THIRD YEAR B. Sc.
COMPUTER SCIENCE

SEMESTER-V

NEW SYLLABUS
CBCS PATTERN

FOUNDATIONS OF DATA SCIENCE

Dr. Ms. MANISHA BHARAMBE
Dr. Mrs. HARSHA PATIL

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

SPPU New Syllabus

A Book Of

FOUNDATIONS OF DATA SCIENCE

For T.Y.B.Sc. Computer Science : Semester – V

[Course Code CS 354 : Credits - 2]

CBCS Pattern

As Per New Syllabus, Effective from June 2021

Dr. Ms. Manisha Bharambe

M.Sc. (Comp. Sci.), M.Phil. Ph.D. (Comp. Sci.)
Vice Principal, Associate Professor, Department of Computer Science
MES's Abasaheb Garware College
Pune

Dr. Mrs. Harsha Patil

M.C.A., M.Phil. Ph.D. (Comp. Sci.)
Asst. Professor, AEF's Ashoka Center for Business and Computer Studies,
Nashik

Price ₹ 240.00

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N5864

FOUNDATIONS OF DATA SCIENCE **ISBN 978-93-5451-187-5**

Second Edition : August 2022

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar,
Off J.M. Road, Pune – 411005
Tel - (020) 25512336/37/39
Email : niralipune@pragationline.com

Polyplate

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Survey No. 10/1A, Ghule Industrial Estate
Nanded Gaon Road
Nanded, Pune - 411041

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan
(For orders outside Pune)
S. No. 28/27, Dhayari Narhe Road, Near Asian College
Pune 411041, Maharashtra
Tel : (020) 24690204; Mobile : 9657703143
Email : bookorder@pragationline.com

Nirali Prakashan
(For orders within Pune)
119, Budhwar Peth, Jogeshwari Mandir Lane
Pune 411002, Maharashtra
Tel : (020) 2445 2044; Mobile : 9657703145
Email : niralilocal@pragationline.com

MUMBAI

Nirali Prakashan
Rasdhara Co-op. Hsg. Society Ltd., 'D' Wing Ground Floor, 385 S.V.P. Road
Girgaum, Mumbai 400004, Maharashtra
Mobile : 7045821020, Tel : (022) 2385 6339 / 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

DELHI

Nirali Prakashan
Room No. 2 Ground Floor
4575/15 Omkar Tower, Agarwal Road
Darya Ganj, New Delhi 110002
Mobile : 9555778814/9818561840
Email : delhi@niralibooks.com

BENGALURU

Nirali Prakashan
Maitri Ground Floor, Jaya Apartments,
No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003
Karnataka; Mob : 9686821074
Email : bengaluru@niralibooks.com

NAGPUR

Nirali Prakashan
Above Maratha Mandir, Shop No. 3,
First Floor, Rani Jhanshi Square,
Sitabuldi Nagpur 440012 (MAH)
Tel : (0712) 254 7129
Email : nagpur@niralibooks.com

KOLHAPUR

Nirali Prakashan
New Mahadvar Road, Kedar Plaza,
1st Floor Opp. IDBI Bank
Kolhapur 416 012 Maharashtra
Mob : 9850046155
Email : kolhapur@niralibooks.com

JALGAON

Nirali Prakashan
34, V. V. Golani Market, Navi Peth,
Jalgaon 425001, Maharashtra
Tel : (0257) 222 0395
Mob : 94234 91860
Email : jalgaon@niralibooks.com

SOLAPUR

Nirali Prakashan
R-158/2, Avanti Nagar, Near Golden
Gate, Pune Naka Chowk
Solapur 413001, Maharashtra
Mobile 9890918687
Email : solapur@niralibooks.com

marketing@pragationline.com | www.pragationline.com

Also find us on www.facebook.com/niralibooks

Preface ...

We take an opportunity to present this Text Book on **"Foundations of Data Science"** to the students of Third Year B.Sc. (Computer Science) Semester-V as per the New Syllabus, June 2021.

The book has its own unique features. It brings out the subject in a very simple and lucid manner for easy and comprehensive understanding of the basic concepts. The book covers theory of Introduction to Data Science, Statistical Data Analysis, Data Preprocessing and Data Visualization.

A special word of thank to Shri. Dineshbhai Furia, and Mr. Jignesh Furia for showing full faith in us to write this text book. We also thank to Mr. Amar Salunkhe and Mrs. Prachi Sawant of M/s Nirali Prakashan for their excellent co-operation.

We also thank Mr. Ravindra Walodare, Mr. Sachin Shinde, Mr. Ashok Bodke, Mr. Moshin Sayyed and Mr. Nitin Thorat.

Although every care has been taken to check mistakes and misprints, any errors, omission and suggestions from teachers and students for the improvement of this text book shall be most welcome.

Authors

Syllabus ...

1. Introduction to Data Science (6 Lectures)

- Introduction to Data Science, The 3 V's: Volume, Velocity, Variety
- Why Learn Data Science?
- Applications of Data Science
- The Data Science Lifecycle
- Data Scientist's Toolbox
- Types of Data
  - Structured, Semi-structured, Unstructured Data, Problems with Unstructured Data
  - Data Sources
  - Open Data, Social Media Data, Multimodal Data, Standard Datasets
- Data Formats
  - Integers, Floats, Text Data, Text Files, Dense Numerical Arrays, Compressed or Archived Data, CSV Files, JSON Files, XML Files, HTML Files, Tar Files, GZip Files, Zip Files, Image Files: Rasterized, Vectorized, and/or Compressed

2. Statistical Data Analysis (10 Lectures)

- Role of Statistics in Data Science
- Descriptive Statistics
  - Measuring the Frequency
  - Measuring the Central Tendency: Mean, Median, and Mode
  - Measuring the Dispersion: Range, Standard Deviation, Variance, Interquartile Range
- Inferential Statistics
  - Hypothesis Testing, Multiple Hypothesis Testing, Parameter Estimation Methods
- Measuring Data Similarity and Dissimilarity
  - Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes, Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Euclidean, Manhattan, and Minkowski Distances, Proximity Measures for Ordinal Attributes
- Concept of Outlier, Types of Outliers, Outlier Detection Methods

3. Data Preprocessing (10 Lectures)

- Data Objects and Attribute Types: What is an Attribute?, Nominal, Binary, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes
- Data Quality: Why Preprocess the Data?
- Data Munging/Wrangling Operations
- Cleaning Data
  - Missing Values, Noisy Data (Duplicate Entries, Multiple Entries for a Single Entity, Missing Entries, NULLs, Huge Outliers, Out-of-Date Data, Artificial Entries, Irregular Spacings, Formatting Issues - Irregular between Different Tables/Columns, Extra Whitespace, Irregular Capitalization, Inconsistent Delimiters, Irregular NULL Format, Invalid Characters, Incompatible Datetimes)

- Data Transformation:
  - Rescaling, Normalizing, Binarizing, Standardizing, Label and One Hot Encoding
- Data Reduction
- Data Discretization

4. Data Visualization (10 Lectures)

- Introduction to Exploratory Data Analysis
- Data Visualization and Visual Encoding
- Data Visualization Libraries
- Basic Data Visualization Tools
  - Histograms, Bar Charts/Graphs, Scatter Plots, Line Charts, Area Plots, Pie Charts, Donut Charts
- Specialized Data Visualization Tools
  - Boxplots, Bubble Plots, Heat Map, Dendrogram, Venn Diagram, Treemap, 3D Scatter Plots
- Advanced Data Visualization Tools - Wordclouds
- Visualization of Geospatial Data
- Data Visualization Types

❖❖❖

Contents ...

❖❖❖

CHAPTER 1

Introduction to Data Science

Objectives ...

- To learn Basic Concepts in Data Science
- To study Data Types, Data Sources, Data Formats
- To understand Life Cycle and Applications of Data Science

1.0 INTRODUCTION

- Today, with the emergence of new technologies, there has been an exponential increase in data and its growth continues. This has created an opportunity or need to analyze and derive meaningful insights from data.
- Now, handling of such huge amount of data is a challenging task. So to handle, process and analysis of this, we required some complex, powerful, and efficient algorithms and technology, and that technology came into existence as Data Science.
- Data science is a new area of research that is related to huge data and involves concepts like collecting, preparing, visualizing, managing and preserving.
- Data science is intended to analyze and understand the original phenomenon related to the data by revealing the hidden features of complex social, human and natural phenomena related to data from another point of view other than traditional methods.
- Data science is a collection of techniques used to extract value from data. Data science has become an essential tool for any organization that collects, stores and processes data as part of its operations.
- Data science is the art and science of acquiring knowledge through data. Data science techniques rely on finding useful patterns, connections and relationship within data.
- Data science is the process of deriving knowledge and insights from a huge and diverse set of data through organizing, processing and analyzing the data.
- Data science involves many different disciplines like mathematical and statistical modeling, extracting data from it source and applying data visualization techniques.

1.1 INTRODUCTION TO DATA SCIENCE

- In today's technology-driven world, the rate at which data is being generated per day is tremendous. To handling of such huge/massive amount of data is a challenging task for every organization.
- Data science is a deep study (analysis) of the massive amount of data, which involves extracting meaningful insights from raw, structured, and unstructured data.
- The term data science appeared in the computer science literature throughout the 1960s-1980s.
- Data science was not until the late 1990s however, after that it began to emerge from the statistics and data mining communities.
- Data science was first introduced as an independent discipline in 2001. Since, that time, there have been countless articles advancing the discipline, culminating with data scientist being declared the hottest job of the 21st century.
- Data Science (DS) refers to an emerging area of work concerned with the collection, preparation, analysis, visualization, management and preservation of large collections of information.
- Although the name data science seems to connect most strongly with areas such as databases and computer science, many different kinds of skills - including non-mathematical skills are needed.
- Our world is now measured, mapped, and recorded in digital bits. Entire lives, from birth to death, are now catalogued in the digital realm.
- These data, originating from such diverse sources as connected vehicles, underwater microscopic cameras, and photos we post to social media, have propelled us into the greatest age of discovery humanity has ever known.
- It is through data science that we are unlocking the secrets hidden within these data. We are making discoveries that will forever change how we live and interact with the world around us. Data science has become the catalyzing force behind our next evolutionary leap.
- Data science is being essentially the systematic study of the extraction of knowledge from data. In short, it's all about the difference between explaining and predicting.
- Data analysis has been generally used as a way of explaining some phenomenon by extracting interesting patterns from individual data sets with well-formulated queries.
- Data science, on the other hand, aims to discover and extract actionable knowledge from the data, that is, knowledge that can be used to make decisions and predictions, not just to explain what's going on.
- The raw materials of data science are not independent data sets, no matter how large they are, but heterogeneous, unstructured data set of all kinds, such as text, images, audio and video.

- The data scientist will not simply analyze the data, but will look at it from many angles, with the hope of discovering new insights.
- Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data.
  - Data science helps to solve complex problems using analytical approach. This happens through exploration of the data including data collection, storage, and processing of data using various tools and techniques, testing hypotheses, and creating conclusions with data and analyses as evidence. Such data may be structured, unstructured or semi structured data and can be generated by humans (surveys, logs, etc.) or machines (weather data, road vision, etc.).
  - Data science is becoming an essential field as companies/organizations produce larger and more diverse datasets. For most enterprises, the data discovery process begins with data scientists diving through massive sets while seeking strategies to focus them and provide better insights for analysis.
  - One of the biggest fields where data analytics software incorporates data science is in Internet search and recommendation engines. Companies like Google use data science and analytics to predict search values based on inputs, recommendations, and even recognition of images, video, and audio.
  - In retail, data science can simplify the process of targeting by improving the discovery part of the analysis and uncovering connections that are not readily visible, leading to better targeting and marketing efforts.
  - It uses techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, information science, and domain knowledge. However, data science is different from computer science and information science.

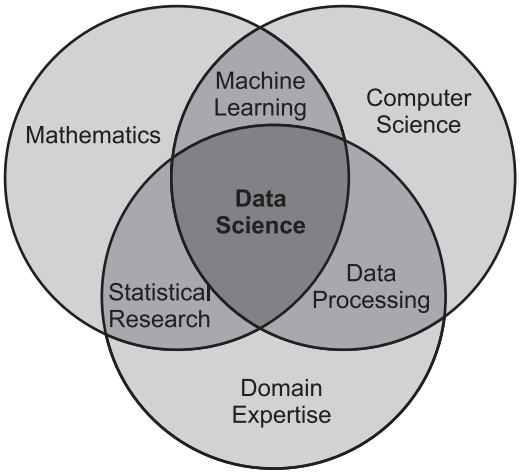


Fig. 1.1: Data Science and Associated Fields

- Data science is the task of scrutinizing and processing raw data to reach a meaningful conclusion.
- Data science and data analytics can gain meaningful insights that help companies in identifying possible areas of growth, streamlining of costs, better product opportunities, and effective company/organisation decisions.
- Data is mined and classified to detect and study behavioural data and patterns and the techniques used for this may vary according to the requirements.
- For data collection, there are two major sources of data – primary and secondary.
  1. **Primary data** is data that is never collected before and can be gathered in a variety of ways such as, participatory or non-participatory observation, conducting interviews, collecting data through questionnaires or schedules, and so on.
  2. **Secondary data**, on the other hand, is data that is already gathered and can be accessed and used by other users easily. Secondary data can be from existing case studies, government reports, newspapers, journals, books and also from many popular dedicated websites that provide several datasets.
- Few standard popular websites for downloading datasets include the UCI Machine Learning Repository, the Kaggle datasets, IMDB datasets and Stanford Large Network Dataset Collection.

Process of Data Science:

- Data science builds algorithms and systems for discovering knowledge, detecting the patterns, and generating useful information from massive data.
- To do so, it encompasses an entire data analysis process that starts with the extraction of data and cleaning, and extends to data analysis, description, and summarization.
- Fig. 1.2 shows process of data science. The process of data science starts with data collection.

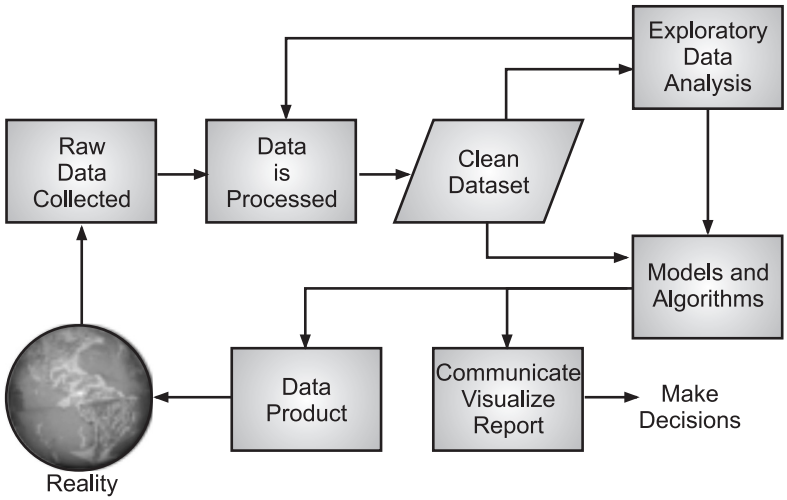


Fig. 1.2: Process of Data Science

- Next, the data is cleaned to select the segment that has the most valuable information. To do so, the user will filter over the data or formulate queries that can erase unnecessary information.
- After the data is prepared, an exploratory analysis that includes visualizing tools will help decide the algorithms that are suitable to gain the required knowledge.
- A data product is a computer application that takes data inputs and generates outputs, feeding them back into the environment.

Advantages of Data Science:

1. Data science helps to extract meaningful information from raw data.
2. Data science is a very versatile used in fields like healthcare, banking and e-commerce, transport industries etc.
3. Data science improves the quality of data.
4. Data science improves quality of services and products

Disadvantages of Data Science:

1. **Data Privacy:** The information or the insights obtained from the data can be misused against any organization or a group of people.
2. **Expensive:** The tools used for data science are more expensive to use to obtain information. The tools are also more complex, so people have to learn how to use them.
3. **Difficult to Selection of Tools:** It is very difficult to select the right tools according to the circumstances because their selection is based on the proper knowledge of the tools as well as their accuracy in analyzing the data and extracting information.

1.1.1 The 3 V's (Volume, Velocity, Variety)

- Why is data science so important now? We have a lot of data, we continue to generate a staggering amount of data at an unprecedented and ever-increasing speed, analyzing data wisely necessitates the involvement of competent and well-trained practitioners, and analyzing such data can provide actionable insights.
- The need for complex data analysis has been immensely felt over these years in main business sectors and companies to discover historical patterns for improving the performance of the business in the future.
- Data science has three specific areas Volume, Variety and Velocity (known as 3V's) dealing with data.
- Due to the expansion of data at the turn of the twenty-first century epitomized by the so-called 3Vs of data science, which are volume, velocity, and variety.

- Volume refers to the increasing size of data, velocity the speed at which data is acquired, and variety the diverse types of data that are available.
- The 3V's are explained below:
  1. **Velocity:** The speed at which data is accumulated.
  2. **Volume:** The size and scope of the data.
  3. **Variety:** The massive array of data and types (structured and unstructured).

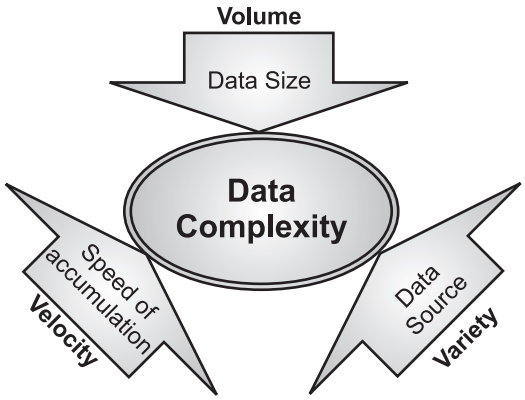


Fig. 1.3: 3V's in Data Science

- Each of these three Vs regarding data has dramatically increased in recent years. Specifically, the increasing volume of heterogeneous and unstructured (text, images, and video) data, as well as the possibilities emerging from their analysis, renders data science evermore essential.

