DAV UT-2

1.explain the seven practices of text analytics 10M

Search and information retrieval (IR): Storage and retrieval of text documents, including search engines and keyword search.

Document clustering: Grouping and categorizing terms, snippets, paragraphs, or documents, using data mining clustering methods.

Document classification: Grouping and categorizing snippets, paragraphs, or documents, using data mining classification methods, based on models trained on labeled examples.

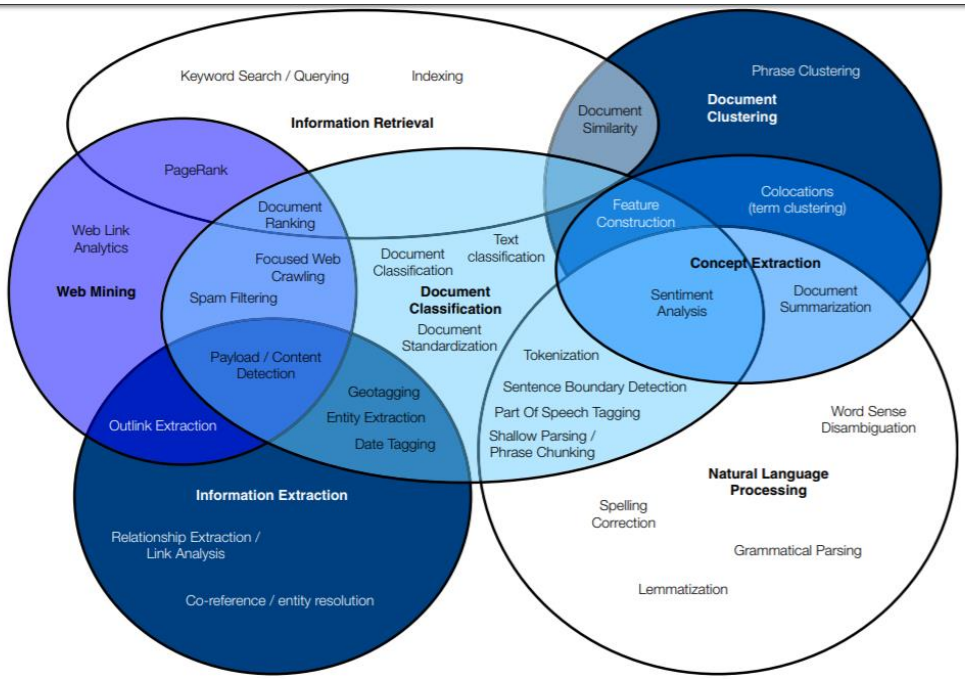
Web mining: Data and text mining on the Internet, with a specific focus on the scale and interconnectedness of the web.

Information extraction (IE): Identification and extraction of relevant facts and relationships from unstructured text; the process of making structured data from unstructured and semistructured text.

Natural language processing (NLP): Low-level language processing and understanding tasks (e.g., tagging part of speech); often used synonymously with computational linguistics.

Concept extraction: Grouping of words and phrases into semantically similar groups

---



2.Discuss below the use cases for text mining

- **Extracting “meaning” from unstructured text**
- **Summarizing Text**

Extracting “meaning” from unstructured text: -

Various methodologies can extract specific contents or contextual meaning from a large corpus of small text documents or from a small corpus of large text documents that cannot be read and summarized in a practical manner.

1.Sentiment Analysis:

Seeks to determine the general sentiments, opinions, and affective states of people reflected in a corpus of text.

Analysis of these sentiments can address:

- What are my customers saying about me?

- What are the emerging areas of concern or interest in a specific target group?

- Analyzing open-ended responses to survey questions

2.Trending Themes in a Stream of Text:

- Used in areas that are more interested in detecting changes, trends, and unusual events.
- Warranty Claim Trends
- Insurance Claims
- Fraud Detection.

Summarizing Text: -

- Quickly summarizes one or a few very large documents.

- Two types of text summarizations:

First: -

- Summarizes themes across the chapters or paragraphs of the text.
- Here, individual paragraphs or chapters can be considered different documents of a larger corpus (the entire text).
- Identifies the different themes across the various documents

- Identifies common dimensions or relationships among individuals, events, and so on.

Second –

- Summarizes the contents of a large text document into a meaningful narrative which cannot be accomplished effectively using automatic text mining methods and algorithms.
- It is not realistic to expect that present computer algorithms are capable of summarizing the “essence” of a very large book into a single paragraph.
- Can be done only in other highly subjective ways.

3. Find out the advantages using R in data Analysis

R is the most popular programming language for statistical modeling and analysis. Like other programming languages, R also has some advantages and disadvantages. It is a continuously evolving language which means that many cons will slowly fade away with future updates to R.

