MGMUNIVERSITY

Practical 1

Aim :- To introduce R-tools, install the R programming environment & basic packages.

Theory

Introduction to R:

R is a programming language & free software environment for statistical computing & graphics supported by the R Foundation for statistical computing. It is widely used among statisticians & data miners for data analysis & developing statistical software. R provides a wide variety of statistical & graphical techniques, including linear & non-linear modeling, classical statistical tests, time-series analysis, classification, clustering & other.

Features:

1] Open source: R is an open-source programming language which means it is free to use, modify & distribute.

2] Comprehensive statistical analysis: R includes a vast array of tools for statistical analysis and data mining.

3] Visualization: R excels at creating detailed & advanced plots & graphs offering extensive visualization capabilities.

4] Community support: R has ~~largest~~ & large & active community that contributes packages & provide support to users.

5] Extensible :- R can be extended easily via packages. There are thousands of packages available on CRAN [Comprehensive R Archive Network] for various applications.

6] Cross-platform :- R is available on multiple platforms including windows, MacOS & Linux.

7] Data handling: R can handle & manipulate large datasets efficiently

8] Interfacing: R can integrate with other language like C/C++, Python & can be used with database management systems.

R	Python
- R is a language & environment for statistical programming which includes statistical computing & graphics.	- Python is a general purpose programming language for data analysis & scientific computing.
- It has many features which are useful for statistical analysis & representation.	- It can be used to develop GUI applications & web applications as well as with embedded systems.
- It has many easy to use packages for performing tasks	- It can easily perform matrix computation as well as optimization.
- Various popular R IDE are Rstudio, RKward, R Commander etc	- Various popular Python IDEs are Syder, Eclipse+Pydev, Atom
- There are many packages & libraries like ggplot 2, Lard	- Some essential packages & libraries are pandas, numpy, scipy.
- It is mainly used for complex data analysis in data science.	- It takes a more streamlined approach for data science projects.

MGMUNIVERSITY

Steps to install R & Basic packages:-

1. Install R:
   - Go to the R project website
   - Click on "CRAN" under "download" to find a CRAN mirror
   - Select a CRAN mirror near you
   - Download the R installer for your OS.

2. Install RStudio:- Download the RStudio Desktop installer for your OS. Run the installer & follow the on-screen instructions to complete the installation

3. Installing basic packages:-
   commands:
   install.packages("data.tables")
   install.packages("dplyr")
   install.packages("ggplot2")
   install.packages("caret")

- data.tables:- It is highly efficient for handling large datasets, making operations like filtering, aggregating & joining data much faster compared to base R or other packages.

- dplyr :- It simplifies data manipulation tasks & makes code more readable & expressive, especially for those coming from a SQL background.

- ggplot 2:- It is powerful, flexible & makes it easy to create ~~as~~ aesthetically pleasing & informative visualizations.

- caret :- It simplifies the process of training & evaluating machine learning models providing a unified interface to a wide range of model types & tool for model comparison & tuning.

Conclusion:
In this practical we introduced the R programming environment & its features. We also covered the difference bet^n R & Python.

25/7/2024

Practical 2:

Aim: Implement different data types & data structures in R, including vectors, lists, matrices, arrays, factors & data frames.

Theory:

1] Vector :- A vector is a basic data structure in R that contains elements of the same type. It can be numeric, integer, character, logical or complex.

Ex:

```
numeric_vector ← c(1.1, 2.2, 3.3)
print (numeric_vector)

char_vect ← c("apple", "banana")
print (char_vect)

logical_vector ← c (TRUE, FALSE, TRUE)
print (logical_vector)
```

2] List : A list is an R data structure that can hold elements of different types number, string, vector. It is similar to a vector but more flexible.

Ex:

```
my_list ← list (name = "johnm", age = 25,
                score = c(35, 40, 39))
print (my_list)
print (my_list$name)
print (my_list$age)
print (my_list$scores)
```

3] Matrices: A matrix is a two dimensional array that contains elements of the same type. It is used for mathematical computations involving rows & columns.