1.2 WHY LEARN DATA SCIENCE?

- In today's fast-growing data-driven world, data science has been providing us the tools for study the behavioural pattern of customers in case of online purchases, stock market investment and advertising products to other customers and so on.
- These activities require an in-depth analysis of existing relevant data and helps for operational and strategic decision making for many Industries like:
  1. **Finance**: The finance market is an emerging field in the data industry. The financial analytics market takes care of risk analysis, fraud detection, shareholders 'upcoming share status, working capital management, and soon.
  2. **Healthcare**: The healthcare sector also nowadays heavily relies on analytics of patient data to predict diseases and health issues. Healthcare industries make an analysis of data-driven patient quality care, improved patient care, classification of the type of symptoms of patients and predicted health deficiencies, and so on.

3. **Ecommerce**: Ecommerce sites hugely involve data science for maximizing revenue and profitability. These sites analyze the shopping and purchasing behavior of customers and accordingly recommend products to customers for more purchases online.
4. **Sports:** Nowadays, sports analytics is often used in international tournaments to analyze the performance of players, the predicted scores, prevention of injuries, and the possibility of winning or losing a match by a particular team.
5. **Retail**: Retail industries take care of a 360-degree view and feedback reviews of customers. The retail analytics market analyzes customers' purchasing trends and demands in order to get products based on customers' liking. Retail industries involve data science for optimal pricing, personalized offers, better marketing strategies, market basket analysis, stock management, and so on.
6. **Human Resource (HR)**: HR analytics involves HR-related data that can be used for building strong leadership, employee acquisition, employee retention, workforce optimization, and performance management.
7. **Education**: The sources of data in education is vast, starting from student centric data, enrollment in various courses, scholarship and fee details, examination results, and so on. Education analytics play a major role in academic institutions for better admission scenario, empowerment of students for successful examination results, and all-round student performance.

- Other number of sectors like telecom industries, sales, supply chain management, risk monitoring, manufacturing industries, and IT companies are also prominent domain for using data science for strategic decision making.
- The recent competitions in businesses and companies consider data science no longer as an optional requirement but rather hire data analysts and data scientists for the same to deal with hidden massive data to provide meaningful results and generate reports to arrive at profit-making decisions.
- Also, in the recent trends in the job market show that data analysts, data scientists, data architect and data engineers have a huge demand in the IT companies and this demand will continue for the next decade.
- Data analyst is an individual, who performs extracting information from massive amount of data.
- The main role of a data analyst is to extract data and interpret the information attained from the data for analyzing the outcome of a given problem.
- A data scientist is a person who works with an enormous amount of data to obtain meaningful insights using deployment tools, techniques, algorithms, etc.
- Data scientists mainly deal with large and complex data that can be of high dimension, and carry out appropriate machine learning and visualization tools to convert the complex data into easily interpretable meaningful information.

- A data engineer works with massive amount of data and responsible for building and maintaining the data architecture of a data science project.
- The role of a data engineer is not to analyze data but rather to prepare, manage and convert data into a form that can be readily used by a data analyst or data scientist.
- The data architect provides the support of various tools and platforms that are required by data engineers to carry out various tests with precision.
- The main task/role of data architects is to design and implement database systems, data models and components of data architecture.

1.3 APPLICATIONS OF DATA SCIENCE

- Traditionally, the data was mostly structured and small in size, which could be analyzed by using the simple BI (Business Intelligence) tools.
- Unlike data in the traditional systems which was mostly structured, today most of the data is unstructured or semi-structured.
- This data is generated from different sources like financial logs, text files, multimedia forms, sensors, and instruments.
- Simple BI tools are not capable of processing this huge volume and variety of data. This is why we need more complex and advanced analytical tools and algorithms for processing, analyzing and drawing meaningful insights out of it.
- Data science is all about uncovering findings from data. Diving in at a granular level to mine and understand complex behaviors, trends, and inferences.
- It is about surfacing hidden insight that can help enable companies to make smarter business decisions.
- For example, Netflix data mines movie viewing patterns to understand what drives user interest, and uses that to make decisions on which Netflix original series to produce.
- Target identifies what are major customer segments within its base and the unique shopping behaviours within those segments, which helps to guide messaging to different market audiences.
- Proctor and Gamble utilizes time series models to more clearly understand future demand, which help plan for production levels more optimally.
- The following are the applications of the data science:
  1. **Image Recognition and Speech Recognition:** Image and speech recognition is a prominent application area for Data Science. When we upload an image on any social media, it automatic tagging suggestion uses image recognition algorithm, which is part of data science.

2. **Gaming World:** In the gaming world, the use of data science is increasing for enhancing user experience.
3. **Internet Search:** When we want to search for something on the internet, then we use different types of search engines such as Google, Yahoo, Bing, Ask, etc. All these search engines use the data science technology to make the search experience better and we can get a search result with a fraction of seconds.
4. **Transport:** Transport industries also using data science technology to create self-driving cars. With self-driving cars, it will be easy to reduce the number of road accidents.
5. **Healthcare:** In the healthcare sector, data science is providing lots of benefits. Data science is being used for tumor detection, drug discovery, medical image analysis, virtual medical bots, etc.
6. **Recommendation Systems:** Most of the companies, such as Amazon, Netflix, Google Play, etc., are using data science technology for making a better user experience with personalized recommendations. Such as, when we search for something on Amazon, and we started getting suggestions for similar products, so this is because of data science technology.
7. **Risk Detection:** Finance industries always had an issue of fraud and risk of losses, but with the help of data science, this can be rescued. Most of the finance companies are looking for the data scientist to avoid risk and any type of losses with an increase in customer satisfaction.

1.4 DATA SCIENCE LIFE CYCLE

- The life cycle of data science outlines the steps/phases, from start to finish, that projects usually follow when they are executed.
- The lifecycle of the data analytics provides a framework for the best performances of each phase from the creation of the project until its completion.
- Fig. 1.4 shows the steps of data science life cycle.
- Each step in the data science life cycle explained above should be worked upon carefully. If any step is executed improperly, it will affect the next step, and the entire effort goes to waste.
- For example, if data is not collected properly, we will lose information and we will not be building a perfect model. If data is not cleaned properly, the model will not work.
- If the model is not evaluated properly, it will fail in the real world. From business understanding to model deployment, each step should be given proper attention, time and effort.

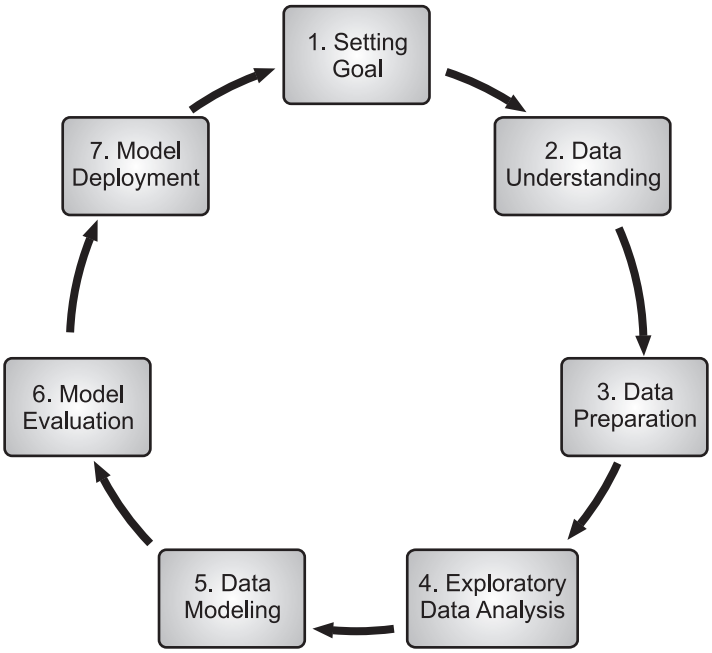


Fig. 1.4: Life Cycle of Data Science

- The lifecycle of the data science project contains following steps:

Step 1: Setting Goal

- The entire cycle revolves around the business or research goal. What will we solve if we do not have a precise problem? It is essential to understand the business objective clearly because that will be the final goal of the analysis.
- Only we can set the specific goal of analysis in synchronised with the business objective after proper understanding.

Step 2: Data Understanding

- Data understanding involves the collection of all the available data.
- We need to understand what data is present and what data could be used for given problem.
- The data understanding step also involves describing the data, their structure, their relevance, their data type.

Step 3: Data Preparation

- The data preparation step includes selecting the relevant data, integrating the data by merging the data sets, cleaning them, treating the missing values by either removing them or imputing them, treating erroneous data by removing them and checking outliers using box plots and handle them.

- This step is used for constructing new data derive new features from existing ones. Format the data into the desired structure, remove unwanted columns and features.
- Data preparation is the most time consuming yet arguably the most important step in the entire life cycle. The model will be as good as the data.

Step 4: Exploratory Data Analysis

- This step involves getting some idea about the solution and factors affecting it before building the actual model.
- The distribution of data within different feature variables is explored graphically using bar-graphs; relations between different features are captured through graphical representations like scatter plots and heat maps.
- Many other data visualization techniques are extensively used to explore every feature individually and combine them with other features.

Step 5: Data Modeling

- Data modeling is the heart of data analysis. A model takes the prepared data as input and provides the desired output.
- Data modeling step includes choosing the appropriate type of model, whether the problem is a classification problem, or a regression problem or a clustering problem.
- After choosing the model family, amongst the various s amongst that family, we need to choose the algorithms to implement and implement them carefully.
- We need to tune the hyper parameters of each model to achieve the desired performance.
- We also need to make sure there is a correct balance between performance and generalizability. We do not want the model to learn the data and perform poorly on new data.

Step 6: Model Evaluation

- In this step, the model is evaluated for checking if it is ready to be deployed. The model is tested on unseen data, evaluated on a carefully thought out set of evaluation metrics.
- We also need to make sure that the model conforms to reality. If we do not obtain a satisfactory result in the evaluation, we must re-iterate the entire modeling process until the desired level of metrics is achieved.
- Any data science solution, a machine learning model, just like a human, should evolve, should be able to improve itself with new data, adapt to a new evaluation metric.
- We can build multiple models for a certain phenomenon, but a lot of them may be imperfect. Model evaluation helps us choose and build a perfect model.

Step 7: Model Deployment

- The model, after a rigorous evaluation, is finally deployed in the desired format and channel. This is the final step in the data science life cycle.
- One goal of a project is to change a process and/or make better decisions. We may still need to convince the business that our findings will indeed change the business process as expected.
- The importance of this step is more apparent in projects on a strategic and tactical level. Certain projects require us to perform the business process over and over again, so automating the project will save time.

1.5 DATA SCIENTIST's TOOLBOX

- A data scientist is a professional who responsible for extracting, manipulating, pre-processing and generating predictions out of data. In order to do so, he/she requires various statistical tools and programming languages.
- There are many popular tools and techniques used by data scientists and data analysts.
- One best thing about these tools is that most of these tools are popular, user-friendly and open-source and provide good performance in the field of data science.
- Let us discuss some open-source tools that can be learned and adapted by any beginner or researcher who wants to explore in the field of data science.

1. Python Programming:

- Choosing the right programming language for Data Science is of utmost importance. Python offers various libraries designed explicitly for Data Science operations.
- Python programming language is an open-source tool and falls under object-oriented scripting language. It was found in the 1980s by Guido van Rossum.
- Python programming language is popular for the implementation of data preprocessing, statistical analysis, machine learning and deep learning, which are the core tasks in any data science project.
- Python programming language is versatile and can run on any platform such as UNIX, Windows, and Mac operating systems. It can also be assembled on any database platform like the SQL server or a MongoDB database.
- The rich set of libraries are (more than 200,000) the core strengths of Python language. Many types of visualization graphs can also be plotted using Python codes.

2. R Programming:

- R programming is a popular language used in the Data Science provides a scalable software environment for statistical analysis.

- R programming is also an open-source tool. It was developed by Ross Ihaka and Robert Gentleman, both of whose first names start with the letter R and hence the name 'R' has been given for this language.
- R programming is versatile and can run on any platform such as UNIX, Windows, and Mac operating systems.
- R programming also has a rich collection of libraries (more than 11,556) that can be easily installed as per requirements.
- This makes the R programming language popular and is widely used for data analytics for handling major tasks such as classical statistical tests, time-series forecasting and machine learning such as classification and regression, and many more.
- The basic visualization graphs can also be effortlessly plotted through R programming codes that make data interpretation easy using this language.

3. SAS (Statistical Analysis System):

- SAS is used by large organizations to analyze data uses SAS programming language which for performing statistical modeling.
- SAS offers numerous statistical libraries and tools that we as a Data Scientist can use for modeling and organizing their data.
- The SAS is a programming environment and language used for advanced data handling such as criminal investigation, business intelligence, and predictive analysis.
- It was initially released in 1976 and has been written in C language. It is supported in various operating systems such as Windows, Unix/Linux, and IBM mainframes.
- It is mainly used for integrating data from multiple sources and generating statistical results based on the input data fed into the environment.
- SAS data can be generated in a wide variety of formats such as PDF, HTML, Excel, and many more.

4. Tableau Public:

- Tableau is data visualization software which has its free version named as Tableau Public. It is data visualization software/tool that is packed with powerful graphics to make interactive visualizations.
- Tableau was developed in 2003 by four founders from the United States. It has an interesting interface that allows connectivity to both local and cloud-based data sources.
- The preparation, analysis, and presentation of input data can be all done in Tableau with various drag and drop features and easy available menus.
- Tableau software is well-suited for big-data analytics and generates powerful data visualization graphs that make it very popular in the data analytics market.

- A very interesting functionality of Tableau software is its ability to plot latitude and longitude coordinates for geospatial data and generate graphical maps based on these coordinate values.

5. Microsoft Excel:

- Microsoft Excel is an analytical tool for Data Science used by data scientists for data visualization.
- Excel represents the data in a simple way using rows and columns and comes with various formulae, filters for data science.
- Microsoft Excel is a data analytics tool widely used due to its simplicity and easy interpretation of complex data analytical tasks.
- Excel was released in the year 1987 by the Microsoft Company to handle numerical calculations efficiently.
- Microsoft Excel is of type spreadsheet and can handle complex numerical calculations, generate pivot tables, and display graphics.
- An analyst may use R, Python, SAS or Tableau and will also still use MS Excel for its simplicity and efficient data modeling capabilities.
- However, Microsoft Excel is not an open-source application and can be used if one has Windows, macOS or Android operating system installed in one's machine.

6. RapidMiner:

- The RapidMiner is a widely used Data Science software tool due to its capacity to provide a suitable environment for data preparation.
- Any Data Science model can be prepared from scratch using RapidMiner. Data scientists can track data in real-time using RapidMiner and can perform high-end analytics.
- RapidMiner is a data science software platform developed by the RapidMiner Company in the year 2006.
- RapidMiner is written in the Java language and has a GUI that is used for designing and executing workflows related to data analytics.
- RapidMiner also has template-based frameworks that can handle several data analysis tasks such as data preprocessing, data mining, machine learning, ETL handling, and data visualization.
- The RapidMiner Studio Free Edition has one logical processor and can be used by a beginner who wants to master the software for data analysis.

7. Apache Spark:

- Apache Spark based on the Hadoop, MapReduce, can handle interactive queries and stream processing. Apache Spark is open-source software developed in 2014 by the Apache Spark developers.

- Apache Spark is versatile and can run on any platform such as UNIX, Windows, and Mac operating systems.
- Spark has a remarkable advantage of having high speed when dealing with large datasets and is found to be more efficient than the MapReduce technique used in a Hadoop framework.
- Many libraries are built on top of the Spark Core that helps in enabling many data analysis tasks such as handling SQL queries, drawing visualization graphs, and machine learning.
- Other than the Spark Core, the other components available in Apache Spark are Spark SQL, Spark Streaming, MLlib (Machine Learning Library), and GraphX.
- Apache Spark has become one of the best Data Science tools in the market due to its in-memory cluster computing.

8. Knime:

- Knime is one of the widely used Data Science tools for data reporting, mining, and analysis.
- It's ability to perform data extraction and transformation makes it one of the essential tools used in Data Science. The Knime platform is open-source and free to use in various parts of the world.
- Knime (Konstanz Information Miner) Analytics platform is an open-source data analytics and reporting platform.
- Knime was developed in 2004 by a team of software engineers from Germany. It is mainly used for applying statistical analysis,data mining, ETL handling, and machine learning.
- The Knime workbench has several components such as Knime Explorer, Workflow editor, Workflow Coach,Node Repository, Description, Outline, Knime Hub Search, and Console.
- The core architecture of Knime is designed in such a way that it practically has almost no limitations on the input data fed into the system.
- This is a big advantage of using Knime as a data science tool as large volumes of data are needed to be dealt with for analysis in data science.

9. Apache Flink:

- It is one of the best Data Science tools offered by the Apache Software Foundation in 2020/2021.
- Apache Flink can quickly carry out real-time data analysis. Apache Flink is an open-source distributed framework that can perform scalable Data Science computations.
- The Flink helps data scientists in reducing complexity while real-time data processing.

1.6 TYPES OF DATA

- Data is a set of raw facts such as descriptions, observations and numbers that needs to be processed to make it meaningful. Processed data in a meaningful way is known as information.
- One purpose of Data Science is to structure data, making it interpretable and easy and simply to work with.
- In data science, we come across many different types of data, and each of them tends to require different tools and techniques.
- We can classify data as structured data, unstructured data and semi-structured data. The structured data resides in predefined formats.
- The unstructured data is stored in its natural format until it's extracted for analysis. The semi-structured data basically is a mix of both structured and unstructured data.

