Hyder Reza
CSE
160318733073

Data Science using R

Assignment - I

① Eucidate how linear algebra is used in data Science?

A: Linear Algebra is a branch of mathematics which involves study of lines, planes, vectors, matrices, equations etc. It has a wide role of applications in data science field, ~~a glimpse of which we will see~~

- Data Representation becomes an important aspect of data science and data is represented usually in a matrix form.
- Data contains several variables of interest and linear algebra provides us with the tools to understand relationships between variables.
- Linear Algebra is used in data processing, data transformation and model evaluation.

① Vectors: LA deals with vector spaces. Vector space is a set of objects called vectors. It can be thought of as a point in a space, numerically represented in form of list.

Example: $A = [1, 2, 3, 4, 5]$ is a row vector

$$B = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$ is a column vector

(2) Matrix: It is an array represented in the form of rows and columns. We often deal with matrix while performing or applying algorithms in ML. Operations like addition, substraction, multiplication, transpose, etc. can be performed in matrices.

- Matrices can be used to represent samples with multiple attributes in a compact form.

Ex: $$A = \begin{bmatrix} 10 & 2 & 0.3 \\ 20 & 1 & 0.3 \\ 40 & 1 & 0.4 \end{bmatrix}$$

- Typically, rows are used to represent the sample and columns represent the variables.

(2) Define hyperplanes, eigen vectors and eigen values. Demonstrate the usage of hyperplane, eigen vectors and eigen values in data science with an example.

A: (a) Hyperplanes

- Hyperplanes are decision boundaries that help classify the datapoints.

- Data points falling on either side of the hyperplane can be attributed to different classes.

- The dimension of the hyperplane depends upon the number of features. If the input features is 2, then the hyperplane is just a line. If the input features is 3, then hyperplane becomes two-dimensional plane.

- ~~If V is an~~ If vector space V is an n-dimensional vector space then hyperplane H is an (n-1) dimensional subspace.

Example: Support Vector Machine

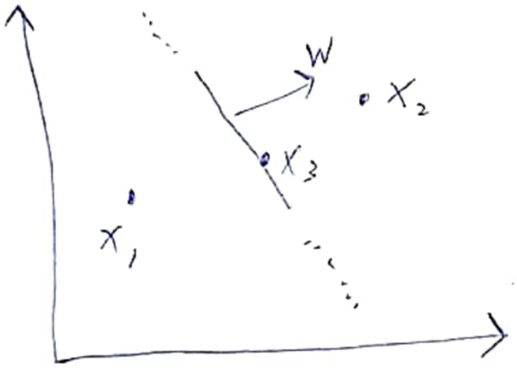
- It is used to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

- The general equation of a hyperplane is $\boxed{W \cdot x + b = 0}$, where W is a vector normal to the hyperplane, b is an offset.

- If $W \cdot x + b < 0$, then point X_1 lies below the hyperplane.

- If $W \cdot x + b > 0$, then point X_2 lies above the hyperplane.

- If $W \cdot x + b = 0$, then point X_3 lies on the hyperplane.



(b) Eigen Vectors

- Eigenvector is a vector that is associated with a set of linear equations.

- It is defined as a non-vector in which when a given matrix is multiplied.

- It is equal to a scalar multiple of that vector.

- Suppose A is an n × n square matrix, and if V be a non-zero vector, then the product of matrix A, and vector V is defined as the product of a scalar quantity λ and the given vector such that

$$AV = \lambda V$$

where, V = Eigen vector

λ = scalar quantity that is termed as Eigen value (EV are particular set of scalar values related to linear equns)

Example:-

- Eigenvectors are used in Physics is simple mode of osillation
- In Mathematics, eigen vector decomposition is widely used in order to solve the linear equation of first order, in ranking matrices, in differential calculus etc.
- widely used in quantum mechanics.
- used to reduce a linear operation to separate or simplify the problems.

(3) Explain R objects vectors, lists and data frames with an example?