1. **Open-source:** R is an open-source language, which means it is free to use, modify, and distribute. This makes it accessible to a wide range of users and promotes collaboration in the data science community.
2. **Large community:** R has a large and active user community that develops and shares packages, tutorials, and tools. This makes it easy to find resources and solutions for various data analysis tasks.
3. **Comprehensive packages:** R has a vast collection of packages that provide functionalities for data cleaning, data visualization, statistical analysis, machine learning, and more. These packages make it easy to perform complex tasks without having to write code from scratch.
4. **Reproducibility:** R code is highly reproducible, which means that the results of data analysis can be easily replicated. This is important for research and data-driven decision-making.
5. **Graphical capabilities:** R has powerful graphical capabilities that enable the creation of high-quality visualizations, such as scatter plots, histograms, heatmaps, and more.

6. **Integration with other tools:** R can be easily integrated with other tools, such as SQL databases, Python, and Excel. This makes it flexible and adaptable to various data analysis workflows.
7. **Data manipulation:** R has powerful data manipulation capabilities, including filtering, merging, and reshaping data. These functionalities enable the user to quickly and efficiently transform data to meet specific analysis requirements.

4. Distinguish between data mining and data analysis

Criteria	Data Mining	Data Analysis
Definition	The process of discovering patterns and knowledge from large datasets	The process of examining and interpreting data to draw insights and conclusions
Purpose	To identify patterns, correlations, and relationships in data that may not be apparent through traditional analysis methods	To understand the nature of data, identify trends, and draw actionable insights
Techniques	Machine learning algorithms, cluster analysis, decision trees, neural networks, association rule mining, etc.	Descriptive statistics, exploratory data analysis, regression analysis, hypothesis testing, etc.
Data Types	Large and complex datasets, typically unstructured or semi-structured	Structured data, such as tables, spreadsheets, and databases
Data Sources	Data warehouses, transactional databases, web logs, sensor data, social media, etc.	Business reports, surveys, sales data, customer feedback, etc.

Criteria	Data Mining	Data Analysis
Goal	To find hidden patterns and relationships that can be used to make informed decisions	To summarize, understand, and explain data in a way that is meaningful and useful to stakeholders
Outcome	Insights that can be used to improve business processes, optimize marketing campaigns, reduce costs, etc.	Recommendations, predictions, and visualizations that can be used to make informed decisions
Tools	Python, R, SAS, Weka, RapidMiner, etc.	Excel, Tableau, Power BI, Google Analytics, etc.

5. Explain indetail about TFIDF 10M

TF-IDF stands for *term frequency-inverse document frequency* and it is a measure, used in the fields of information retrieval (IR) and machine learning, that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc) in a document amongst a collection of documents (also known as a corpus).

What is TF (term frequency)?

Term frequency works by looking at the frequency of a *particular term* you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency:

- Number of times the word appears in a document (raw count).
- Term frequency adjusted for the length of the document (raw count of occurences divided by number of words in the document).
- Logarithmically scaled frequency (e.g. log(1 + raw count)).
- Boolean frequency (e.g. 1 if the term occurs, or 0 if the term does not occur, in the document).

What is IDF (inverse document frequency)?

Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. IDF is calculated as follows where ***t*** is the term (word) we are looking to measure the commonness of and ***N*** is the number of documents (d) in the corpus (D).. The denominator is simply the number of documents in which the term, ***t***, appears in.

$$idf\,(t, D) = log\,(\ \frac{N}{count\ (d \in D : t \in d)}\ )$$

The reason we need IDF is to help correct for words like “of”, “as”, “the”, etc. since they appear frequently in an English corpus. Thus by taking inverse document frequency, we can minimize the weighting of frequent terms while making infrequent terms have a higher impact.

Putting it together: TF-IDF

To summarize the key intuition motivating TF-IDF is the importance of a term is inversely related to its frequency across documents.TF gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. By multiplying these values together we can get our final TF-IDF value.

$$tf\,idf\,(t, d, D) = tf\,(t, d)\ .\ idf\,(t, D)$$

Here are some common applications of TF-IDF:

1. **Information retrieval:** TF-IDF is used in search engines to rank the relevance of documents to a query. Documents that contain more occurrences of the query terms receive a higher score than those that contain fewer occurrences.
2. **Text classification:** TF-IDF can be used to classify documents into different categories. A document is represented as a vector of TF-IDF scores for each term in the vocabulary, and a classifier can be trained to predict the category based on the vector.
3. **Keyword extraction:** TF-IDF can be used to extract important keywords from a document. Terms with high TF-IDF scores are considered to be

more important and relevant to the document than those with low scores.

4. **Recommender systems:** TF-IDF can be used to recommend items to users based on their preferences. User profiles can be represented as vectors of TF-IDF scores for terms that appear in items they have interacted with, and items can be recommended based on their similarity to the user profile.