1.6.1 Structured Data

- Structured data as name suggest type of data is well organized. Structured data is data that depends on a data model and resides in a fixed field within a record.
- Structured data is comprised of clearly defined data types whose pattern makes them easily searchable. It is often easy to store structured data in tables within databases or Excel files.
- Fields store length-delineated data phone numbers, Social Security numbers, or ZIP codes. Even text strings of variable length like names are contained in records, making it a simple matter to search.
- Data may be human- or machine-generated as long as the data is created within an RDBMS structure.
- This format is eminently searchable both with human generated queries and via algorithms using type of data and field names, such as alphabetical or numeric, currency or date.
- SQL or Structured Query Language is the preferred way to manage and query data that resides in databases.
- PostgreSQL is a tool for structured data supports SQL and JSON querying as well as high-tier programming languages (C/C+, Java, Python, etc.).

1.6.2 Unstructured Data

- Unstructured data is a data that is not organized in a pre-defined manner or does not have a pre-defined data model.

- Unstructured data is data that does not fit into a data model because the content is context-specific or varying. One example of unstructured data is the regular email message.
- Unstructured data has internal structure but is not structured via pre-defined data models or schema.
- It may be textual or non-textual and human- or machine-generated. It may also be stored within a non-relational database like NoSQL.
- Natural language is a special type of unstructured data. It is challenging to process unstructured data because it requires knowledge of specific data science techniques and linguistics.

Typical Human-generated Unstructured Data:

- **Text Files:** Word processing, spreadsheets, presentations, email, logs.
- **Email:** Email has some internal structure thanks to its metadata, and we sometimes refer to it as semi-structured. However, its message field is unstructured and traditional analytics tools cannot parse it.
    1. **Social Media:** Data from Facebook, Twitter, LinkedIn.
    2. **Website:** Data from YouTube, Instagram, photo sharing sites.
    3. **Mobile Data:** Data from Text messages, locations.
    4. **Communications:** Chat, IM, phone recordings, collaboration software.
    5. **Media:** Data from MP3, digital photos, audio and video files.
    6. **Business Applications:** Data from MS Office documents, productivity applications.

Typical Machine-generated Unstructured Data:

- Machine-generated data is information that is automatically created by a computer, process, application, or other machine without human intervention. Machine-generated data is becoming a major data resource and will continue to do so.
- For example, it includes:
    1. **Satellite Imagery:** Data from Weather data, land forms, military movements.
    2. **Scientific Data:** Data from Oil and gas exploration, space exploration, seismic imagery, atmospheric data.
    3. **Digital Surveillance:** Data from Surveillance photos and video.
    4. **Sensor Data:** Data from Traffic, weather, oceanographic sensors.

Unstructured Data Tools:

1. **MongoDB**: Uses flexible documents to process data for cross-platform applications and services.

2. **Hadoop**: Provides distributed processing of large data sets using simple programming models and no formatting requirements.
3. **Azure**: Enables agile cloud computing for creating and managing apps through Microsoft's data centers.

1.6.3 Semi-structured Data

- Semi-structured data is a data type that contains semantic tags, but does not conform to the structure associated with typical relational databases.
- It maintains internal tags and markings that identify separate data elements, which enables information grouping and hierarchies.
- Both documents and databases can be semi-structured. This type of data only represents about 5-10% of the structured/semi-structured/unstructured data pie, but has critical business usage cases.
- Email is a very common example of a semi-structured data type. Examples of semi-structured data includes:
  1. **Markup Language XML:** This is a semi-structured document language. XML is a set of document encoding rules that defines a human- and machine-readable format. Its value is that its tag-driven structure is highly flexible, and coders can adapt it to universalize data structure, storage, and transport on the Web.
  2. **Open Standard JSON (JavaScript Object Notation) JSON:** It is another semi-structured data interchange format. Java is implicit in the name but other C-like programming languages recognize it. Its structure consists of name/value pairs (or object, hash table, etc.) and an ordered value list (or array, sequence, list). Since the structure is interchangeable among languages, JSON excels at transmitting data between web applications and servers.
  3. **NoSQL:** Semi-structured data is also an important element of many NoSQL (Not only SQL) databases. NoSQL databases differ from relational databases because they do not separate the organization (schema) from the data. This makes NoSQL a better choice to store information that does not easily fit into the record and table format, such as text with varying lengths. It also allows for easier data exchange between databases.
- Above databases are common in big data infrastructure and real-time Web applications like LinkedIn.
- On LinkedIn, hundreds of millions of business users freely share job titles, locations, skills, and more; and LinkedIn captures the massive data in a semi-structured format.
- When job seeking users create a search, LinkedIn matches the query to its massive semi-structured data stores, cross-references data to hiring trends, and shares the resulting recommendations with job seekers.

- The same process operates with sales and marketing queries in premium LinkedIn services like Salesforce. Amazon also bases its reader recommendations on semi-structured databases.

Differences between Structured, Semi-structured and Unstructured Data:

Sr. No.	Key	Structured Data	Semi Structured Data	Unstructured Data
1.	Level of organizing	Structured data as name suggest this type of data is well organized and hence level of organizing is highest in this type of data.	Semi structured data the data is organized up to some extent only and rest is non organized hence the level of organizing is less than that of Structured Data and higher than that of Unstructured Data.	Unstructured data is non organized, hence level of organizing is lowest in case of Unstructured Data.
2.	Means of data organization	Structured data is get organized by the means of Relational Database.	Semi structured data is partially organized by the means of XML/RDF.	Unstructured data is based on simple character and binary data.
3.	Transaction management	In structured data management and concurrency of data is present and hence mostly preferred in multitasking process.	In semi structured data transaction is not by default but is get adapted from DBMS but data concurrency is not present.	While in unstructured data no transaction management and no concurrency are present.
4.	Versioning	Structured data supports in RDB so versioning is done over tuples, rows and table as well.	Semi structured data versioning is done only where tuples or graph is possible as partial database is supported in case of semi structured data.	Versioning in case of unstructured data is possible only as on whole data as no support of database at all.

Contd...

5.	Flexible and Scalable	As structured data is based on relational database so it becomes schema dependent and less flexible as well as less scalable.	While in case semi structured data data is more flexible than structured data but less flexible and scalable as compare to unstructured data.	As there is no dependency on any database so unstructured data is more flexible and scalable as compare to structured and semi structured data.
6.	Performance	In structure data we can perform structured query which allow complex joining and thus performance is highest as compare to that of semi structured and unstructured data.	In semi structured data only queries over anonymous nodes are possible so its performance is lower than structured data but more than that of unstructured data	In unstructured data only textual query are possible so performance is lower than both structured and semi structured data.
7.	Technology	It is based on Relational database table.	It is based on XML/RDF.	It is based on character and binary data.
8.	Transaction management	Matured transaction and various concurrency techniques.	Transaction is adapted from DBMS not matured.	No transaction management and no concurrency.
9.	Flexibility	It is schema dependent and less flexible.	It is more flexible than structured data but less than flexible than unstructured data.	It very flexible and there is absence of schema.
10.	Robustness	Very robust.	New technology, not very spread.	–
11.	Query performance	Structured query allow complex joining.	Queries over anonymous nodes are possible.	Only textual query are possible.

1.6.4 Problems with Unstructured Data

- Unstructured data is not organized in a predefined manner. It generates immense business value, but most organizations have not been able to yield insights because there are simply so many challenges involved in analyzing unstructured data.
- Following are some common problems associated with unstructured data:
  1. **Unstructured Data Keeps Expanding:** Unstructured data continues to grow at an exponential rate and experts believe that it will make up over 93% of data by 2022. This large volume is going to be a huge challenge in analysing this type of data because the larger the data set, the harder it is to store and analyse data in a way that is timely and efficient.
  2. **Time Consuming:** The lack of structure makes compilation and organizing unstructured data a time- and energy-consuming task.
  3. **Not all Unstructured Data is High Quality:** Unstructured data can be very uneven when it comes to quality. The lack of consistency in quality occurs because data is difficult to verify and therefore, is not always accurate. For example, Facebook status updates, images and videos all qualify as unstructured data, but that does not make it useful for organizations.
  4. **Data cannot be Analysed with Conventional Systems:** Unstructured data cannot be analysed with current databases because most data analytics databases are designed for structured data, and are not equipped for unstructured data. Therefore, data analytics experts need to find new methods to locate, extract, organise and store data. Unstructured data comes in different formats and databases that need to reflect the freeform state of the data.

1.7 DATA SOURCES

- A data source in data science is the initial location where data that is being used come from.
- Data collection is the process of acquiring, collecting, extracting, and storing the huge amount of data which may be in the structured or unstructured form like text, video, audio, XML files, records, or other image files used in later stages of data analysis.
- In the process of data analysis, “Data collection” is the initial step before starting to analyze the patterns or useful information in data.
- The data which is to be analyzed must be collected from different valid sources. Some of these data sources are Open Data Source, Social Media Data Source, Multimodal Data Source and Standard Datasets Source.

1.7.1 Open Data Source

- The idea behind open data is that some data should be freely available in a public domain that can be used by anyone as they wish, without restrictions from copyright, patents, or other mechanisms of control.
- Local and federal governments, Non-Government Organizations (NGOs) and academic communities all lead open data initiatives. For example, Open Government Data Platform India is a platform for supporting Open Data initiative of Government of India. Open Government Data Platform India is also packaged as a product and made available in open source for implementation by countries globally.
- The National Data Sharing and Accessibility Policy (NDSAP) that came into existence in 2012, with the approval of the union cabinet, is taken effort to associate the principles with Open data accessibility.
- The NDSAP is empowered by the Section 4(2) of the Right to Information (RTI) Act, and makes it the responsibility of every public authority to share their data and information at regular intervals.
- The NDSAP is applicable to all non-personal, non-sensitive data produced using public funds by the central, state, and local governments, and their departments. It covers data in all formats, digital, analog, machine, and human-readable formats.
- The NDSAP uses the principles of open data such as openness, transparency, quality, privacy, and machine readability.
- As a policy, NDSAP encourages and facilitates the sharing of government-owned data to achieve two primary goals: transparency and the accountability of the government, and innovation and the economic development of the country.

1.7.2 Social Media Data Source

- Social media channels has abundant source of data. Social media are interactive Web 2.0 Internet-based applications. Social media are reflection of public.
- Social media are interactive technologies that allows creation or sharing/exchange of information, ideas, career interests and other forms of expression via virtual communities and networks.
- Social media data is very useful for research or marketing purposes. Analysis of social data helps to take appropriate decision for marketing strategies, security policies, and prediction and consumer analysis.
- This is facilitated by the Application Programming Interface (API), which are provided by social media companies, to leverage the vast amounts of data available through various social media channels.

- For various data-related needs (e.g., retrieving a user's profile picture), one could send API requests to a particular social media service.
- This is typically a programmatic call that results in that service sending a response in a structured data format, such as an XML.
- Following table summarize of various social Media APIs with their feature:

Sr. No.	Social Media API	Social Media Description	API Features
1.	Twitter API	• Twitter allows users to find the latest world events and interact with other users. • Use various types of messaging content (called tweets). • Twitter can be accessed via its website interface, applications installed on mobile devices, or a short message service (SMS).	Twitter provides various API endpoints for completing various tasks. • The Search API can be use to retrieve historical tweets, • The Account Activity API to access account activities, • the Direct Message API to send direct messages, • Ads API to create advertisement campaigns. • Embed API to insert tweets on your web application.
2.	Facebook API	• Facebook is a social networking platform that allows users to communicate using messages, photos, comments, videos, news, and other interactive content.	Facebook provides various APIs and SDKs that allow developers to access its data • The Facebook Graph API is an HTTP-based API that provides the main way of accessing the platform's data. • With the API, you can query data, post images, access pages, create new stories, and carry out other tasks. • The Facebook Marketing API allows you to create applications for automatically marketing your products and services on the platform.

Contd...

3.	Instagram API	• Instagram is a Facebook-owned social networking platform that lets users share photos and videos.	Facebook offers many APIs to allow developers to create tools that enhance users' experience on the Instagram platform. • With the APIs, you can enable users to share their favorite stories and daily highlights from your application to Instagram. • The Instagram Graph API that allows developers to access the data of businesses operating Instagram accounts. With the Graph API, you can conveniently manage and publish media objects, discover other businesses, track mentions, analyze valuable metrics, moderate comments, and search hashtags.
4.	YouTube API	• YouTube is a Google-owned popular platform for sharing videos, music, and other visual images.	The YouTube API lets developers embed YouTube functionalities into their websites and applications. • The API enable users to play YouTube videos directly on user's application. • find YouTube content, manage playlists, upload videos, and complete other tasks. • The API also allows to analyze the performance of your videos, schedule live streaming broadcasts, and add the YouTube subscribe button.
5.	**Pinterest API**	• Pinterest is a social media platform that lets users share and discover online information, majorly in the form of images and videos (called pins). Users can post the media content to their own or others' pinboards.	The Pinterest API allows developers to access Pinterest's data and boost the capabilities of their applications. • With the API, you can access users' boards, followers, pins, and other data. • We can also add Pinterest buttons, widgets, or RSS (Really Simple Syndication) feeds to our application,

1.7.3 Multi-model Data

- Today explosion of unstructured data evolving as a big challenge for industry and researchers.
- IoT (Internet of Things) has allowed us to always remain connected with the help of different electronics gadgets. This communication network generates huge data having different formats and data types.
- When dealing with such contexts, we may need to collect and explore multimodal (different forms) and multimedia (different media) data such as images, music and other sounds, gestures, body posture, and the use of space.
- Once, the sources are identified, the next thing to consider is the kind of data that can be extracted from those sources.
- Based on the nature of the information collected from the sources, the data can be categorized into two types: structured data and unstructured data.
- One of the well-known applications of such multimedia data is analysis of brain imaging data sequences – where the sequence can be a series of images from different sensors, or a time series from the same subject.
- The typical dataset used in this kind of application is a multimodal face dataset, which contains output from different sensors such as EEG, MEG, and fMRI (medical imaging techniques) on the same subject within the same paradigm.
- In this field, Statistical Parametric Mapping (SPM) is a well-known statistical technique, created by Karl Friston that examines differences in brain activity recorded during functional neuro imaging experiments.

1.7.4 Standard Datasets

- A dataset or data set is simply a collection of data.
- In the case of tabular data (in the form of table), a data set corresponds to one or more database tables, where every column of a table represents a particular variable and each row corresponds to a given record of the data set in question.
- In the open data discipline, data set is the unit to measure the information released in a public open data repository.
- The simplest and most common format for datasets is a spreadsheet or CSV format - a single file organized as a table of rows and columns. Sometimes a dataset may be a zip file or folder containing multiple data tables with related data.
- Uploading datasets as Open Access helps both individuals and institutions meet. Availability of Authenticate and standardized data sets provides supports research reproducibility, fosters innovations and discoverability.

- Some of the datasets are given in the following table:

Sr. No.	Dataset	Details	Link
1.	Quandl	It is a massive repository for Economic and Financial data. Most of the datasets are free but some are available to purchase as well.	https://www.quandl.com/search
2.	Academic Torrents	It has data used to publish scientific research papers. The variety of datasets is massive with availability of free download.	https://academictorrents.com/browse.php?cat=6
3.	Data.gov	It consists of a variety of datasets from US Government agencies. Domains include Education, Climate, Food, Chronic disease and what not.	https://www.data.gov/
4.	UCI Machine Learning Repository	This site consists of datasets hosted by the University of California, Irvine. It has a collection of about 400+ datasets aimed towards the Machine Learning community.	http://archive.ics.uci.edu/ml/index.php
5.	Google Public Datasets	Google has hosted tons of datasets on Google Public Datasets which is basically their Cloud Platform. You can browse through their dataset collection using BigQuery. The first 1 Terabyte of queries you make are basically free.	https://cloud.google.com/bigquery/public-data/
6.	Datasets on Github	It hosts tons of awesome datasets. This github boasts a variety of datasets such as Climate Data, Time Series data, Plane crash data etc. Feel free to dig in.	https://github.com/awesomedata/awesome-public-datasets
7.	Socrata	Socrata hosts cleaned datasets across domains such as Government data, Radiation data, Workplace related data etc.	https://opendata.socrata.com/

Contd...

8.	Kaggle datasets	Kaggle is a house-hold name by now amongst data professionals. Kaggle hosts massive open source public data across various domains.	https://www.kaggle.com/datasets
9.	World Bank	These datasets are offered by the World Bank. They also provide several tools such as Education Indices, Open Data Catalog etc.	http://data.worldbank.org/
10.	Reserve Bank of India	RBI provides number of datasets related to Money Market Operations, Banking products etc	https://www.rbi.org.in/Scripts/Statistics.aspx
11.	FiveThirtyEight	They have a wide variety of datasets on their Github. The specialty of this site is that they have a detailed data dictionary explaining each of the dataset which is very beneficial.	https://github.com/fivethirtyeight/data
12.	AWS datasets	The big has entered with hundreds of datasets. It's no surprise if AWS hosts the largest datasets in the coming days.	https://registry.opendata.aws/
13.	YouTube Video dataset	This is a YouTube labeled video dataset. It consists of 8 million video IDs with related data.	https://research.google.com/youtube8m/
14.	MNIST dataset	This is a repository containing a hand-written digit dataset (About 60,000 samples).	http://yann.lecun.com/exdb/mnist/
15.	ImageNet	ImageNet is an image database consisting of images organized according to the WordNet hierarchy.	http://image-net.org/
16.	Yelp dataset	This dataset contains over 8 million + yelp reviews. This dataset is perfect for Text Classification use cases.	https://www.yelp.com/dataset
17.	Airbnb dataset	The data behind the Inside Airbnb site is sourced from publicly available information from the Airbnb site. The data has been analyzed,	http://insideairbnb.com/get-the-data.html

Contd...

