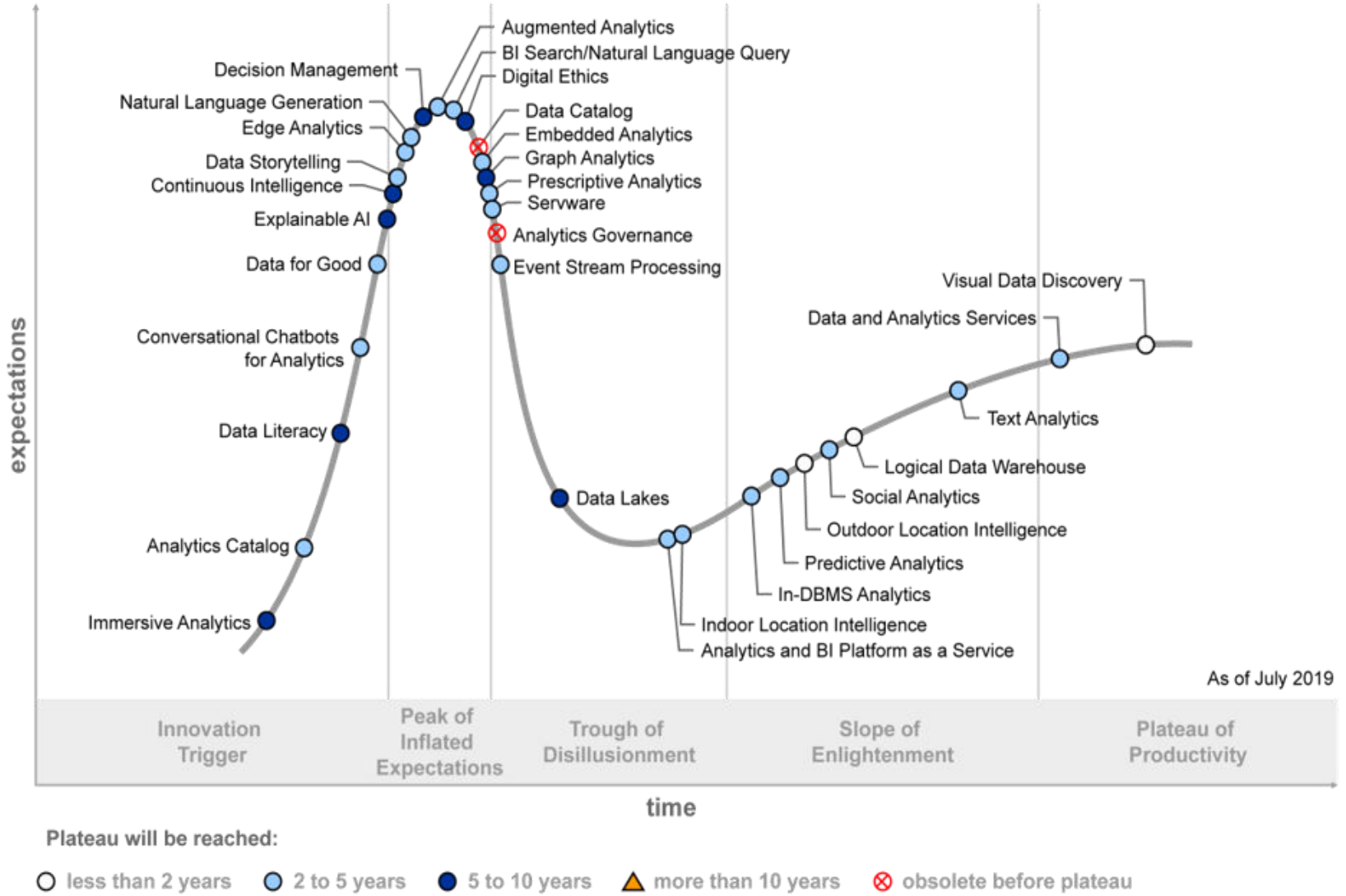
The beauty of Analytics is that two data scientist with same problem may come up with two different new solutions.

Gartner's Hype Cycle for Advanced Analytics and Data Science -2015



Compiled By: Dr. Nilamadhab Mishra [(PhD- CSIE) Taiwan]

Hype Cycle for Analytics and Business Intelligence, 2019



Source: Gartner

Compiled By: Dr. Nilamadhab Mishra [(PhD- CSIE) Taiwan]

Time to explore[Activity-01]

Investigate the numerous Data mining and analytic Core Tasks, Applications & Algorithms and prepare your investigation report.

RBT – Revised Bloom's Taxonomy
KL1 – Remember,
KL2-Understand,
KL3-Apply,
KL4-Analyse,
KL5-Evaluate,
KL6-Create
CO – Course Outcome

Cheers For the Great Patience!

Query Please?

Compiled By: Dr. Nilamadhab Mishra [(PhD- CSIE) Taiwan]