Ex:

```
my_matrix ← matrix(1:9, nrows=3, ncol=3)
print(my_matrix)
print (my_matrix[1,2])
print (my_matrix[,3])
print (my_matrix[2,])
```

4] Array :- An array is a multi-dimensional data structure that can hold elements of the same type. It can be considered as a generalized form of a matrix

Ex.

```
my_array ← array(1:12, dim=c(3,2,2))
print(my_array)
print(my_array[1,2,1])
print(my_array[,,2])
```

5] Factor: A factor is a data structure used for fields that take only pre-defined, finite no. of values [categorical data]. Factors are useful in statistical modeling.

Ex:

```
my_factor ← factor(c("low", "medium", "high"))
print(my_factor)
print(levels(my_factor))
my_vector ← c("yes", "no", "yes")
factor_vector ← as.factor(my_vector)
print(factor_vector)
```

6] Data Frame: A data frame is a table or a 2-dimensional array like structure in which each column contains values of one variable & each row contains one set of values from each column. Its similar to a matrix but can contain different types of data.

Ex:

```
my_df ← data.frame(
    name = c("Alice", "Bob", "Charlie"),
    age = c(20, 21, 22),
```

```
score = c(90, 91, 92))
Print ( my_df)
print ( my df $name)
print ( my_df [1,])
print (my_df [, "age"])
```

conclusion:-

In this practical, we explored & implemented different data types & data structure in R. Including vector, lists, matrices, arrays & data frame. Understanding these structures is fundamental for performing data analysis & manipulation tasks in R.

25/7/2024

MGMUNIVERSITY

Practical 3

Aim: To implement a program in R that calculates basic statistics including mean, median, mode, variance, std deviation, range, summary, skewness & kurtosis

Theory:

Descriptive statistics:
It is used to summarize & describe the main features of a dataset. They provide simple summaries about the sample & the measures. Together with simple graphics analysis, they form the basis of virtually every quantitative analysis of data.

1. Mean: The mean is the sum of all the numbers in a dataset divided by the count of number in that dataset.

2. Median: The median is the middle value in a data set when the numbers are arranged in ascending or descending order. If the dataset has an even number of observations, the median is the avg of the two middle numbers.

3. Mode: The mode is the value that appears most frequently in a data set.

4. Variance: Variance measures how far a set of numbers are spread out from their average values. It is the average of the squared differences from the mean.