Pros of using TF-IDF

The biggest advantages of TF-IDF come from how simple and easy to use it is. It is simple to calculate, it is computationally cheap, and it is a simple starting point for similarity calculations (via TF-IDF vectorization + cosine similarity).

Cons of using TF-IDF

Something to be aware of is that TF-IDF cannot help carry semantic meaning. It considers the importance of the words due to how it weighs them, but it cannot necessarily derive the contexts of the words and understand importance that way.

6. Difference between data exploration and data presentation

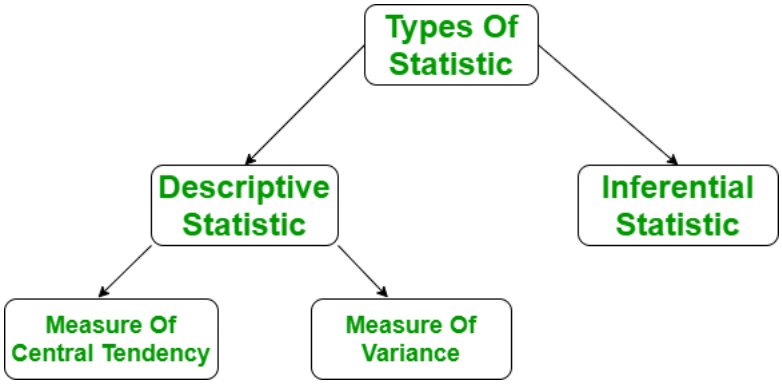
Data Exploration	Data Presentation
The process of analyzing and understanding data before any formal analysis is conducted	The process of presenting insights or findings from data analysis
Goal is to identify patterns, relationships, and trends in the data	Goal is to communicate insights and findings to an audience
Often involves visual exploration of the data through graphs, charts, and plots	Often involves creating static or interactive visualizations of the data
Typically an iterative process that involves refining and adjusting analyses as new insights are discovered	Typically a final step in the data analysis process that summarizes the key insights and findings
Can involve descriptive statistics such as mean, median, and standard deviation	Can involve a variety of visual aids such as tables, charts, infographics, and dashboards
Focuses on identifying potential biases, outliers, or missing values in the data	Focuses on presenting data in a clear, concise, and compelling way that is appropriate for the audience
Can involve hypothesis generation to guide further analysis	Can involve annotation and contextualization of the data to aid understanding

8.Explain about descriptive analysis in R

In Descriptive analysis, we are describing our data with the help of various representative methods like using charts, graphs, tables, excel files, etc. In the descriptive analysis, we describe our data in some manner and present it in a meaningful way so that it can be easily understood. Most of the time it is performed on small data sets and this analysis helps us a lot to predict some future trends based on the current findings. Some measures that are used to describe a data set are measures of central tendency and measures of variability or dispersion.

Process of Descriptive Analysis

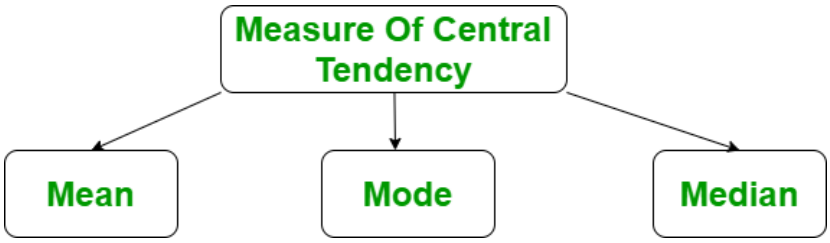
- The measure of central tendency
- Measure of variability



Measure of central tendency

It represents the whole set of data by a single value. It gives us the location of central points. There are three main measures of central tendency:

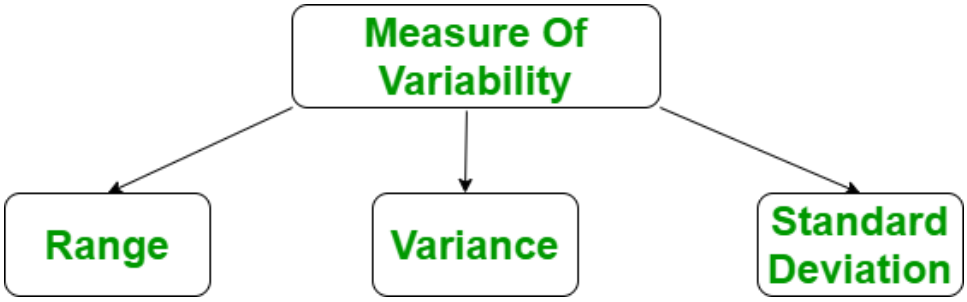
- Mean
- Mode
- Median



Measure of variability

Measure of variability is known as the spread of data or how well is our data is distributed. The most common variability measures are:

- Range
- Variance
- Standard deviation



Need of Descriptive Analysis

Descriptive Analysis helps us to understand our data and is a very important part of Machine Learning. This is due to Machine Learning being all about making predictions. On the other hand, statistics is all about drawing conclusions from data, which is a necessary initial step for Machine Learning. Let's do this descriptive analysis in R.