A:- R Objects

- Objects are the instance of the class. Everything in R is an object. They can have their attributes like class, attributes, dimnames, names, etc.

- There are many type of R objects.

(i) Vectors
(ii) Lists
(iii) Matrices
(iv) Arrays
(v) Factors
(vi) Data Frames

I Vectors

- Vectors are the most basic R data object and there are six types of atomic vectors. They are logical, integer, double, complex, character and raw.

- It is an ordered collection of same datatypes.

Example: (i)

```
x = c(2.3, 4.6, 1.2, 7.8)
print(x)

[1]  2.3  4.6  1.2  7.8
```

(ii)

```
Y <- 3.0
print(Y)

[1]  3.0
```

II. Lists

- List is an ordered collection of object themselves.
- It is a generic object consisting of ordered collections of objects.
- List can be a list of vectors, list of matrices list of characters and list of function and so on.

Example: # List Example: Employee detail:

```
> ID = c(1,2,3,4)
> EMP.NAME = c("man", "RAG", "Kat", "SHA")
> NUM.EMP = 4
> EMP.LIST = (list(ID, EMP.NAME, NUM.EMP))

> EMP.LIST

[[1]]
[1]  1  2  3  4

[[2]]
[1]  "man"  "RAG", "Kat"   "SHA"

[[3]]
[1]  4
```

III. Dataframe

- It is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

- Characteristics of Dataframe

- The column names should be non-empty
- The row names sholld be unique.
- The data stored in a data frame can be of numeric, factor or character type.
- Each column should contain same number of data items.

Example:

```
> pd = data.frame("Name" = c("senthill", "senthil", "sam", "sam"),
                  "Month" = c("Jan", "Feb", "Jan", "Feb"),
                  "BS" = c(140.2, 139.4, 138.3, 142.2),
                  "BP" = c(90, 88, 87, 85))
```

```
> pd
                                  BP
    Name      Month      BS
                                  90
1   senthil   Jan        140.2    88
2   senthil   Feb        139.4    87
3   ~~[illegible]~~sam   Jan        138.3    85
4   sam       Feb        142.2
```

④ With examples describe data visualization in R?

A: R programming provides comprehensive sets of tools such as in-built funtions and a wide range of packages to perform data analysis, represent data and build visualizations.

I. Barplot

- It is used to represent data in the form of rectangular bars, both in vertical and horizontal ways, and the length of the bar is proportional to the value of the variable

Example: barplot(mtcars$hp, ylab = "Horsepower", col="cyan", horiz =FALSE)

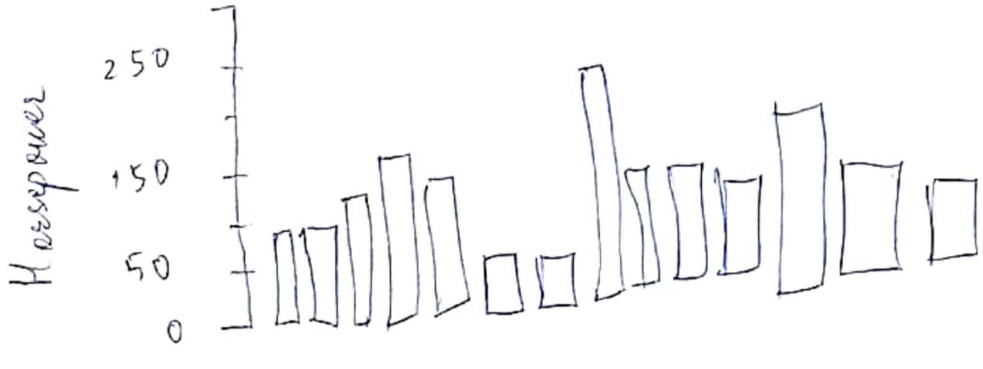


Fig. Barplot

II Histogram

- It is used to divide values into groups of continuous ranges measured against the frequency range of the variable.