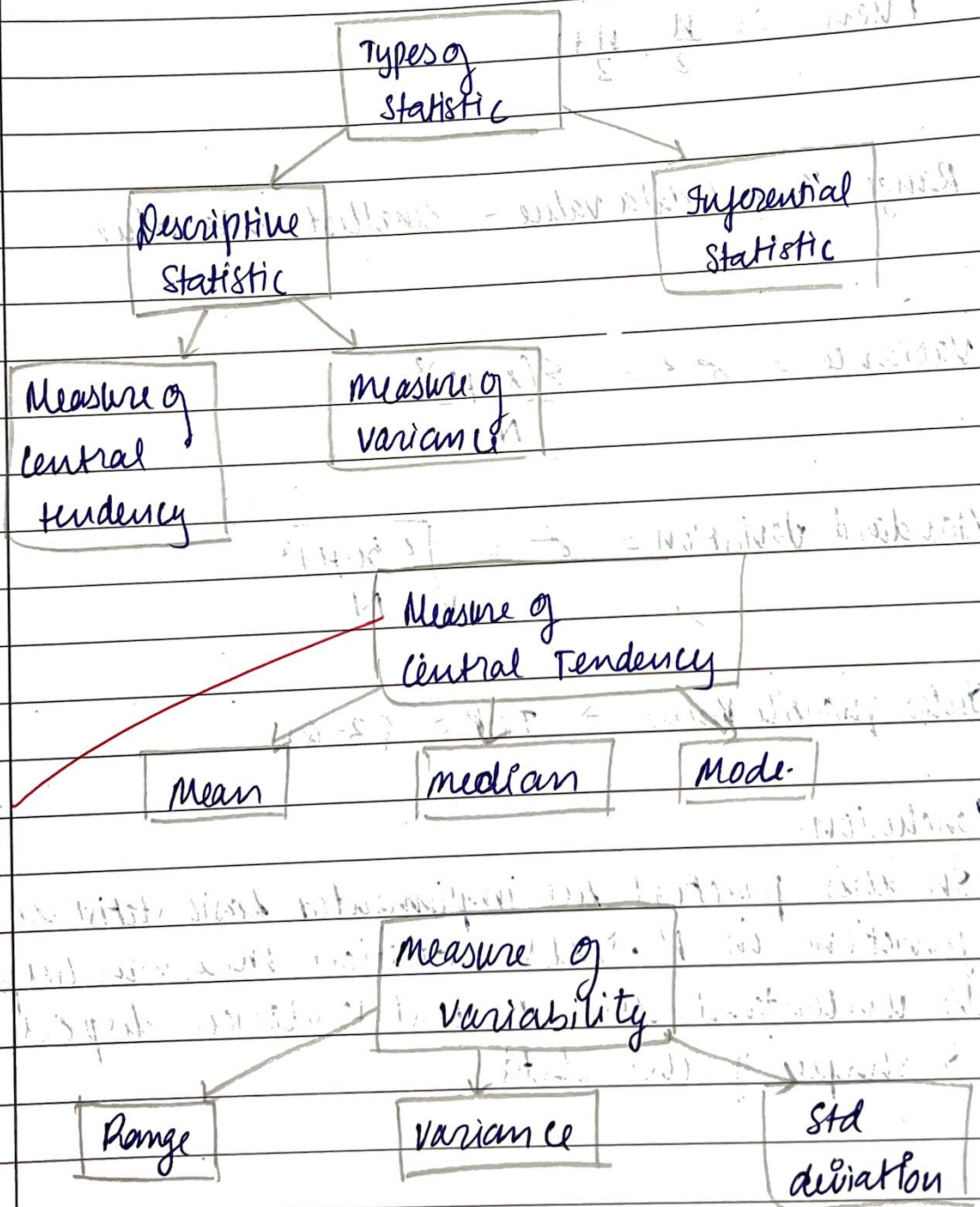
5. Standard deviation: It is the square root of the variance. It indicates the extent to which the value in a dataset deviate from the mean.

6. Range: The range is the difference between the highest & lowest values in a data set.

7. Summary: The summary function in R provides a summary of the data, including the min, 1st quartile, median, mean, 3rd quartile & maximum.

8. Skewness: It measures the asymmetry of the data distribution. Positive skewness indicates a distribution with a long right tail, while negative skewness indicates a distribution with a long left tail.

9] Kurtosis: It measures the tailedness of the data distribution. High kurtosis means more of the variance is due to infrequent extreme deviations, as opposed to frequent modestly sized deviations.



Mean ($\bar{x}$) = $\frac{\Sigma x}{n}$

Median

odd ⇒ $\frac{n+1}{2}$

Even ⇒ $\frac{n}{2}, \frac{n}{2}+1$

Range = Largest data value - Smallest data value

Variance = $\sigma^2 = \frac{\Sigma(x-\mu)^2}{N}$

Standard deviation = $\sigma = \sqrt{\frac{\Sigma(x-\mu)^2}{N}}$

Inter quartile Range ⇒ IQR = Q3 - Q1

Conclusion:

In this practical we implemented basic statistical functions in R. These statistical measures help to understand the central tendency, dispersion & shape of the data.

Practical 4

Aim: Write a program in R for data visualization using histogram & pie.

Theory

Data visualization is the graphical representation of information & data. By using visual elements like charts, graphs & maps. Data visualization tools provide an accessible way to see & understand trends, outliers & patterns in data.

1. Histogram:
It is a graphical representation that organizes a group of data point into user-specified ranges.
Similar in appearance to a bargraph, the histogram condenses a data series interpreted visual by taking many points & grouping them into logical ranges or bins.