18.	Walmart dataset	This dataset has details about Sales transactions from about 45 Walmart stores in the US.	https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data
19.	LendingClub	The site hosts massive datasets about loan related data. You have to create an account to access the data.	https://www.lendingclub.com
20.	Wikipedia Database	This dataset is perfect for Natural Language Processing Tasks.	https://en.wikipedia.org/wiki/Wikipedia:Database_download#English-language_Wikipedia
21.	Reddit	This site hosts all the comments millions of users made on Reddit from 2005 to 2017	https://www.reddit.com/r/datasets/comments/65o7py/updated_reddit_comment_dataset_as_torrents/
22.	UNICEF	This site hosts data about the lives of children with details such as Nutrition, Education etc.	https://data.unicef.org/

1.8 DATA FORMATS

- In data science the data appears in different sizes and shapes, it can be numerical data, text, audio, video or a few other types of data. The data format is said to be a kind of format which is used for coding the data.
- Some data formats in data science are explained below:

1. Integers:

- An integer is a datum of integral data type, a data type that represents some range of mathematical integers.
- Integral data types may be of different sizes and may or may not be allowed to contain negative values.

2. Floats:

- A floating point (known as a float) number has decimal points even if that decimal point value is 0.
- For example: 1.13, 2.0 and 1234.345.

3. Text Data:

- Text data type is known as Strings in Python, or Objects in Pandas. Strings can contain numbers and/or characters.
- For example, a string might be a word, a sentence, or several sentences. A string can also contain or consist of numbers.
- For example, '1234' could be stored as a string, as could '10.23'. However strings that contain numbers cannot be used for mathematical operations.

4. Dense Numerical Arrays:

- For storing large arrays of numbers, it is much more space- and performance- efficient to store them in something such as the native format that computers use for processing numbers.
- Most image files or sound files consist mostly of dense arrays of numbers, packed adjacent to each other in memory.
- Many scientific datasets fall into this category.

5. Compressed or Archived Data:

- Many data files, when stored in a particular format, take up a lot more space compared to the file in question logically needs.
- For example, if most lines in a large text file are exactly the same or a dense numerical array consists mostly of 0s. In these cases, we want to compress the large file into a smaller one, so that it can be stored and transferred more easily.
- A related problem is when we have a large collection of files that we want to condense into a single file for easier management, often called data archiving.
- There are a variety of ways that we can encode the raw data into these more manageable forms. There is a lot more to data compression than just reducing the size.
  (i) It generally reduces the size of the data, easing storage requirements.
  (ii) If it can't compress the data much (or at all), then at least it doesn't balloon it to take up much MORE space.
  (iii) Decompression process should be quick. So, it might take less time to load the compressed data compared to the raw data itself, even with the decompression step. This is because decompression in RAM can be fairly quick, but it takes a long time to pull extra data off the disk.
  (iv) Decompression can be done "one line at a time," rather than loading the entire file. This helps to deal with corrupt data and typically makes decompression go faster since you're operating on less data at a time.
  (v) We can recompress it quickly.

6. CSV Format:

- CSV stands for Comma Separated Values which is a text-based file format that store data in a tabular form similar to a spreadsheet or a database table and generally use a comma to separate values and has an extension of .csv.
- CSV files are the commonly used data format for data science. "CSV" stands for "Comma Separated Value," but it really should be "Character Separated Value" since characters other than commas do get used.
- Some components of CSV files are as follows:
  (i) Headers: Sometimes, the first line gives names for all the columns, and sometimes, it gets right into the data.

(ii) Quotes: In many files, the data elements are surrounded in quotes or another character. This is done largely so that commas (or whatever the delimiting character is) can be included in the data fields.

(iii) Nondata Rows: In many file formats, the data itself is CSV, but there are a certain number of nondata lines at the beginning of the file. Typically, these encode metadata about the file and need to be stripped out when the file is loaded into a table.

(iv) Comments: Many CSV files will contain human readable comments, as source code does. Typically, these are denoted by a single character, such as the # in Python.

Example:

Year	Make	Model	Description	Price
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

- The above table of data may be represented in CSV format as follows:

```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""","",4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!,air, moon roof, loaded",4799.00
```

7. **HTML Files:**

- HTML stands for Hyper Text Markup Language. An HTML file is a text file containing small markup tags.
- The markup tags tell the Web browser how to display the page An HTML file must have an .htm or .html file extension.
- An HTML file can be created using a simple text editor like Notepad.
- HTML comprises two major parts that give a document a well-structured look. They are head and body.
  - The head contains the title of the document, and the heading - which is the heading of that particular page in the document.
  - The body contains the entire content of the document.

- HTML tags are used to mark-up the file content. HTML tags are surrounded by the two characters < and > called angle brackets. HTML tags normally come in pairs like <html> and </html>, but there are single tags as well like <hr>
- The first tag in a pair is the start tag the second tag is the end tag. The text between the start and end tags is the element content. HTML tags are not case sensitive; <html> means the same as <HTML>.

8. JSON Files:

- JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is not only easy for humans to read and write, but also easy for machines to parse and generate.
- JSON is built on following two structures:
  - **A collection of name–value pairs:** In various languages, this is realized as an object, record, structure, dictionary, hash table, keyed list, or associative array.
  - **An ordered list of values:** In most languages, this is realized as an array, vector, list or sequence.
- When exchanging data between a browser and a server, the data can be sent only as text.
- JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects.
- This way we can work with the data as JavaScript objects, with no complicated parsing and translations.
- Let us look at examples of how one could send and receive data using JSON:

  Sending Data: If the data is stored in a JavaScript object, we can convert the object into JSON, and send it to a server. Below is an example:

```
<!DOCTYPE html>
   <html>
      <body>
      <p id="demo"></p>
   <script>
      var obj = {"name":"John", "age":25, "state": "New Jersey"};
      var obj_JSON = JSON.stringify(obj);
      window.location = "json_Demo.php?x=" + obj_JSON;
   </script>
      </body>
</html>
```

Receiving Data: If the received data is in JSON format, we can convert it into a JavaScript object. For example:

```
<!DOCTYPE html>
   <html>
   <body>
      <p id="demo"></p>
   <script>
      var obj_JSON = "{"name":"John", "age":25, "state":
      "New Jersey"}";
      var obj = JSON.parse(obj_JSON);
      document.getElementById("demo").innerHTML=obj.name;
</script>
</body>
</html>
```

9. XML Files:

- An XML (eXtensible Markup Language) file was designed to be both human- and machine readable, and can thus be used to store and transport data.
- In the real world, computer systems and databases contain data in incompatible formats.
- As the XML data is stored in plain text format, it provides a software- and hardware-independent way of storing data. This makes it much easier to create data that can be shared by different applications.
- Here is an example of a page of XML:

```
<?xml version="1.0" encoding="UTF-8"?>
   <bookstore>
      <book category="information science" cover="hardcover">
      <title lang="en">Social Information Seeking</title>
      <author>Chirag Shah</author>
      <year>2017</year>
      <price>62.58</price>
      </book>
      <book category="data science" cover="paperback">
      <title lang="en">Hands-On Introduction to Data Science</title>
      <author>Chirag Shah</author>
      <year>2019</year>
```

```
        <price>50.00</price>
        </book>
    </bookstore>
```

- For instance, one could develop a website that runs in a Web browser and uses the above data in XML, whereas someone else could write a different code and use this same data in a mobile app.
- In other words, the data remains the same, but the presentation is different. This is one of the core advantages of XML and one of the reasons XML is becoming quite important as we deal with multiple devices, platforms, and services relying on the same data.

10. Tar Files:

- The origin of TAR extension is “Tape Archive”.
- It's a UNIX based file archiving format widely used to archive multiple files and sharing them over the internet.
- TAR files can contain different files like videos and images, even software installation files which can be distributed online.
- The tar is a computer software utility for collecting many files into one archive file, often referred to as a tarball, for distribution or backup purposes

11. GZip Files:

- Files with GZ extension are compressed archives that are created by the standard GNU zip (gzip) compression algorithm.
- This archive format was initially created by two software developers to replace file compression program of UNIX.
- It’s still one of the most common archive file formats on UNIX and Linux systems.
- GZ files have handy features like storing original file name and timestamp, enabling users to recover original file information even after the file was transferred.
- Also, gz format is often used to compress elements of web pages for faster page loading.

12. Zip Files:

- The Zip archive format makes it easier to send and back up large files or groups of files.
- A Zip file is a single file containing one or more compressed files, offering an ideal way to make large files smaller and keep related files together.
- The most popular compression format for Windows, Zip is commonly used for emailing and sharing files over the Internet. ZIP is one of the most widely used compressed file formats.
- It is universally used to aggregate, compress, and encrypt files into a single Inter operable container.

13. Image Files:

- Image file formats are standardized means of organizing and storing digital images. Image files are Rasterized, Vectorized, and/or Compressed.
- An image file format may store data in an uncompressed format, a compressed format (which may be lossless or lossy), or a vector format.
- Image files are composed of digital data in one of these formats so that the data can be rasterized for use on a computer display or printer.
- Rasterization converts the image data into a grid of pixels. Each pixel has a number of bits to designate its color (and in some formats, its transparency).
- Rasterizing an image file for a specific device takes into account the number of bits per pixel (the color depth) that the device is designed to handle.
- Rasterizing image file format includes .jpeg, .gif, .psd, .png and so on.
- Vectorized images are digital artwork in which points, lines and curves are calculated by the computer. Vector images are typically used for logos, icons, typesetting and digital illustrations.
- Vectorized image file format includes .svg, .eps, .ai, .pdf, .cdr, .wmf and so on.

PRACTICE QUESTIONS

Q.I Multiple Choice Questions:

1. Which is the task of scrutinizing and processing raw data to reach meaningful information?
   (a) Data Science (b) Data Preprocessing
   (c) Statistics Analysis (d) Data Analysis
2. The type of data analysis that emphasizes on recommending actions based on the forecast is called as,
   (a) prescriptive analysis (b) diagnostic analysis
   (c) predictive analysis (d) None of the mentioned
3. Which is an open-source tool and falls under object-oriented scripting language?
   (a) SAS (Statistical Analysis System) (b) Python
   (c) Tableau (d) RapidMiner
4. Data science related to huge data and involves concepts like,
   (a) data collection and preparation (b) data managing and storing
   (c) data cleaning and visualizing (d) All of the mentioned
5. Which is a data that is which is not organized in a pre-defined manner?
   (a) Structured data (b) Unstructured data
   (c) Semi-structured data (d) All of the mentioned

6. Which data science tool is written in the Java language?
   (a) Apache Spark (b) Tableau
   (c) RapidMiner (d) Knime
7. Which file is a text file in which the values in the columns are separated by a comma?
   (a) csv (b) json
   (c) html (d) xlsx
8. The 3V's in data science is,
   (a) Volume (amount of data) (b) Velocity (speed of data)
   (c) Variety (different types of data) (d) All of the mentioned
9. Which file format is use for compress file size?
   (a) XML file (b) HTML Files
   (c) Zip File (d) Text File
10. Well defined format data is knows as,
   (a) Semi-structured data (b) Structured data
   (c) Unstructured data (d) Quasi-structured data
11. XML is an example of,
   (a) Semi-structured data (b) Structured data
   (c) Unstructured data (d) Quasi-structured data
12. Which of the following is not a component of 3V's,
   (a) Volume (b) Velocity
   (c) Vacuum (d) Variety
13. Data science is used in,
   (a) banking and retail (b) ecommerce and finanace
   (c) healthcare and education (d) All of the mentioned
14. Which data refers to data that are freely available without restrictions from copyright, patents or other mechanisms of control?
   (a) open (b) closed
   (c) semi-open (d) All of the mentioned
15. To work with social media data, it is required to get access to data source that is generated from various social media sites. Such sites include social networking sites such as,
   (a) Facebook (b) Twitter
   (c) LinkedIn (d) All of the mentioned

Answers

1. (a)	2. (c)	3. (b)	4. (d)	5. (b)	6. (c)	7. (a)	8. (d)	9. (c)	10. (b)
11. (a)	12. (c)	13. (d)	14. (a)	15. (d)					

Q.II Fill in the Blanks:

1. _____ involves the use scientific methods, processes, analysis, algorithms and systems to extract information from data and create insights.
2. Activities like online purchases, stock market investment, healthcare requires an in-depth analysis of existing relevant data which makes data science a promising field of study in today's fast-growing _____ -driven world.
3. _____ is a data science tool widely used due to its simplicity and easy interpretation of complex data analytical tasks.
4. The _____ of the data analytics provides a framework for the best performances of each phase from the creation of the project until its completion.
5. A data _____ in data science is the location where data that is being used come from.
6. XML and NoSQL data is an example of _____ data.
7. _____ programming is an open-source tool that is often used for data science for data handling and manipulation.
8. A _____ is a collection of data.
9. Reading data from csv (comma separated values) is a fundamental necessity in data science with _____ file extension.
10. Data science is a combination of multiple disciplines that uses statistics, data analysis, and machine learning to analyze data and to extract knowledge and _____ from it.
11. _____ is a lightweight, text-based open standard designed for exchanging the data over the web.
12. Data _____ is the process of acquiring, collecting, extracting, and storing the huge amount of data in the form of text, video, audio, XML files, images and so on.
13. The _____ in data science indicates that nowadays data collections are huge/massive (Volume), data grow at an extremely fast rate (Velocity) and data in the form of different types (Variety).
14. Social Media data is an example of _____ data.

Answers

1. Data science	2. data	3. Excel	4. lifecycle
5. source	6. semi-structured	7. R	8. dataset
9. .csv	10. insights	11. JSON	12. collection
13. 3V's	14. unstructured		

Q.III State True or False:

1. Data science is a rapidly growing field of study that uses scientific methods to extract meaningful insights from raw data.
2. RapidMinner is written in Python language.
3. Data science is a collection of techniques and tools used to extract value/information/insights from raw data.
4. A relation data is unstructured type of data.
5. Tableau is data visualization software.
6. JSON file stores data as text in human-readable format with with .json file extension.
7. Structured data is a data that is which is well organized in a pre-defined manner/format.
8. The first phase of the data analytics lifecycle is visualization.
9. ZIP and GZIP are two very popular methods of compressing files, in order to save space, or to reduce the amount of time needed to transmit the files across the network, or internet.
10. Knime stands for Konstanz Information Miner is an analytics platform is an open-source data analytics and reporting platform.
11. An image file format is a standard way to organize and store image data.
12. The name tar file is derived from "tape archive", used for or collecting many files into one archive file.
13. The 3V's containd Volume (vast amounts of generated data), Variety (different types of data) and Velocity (the speed at which the data is generated).
14. The phase of the data analytics lifecycle in which the team works on developing datasets for training and testing is deployment.
15. Apache Spark is open-source software can run on any platform such as UNIX, Windows and Mac operating systems.

Answers

1. (T)	2. (F)	3. (T)	4. (F)	5. (T)	6. (T)	7. (T)	8. (F)	9. (T)	10. (T)
11. (T)	12. (T)	13. (T)	14. (F)	15. (T)					

Q.IV Answer the following Questions:

(A) Short Answer Questions:

1. Define data science.
2. What is role of data science?
3. Define unstructured Data.
4. Define data source?

5. What is data set?
6. What is CSV format?
7. What are uses of Zip files?
8. What is open data?
9. What is meant by semi-structured data?
10. What is rasterized and vectorized image files?
11. What is SAS?
12. List application of data science.
13. What is social media data?
14. List tools for data scientist.
15. What is 3V's?
16. What is data science life cycle?
17. Give the use of tar files.
18. What is compressed data?
19. What is data source?

(B) Long Answer Questions:

1. What is data science? Why the learning of data science is important?? Explain in detail.
2. Explain 3v's of data science with diagram.
3. What are components of data scientist's toolbox? Explain two of them.
4. What are applications of data science?
5. With the help of diagram describe lifecycle of data science.
6. What are different types of data? Explain in detail with appropriate examples.
7. Distinguish between structured and unstructured data?
8. What are different sources of data in data science? Explain in detail.
9. Explain different data formats in brief.
10. Explain unstructured data with example. What are problems with unstructured data?

❖❖❖

CHAPTER 2

Statistical Data Analysis

Objectives...

- To learn Statistical Data Analysis
- To study Descriptive Statistics, Inferential statistics and Concept of Outlier

2.0 INTRODUCTION

- Statistics is a way to collect and analyze the numerical data in a large amount and finding meaningful insights from it.
- Statistics is the discipline that concerns the collection, organization, analysis, interpretation and presentation of data. Statistical data analysis is a procedure of performing various statistical operations.
- Two main statistical methods are used in data analysis are **descriptive statistics**, which summarize data from a sample using indexes such as the mean or standard deviation and **inferential statistics**, which draw conclusions from data that are subject to random variation (e.g., observational errors, sampling variation).
- A data analyst or data scientist needs to perform a lot of statistical analysis to analyze and interpret data to gain meaningful results.
- A few of the fundamental aspects of statistical analysis are:

1. Classification of Data Samples:

- This is a statistical method that is used by the same name in the data science and mining fields.
- Classification is used to categorize available data into accurate, observable analyses. Such an organization is key for companies who plan to use these insights to make predictions and form business plans.

2. Probability Distribution and Estimation:

- These statistical methods are helps to learning the basics of machine learning and algorithms like logistic regressions.
- Cross-validation and LOOCV (Leave One Out Cross Validation) techniques are also inherently statistical tools that have been brought into the Machine Learning and Data Analytics world for inference-based research, A/B and hypothesis testing.

3. Finding Data Patterns:

- Companies/organizations often find themselves having to deal with enormous dumps of data from panoply of sources, each more complicated than the last.
- Statistics can help to find anomalies and trends in this data, further allowing researchers to discard irrelevant data at a very early stage instead of keen observation through data and wasting time, effort and resources.

4. Statistical Modeling:

- Data is made up of series of factors and variables. To model these or display them in a coherent manner, statistical modeling using graphs and networks is solution.
- This also helps to identify and account for the influence of hierarchies in global structures and escalate local models to a global scene.