Descriptive Analysis in R

Descriptive analyses consist of describing simply the data using some summary statistics and graphics. Here, we'll describe how to compute summary statistics using R software.

Import your data into R:

Before doing any computation, first of all, we need to prepare our data, save our data in external .txt or .csv files and it's a best practice to save the file in the current directory. After that import, your data into R as follow:

Print the first 6 rows: -

```
# R program to illustrate
# Descriptive Analysis

# Import the data using read.csv()
myData = read.csv("CardioGoodFitness.csv",
                  stringsAsFactors = F)
# Print the first 6 rows
print(head(myData))
```

the mean value: -

```
# R program to illustrate
# Descriptive Analysis

# Import the data using read.csv()
myData = read.csv("CardioGoodFitness.csv",
                  stringsAsFactors = F)

# Compute the mean value
mean = mean(myData$Age)
print(mean)
```

the median value: -

```
# R program to illustrate
# Descriptive Analysis

# Import the data using read.csv()
myData = read.csv("CardioGoodFitness.csv",
                  stringsAsFactors = F)

# Compute the median value
median = median(myData$Age)
print(median)
```

Calculating variance: -

```
# R program to illustrate
# Descriptive Analysis

# Import the data using read.csv()
myData = read.csv("CardioGoodFitness.csv",
                  stringsAsFactors = F)

# Calculating variance
variance = var(myData$Age)
print(variance)
```

Calculating Standard deviation: -

```
# R program to illustrate
# Descriptive Analysis

# Import the data using read.csv()
myData = read.csv("CardioGoodFitness.csv", stringsAsFactors = F)

# Calculating Standard deviation
std = sd(myData$Age)
print(std)
```

Calculating Quartiles: -

```
# R program to illustrate
# Descriptive Analysis

# Import the data using read.csv()
myData = read.csv("CardioGoodFitness.csv", stringsAsFactors = F)

# Calculating Quartiles
```

```
quartiles = quantile(myData$Age)
print(quartiles)
```

8.what are the essential libraries for data analysis

There are many essential libraries for data analysis, but some of the most commonly used and important ones are:

1. **NumPy:** NumPy is a library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
2. **Pandas:** Pandas is a library for data manipulation and analysis. It provides easy-to-use data structures like DataFrames and Series, which make it easy to work with tabular data.
3. **Matplotlib:** Matplotlib is a library for creating visualizations and plots in Python. It provides a wide range of charts, graphs, and other visualizations to help you explore and communicate your data.
4. **Seaborn:** Seaborn is a library for creating more advanced visualizations and statistical graphics in Python. It provides a range of tools for visualizing relationships between variables and for exploring patterns in data.
5. **Scikit-learn:** Scikit-learn is a library for machine learning in Python. It provides a wide range of algorithms for classification, regression, clustering, and other tasks, along with tools for model selection and evaluation.
6. **Statsmodels:** Statsmodels is a library for statistical modeling and testing in Python. It provides a range of tools for fitting regression models, analyzing time series data, and conducting hypothesis tests.

These are just a few examples of the many libraries available for data analysis in Python. The choice of which libraries to use will depend on the specific needs of your analysis and the nature of the data you are working with.

9. Discuss different basic plotting with matplotlib

Matplotlib is a Python library that helps in visualizing and analyzing the data and helps in better understanding of the data with the help of graphical, pictorial visualizations that can be simulated using the matplotlib library. Matplotlib is a comprehensive library for static, animated and interactive visualizations.

plot(): -

it creates the plot at the background of computer, it doesn't displays it. We can also add a label as it's argument that by what name we will call this plot – utilized in legend()