2. Scatter plot:
It is a type of data visualization that displays values for typically two variable for a set of data.
The data is displayed as a collection of points, each having the value of one variable determining the position

on the horizontal axis & the value of the other variable determining the position on the vertical axis.

3. Dot plot:
A dot plot is a simple statistical chart that contains / consists of data points plotted as dots on a simple scale, used to show frequency distributions & patterns in data.

4. Line chart:
A line chart is a type of chart that displays information as a series of data points called "markers" connected by straight line segments. It is similar to a scatter plot except that the measurement points are ordered & joined with straight line segments.

5. Pie chart:
A pie chart is a circular graphic which is divided into ~~set~~ slices to illustrate numerical proportions. In a pie chart, the arc length of each slice is proportional to the quantity it represents.

6. Box plot:
It is a standardized way of displaying the distribution of data based on five number summary: min, first quartile (Q1), median, third quartile (Q3) & maximum.

7. Density plot:
It is a smoothed continuous version of a histogram estimated from the data. It is used to visualize the distribution of the data.

8. Normal QQ Plot:-
It is a graphical tool to help assess if a set of data plausibly came from some theoretical distribution such as a normal or exponential.
The data are plotted against a theoretical distribution in such a way that the points should form an approximate straight line

Conclusion:
In this lab we successfully created various types of data visualizations using the base R functions.
These visualizations help us to understand the underlying patterns, distribution & relationship of the dataset.
These plots help to obtain different perspective making it easier to derive meaningful insights from the data.

Practical 5

Aim: Implementation of data processing and EDA in R.

Theory:-

1. Data processing:
Data processing is a critical step in ~~that~~ the data analysis pipeline that involves preparing raw data for analysis. Data preprocessing includes several tasks such as data cleaning, data transformation & feature scaling

2. Data cleaning:
Data cleaning involves detecting & correcting errors & inconsistencies in the data to improve its quality. It is an essential step to ensure that the dataset is reliable & accurate for analysis.

Key steps:
- Detecting missing values: Identify missing values in the dataset, which can occur due to various reasons such as data entry error or missing information

- Handling missing values: Decide on a strategy to handle missing values. Common methods include:-

1) Removal: Deleting rows or columns with missing values if they are few & not critical to the analysis.

2) Imputation: Filling in missing values using various methods such as mean, median, mode or more advanced techniques like regression or ML models.

- Detecting Outliers :- Identify outliers in the data which can skew the analysis. Outliers can be detected using statistical methods or visualization techniques

- Handling Outliers: Outliers can be handled by removing them, transforming them or using robust statistical method that are less sensitive to outliers

- Correcting Data Entry Errors: Detect & correct errors such as typos, incorrect values or inconsistent format.

- Standardizing data format: Ensure consistency in data formats, such as date formats, numeric precision & categorical labels.

3. Data Transformation

Data transformation involves converting data into a suitable format or structure for analysis. This process helps in enhancing the quality & usability of the data.

Steps:-

- Encoding Categorical Variables: Convert categorical variables into numerical format for analysis. common techniques:
  1) Label Encoding: Assigning a unique integer to each category.
  2) One-hot Encoding: Creating binary columns for each category.

- Creating New features: Derive new features from existing data to capture additional information.

- Scaling numerical Features:
  Standardize the range of numerical features to ensure that they contribute equally to the analysis

  1) Min-Max Scaling: Rescaling the features to a fixed range, usually [0,1]

2) Standardization (Z-score scaling):
Transforming the feature to have a mean of 0 & a std deviation of 1.

- Normalization: Adjusting the data to a common scale without distorting differences in the ranges of values

4. Feature Scaling:
Feature scaling is the process of normalizing the range of independent variables or features of data. It is crucial for dl algorithms that are sensitive to the scale of data, such as gradient descent-based methods & distance based algo.

Conclusion:
In this practical we have performed data pre-processing & EDA, that are fundamental steps in the data analysis process

Practical 6:

Aim: To implement analysis of variance (ANOVA) on a given dataset & calculate the difference of variances bet^n the groups.

Theory:
ANOVA is a statistical method used to compare the means of three or more samples to understand if at least one sample mean is significantly different from the others. It helps in determining whether the observed variations among sample means are due to actual differences in the population or merely due to random chance.