5. Data Visualization:

- Visualization in data is the representation and interpretation of found structures, models and insights in interactive, understandable, and effective formats.
- Beyond this, data analytics representations also use the same display formats as statistics - graphs, pie charts, histograms and so on.
- Not only does this make data more readable and interesting, but it also makes it much easier to find trends or flaws and offset or enhance them as required.

6. Facilitates understanding of Distributions in Model-based Data Analytics:

- Statistics can help to identify clusters in data or even additional structures that are dependent on space, time, and other variable factors.
- Reporting on values and networks without statistical distribution methods can lead to estimates that don't account for variability, which can make or break your results.
- Small wonder, then, that the method of distribution is a key contributor to statistics and to data analytics and visualization as a whole.

7. Mathematical Analysis:

- The basics of mathematical analysis differentiability and continuity– also form the base of many major Machine Learning, Artificial Intelligence and data analytics algorithms.
- Neural networks in deep learning are effectively guided by the shift in perspective that is differential programming.
- Predictive power is key in how effective a data analytics algorithm or model is. The rule of thumb is that the lesser the assumptions made, the higher the model's predictive power.

- Statistics help to bring down the rate of assumptions, thereby making models a lot more accurate and usable.
- Statistical data analysis deals with data that is essential of two types, namely, continuous data and discrete data.
- The fundamental difference between continuous data and discrete data is that continuous data cannot be counted, whereas discrete data can be counted.
- One example of continuous data could be the time taken by athletes to complete a race. The time in a race can be measured but cannot be counted.
- An example of discrete data is the number of students in a class or the number of students in a University that can be counted.
- While continuous data is distributed under continuous distribution function (also called as Probability Density Function (PDF)), discrete data is distributed under discreet distribution function (also called as Probability Mass Function (PMF)).

2.1 ROLE OF STATISTICS IN DATA SCIENCE

- Statistics has evolved along with technology and the growth of data. Context of Statistics and its applications are tremendously changed by time. Strategies for taking business decisions using statistical results are now more expanded.
- With concept of Big data, a major community is now depends on data driven decisions an predictions. The statistical community has been committed to the almost exclusive use of data models and algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics.
- It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets.
- Framing questions statistically allows researchers to leverage data resources to extract knowledge and obtain better answers.
- The central dogma of statistical inference, that there is a component of randomness in data, enables researchers to formulate questions in terms of underlying processes and to quantify uncertainty in their answers.
- A statistical framework allows researchers to distinguish between causation and correlation and thus to identify interventions that will cause changes in outcomes.
- It also allows researchers to establish methods for prediction and estimation, to quantify their degree of certainty, and to do all of this using algorithms that exhibit predictable and reproducible behavior.
- In this way, statistical methods aim to focus attention on findings that can be reproduced by other researchers with different data resources. Simply put, statistical methods allow researchers to accumulate knowledge.

- Data science involves techniques that emphasize heavily on using statistical analysis by providing appropriate statistical tools that can imbibe in user precise statistical thinking patterns.
- Some roles in which Statistics helps in Data Science are explained below:
  1. **Prediction and Classification:** Statistics help in prediction and classification of data whether it would be right/correct for the clients viewing by their previous usage of data.
  2. **Helps to Create Probability Distribution and Estimation:** Probability distribution and estimation are crucial in understanding the basics of machine learning and algorithms like logistic regressions.
  3. **Cross-validation and LOOCV Techniques:** They are also inherently statistical tools that have been brought into the Machine Learning and Data Analytics world for inference-based research, A/B and hypothesis testing.
  4. **Pattern Detection and Grouping:** Statistics help in picking out the optimal data and weeding out the unnecessary dump of data for companies who like their work organized. It also helps spot out anomalies which further helps in processing the right data.
  5. **Powerful Insights:** Dashboards, charts, reports and other data visualizations types in the form of interactive and effective representations give much more powerful insights than plain data and it also makes the data more readable and interesting.
  6. **Segmentation and Optimization:** It also segments the data according to different kinds of demographic or psychographic factors that affect its processing. It also optimizes data in accordance with minimizing risk and maximizing outputs.

2.2 DESCRIPTIVE STATISTICS

- The study of numerical and graphical ways to describe and display the data is called descriptive statistics.
- Descriptive statistics use data to carry out descriptions of the population in the form of numerical calculations, visualization graphs, or tables.
- Descriptive statistics are used to describe the basic features of the data in a study. They provide simple summaries about the sample and the measures.
- Together with simple graphics analysis, they form the basis of virtually every quantitative analysis of data.
- Descriptive statistics are used to present quantitative descriptions in a manageable form. In a research study we may have lots of measures.

- Data using descriptive statistics can be expressed in a quantifiable form that can be easily managed and understood.
- Descriptive statistics help us to simplify large amounts of data in a sensible way. Each descriptive statistic reduces lots of data into a simpler summary.
- Descriptive statistics is summarizing the data at hand through certain numbers like mean, median etc. so as to make the understanding of the data easier.
- For example, consider a simple number used to summarize how well a batter is performing in baseball, the batting average.
- This single number is simply the number of hits divided by the number of times at bat (reported to three significant digits).
- A batter who is hitting .333 is getting a hit one time in every three at bats. One batting .250 is hitting one time in four.
- The single number describes a large number of discrete events. Consider the Grade Point Average (GPA) of any student from the university.
- This single number describes the general performance of a student across a potentially wide range of course experiences.
- There are mainly four types of descriptive statistics namely, Measures of frequency, Measures of central tendency, Measures of dispersion and Measures of position as shown in Fig. 2.1.

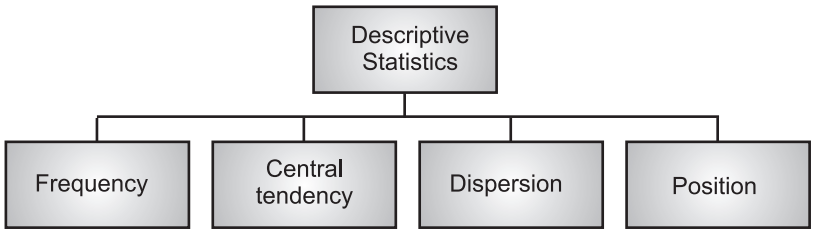


Fig. 2.1: Types of Descriptive Statistics

2.2.1 Measures of Frequency

- The measures of frequency are widely used in statistical analysis (analyze and interpret data to gain meaningful insights) to analyze how often a particular data value or a feature occurs.
- The frequency distribution can be tabulated as a frequency chart or it can be graphically represented by drawing a bar chart or a histogram.
- The measures of frequency simply count the number of times each feature occurs, such as the number of students passed and the number of students who failed in an examination.

- For example, if we assume the total number of students in a class is 80, out of which 60 passed and 20 failed, the frequency/count and percentage can be represented in the form of a frequency chart as given in following table:

	Frequency	Percentage
Pass	60	75%
Fail	20	25%
Total	80	100%

- Following program shows the Python code for calculating the frequency and percentage for two features – Gender and Result - of a given dataset.
- The groupby() function is used in the Python code to split the data into groups based on the values of Gender and Result.
- Next, the Total column is created to display the total number of data for a particular gender value. Also, the percentage of students passed and failed for each gender is displayed in another two columns namely, Pass_Percentage and Fail_Percentage.
- Lastly, a bar chart is displayed for a visual representation of each set of four groups of features formed from the combination of given two features:

```
import pandas as pd
file = pd.read_csv("frequency.csv")
print("\n DATASET VALUES")
print("----------------");
print(file)
#Displaying Frequency Distribution
dframe = pd.DataFrame(file)
print("\n FREQUENCY DISTRIBUTION")
print("------------------------ \n")
data = dframe.groupby(['Gender','Result']).size().unstack().reset_index()
data['Total'] = (data['P'] + data['F'])
data['Pass_Percent'] = data['P'] / data['Total']
data['Fail_Percent'] = data['F'] / data['Total']
print(data[:5])
```

- The output of the above program initially displays the entire dataset consisting of 11 records and then displays the frequency distribution in the form of a table.

```
 DATASET VALUES
----------------
   Gender Result
0       M      P
1       F      P
2       F      F
3       F      F
4       M      P
5       M      P
6       M      F
7       M      F
8       M      P
9       M      F
10      M      P
11      F      P
12      F      F

 FREQUENCY DISTRIBUTION
-----------------------

Result Gender  F  P  Total  Pass_Percent  Fail_Percent
0           F  3  2      5         0.400         0.600
1           M  3  5      8         0.625         0.375
```

- A frequency distribution is a common statistical measure used to find the number of times a particular feature occurs for a given population.

2.2.2 Measures of Central Tendency

- One of the simplest and yet important measures of statistical analysis is to find one such value that describes the characteristic of the entire huge set of data.
- This single value is referred to as a central tendency that provides a number to represent the whole set of scores of a feature.
- A measure of central tendency is a summary statistic that represents the center point or typical value of a dataset.
- These measures indicate where most values in a distribution fall and are also referred to as the central location of a distribution.
- There are mainly three ways to measure central tendency namely, mean, median, and mode as described below:

1. Mean:

- Mean is the most popular and widely used measure of representing the entire data by one value.
- Mean is the ratio of the sum of all the observations in the data to the total number of observations.
- Fig. 2.2 shows types of mean.

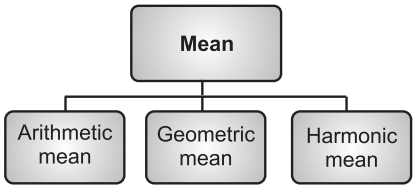


Fig. 2.2: Types of Mean

- For a given set of values, the mean can be found by using any of the following:

(i) Arithmetic Mean:

- Arithmetic mean is by far the most commonly used method for finding the mean. The arithmetic mean is obtained by adding all the values and then dividing the sum by the total number of digits.
- Arithmetic mean is the most common and effective numeric measure of the "center" of a set of data.
- Let $x_1, x_2, : : : , x_N$ be a set of N values or observations, such as for some numeric attribute X, like salary. The mean of this set of values is,

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{x_1 + x_2 + \ldots + x_N}{N} \quad \ldots (2.1)$$

Example: Suppose we have the following values for *salary* (in thousands of rupees), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

Sol.:

Using equation (2.1), we have

$$\bar{x} = \frac{30 + 36 + 47 + 50 + 52 + 52 + 56 + 60 + 63 + 70 + 70 + 110}{12}$$

$$= \frac{696}{12} = 58$$

Thus, the mean salary is Rs. 58,000.

- The arithmetic mean is best used in situations when the set of data values has no outliers (extreme values that do not match with the majority of the values in the set), as well as the individual data values, are not dependent on each other.
- There is a significant drawback to using the mean as a central statistic as it is susceptible to the influence of outliers.
- Also, mean is only meaningful if the data is normally distributed, or at least close to looking like a normal distribution.
- The arithmetic mean is useful in machine learning when summarizing a variable, e.g. reporting the most likely value. The arithmetic mean can be calculated using the mean() NumPy function.

(ii) Harmonic Mean:

- Harmonic mean is used in situations when the set of data values has one or more extreme outliers.
- The harmonic mean is obtained by dividing the total number of digits with the sum of the reciprocal of all numbers.

- For example, if we consider the set of values X = {1, 2, 4}, the harmonic mean X_{HM} is calculated as follow:

$$X_{HM} = \frac{3}{\frac{1}{1}+\frac{1}{2}+\frac{1}{4}} = 1.7143$$

(iii) Geometric Mean:

- Geometric mean is used in situations when the set of data values are inter-related.
- The geometric mean is obtained by finding the nth root of the product of all numbers of a given dataset.
- For example, if we consider the set of values X = {1, 4, 16}, the geometric mean X_{GM} is calculated as follows:

$$X_{GM} = \sqrt[3]{1.4.16} = 4$$

2. Median:

- It refers to the middle value in a sorted distribution of numbers. The entire set of observations is split into two halves and the mid-value is extracted to calculate the median.
- The median of a distribution with a discrete random variable depends on whether the number of terms in the distribution is even or odd.
- If the number of terms is odd, then the median is the value of the term in the middle. This is the value such that the number of terms having values greater than or equal to it is the same as the number of terms having values less than or equal to it.
- If the number of terms is even, then the median is the average of the two terms in the middle, such that the number of terms having values greater than or equal to it is the same as the number of terms having values less than or equal to it.
- For instance, if the numbers considered are 3, 18, 2, 12, and 5, the sorted list will be 2, 3, 5, 12, and 18. In this sorted list of numbers, the mid-value or the median is 5.
- Let us now consider another case where the numbers are, 2, 15, 8, 10, 4, 12. The middle values in the sorted list 2, 4, 8, 10, 12, and 15 are 8 and 10, and the median is then calculated as ((8 + 10) / 2) = 9.
- The median of a distribution with a continuous random variable is the value m such that the probability is at least 1/2 (50%) that a randomly chosen point on the function will be less than or equal to m, and the probability is at least 1/2 that a randomly chosen point on the function will be greater than or equal to m.
- Unlike mean, median is not sensitive for outliers or extremes values.
- The other good case for median is the interpretation of data. Median splits data perfectly into two halves, so if median income in Howard County is $100,000 per year, we could simply say that half the population has higher and the remaining half has lowers than $100k income in the county.

- However, there is an obvious disadvantage. Median uses the position of data points rather than their values.
- That way some valuable information is lost and we have to rely on other kinds of measures such as measures of dispersion to get more information about the data.
- The median is expensive to compute when we have a large number of observations. For numeric attributes, however, we can easily approximate the value.
- Assume that data are grouped in intervals according to their xi data values and that the frequency (i.e., number of data values) of each interval is known.
- For example, employees may be grouped according to their annual salary in intervals such as $10–20,000, $20–30,000, and so on.
- Let the interval that contains the median frequency be the median interval. We can approximate the median of the entire data set (e.g., the median salary) by interpolation using the following formula:

$$\text{Median} = L_1 + \left(\frac{N/2 - (\sum \text{freq})_l}{\text{Freq}_{\text{median}}} \right) \text{width}$$

where, L_1 is the lower boundary of the median interval, N is the number of values in the entire data set, $(\sum \text{freq})_l$ is the sum of the frequencies of all of the intervals that are lower than the median interval, $\text{freq}_{\text{median}}$ is the frequency of the median interval, and width is the width of the median interval.

3. Mode:

- It refers to the modal value which is the value in a series of numbers that has the highest frequency.
- Mode is the number which has the maximum/highest frequency in the entire data set.
- Understanding mode of a distribution is important because frequently occurring values are more likely to be picked up in a random sample.
- **For example:** Consider 21, 34, 56, 34, 25, 34, 11, 89. Mode is 34, because it has more than the rest of the values, i.e. thrice.
- The mode is another measure of central tendency. The mode for a set of data is the value that occurs most frequently in the set.
- Therefore, it can be determined for qualitative and quantitative attributes. It is possible for the greatest frequency to correspond to several different values, which results in more than one mode.
- Data sets with one, two, or three modes are respectively called unimodal, bimodal and trimodal.

- In general, a data set with two or more modes is multimodal. At the other extreme, if each data value occurs only once, then there is no mode.
- Following program shows the Python code for measuring the central tendencies for a given dataset that consists of two set of score values for English and Science papers.
- The various measures of central tendencies such as arithmetic mean, harmonic mean, geometric mean, median and mode are found for both the set of papers using the in-built functions tmean(), hmean(), gmean(), median(),and mode().

```
import pandas as pd
import scipy.stats as s
score={'English': [73,58,68,85,42,67,96,74,53,43,80,68],
                        'Science' : [90,98,78,66,58,83,87,80,55,90,82,68]}
#print the DataFrame
dframe=pd.DataFrame(score)
print(dframe)
# Arithmetic Mean of the Score Columns in DataFrame
print("\n\n Arithmetic Mean Values in the Distribution")
print("Score 1 ", s.tmean(dframe["English"]).round(2))
print("Score 2 ", s.tmean(dframe["Science"]).round(2))
# Harmonic Mean of the Score Columns in DataFrame
print("\n Harmonic Values in the Distribution")
print("Score 1 ", s.hmean(dframe["English"]).round(2))
print("Score 2 ", s.hmean(dframe["Science"]).round(2))
# Geometric Mean of the Score Columns in DataFrame
print("\n Geometric Values in the Distribution")
print("Score 1 ", s.gmean(dframe["English"]).round(2))
print("Score 2 ", s.gmean(dframe["Science"]).round(2))
# Median of the Score Columns in DataFrame
print("\n Median Values in the Distribution")
print("Score 1 ", dframe["English"].median())
print("Score 2 ", dframe["Science"].median())
```

```
# Mode of the Score Columns in DataFrame
print("\n Mode Values in the Distribution")
print("Score 1 ", dframe["English"].mode())
print("Score 2 ", dframe["Science"].mode())
```

- The output of above program initially displays the entire dataset consisting of 12 records and 2 columns (indicating English and Science marks) then all the measures of central tendencies, namely, arithmetic mean, harmonic mean, geometric mean, mode, and median are displayed for both the column values:

```
    English  Science
0        73       90
1        58       98
2        68       78
3        85       66
4        42       58
5        67       83
6        96       87
7        74       80
8        53       55
9        43       90
10       80       82
11       68       68


 Arithmetic Mean Values in the Distribution
Score 1  67.25
Score 2  77.92

 Harmonic Values in the Distribution
Score 1  63.36
Score 2  75.58

 Geometric Values in the Distribution
Score 1  65.35
Score 2  76.78

 Median Values in the Distribution
Score 1  68.0
Score 2  81.0

 Mode Values in the Distribution
Score 1  0    68
dtype: int64
Score 2  0    90
dtype: int64
```

- It is important to note that all these measures of central tendencies should be used based on the type of data (nominal, ordinal, interval or ratio) as shown in Fig. 2.3.