show(): -

it displays the created plots

xlabel(): -

it labels the x-axis

ylabel(): -

it labels the y-axis

title(): -

it gives the title to the graph

gca(): -

it helps to get access over the all the four axes of the graph

xticks(): -

it decides how the markings are to be made on the x-axis

yticks(): -

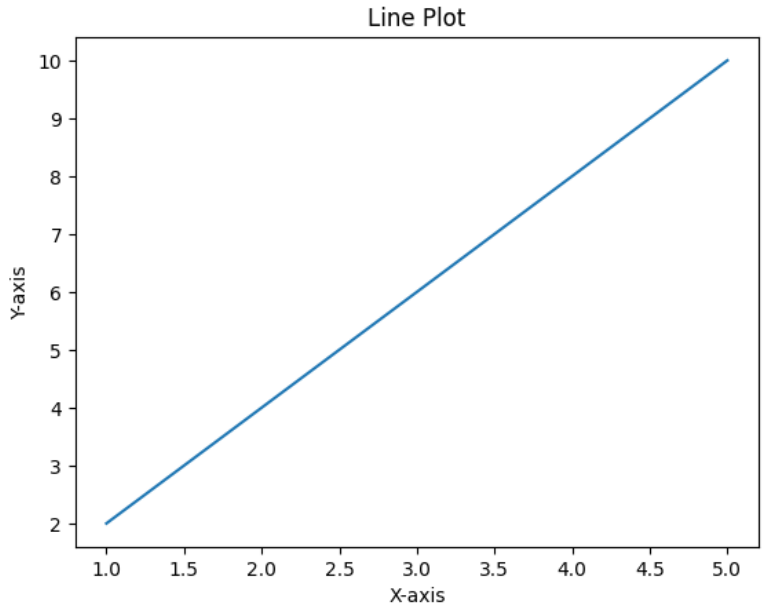
it decides how the markings are to be made on the y-axis

annotate(): -

it is use to write comments on the graph at the specified position

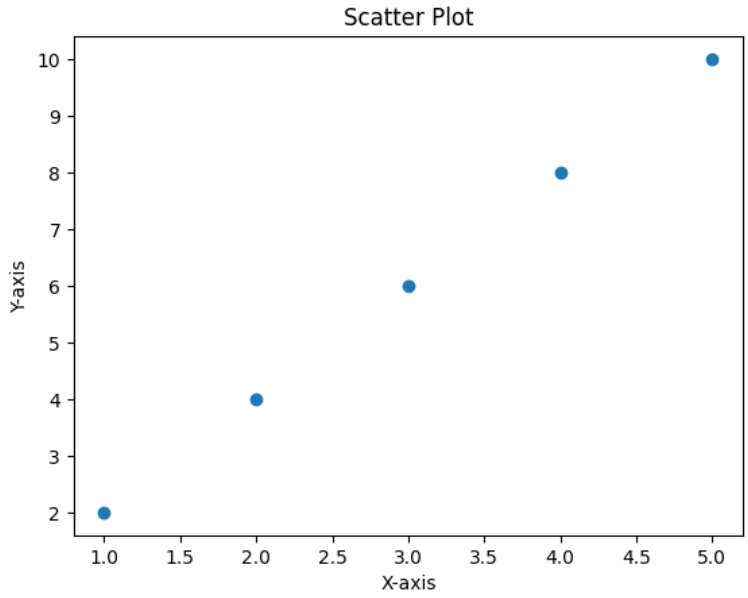
LINE PLOT

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.show()
```



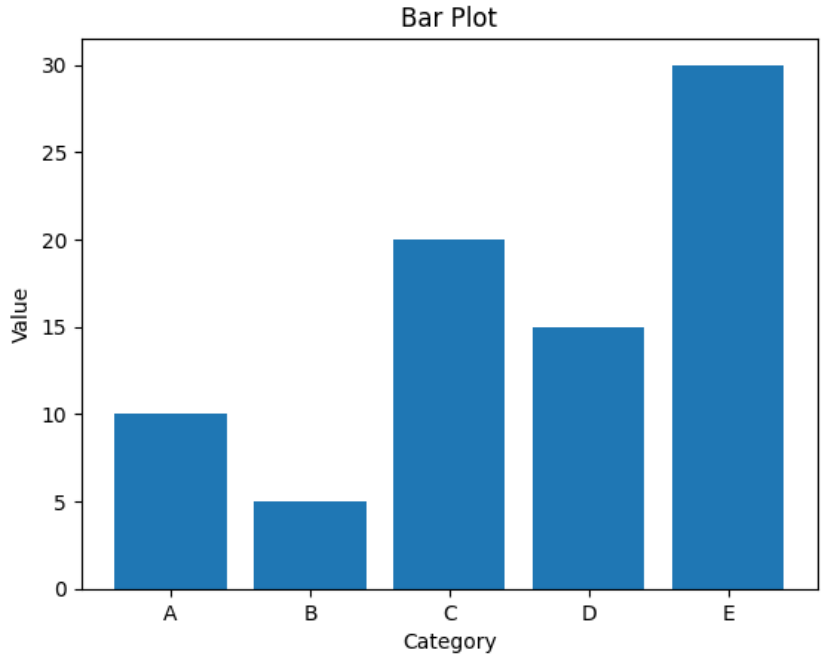
SCATTER PLOT

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.scatter(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()
```