Key Concepts:-
1. Null Hypothesis (H_0):- Assumes that all group means are equal.
2. Alternative hypothesis (H_1):- Assumes that at least one group mean is different.
3. Between Group variance:- The variation due to the interaction bet^n the different groups.
4. Within Group variance:- The variation within each groups
5. F-statistic:- A ratio of bet^n-group variance to within group variance. A higher-F-statistic indicates a greater disparity bet^n grp means.

Steps in ANOVA:

1. Calculates the group means: Determine the mean of each group in the dataset.
2. Calculate the Overall mean: Determine the mean of all data points combined.
3. Compute the bet^n-Group Sum of square (SSB) Measure the variance bet^n the group means & the overall mean.
4. Compute the within Group sum of Squares (SSW) Measure the variance within each group.
5. Calculate the Mean Square bet^n (MSB): Divide SSB by the degrees of freedom bet^n the group.
6. Calculate the mean square within (MSW): Divide SSW by the degree of freedom within the groups.
7. Compute the F-statistic: Divide MSB by MSW to obtain the F-value.
8. Compare with critical value: Compare the F-value with the critical value from the F-distribution table to determine the significance.

Conclusion:-

ANOVA allows us to statistically test if there are any significant differences in the variances bet^n groups, aiding in understanding the impact of different factors on the observed data.

Practical 7

Aim:- To implement simple linear regression on a given data set

Theory:
Simple linear regression is a statistical method used to model the relationship bet^n a dependent variable & an independent variable by fitting a linear eq^n to the observed data. The objective is to predict the value of the dependent variable based on the value of the independent variable.

Key concept:

1] Dependent Variable (Y):- The variable we are trying to predict or explain.

2] Independent variable (X):- The variable we are using to make predictions about the dependent variables.

3] Linear Relationship:- The relationship bet^n x & y is assumed to be linear, meaning it can be represented by a st. line.

4] Regression line: The line that best fits the data points, described by the eqn [$y = b_0 + b_1 x$] where b_0 is the intercept & b_1 is the slope

5] Least square method :- A method used to determine the best fitting line by minimizing the sum of the squares of the vertical distances of the points from the line.

Steps in Simple Linear Regression:

1] Plot the data: Visualize the data points on a scatter plot to observe the relationship bet^n x & y.

2] Calculate the Mean of x & y: Determine the average values of the independent & dependent variables.

3] Determine the slope (b1): Use the formula

$$b1 = \frac{\sum_{i=1}^{n} (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

to calculate the slope of the regression line.

4] Determine the intercept (b_0): Use the formula, to calculate the intercept of regression line.

$$b_0 = \bar{y} - b_1 \bar{x}$$

5] Form the regression eqn: Combine the slope & intercept to form the regression eqn

$$y = b_0 + b_1 x$$

6] Make predictions: use the regression equation to predict the value of the dependent variable based on new values of the independent variable.

7] Evaluate the model: Assess the goodness-of fit using metrics such as the coefficient of determination (R^2) & the residuals (differences b/t observed & predicted values)

Conclusion:-

Simple linear regression provides a straightforward method to model the relationship b/t two variables & make prediction, facilitating data driven decision-making.

Practical 8:

Aim: To implement simple logistic regression on a given dataset.

Theory:
Simple logistic regression is a statistical method used to model the relationship betn a binary dependent variable & one independent variable. Unlike linear regression, which predicts a continuous outcome logistic regression predicts the probability of a binary outcome [eg: success/failure, yes/No]

Key Concepts:
1. Dependent Variable (Y): The binary outcome we are trying to predict, often coded as 0 or 1.
2. Independent Variable (X): The variable used to predict the binary outcome.
3. Logit Function: The natural logarithm of the odds of the dependent variable being 1. The logistic regression model is based on the logit function.

$$\text{logit}(P) = \log\left(\frac{p}{1-p}\right) = b_0 + b_1 x.$$

Where p is the probability that the outcome equals 1.