Example:

```
> hist(mtcars$mpg, xlab ="Miles per Gallon", main =" Histogram
      for MPG", col = "yellow")
```

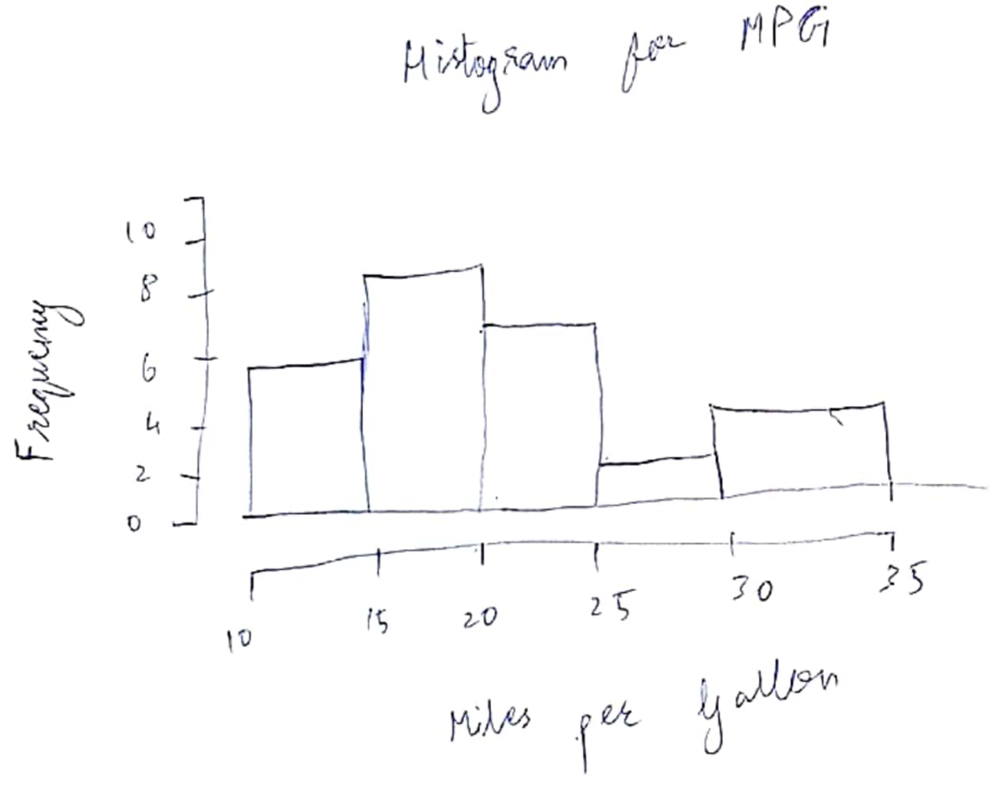


Fig: Histogram

III. Boxplot

- It is used to represent descriptive statistics of each variable in a dataset.
- It represents the minimum, first quartile, median, third quartile and the maximum values of a variable.

Examples: boxplot (mtcars [, 3:4])