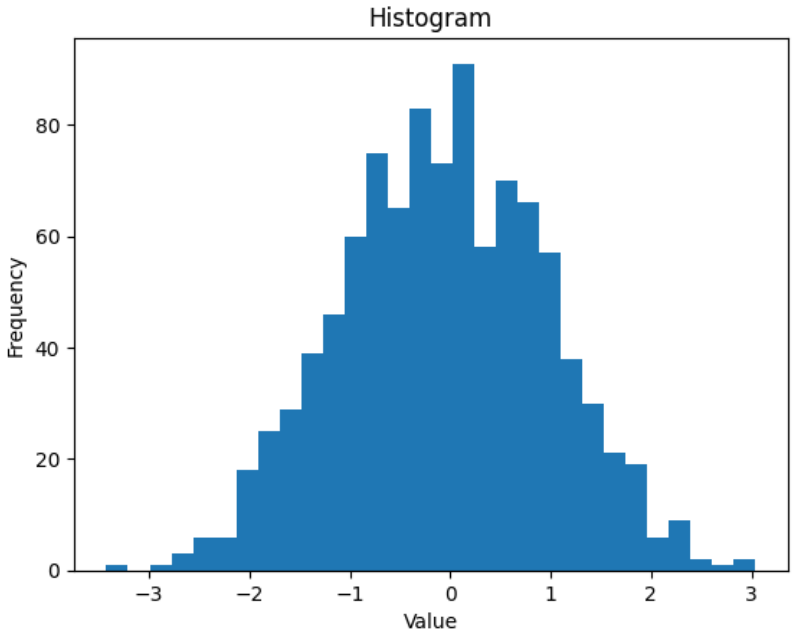
BAR PLOT

```
import matplotlib.pyplot as plt

x = ['A', 'B', 'C', 'D', 'E']
y = [10, 5, 20, 15, 30]

plt.bar(x, y)
plt.xlabel('Category')
plt.ylabel('Value')
plt.title('Bar Plot')
plt.show()
```



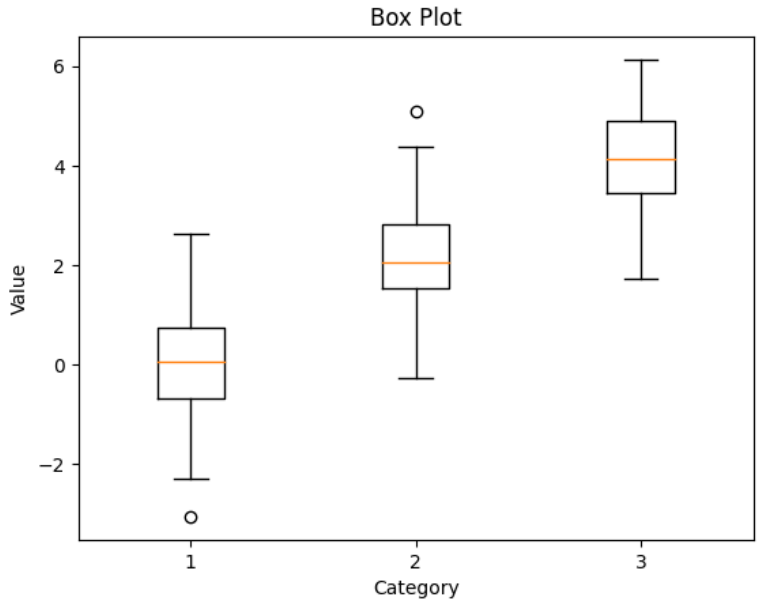
HISTOGRAM

```
import matplotlib.pyplot as plt
import numpy as np

data = np.random.normal(0, 1, 1000)

plt.hist(data, bins=30)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```



BOXPLOT

```
import matplotlib.pyplot as plt
import numpy as np

data = [np.random.normal(0, 1, 100),
        np.random.normal(2, 1, 100),
        np.random.normal(4, 1, 100)]

plt.boxplot(data)
plt.xlabel('Category')
plt.ylabel('Value')
plt.title('Box Plot')
plt.show()
```



10. Classify the exploratory data analysis in R 10M

Exploratory Data Analysis in R

In R Language, we are going to perform EDA under two broad classifications:

- **Descriptive Statistics**, which includes mean, median, mode, inter-quartile range, and so on.
- **Graphical Methods**, which includes histogram, density estimation, box plots, and so on.

Before we start working with EDA, we must perform the data inspection properly. Here in our analysis, we will be using the loafercreek from the soilDB package in R. We are going to inspect our data in order to find all the typos and blatant errors. Further EDA can be used to determine and identify the outliers and perform the required statistical analysis. For performing the EDA, we will have to install and load the following packages:

- “aqp” package
- “ggplot2” package
- “soilDB” package

We can install these packages from the R console using the install.packages() command and load them into our R Script by using the library() command. We will now see how to inspect our data and remove the typos and blatant errors.

Data Inspection for EDA in R

To ensure that we are dealing with the right information we need a clear view of your data at every stage of the transformation process. Data Inspection is the act of viewing data for verification and debugging purposes, before, during, or after a translation. Now let's see how to inspect and remove the errors and typos from the data.

Descriptive Statistics in EDA

For Descriptive Statistics in order to perform EDA in R, we will divide all the functions into the following categories:

- Measures of central tendency
- Measures of dispersion
- Correlation

We will try to determine the mid-point values using the functions under the Measures of Central tendency. Under this section, we will be calculating the mean, median, mode, and frequencies.

Graphical Method in EDA

Since we have already checked our data for missing values, blatant errors, and typos, we can now examine our data graphically in order to perform EDA. We will see the graphical representation under the following categories:

- Distributions
- Scatter and Line plot

Under the Distribution, we shall examine our data using the bar plot, Histogram, Density curve, box plots, and QQplot.

11. write a program to display boxplot, violin plot , heatmaps.

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Generate some random data
data = np.random.normal(size=(100, 5))

# Box plot
plt.figure()
plt.boxplot(data)
plt.title('Box plot')
plt.show()

# Violin plot
plt.figure()
sns.violinplot(data=data)
plt.title('Violin plot')
plt.show()

# Heatmap
plt.figure()
corr_matrix = np.corrcoef(data, rowvar=False)
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap')
plt.show()
```