4. Sigmoid function :-
The function that converts the logit back to a probability given by

$$p = \frac{1}{1+e^{-x}}$$

5. Maximum likelihood Estimation [MLE] :
A method used to estimate the parameters [coefficients] of the logistic regression model by maximizing the likelihood function.

Steps in simple Logistic Regression :-

1. Plot the data:
Visualize the data points on a scatter plot to observe the relationship bet^n x & the binary outcome y.

2. Initialize the model: Define the logistic regression model with the logit function.

3. Estimate parameters (b_0 & b_1): Use maximum likelihood estimation to find the coefficients that best fit the data.

4. Form the logistic regression eq^n :- Combine the coefficients to form the logistic regression eq^n.

5. Make predictions: use the logistic regression equation to predict the probability of the binary outcome for new values of the independent variable.

6. Classify Outcomes: Determine the binary outcome by applying a threshold [eg: if $p > 0.5$, predict 1 else 0]

7. Evaluate the model: Assess the model's performance using metrics such as accuracy, precision, recall & the area under the ROC curve [AUC]

Conclusion:-

Simple logistic regression allows us to model the probability of a binary outcome based on one predictor variable making it useful for classification tasks.

Practical 9

Aim: Building a classification Model using K-nearest neighbors (KNN)

Theory:-

K-Nearest Neighbors [KNN] is a simple yet powerful classification & regression algorithm that is a non-parametric & instance-based. It belongs to a category of algorithms called Lazy-learners because, unlike many algo. that explicitly build models during training, KNN does not learn a model. Instead, it memorizes the training data & performs classification or regression only when it receives a query instance. The fundamental assumption of KNN is that similar data points are likely to belong to the same class, making it a strong tool for classification tasks.

Key concepts:-

1] Instance based learning:- It means that it stores the training instances rather than constructing an explicit model. For each new data point, it finds ~~the~~ to those stored instances. When a prediction is made KNN uses the training instances that are closest to the new data points to classify it.

2. Distance Metric: -

KNN relies on a distance metric to measure the similarity bet^{n} data points.

i) Euclidean distance: The dist. bet^{n} 2 points in space. [st. line]

ii) Manhattan distance: - The sum of the absolute differences bet^{n} the co-ordinates of two points.

3. Majority voting: -

In KNN classification the class label of a new data point is determined by a ~~voit~~ voting mech^{m}. Once the 'k' nearest neighbors are identified their class labels are observed & the new point is assigned to the class that is most frequent among these 'k' neighbors. This is the essence of how KNN makes predictions.

4. Parameter 'k': -

The parameter 'k' determines how many neighbors are considered for the majority voting process.

- if 'k' is small makes the algo. more sensitive to local variations in the data & lead to overfitting.
- if 'k' is large, it smoothens the decision boundary, reducing the risk of overfitting but it can potentially ignore subtle but imp. structures in the data.

Steps in building a KNN classification model:

1. Data preprocessing:-
Before applying KNN, the data must be preprocessed to ensure that the distance metric works correctly. Apply Normalization or standardization to scale each feature to a common range. Handle the missing values by either imputing the missing values or by removing affected rows & columns.

2. Selecting the value of 'k':-
Choosing the right 'k' is critical. A small 'k' leads to more flexible decision boundaries but can make the model sensitive to noise. A large 'k' produces smoother more generalized boundaries but may miss imp. local patterns.

3. Computing distance:-
Once the model receives a new data point to classify, it computes the distance from this point to all the data points in the training set using the chosen distance metric

4. Identifying nearest neighbors:-
After calculating the distances the model selects the 'k' [illegible] training instances with the smallest

distances to the new data point.

5. Majority Voting:
The model then examines the class labels of the identified 'k' nearest neighbors. The class with the most votes is assigned as the predicted label for the new data points.

6. Model Evaluation:
Once the KNN model has made predictions, it needs to be evaluated to assess its performance.

Conclusion:-
The ~~K-means~~ KNN builds a classification model by assigning class labels based on the majority votes among the nearest neighbors, making it a straightforward & versatile classifier.