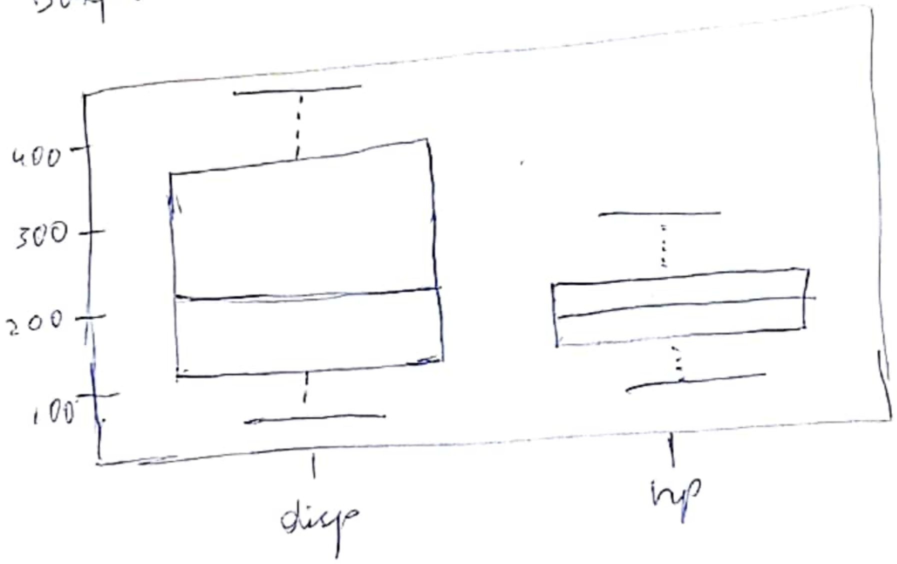


Fig. Boxplot

IV Pie chart

- A pie-chart is a representation of values as slices of a circle with different colors.
- The slices are labelled and the numbers corresponding to each slice is also represented in the chart.

Example:

```
> x <- c( 21, 62, 10 , 53)
> labels <- c ( "London", "New York" , "Singapore", "Mumbai")
> pie ( x, labels)
```

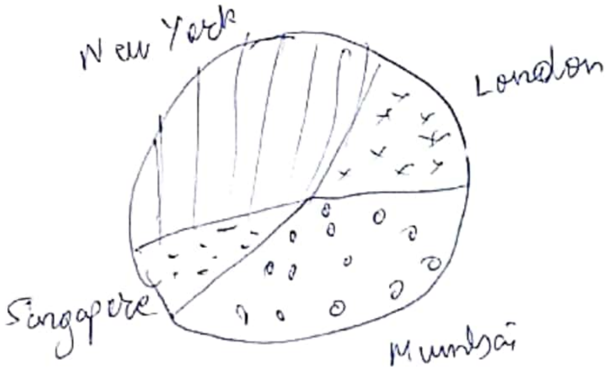


Fig. Pie Chart

(5) Describe different Matrix operations in R with examples? explain about functions ?

A: - There are four basic operations i.e DMAS (Division, Multiplication, Addition, Subtraction) that can be done with matrices. Both the matrices involved in the operation should have the same number of rows and columns.

- Matrix can be created using the matrix() function

Syntax:

```
matrix (data, nrow, ncol , byrow, dimnames)
```

~~I Addition~~

Example:

```
B = matrix(c(1, 2, 3, 4, 5, 6), nrow=2, ncol=3)
C = matrix(c(7, 8, 9, 10, 11, 12), nrow = 2, ncol = 3)
```

I. Addition

- R has an inbuilt operator '+' for performing matrix addition

Ex: print (B + C)

```
      [,1] [,2] [,3]
[1,]   8    12   16
[2,]   10   14   18
```

II Subtraction

- R has an inbuilt operator '-' for performing matrix subtraction.

Ex: print (B - C)

```
      [,1] [,2] [,3]
[1,]   -6   -6   -6
[2,]   -6   -6   -6
```

III. Multiplication

- R has an inbuilt '*' operator for performing matrix multiplication.

Example:- print (B*C)

```
      [,1]   [,2]   [,3]
[1,]    7     27     55
[2,]   16     40     72
```

IV. Division

- R has an inbuilt '/' operator for performing matrix division.

Example: print (B/C)

```
        [,1]          [,2]          [1,3]
[1,]   0.1428571     0.3333333     0.4545455
[2,]   0.2500000     0.4000000     0.5000000
```

Functions

- A function is a set of statements organized together to perform a specific task. R has a large number of in-built functions and the user can create their own function.

- An R function is created by using the Keyword function. Basic syntax of an R function definition is as follows -

```
function_name = function(arg-1, arg-2, ....) {
    Funtion body
}
```

Example:

```
volCylinder = function( dia = 5, len = 100) {
        volume = pi * dia^2 * len/4
        return (volume)
}
```

- Once you load the function, you can invoke the function from the console as follows you want the volume to be saved in the variable v and they you are calling the function volcylinder to calculate the volume.

Example:
```
> v = volCylinder(5,10)
> v
[1]  196.3495
```