12. Explain in detail about.

- **Regressionplot**
- **Multipleplot**

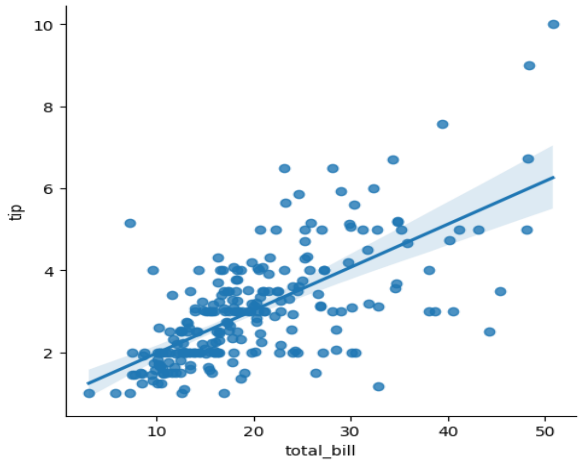
1.Regressionplot

A regression plot is a type of visualization that shows the relationship between two variables by fitting a linear regression model to the data. In other words, it helps to determine how strongly two variables are related and whether they have a positive or negative correlation.

A regression plot typically shows a scatter plot of the data points, along with a line of best fit that represents the regression model. It can also display confidence intervals around the regression line to show the level of uncertainty in the model. Regression plots are commonly used in data analysis, machine learning, and statistical modeling to explore and visualize relationships between variables.

Code that generates a simple regression plot:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the "tips" dataset from Seaborn library
tips = sns.load_dataset("tips")

# Create a regression plot using the "lmplot" function
sns.lmplot(x="total_bill", y="tip", data=tips)

# Show the plot
plt.show()
```



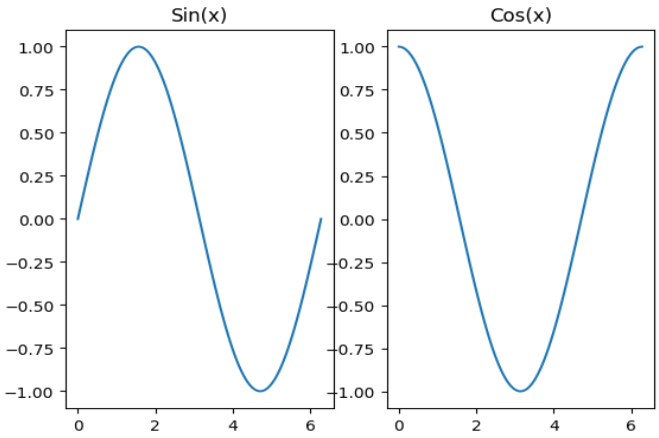
2.Multipleplot

A multiple plot is a type of visualization that shows multiple graphs or charts in the same figure, allowing for easy comparison and analysis of different data sets or variables. In other words, it helps to visualize multiple aspects of data in a single figure.

Multiple plots can be created using various techniques, depending on the programming language and visualization library used. In Python, we can use the Matplotlib library to create multiple plots using the subplot() function.

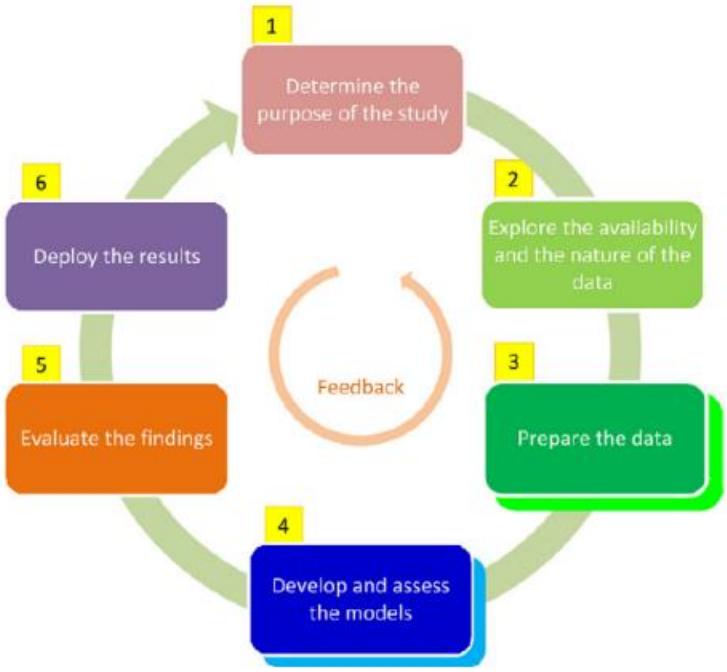
Here's an example code that generates a simple figure with multiple plots:

```
import matplotlib.pyplot as plt
import numpy as np

# Generate some random data
x = np.linspace(0, 2*np.pi, 100)
y1 = np.sin(x)
y2 = np.cos(x)

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2)

# Plot the first subplot
ax1.plot(x, y1)
ax1.set_title("Sin(x)")

# Plot the second subplot
ax2.plot(x, y2)
ax2.set_title("Cos(x)")

# Show the figure
plt.show()
```



13. Explain the text analysis steps in detail 10 M

Text analysis is the process of using computer systems to read and understand human-written text for business insights. Text analysis software can independently classify, sort, and extract information from text to identify patterns, relationships, sentiments, and other actionable knowledge. You can use text analysis to efficiently and accurately process multiple text-based sources such as emails, documents, social media content, and product reviews, like a human would.



Phase 1: Determine the Purpose of the Study

- Requires a thorough understanding of the business case and what the study aims to accomplish.
- To achieve this understanding and define the aims precisely, assess the nature of the problem (or opportunity) that initiated the study.
- Interact closely with the domain experts in order to develop an in-depth appreciation of the underlying system, its structure, its system constraints and the available resources.
- Develop a set of realistic goals and objectives to govern the direction of the study.

Phase 2: Explore the Availability and the Nature of the Data

- Ready to assess the availability, obtainability, and applicability of the necessary data in the context of the specific study.
- Identification of the textual data sources (digitized or paper-based; internal or external to the organization).
- Assessment of the accessibility and usability of the data
- Collection of an initial set of data
- Exploration of the richness of the data
- Assessment of the quantity and quality of the data.

Phase 3: Prepare the Data

Phase 4: Develop and Assess the Models

- The context diagram draws the boundaries around the process to explicitly show what is to be included (and/or excluded) from the representation of the text mining process.
- The primary purpose of text mining is to process unstructured (textual) data and structured and semistructured data (if relevant to the problem being addressed) to extract novel, meaningful, and actionable knowledge/information for better decision making.
- The inputs arrow in fig. to the text-based knowledge discovery process box are the unstructured, semistructured or structured data that are collected, stored and made available to the process.
- The outputs arrow represents the context-specific knowledge products that can be used for decision making.
- The constraints (or controls) arrow entering at the top edge of the box represents software and hardware limitations, privacy issues and the difficulties related to processing of the text that is presented in the form of natural language.
- The enablers entering the bottom of the box represents software tools, fast computers, domain expertise and natural language processing (NLP) methods

Phase 5: Evaluate the Results

- Verify and validate the proper execution of all of the activities.
- E.g. – Verify sampling was done properly and then repeat the steps to validate.
- Such a comprehensive assessment of the process helps to mitigate the possibility of error propagating into the decision-making process, potentially causing irreversible damage to the business.
- This assessment step is meant to make that connection one more time to ensure that the models developed and verified are actually addressing the business problem and satisfying the objectives they were built to satisfy.
- If this assessment leads to the conclusion that one or more of the business objectives are not satisfied, or there still is some important business issue that has not been sufficiently considered, then go back and correct these issues before moving into the deployment phase.

Phase 6: Deploy the Results

- Once the models and the modeling process successfully pass the assessment process, they can be deployed.
- Deployment of these models can be as simple as writing a report that explains the findings of the study in a way that appeals to the decision makers, or it can be as complex as building a new business intelligence system around these models (or integrating them into an existinbusiness intelligence system) so they can be used repetitively for better decision making.
- Some of the models will lose their accuracy and relevancy over time and needs to be updated (or refined) periodically with new data.
- This can be accomplished by executing a new analysis process every so often to re-create the models, or, more preferably, the business intelligence system itself can be designed in a way that it refines its models automatically as new and relevant data become available.
- Developing such a sophisticated system that is capable of self-assessing and self-adjusting is a challenging undertaking, once accomplished, the results would be very satisfying.