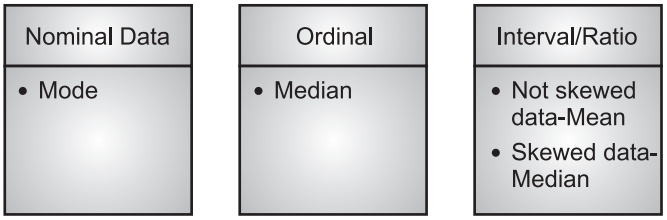


Fig. 2.3: Central Tendency Measures for varying Type of Data

2.2.3 Measures of Dispersion

- The measures of central tendency may not be adequate to describe data unless we know the manner in which the individual items scatter around it.
- In other words, a further description of a series on the scatter or variability known as dispersion is necessary, if we are to gauge how representative the average is.
- Let us take the following three sets.

Students	Group X	Group Y	Group Z
1	50	45	05
2	50	50	45
3	50	55	100
Mean	50	50	50

- Thus, the three groups have same mean i.e. 50. In fact the median of group X and Y are also equal. Now if one would say that the students from the three groups are of equal capabilities, it is totally a wrong conclusion.
- Close examination reveals that in group X students have equal marks as the mean, students from group Y are very close to the mean but in the third group Z, the marks are widely scattered. It is thus, clear that the measures of the central tendency is alone not sufficient to describe the data.
- The measure of dispersion helps us to know the degree of variability in the data and provide a better understanding of the data.

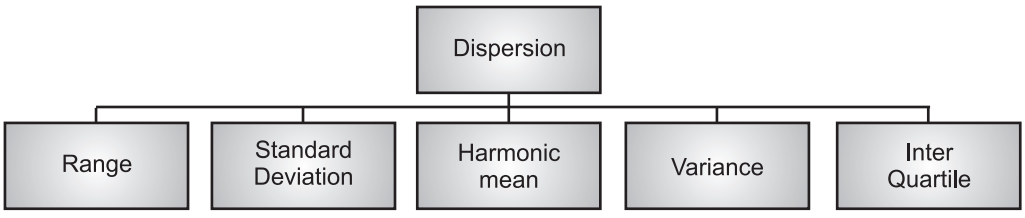


Fig. 2.4: Measures of Dispersion

- So, measures of dispersion or variability indicate the degree to which scores differ around the average. It is one single number that indicates how the data values are dispersed or spread out from each other.
- There are following ways in which the dispersion of data values can be measured:

1. Range:

- The value of the range is the simplest measure of dispersion and is found by calculating the difference between the largest data value (L) and the smallest data value (S) in a given data distribution. Thus, Range (R) = L – S.
- For instance, if the given data values are 8, 12, 3, 24, 16, 9, and 20, the value of range will be 21 (that is, the difference between 3 and 24). The range is rarely used in statistical and scientific work as it is fairly insensitive.

  Coefficient of Range: It is a relative measure of the range. It is used in the comparative study of the dispersion,

  $$\text{Co-efficient of Range} = \frac{L - S}{L + S}$$

- In case of continuous series Range is just the difference between the upper limit of the highest class and the lower limit of the lowest class.
- Range is very simple to understand and easy to calculate. However, it is not based on all the observations of the distribution and is unduly affected by the extreme values.
- Any change in the data not related to minimum and maximum values will not affect range. It cannot be calculated for open-ended frequency distribution.

Example: The amount spent (in Rs.) by the group of 10 students in the school canteen is as: 110, 117, 129, 197, 190, 100, 100, 178, 255, 790. Find the range and the co-efficient of the range.

Solution: R = L – S = 790 – 100 = Rs. 690

$$\text{Co-efficient of Range} = \frac{L - S}{L + S} = \frac{790 - 100}{790 + 100} = \frac{690}{890} = 0.78$$

---

2. Standard Deviation:

- The standard deviation is the measure of how far the data deviates from the mean value.
- Standard deviation is the most common measure of dispersion and is found by finding the square root of the sum of squared deviation from the mean divided by the number of observations in a given dataset.
- In short, the square root of variance is called the standard deviation.

- The standard formula for calculating the standard deviation (σ) is given by,

$$\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{N}}$$

- Here, x is each value provided in the data distribution, x is the arithmetic mean of the data values, and N is the total number of values in the data distribution.
- For example, if the given values in the data distribution are 19, 21, 24, 22, 18, 25, 23, 20, 21 and 23, the standard deviation can be calculated as:

$$\sigma = \sqrt{\frac{(19 - 21.6)^2 + (21 - 21.6)^2 + (24 - 21.6)^2 + (22 - 21.6)^2 + (18 - 21.6)^2 + (25 - 21.6)^2 + (23 - 21.6)^2 + (20 - 21.6)^2 + (21 - 21.60)^2 + (23 - 21.6)^2}{10}}$$

$\therefore \quad \sigma = 2.1071$

4. Variance:

- The variance is a measure of variability. It is the average squared deviation from the mean. Variance measures how far are data points spread out from the mean.
- Variance is a measure of dispersion that is related to the standard deviation. It is calculated by finding the square of the standard deviation of given data distribution.
- Hence, in the previous example, as the standard deviation for the data values 19, 21, 24, 22, 18, 25, 23, 20, 21, and 23 is 2.1071, the value of variance will be 4.4399.

5. Inter Quartile:

- Let x_1, x_2, ..., x_N be a set of observations for some numeric attribute, X. The range of the set is the difference between the largest (max() and min()) values. Suppose that the data for attribute X are sorted in increasing numeric order.
- Suppose we want to choose certain data points so as to split the data distribution into equal-size consecutive sets.
- These data points are called quantiles. Quantiles are points taken at regular intervals of a data distribution, dividing it into essentially equalsize consecutive sets.
- The k^{th} q-quantile for a given data distribution is the value x such that at most k/q of the data values are less than x and at most (q-k)/q of the data values are more than x, where k is an integer such that $0 < k < q$. There are q-1 q-quantiles.
- The 2-quantile is the data point dividing the lower and upper halves of the data distribution. It corresponds to the median.
- The 4-quantiles are the three data points that split the data distribution into four equal parts; each part represents one-fourth of the data distribution. They are more commonly referred to as quartiles.

- The 100-quantiles are more commonly referred to as percentiles; they divide the data distribution into 100 equal-sized consecutive sets. The median, quartiles and percentiles are the most widely used forms of quantiles.

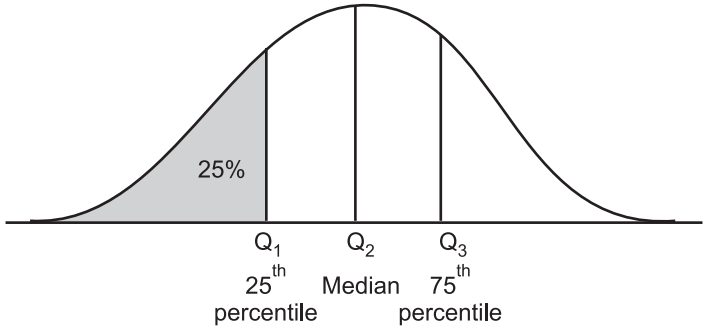


Fig. 2.5

- Fig. 2.5 shows a plot of the data distribution for some attribute X. The quantiles plotted are quartiles. The three quartiles divide the distribution into four equal-size consecutive subsets. The second quartile corresponds to the median.
- The quartiles give an indication of a distribution's center, spread, and shape. The first quartile, denoted by Q_1, is the 25^{th} percentile.
- It cuts off the lowest 25% of the data. The third quartile, denoted by Q_3, is the 75^{th} percentile - it cuts off the lowest 75% (or highest 25%) of the data. The second quartile is the 50^{th} percentile
- As the median, it gives the center of the data distribution. The distance between the first and third quartiles is a simple measure of spread that gives the range covered by the middle half of the data. This distance is called the InterQuartile Range (IQR) and is defined as,

$$IQR = Q_3 - Q_1$$

- This can be demonstrated by considering an example for the sorted data values as shown below,

2, 3, | 4, | 7, 10, | 15, 22, 26, | 27, | 30, 32

Q_1 **Q_2** **Q_3**

- Now, the IQR can be calculated for the above data distribution as follows:

$$IQR = Q_3 - Q_1 = 27 - 4 = 23$$

- Similarly, the semi-interquartile range can be calculated as given below:

$$SIR = \frac{Q_3 - Q_1}{2} = \frac{27 - 4}{2} = 11.5$$

- Following program shows the Python code for measuring the position and dispersion values for a given dataset that consists of a set of score values for Physics.
- The various measures of position and dispersion values such as percentile rank, range, standard deviation, variance, and interquartile range are found for the score column using the in-built functions rank(), max(), min(), std(), var() and iqr().

```
# Program to Measures of Position and Dispersion
import pandas as pd
from scipy.stats import iqr
import matplotlib.pyplot as plt
#Create a Dataframe
data={'Name':['Geeta','Rani','Rohini','Rita','Rohan','Subham','Rishi',
                          'Ram','Dinesh','Arysn','Raja','Janavi'],
'Marks':[73,58,75,85,51,65,87,74,53,47,89,75]}
#print the Dataframe
dframe = pd.DataFrame(data)
#Percentile Rank of the Score1 Column in DataFrame
dframe['Percentile_rank']=dframe.Marks.rank(pct=True)
print("\n Values of Percentile Rank in the Distribution")
print("-------------------------------------------------")
print(dframe)
print("\n Measures of Dispersion and Position in the Distribution")
print("-----------------------------------------------------------------")
#Range of the Score1 Column in Dataframe
rng=max(dframe["Marks"]) - min(dframe["Marks"])
print("\n Value of Range in the Distribution = ", rng)
#Standard Deviation of the Score1 Column in DataFrame
std=round(dframe["Marks"].std(),3)
print("Value of Standard Deviation in the Distribution = ", std)
#Variance of the Score Column1 in DataFrame
var=round(dframe["Marks"].var(),3)
print("Value of Variance in the Distribution = ", var)
#Interquartile Range of the Score1 Column in DataFrame
iq = iqr(dframe["Marks"])
print("Value of Interquartile Range in the Distribution = ", iq)
```

```
#Boxplot Representation of the Score1 Column
print("\n Boxplot Representation of the Score1 Column")
print("-------------------------------------------------")
dframe.boxplot(column=["Marks"],grid=True, figsize=(7,7))
plt.text(x=0.75, y=dframe["Marks"].quantile(0.75), s="3rd Quartile")
plt.text(x=0.75, y=dframe["Marks"].median(), s="Median")
plt.text(x=0.75, y=dframe["Marks"].quantile(0.25), s="1st Quartile")
plt.text(x=0.75, y=dframe["Marks"].min(), s="Min")
plt.text(x=0.75, y=dframe["Marks"].max(), s="Max")
plt.text(x=0.6, y=dframe["Marks"].quantile(0.50),
                                        s="IQR", rotation=90,size=15)
```

- The output initially displays the entire dataset consisting of 12 records and three columns, namely, Name, Marks and the calculated Percentile_rank. Then, all the various measures of dispersion and position are displayed for the Score1 column.

```
 Values of Percentile Rank in the Distribution
-------------------------------------------------
      Name  Marks  Percentile_rank
0    Geeta     73        0.500000
1     Rani     58        0.333333
2   Rohini     75        0.708333
3     Rita     85        0.833333
4    Rohan     51        0.166667
5   Subham     65        0.416667
6    Rishi     87        0.916667
7      Ram     74        0.583333
8   Dinesh     53        0.250000
9    Arysn     47        0.083333
10    Raja     89        1.000000
11  Janavi     75        0.708333

 Measures of Dispersion and Position in the Distribution
-------------------------------------------------------------

 Value of Range in the Distribution =  42
Value of Standard Deviation in the Distribution =  14.437
Value of Variance in the Distribution =  208.424
Value of Interquartile Range in the Distribution =  20.75
```

2.3 INFERENTIAL STATISTICS

- In Inferential statistics, we make an inference from a sample about the population. The main aim of inferential statistics is to draw some conclusions from the sample and generalize them for the population data.

- For example, we have to find the average salary of a data analyst across India. There are following two options:
  1. The first option is to consider the data of data analysts across India and ask them their salaries and take an average.
  2. The second option is to take a sample of data analysts from the major IT cities in India and take their average and consider that for across India.
- The first option is not possible as it is very difficult to collect all the data of data analysts across India. It is time-consuming as well as costly.
- So, to overcome this issue, we will look into the second option to collect a small sample of salaries of data analysts and take their average as India average. This is the inferential statistics where we make an inference from a sample about the population.
- Inferential statistics draw inferences and predictions about a population based on a sample of data chosen from the population in question.
- In statistics, a sample is considered as a representative of the entire universe or population and is often used to draw inferences about the population.
- This is illustrated in following figure in which a portion of the population, termed as the sample, is used to represent the population for statistical analysis.
- Dealing with the right sample that represents a subset of the population is very important as most of the time it is impractical and impossible to conduct a census survey representing the entire population.
- Hence, choosing an appropriate sample for a population is a practical approach used by data analysts which is done by removing sampling bias as much as possible.
- Also, choosing an appropriate sample size is important so as to lessen the variability of the sample mean.

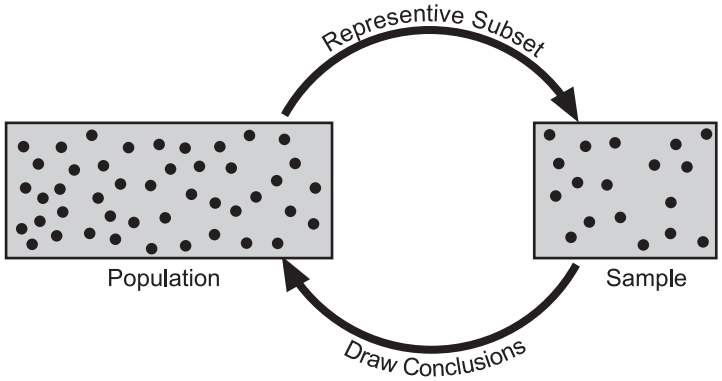


Fig. 2.6: Inferential Statistical Analysis

- Statistical inference mainly deals with two different kinds of problems – hypothesis testing and estimation of parameter values.

2.3.1 Hypothesis Testing

- The hypothesis testing is the one of the most promising inferential statistical techniques used in data analysis to check whether a stated hypothesis is accepted or rejected.
- The process to determine whether the stated hypothesis is accepted or rejected from sample data is called hypothesis testing.
- Hypothesis testing is mainly used to determine whether there is sufficient evidence in a data sample to conclude that a particular condition holds for an entire population.
- In the case of hypothesis testing, there can be two hypotheses namely, the null hypothesis (denoted by H_0) and the alternative hypothesis (denoted by H_a).
  1. The **null hypothesis** states that the sample statistic is equal to the population statistic. For example, a null hypothesis can be stated as there has been no difference found in the exam grade of students even after conducting coaching classes for them.
  2. The **alternate hypothesis** states that there is a variation in the sample statistic and population statistics. For example, an alternate hypothesis can be stated as there has been a significant improvement found in the exam grade of students after conducting coaching classes for them.
- It should be noted that once a null hypothesis for a problem is stated, the statistical calculation is made to conclude as to whether the null hypothesis is rejected or failed to be rejected based on the evaluation of statistical measures.
- In general, there are four basic steps to be followed for hypothesis testing:

  Step 1: State the null and alternative hypotheses.
  Step 2: Select the appropriate significance level and check the specified test assumptions.
  Step 3: Analyze the data by computing appropriate statistical tests.
  Step 4: Interpret the result.
- Statistical inferences begin with the assumption that the null hypothesis is true. The procedure is followed to determine whether there is enough evidence to prove that the alternative hypothesis is true. If not, the null hypothesis is considered to be true.
- Thus, there is a possibility of only two conclusions that can be inferred:
  - Reject the null hypothesis by showing enough evidence to support the alternative hypothesis.
  - Accept the null hypothesis by showing evidence to prove that there is not enough evidence to support the alternative hypothesis.

- While carrying out experiments on hypothesis, there are two types of errors – Type I and Type II - that can be encountered as mentioned in Table 2.1.

Table 2.1

H_0	True	False
Rejected	Type I Error	√
Not Rejected	√	Type II Error

- The Type I Error occurs when we reject a true null hypothesis as shown in above table. Here, H_0 is rejected though it is True.
- Again the Type II Error occurs when we do not reject a false null hypothesis as shown in above table. Here, H_0 is not rejected though it is False.
- The other two cases or possibilities are shown in the table (marked as √) are correctly predicted hypothesis.

Parametric Hypothesis Tests:

- Hypothesis testing can be classified as parametric tests and non-parametric tests.
  1. In the case of **parametric tests**, information about the population is completely known and can be used for statistical inference.
  2. In the case of **non-parametric tests**, information about the population is unknown and hence no assumptions can be made regarding the population.
- Let us discuss a few of the important most commonly used parametric tests and their significance in various statistical analyses.
- In each of these parametric tests, there is a common step of procedures followed as shown in Fig. 2.7.
- The initial step is to state the null and alternate hypotheses based on the given problem. The level of significance is chosen based on the given problem.
- The type of parametric test to be considered is an important decision-making task for correct analysis of the problem. Next, a decision rule is formulated to find the critical values and the acceptance/rejection regions.

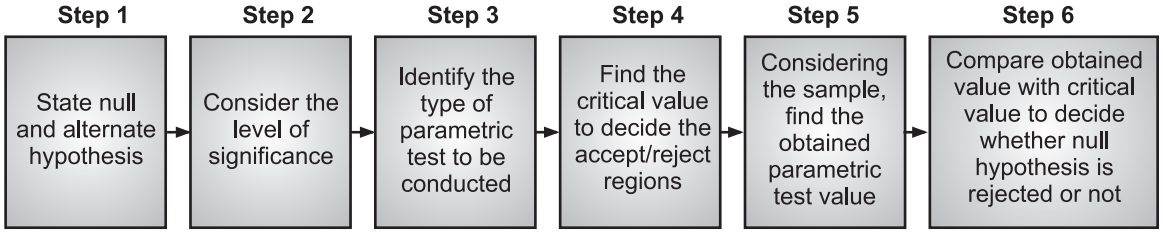


Fig. 2.7

- Lastly, the obtained value of the parametric test is compared with the critical test value to decide whether the null hypothesis (H_0) is rejected or accepted.
- The null hypothesis (H_0) and the alternate hypothesis (H_a) are mutually exclusive.
- At the beginning of any parametric test, is always assumed to be true and the alternate hypothesis H_0 or H_a carries the burden to be proved by following the above-mentioned steps as given in Fig. 2.7.
- Before we perform any type of parametric tests, let us try to understand some of the core terms related to any parametric tests that are required to be known:

1. Acceptance and Critical Regions:

- All the set of possible values which a test-statistic can be divided fall into two mutually exclusive groups:
  - 1^{st} group, called the **acceptance region**, consists of values that appear to be consistent with the null hypothesis.
  - 2^{nd} group, called the **rejection region** or the **critical region,** consists of values that are unlikely to occur if the null hypothesis is true.
- The value(s) that separates the critical region from the acceptance region is called the critical value(s).

2. One-tailed Test and Two-tailed Test:

- For some parametric tests like z-test, it is important to decide if the test is one-tailed or two-tailed test.
- If the specified problem has an equal sign, it is a case of a two-tailed test, whereas if the problem has a greater than (>) or less than (<) sign, it is a one-tailed test.
- Fig. 2.8 shows the differences between a two-tailed test and a one-tailed test. For example, let us consider the following three cases for a problem statement:
  - **Case 1:** A government school states that the dropout of female students between ages 12 and 18 years is 28%.
  - **Case 2:** A government school states that the dropout of female students between ages 12 and 18 years is greater than 28%.
  - **Case 3:** A government school states that the dropout of female students between ages 12 and 18 years is less than 28%.
- Case 1 is an example of the two-tailed test as it states that dropout rate = 28%. Again, Case II and Case III are both examples of one-tailed tests as Case II states that dropout rate > 28% and Case III states that dropout rate< 28%.

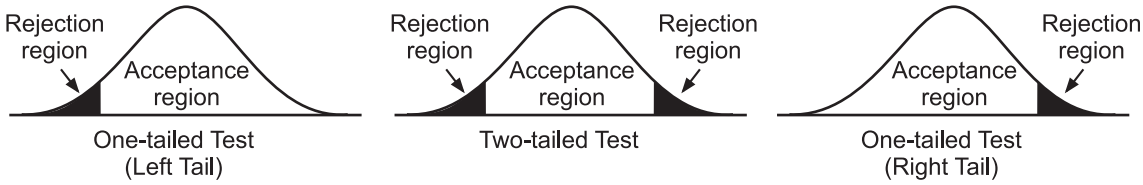


Fig. 2.8: One Tailed and Two Tailed Test

- The alternate hypothesis can take one of three forms – either a parameter has increased, or it has decreased or it has changed (may increase or decrease). This can be illustrated as shown below:
  - **Ha: μ > μ0:** This type of test is called an upper-tailed test or right-tailed test.
  - **Ha: μ < μ0:** This type of test is called a lower-tailed test or left-tailed test.
  - **Ha: μ ≠ μ0:** This type of test is called the two-tailed test.
- To summarize, while a one-tailed test checks for the effect of a change only in one direction, a two-tailed test checks for the effect of a change in both the directions.
- Thus, a two-tailed test considers both positive and negative effects for a change that is being studied for statistical analysis.

Significance Level (a):

- It is denoted by a, is the probability of the null hypothesis being rejected even if it is true. This is so because 100% accuracy is practically not possible for accepting or rejecting a hypothesis.
- For example, a significance level of 0.03 indicates that a 3% risk is being taken that a difference in values exists when there is no difference.
- Typical values of significance level are 0.01, 0.05, and 0.1 which are significantly small values chosen to control the probability of committing a Type I error.

Calculated Probability (r):

- The r-value is a calculated probability that states that when the null hypothesis is true, the statistical summary will be greater than or equal to the actual observed results.
- It is the probability of finding the observed or more extreme results when the null hypothesis is true.
- Low r-values indicate that there is little likelihood that the statistical expectation is true.
- Some of widely used hypothesis testing types are t-test, z-test, ANOVA-test and Chi-square test.

- Let us see above tests in detail:

1. t-test:

- A t-test is a type of inferential statistic which is used to determine if there is a significant difference between the means of two groups which may be related in certain features.
- It is mostly used when the data sets, like the set of data recorded as outcome from flipping a coin a 100 times, would follow a normal distribution and may have unknown variances.
- The t-test is used as a hypothesis testing tool, which allows testing of an assumption applicable to a population.
- The t-test has two types namely, one sampled t-test and two-sampled t-test.

(i) One Sample t-test:

- The one sample t-test determines whether the sample mean is statistically different from a known or hypothesized population mean. The one sample t-test is a parametric test.

Example: We have 10 ages and you are checking whether avg age is 30 or not. Following program illustrate the code for one sample t-test.

```
from scipy.stats import ttest_1samp
import numpy as np
ages = np.genfromtxt("ages.csv")
print(ages)
ages_mean = np.mean(ages)
print(ages_mean)
tset, pval = ttest_1samp(ages, 30)
print("p-values",pval)
if pval< 0.05:    # alpha value is 0.05 or 5%
   print(" we are rejecting null hypothesis")
else:
   print("we are accepting null hypothesis")
```

Output:

```
[34. 32. 29. 22. 39. 29. 37. 36. 38. 30. 26. 22. 21.]
30.384615384615383
p-values 0.829266573264052
we are accepting null hypothesis
```

(ii) Two sampled t-test:

- The independent samples t-test or 2-sample t-test compares the means of two independent groups in order to determine whether there is statistical evidence that the associated population means are significantly different.
- The independent samples t-test is a parametric test. This test is also known as Independent t-test.

```
#Program for Two sample t-test

import statistics
import numpy as np
import scipy.stats as sc
#Creating two independent samples
d1=[11,16,6, 17, 12,13,8, 15,14,9]
d2=[7, 22,13, 14, 15,12, 8,18,21, 17,10,23]
# calculate means
mean1, mean2 = statistics.mean(d1), statistics.mean(d2)
# calculate standard errors
se1, se2 = sc.sem(d1), sc.sem(d2)
# standard error on the difference between the samples
sed = np.sqrt(se1**2.0 + se2**2.0)
# calculate the t statistic
t_stat = (mean1 - mean2) / sed
# degrees of freedom
df = len(d1) + len(d2) - 2
# storing the critical value
alpha=0.05
t_crit1 = -2.09
t_crit2 = +2.09
print("TWO SAMPLES t-TEST RESULTS")
#print("TWO SAMPLES t-TEST RESULTS")
print(".....................\n")
print("Standard error of two sample:",round(se1,2),
                                          "and", round(se2,2))
print("Sample Mean of Two Samples:",mean1, "and",mean2)
print("t-test:",round(t_stat,2))
print("t-critical",round(t_crit1,2),"and",round(t_crit2,2))
if (t_stat>t_crit1) and (t_stat<t_crit2):
   print("\n Null hypothesis is accepted.")
else:
   print("\n Null hypothesis is rejected.")
```

Output:

```
TWO SAMPLES t-TEST RESULTS
.....................

Standard error of two sample: 1.14 and 1.54
Sample Mean of Two Samples: 12.1 and 15
t-test: -1.51
t-critical -2.09 and 2.09
hypothesis accepted
```

Paired Sampled t-test:

- The paired sample t-test is also called dependent sample t-test. It's an uni variate test that tests for a significant difference between two related variables.

```
import statistics
import numpy as np
import math
#Sample Details
d1=[3.8,5.2,3.9,4.1,4.3,4.4,4.2,5.6]
#Sample Size
n=10
# calculate means
s_mean = statistics.mean(d1)
# calculate standard deviation
sd=0
for e in d1:
   sd=sd+(float(e) - s_mean)**2
sigma = math.sqrt(sd / (n-1))
den = sigma/np.sqrt(n)
num = s_mean-4
t_stat = num / den
alpha=0.05
t_crit1 = -2.262
t_crit2 = +2.262
print("two sample paired t-test results")
print(".........................")
```

```
print("sample mean:",s_mean)
print("standard deviation:",round(sigma,2))
print("t-test:",round(t_stat,2))
print("t-critical",round(t_crit1,2),"and",round(t_crit2,2))
if (t_stat>t_crit1) and (t_stat<t_crit2):
   print("null hypothesis accepted")
else:
   print("null Hypothesis rejected")
```

Output:

```
two sample paired t-test results
.........................
sample mean: 4.4375
standard deviation: 0.56
t-test: 2.47
t-critical -2.26 and 2.26
null Hypothesis rejected
```

- The output displays the values of standard deviation 0.56 and the sample Mean 4.4375. Then, the t-test is calculated (2.47) and compared with the critical value of t. A conclusion is made on whether the null hypothesis is accepted or rejected based on whether the obtained t-test value is less than or greater than the critical t-value.

2. z-test:

- A z-test is mainly used when the data is normally distributed. The z-test is mainly used when the population mean and standard deviation are given.
- The **one-sample z-test** is mainly used for comparing the mean of a sample to some hypothesized mean of a given population, or when the population variance is known.
- The main analysis is to check whether the mean of a sample is reflective of the population being considered.
- We would use a z test if:
  - We sample size is greater than 30. Otherwise, use a t test.
  - Data points should be independent from each other. In other words, one data point isn't related or doesn't affect another data point.
  - Our data should be normally distributed. However, for large sample sizes (over 30) this doesn't always matter.
  - Our data should be randomly selected from a population, where each item has an equal chance of being selected.
  - Sample sizes should be equal if at all possible.

Example: A Institute stated that the students' study that is more intelligent than the average Institute. On calculating the IQ scores of 50 students, the average turns out to be 11. The mean of the population IQ is 100 and the standard deviation is 15. State whether the claim of Institute is right or not at a 5% significance level.

First, we define the null hypothesis and the alternate hypothesis. Our null hypothesis will be:

H_0:μ = 100 and alternate hypothesis H_a : μ >100 state the level of significance. Here, our level of significance given in this question (α = 0.05), if not given then we take α = 0.05

Now, we look up to the z-table. For the value of α = 0.05, the z-score for the right-tailed test is 1.645.

Now, we perform the Z-test on the problem:

where:

- X = 110
- Mean (mu) = 100
- Standard deviation (sigma) = 15
- Significance level (alpha) = 0.05
- N=50

$$= \frac{(110-100)}{15/\text{sqrt}(50)}$$

$$= 4.71$$

Here, 4.71 >1.645, so we reject the null hypothesis. If z-test statistics is less than z-score, then we will not reject the null hypothesis.

```
#Program for Sample python code for Z test
import math
import numpy as np
from numpy.random import randn
from statsmodels.stats.weightstats import ztest

# Consider a random array of 50 numbers having mean 110 and sd 15
mean_iq = 110
sd_iq = 15/math.sqrt(50)
null_mean =100
data = sd_iq*randn(50)+mean_iq
# print mean and sd
print('mean=%.2f stdv=%.2f' % (np.mean(data), np.std(data)))
```

```
ztest_Score, p_value= ztest(data,value = null_mean, alternative='larger')
if(p_value <  0.05):
   print("Reject Null Hypothesis")
else:
   print("Fail to Reject NUll Hypothesis")
```

Output:

```
mean=109.75 stdv=2.34
Reject Null Hypothesis
```

- In **two sample z-test**, similar to t-test here we are checking two independent data groups and deciding whether sample mean of two group is equal or not.

$$z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_{p1}^2}{n_1} + \frac{\sigma_{p2}^2}{n_2}}}$$

- In this case, $\bar{X}_1$ and $\bar{X}_2$ are the sample means, σ_{p1} and σ_{p2} are the standard deviation of the populations, and n_1 and n_2 are the sample sizes.
- In case the samples happen to be large but presumed to have been taken from the same population whose variance is known, then the method used for carrying out the z-test for two samples is:

$$z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\sigma_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

3. ANOVA Test (f-test):

- The t-test works well when dealing with two groups, but sometimes we want to compare more than two groups at the same time.
- For example, if we wanted to test whether voter age differs based on some categorical variable like race, we have to compare the means of each level or group the variable.
- We could carry out a separate t-test for each pair of groups, but when you conduct many tests you increase the chances of false positives.
- The ANalysis Of VAriance or ANOVA is a statistical inference test that lets you compare multiple groups at the same time.

  f = Between group variability / Within group variability
- Unlike the z and t-distributions, the f-distribution does not have any negative values because between and within-group variability are always positive due to squaring each deviation.
- One way f-test (ANOVA) tell whether two or more groups are similar or not based on their mean similarity and f-score.

Example: Consider the dataset of Plant growth following table. There are three different categories of plant and their weight and need to check whether all three groups are similar or not. Following is content of the 'plant_g.csv' file.

Sr. No.	Weight	Group
1	4.17	ctrl
2	5.58	ctrl
3	5.18	ctrl
4	6.11	ctrl
5	4.5	ctrl
6	4.61	ctrl
7	5.17	ctrl
8	4.53	ctrl
9	5.33	ctrl
10	5.14	ctrl
11	4.81	trt1
12	4.17	trt1
13	4.41	trt1
14	3.59	trt1
15	5.87	trt1
16	3.83	trt1
17	6.03	trt1
18	4.89	trt1
19	4.32	trt1
20	4.69	trt1
21	6.31	trt2
22	5.12	trt2
23	5.54	trt2
24	5.5	trt2
25	5.37	trt2
26	5.29	trt2
27	4.92	trt2
28	6.15	trt2
29	5.8	trt2
30	5.26	trt2

```
#Program for ANOVA Test
import pandas as pd
import scipy.stats
dframe_anova = pd.read_csv('plant_g.csv')
dframe_anova = dframe_anova[['weight','group']]
groups = pd.unique(dframe_anova.group.values)
d1_data = {groups:dframe_anova['weight'][dframe_anova.group ==
                                          groups] for groups in groups}
F, p1 = scipy.stats.f_oneway(d1_data['ctrl'],
                                    d1_data['trt1'], d1_data['trt2'])
print("p-value for significance is: ", p1)
if p1<0.05:
   print("reject null hypothesis")
else:
   print("accept null hypothesis")
```

Output:

```
p-value for significance is:  0.0159099583256229
reject null hypothesis
```

- The two way f-test is extension of 1-way f-test, it is used when we have 2 independent variable and 2+ groups. 2-way f-test does not tell which variable is dominant. If we need to check individual significance then Post-hoc testing need to be performed.
- Now let's take a look at the Grand mean crop yield (the mean crop yield not by any sub-group), as well the mean crop yield by each factor, as well as by the factors grouped together.

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
df_anova2 = pd.read_csv(
          "https://raw.githubusercontent.com/Opensourcefordatascience/Data-
         sets/master/crop_yield.csv")
model = ols('Yield ~ C(Fert)*C(Water)', df_anova2).fit()
print(f"Overall  model  F({model.df_model:  .0f},{model.df_resid:  .0f})  =
{model.fvalue: .3f}, p = {model.f_pvalue: .4f}")
res = sm.stats.anova_lm(model, typ= 2)
res
```

Output:

```
Overall model F( 3, 16) =  4.112, p =  0.0243
```

4. Chi-square Test:

- The test is applied when we have two categorical variables from a single population. It is used to determine whether there is a significant association between the two variables.

Example: Consider the Ctest.CSV file as per followings:

Gender	Shopping
M	1000
M	2000
F	1500
M	3000
F	5000
M	4000
F	2500
F	3500
M	4500

```
# Program for chi-square test Hypothesis Testing
import pandas as pd
import scipy
from scipy.stats import chi2
df_chi = pd.read_csv('ctest.csv')
contingency_table=pd.crosstab(df_chi["Gender"],df_chi["Shopping"])
print('contingency_table :-\n',contingency_table)#Observed Values
Observed_Values = contingency_table.values
print("Observed Values :-\n",Observed_Values)
b=scipy.stats.chi2_contingency(contingency_table)
Expected_Values = b[3]
print("Expected Values :-\n",Expected_Values)
no_of_rows=len(contingency_table.iloc[0:2,0])
no_of_columns=len(contingency_table.iloc[0,0:2])
ddof=(no_of_rows-1)*(no_of_columns-1)
print("Degree of Freedom:-",ddof)
alpha = 0.05
chi_square=sum([(o-e)**2./e for o,e
                                in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]
print("chi-square statistic:-",chi_square_statistic)
```

```
critical_value=chi2.ppf(q=1-alpha,df=ddof)
print('critical_value:',critical_value)#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=ddof)
print('p-value:',p_value)
print('Significance level: ',alpha)
print('Degree of Freedom: ',ddof)
print('chi-square statistic:',chi_square_statistic)
print('critical_value:',critical_value)
print('p-value:',p_value)
if chi_square_statistic>=critical_value:
   print("Reject H0,There is a relationship between
                                          2 categorical variables")
else:
   print("Retain H0,There is no relationship between
                                          2 categorical variables")
if p_value<=alpha:
   print("Reject H0,There is a relationship between
                                          2 categorical variables")
else:
   print("Retain H0,There is no relationship between
                                          2 categorical variables")
```

Output:

```
contingency_table :-
 Shopping  1000  1500  2000  2500  3000  3500  4000  4500  5000
Gender
F            0     1     0     1     0     1     0     0     1
M            1     0     1     0     1     0     1     1     0
Observed Values :-
 [[0 1 0 1 0 1 0 0 1]
 [1 0 1 0 1 0 1 1 0]]
Expected Values :-
 [[0.44444444 0.44444444 0.44444444 0.44444444 0.44444444 0.44444444
  0.44444444 0.44444444 0.44444444]
 [0.55555556 0.55555556 0.55555556 0.55555556 0.55555556 0.55555556
  0.55555556 0.55555556 0.55555556]]
Degree of Freedom:- 1
chi-square statistic:- 2.05
critical_value: 3.841458820694124
p-value: 0.1522061897871283
Significance level:  0.05
Degree of Freedom:  1
chi-square statistic: 2.05
critical_value: 3.841458820694124
p-value: 0.1522061897871283
Retain H0,There is no relationship between 2 categorical variables
Retain H0,There is no relationship between 2 categorical variables
```

2.3.2 Estimation of Parameter Values

- Parameter estimation plays a vital role in statistics. In statistics, finding estimation or inference refers to the task of drawing conclusions about a population, based on the information provided about the sample.
- This means that the task of estimation of parameter values involves making inferences from a given sample about an unknown population parameter.
- This can be done in two ways namely, using point estimate and using the interval estimate. Both of these ways of estimation of parameter values.

1. Point Estimate:

- The point estimation of a population parameter considers only a single value of a statistic.
- As point estimation is based on a single random sample, its value will vary when different random samples are considered from the sample population.
- An example of point estimation can be the sample means X for the population mean m. While carrying out point estimation it is important to maintain consistency.
- For this, to get more accurate point estimation, we need to consider a considerable large sample size.
- There is no single point estimation method that is universally the best or always proves appropriates in all situations.
- Few of the standard methods for point estimation include:
  (i) Maximum Likelihood Estimator (MLE).
  (ii) Minimum-Variance mean-Unbiased Estimator (MVUE).
  (iii) Minimum Mean Squared Error (MMSE).
  (iv) Best Linear Unbiased Estimator (BLUE).

2. Interval Estimate:

- The interval estimation of a population parameter considers two values between which the population parameter is likely to lie.
- The two values allow setting the interval range within which the parameter value of a population has the probability of occurring.
- If we consider a case where the mean value of a population (μ) lies within the range between 50to 100 ($50 < \mu < 100$), it is a case of interval estimation.
- Contrary to that, if we consider a case where the mean value of a population (μ) is 78, it is a case of point estimation.

- Interval estimation uses sample data to calculate an interval of probable values of an unknown population parameter. For doing so, the following formula is used,

$$\mu = \bar{x} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

- This is the formula for finding the confidence intervals (μ) for the population mean depending on the sample size (n). Here, $\bar{x}$ is the sample means, z is the confidence coefficient, α is the confidence level, σ is the standard deviation, and n is the sample size. The value $\frac{\sigma}{\sqrt{n}}$ is called the standard error for mean.

2.4 MEASURING DATA SIMILARITY AND DISSIMILARITY

- In data science, the similarity measure is a way of measuring how data samples are related or closed to each other. The dissimilarity measure is to tell how much the data objects are distinct.
- In data mining applications, such as clustering, outlier analysis, and nearest-neighbor classification, we need ways to assess how alike or unalike objects are in comparison to one another.
- For example, a store may want to search for clusters of customer objects, resulting in groups of customers with similar characteristics (e.g., similar income, area of residence, and age). Such information can then be used for marketing.
- A cluster is a collection of data objects such that the objects within a cluster are similar to one another and dissimilar to the objects in other clusters.
- Outlier analysis also employs clustering-based techniques to identify potential outliers as objects that are highly dissimilar to others.
- Knowledge of object similarities can also be used in nearest-neighbor classification schemes where a given object (e.g., a patient) is assigned a class label (relating to, say, a diagnosis) based on its similarity toward other objects in the model.
- Similarity and dissimilarity measures are referred as measures of proximity. Similarity and dissimilarity are related.
- A similarity measure for two objects, i and j, will typically return the value 0 if the objects are unalike.
- The higher the similarity value, the greater the similarity between objects, (typically, a value of 1 indicates complete similarity, that is, the objects are identical.)
- A dissimilarity measure works the opposite way. It returns a value of 0 if the objects are the same (and therefore, far from being dissimilar). The higher the dissimilarity value, the more dissimilar the two objects are.

2.4.1 Data Matrix versus Dissimilarity Matrix

- Consider the objects described by multiple attributes. Suppose that we have n objects (e.g., persons, items, or courses) described by p attributes (also called measurements or features, such as age, height, weight, or gender).
- The objects are $x_1=(x_{11}, x_{12}, : : : , x_{1p})$, $x_2 =(x_{21}, x_{22}, : : : , x_{2p})$, and so on, where x_{ij} is the value for object x_i of the j^{th} attribute.
- For brevity, we hereafter refer to object xi as object i. The objects may be tuples in a relational database, and are also referred to as data samples or feature vectors.
- Main memory-based clustering and nearest-neighbor algorithms typically operate on either of the following two data structures:

1. Data Matrix (or Object-by-attribute Structure):

- Data matrix (or object-by-attribute structure) structure stores the n data objects in the form of a relational table, or n-by-p matrix (n objects × p attributes):

$$\begin{bmatrix} x_{ll} & \cdots & x_{lf} & \cdots & x_{lp} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{il} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{nl} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Each row corresponds to an object. As part of our notation, we may use f to index through the p attributes.

2. Dissimilarity Matrix (or Object-by-object Structure):

- This structure stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

where, d.(i, j) is the measured dissimilarity or "difference" between objects i and j.

- In general, d.(i, j) is a non-negative number that is close to 0 when objects i and j are highly similar or "near" each other, and becomes larger the more they differ.
- Note that d.(i, i)= 0; that is, the difference between an object and itself is 0. Furthermore, d(i, j) = d(j, i). (For readability, we do not show the d(j, i) entries; the matrix is symmetric.)

- Measures of similarity can often be expressed as a function of measures of dissimilarity.
- For example, for nominal data,

$$sim(i, j) = 1 - d(i - j) \quad \ldots(2.1)$$

where, sim (i, j) is the similarity between objects i and j.

- A data matrix is made up of two entities or "things," namely, rows (for objects) and columns (for attributes).
- Therefore, the data matrix is often called a **two-mode matrix**. The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a **one-mode matrix**.

2.4.2 Proximity Measures for Nominal Attributes

- A nominal attribute can take on two or more states. For example, map color is a nominal attribute that may have, say, five states namely, red, yellow, green, pink and blue.
- Let the number of states of a nominal attribute be M. The states can be denoted by letters, symbols, or a set of integers, such as 1, 2, ... , M.
- Notice that such integers are used just for data handling and do not represent any specific ordering
- The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p} \quad \ldots (2.2)$$

where, m is the number of matches (i.e., the number of attributes for which i and j are in the same state), and p is the total number of attributes describing the objects.

- Weights can be assigned to increase the effect of m or to assign greater weight to the matches in attributes having a larger number of states.

Dissimilarity between Nominal Attributes:

- Suppose that we have the sample data of in Table 2.2, except that only the object-identifier and the attribute test-1 are available, where test-1 is nominal.
- Let us compute the dissimilarity matrix,

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

- Since, here we have one nominal attribute, test-1 we set p = 1, in Equation (2.2) so that Ai, j) evaluates to 0 if objects i and j match, and 1 if the objects differ. Thus, we get

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

- From this, we see that all objects are dissimilar except objects 1 and 4 (i.e., d(4, 1) = 0).

Table 2.2: A Sample Data Table containing Attributes of Mixed Type

Object Identifier	Test-1 (Nominal)	Test-2 (Ordinal)	Test-3 (Numeric)
1.	case A	outstanding	45
2.	case B	fair	22
3.	case C	good	64
4.	case A	outstanding	28

- Alternatively, similarity can be computed as,

$$\text{sim}(i, j) = 1 - d(i, j) = \frac{m}{p}$$

- Proximity between objects described by nominal attributes can be computed using an alternative encoding scheme.
- Nominal attributes can be encoded using asymmetric binary attributes by creating a new binary attribute for each of the M states.
- For an object with a given state value, the binary attribute representing that state is set to 1, while the remaining binary attributes are set to 0.
- For example, to encode the nominal attribute map color, a binary attribute can be created for each of the five colors previously listed.
- For an object having the color yellow, the yellow attribute is set to 1, while the remaining four attributes are set to 0.

2.4.3 Proximity Measures for Binary Attributes

- A binary attribute has only one of two states: 0 and 1, where 0 means that the attribute is absent and 1 means that it is present.
- To compute the dissimilarity between two binary attributes approach involves computing a dissimilarity matrix from the given binary data.
- If all binary attributes are thought of as having the same weight, we have the 2x2 contingency table of Table 2.3, where q is the number of attributes that equal 1 for

both objects i and j, r is the number of attributes that equal 1 for object i but equal 0 for object j, s is the number of attributes that equal 0 for object i but equal 1 for object j, and t is the number of attributes that equal 0 for both objects i and j.

- The total number of attributes is p, where p = q + r + s + t.
- Recall that for symmetric binary attributes, each state is equally valuable. Dissimilarity that is based on symmetric binary attributes is called symmetric binary dissimilarity.
- If objects i and j are described by symmetric binary attributes, then the dissimilarity between i and j is,

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

Table 2.3: Contingency Table for Binary Attributes

		Object j		
		1	0	sum
Object i	1	q	r	q + r
	0	s	t	s + t
	sum	q + s	r + t	P

- For asymmetric binary attributes, the two states are not equally important, such as the positive (1) and negative (0) outcomes of a disease test.
- Given two asymmetric binary attributes, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match).
- Therefore, such binary attributes are often considered “monary” (having one state).
- The dissimilarity based on these attributes is called asymmetric binary dissimilarity, where the number of negative matches, t is considered unimportant and is thus ignored in the following computation:

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

- Complementarily, we can measure the difference between two binary attributes based on the notion of similarity instead of dissimilarity.
- For example, the asymmetric binary similarity between the objects i and j can be computed as,

$$sim(i, j) = \frac{q}{q + r + s} = 1 - d(i, j)$$

- The coefficient sim(i, j) in above equation is called the Jaccard coefficient. When both symmetric and asymmetric binary attributes occur in the same data set.

Dissimilarity between Binary Attributes:

- Suppose that a patient record table contains the attributes Name, Gender, Fever, Cold, Test-1, Test-2, Test-3 and Test-4, where name is an object identifier, gender is a symmetric attribute and the remaining attributes are asymmetric binary.
- For asymmetric attribute values, let the values Y (yes) and P (positive) be set to 1 and the value N (no or negative) be set to 0. Suppose that the distance between objects (patients) is computed based only on the asymmetric attributes.

Relational table where Patients are described by Binary Attributes

Name	Gender	Fever	Cold	Test-1	Test-2	Test-3	Test-4
Varun	M	Y	N	P	N	N	N
Akshay	M	Y	Y	N	N	N	N
Sara	F	Y	N	P	N	P	N
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

- According to following equation the distance between each pair of the three patients namely, Varun, Sara and Akshay is,

$$d(\text{Varun, Akshay}) = \frac{1+1}{1+1+1} = 0.67,$$

$$d(\text{Varun, Sara}) = \frac{0+1}{2+0+1} = 0.33,$$

$$d(\text{Akshay, Sara}) = \frac{1+2}{1+1+2} = 0.75$$

- Above measurements suggest that Akshay and Sara are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs. Of the three patients, Varun and Sara are the most likely to have a similar disease.

2.4.4 Dissimilarity of Numeric Data

- The distance measures that are commonly used for computing the dissimilarity of objects described by numeric attributes. These measures include the Euclidean, Manhattan and Minkowski distances.

1. Minkowski Distance:

- In some cases, the data are normalized before applying distance calculations. This involves transforming the data to fall within a smaller or common range, such as [-1, 1] or [0.0, 1.0].

- Consider a height attribute, for example, which could be measured in either meters or inches. In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such attributes greater effect or "weight."
- Normalizing the data attempts to give all attributes an equal weight. It may or may not be useful in a particular application.

2. Euclidean Distance:

- The most popular distance measure is Euclidean distance (i.e., straight line or 'as the crow flies").
- Let i = $(x_{i1}, x_{i2}, ..., x_{ip})$ and j = $(x_{j1}, x_{j2}, ..., x_{jp})$ be two objects described by p numeric attributes. The Euclidean distance between objects i and j is defined as follows:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2}$$

3. Manhattan Distance:

- Another well-known measure is the Manhattan (or city block) distance, named so because it is the distance in blocks between any two points in a city (such as 2 blocks down and 3 blocks over for a total of 5 blocks).
- It is defined as follows,

$$d(i, j) = |x_{i1}, x_{j1}| + |x_{i2}, x_{j2}| + ... + |x_{ip} - x_{jp}|$$

- Both the Euclidean and the Manhattan distance satisfy the following mathematical properties:
  - **Non-negativity:** $d(i, j) \geq 0$: Distance is a non-negative number.
  - **Identity of Indiscernibles:** $d(i, i) = 0$: The distance of an object to itself is 0.
  - **Symmetry:** $d(i, j) = d(j, i)$: Distance is a symmetric function.
  - **Triangle inequality:** $d(i, j) \leq d(i, k) + d(k, j)$: Going directly from object i to object j in space is no more than making a detour over any other object k.
- A measure that satisfies these conditions is known as metric. Note that the non-negativity property is implied by the other three properties.

Euclidean Distance and Manhattan Distance:

- Let $x_1 = (3, 2)$ and $x_2 = (5, 6)$ represent two objects. The Euclidean distance between the two is $\sqrt{2^2 + 4^2} = 4.47$. The Manhattan distance between the two is 2 + 4 = 6.
- Minkowski distance is a generalization of the Euclidean and Manhattan distances and defined as follows:

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

where, h is a real number such that h ≥ 1, (such a distance is also called L_p norm in some literature, where the symbol p refers to our notation of h. We have kept p as the number of attributes to be consistent). It represents the Manhattan distance when h = 1 (i.e., L_1 norm) and Euclidean distance when h = 2 (i.e., L_2 norm).

2.4.5 Proximity Measures for Ordinal Attributes

- Ordinal attributes may also be obtained from the discretization of numeric attributes by splitting the value range into a finite number of categories.
- These categories are organized into ranks. That is, the range of a numeric attribute can be mapped to an ordinal attribute f having M_f states.
- For example, the range of the interval-scaled attribute temperature (in Celsius) can be organized into the following states: -30 to -10, -10 to 10, 10 to 30, representing the categories cold temperature, moderate temperature, and warm temperature, respectively.
- Let M represent the number of possible states that an ordinal attribute can have. These ordered states define the ranking 1, ... , M_f.
- "How are ordinal attributes handled?" The treatment of ordinal attributes is quite similar to that of numeric attributes when computing dissimilarity between objects.
- Suppose that f is an attribute from a set of ordinal attributes describing n objects. The dissimilarity computation with respect to f involves the following steps:
  1. The value of f for the i^{th} object is x_{if} and f has M_f ordered states, representing the ranking 1,..., M_f. Replace each x_{if} by its corresponding rank, $r_{if} \in \{1, ..., M_f\}$.
  2. Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto [0.0, 1.0] so that each attribute has equal weight. We perform such data normalization by replacing the rank r_{if} of the i^{th} object in the f^{th} attribute by,

$$Z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  3. Dissimilarity can then be computed using any of the distance measures described in Section 2.4.4 for numeric attributes, using z_{if} to represent the f value for the i^{th} object.

Dissimilarity between Ordinal Attributes:

- Suppose that we have the sample data shown earlier in Table 2.2, except that this time only the object-identifier and the continuous ordinal attribute, test-2 are available.
- There are three states for test-2 namely, fair, good and outstanding, i.e., $M_f = 3$.

- For step 1, if we replace each value for test-2 by its rank, the four objects are assigned the ranks 3, 1, 2 and 3, respectively.
- Step 2 normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5 and rank 3 to 1.0.
- For step 3, we can use, say, the Euclidean distance, which results in the following dissimilarity matrix:

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

- Therefore, objects 1 and 2 are the most dissimilar, as are objects 2 and 4 (i.e., d(2, 1) = 1.0 and d(4, 2) = 1.0). This makes intuitive sense since objects 1 and 4 are both outstanding. Object 2 is fair, which is at the opposite end of the range of values for test-2.
- Similarity, values for ordinal attributes can be interpreted from dissimilarity as sim(i, j) = 1 – d(i op, j).

2.5 CONCEPT OF OUTLIERS

- Outliers are a very important aspect of data analysis. This has many applications in determining fraud and potential new trends in the market.
- In purely statistical sense, an outlier is an observation point that is distant from other observations.
- The probably first definition was given by Grubbs in 1969 as “an outlying observation, or outlier is one that appears to deviate markedly from other members of the sample in which it occurs”.
- Outliers are different from the noise data:
  - Noise is random error or variance in a measured variable.
  - Noise should be removed before outlier detection.
- An outlier may indicate an experimental error, or it may be due to variability in the measurement.
- In data mining, outlier detection aims to find patterns in data that do not conform to expected behavior. It is extensively used in many application domains such as:
  - Fraud detection for credit cards, insurance and healthcare.
  - Telecom fraud detection.

- Intrusion detection in cyber-security.
- Medical analysis.
- Fault detection in safety-critical systems.

2.5.1 Types of Outliers

- Outliers can be classified into following three categories:

1. Global Outlier (or Point Outliers):

- If an individual data point can be considered anomalous with respect to the rest of the data, then the datum is termed as a point outlier.
- For example, Intrusion detection in computer networks.

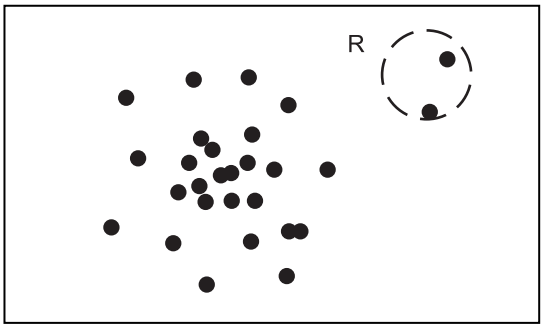


Fig. 2.9: The Objects in Region R are Outliers

- In Fig. 2.9 the points in region R significantly deviate from the rest of the data set and hence are examples of global outliers.
- To detect global outliers, a critical issue is to find an appropriate measurement of deviation with respect to the application in question.
- Various measurements are proposed, and, based on these, outlier detection methods are partitioned into different categories.
- Global outlier detection is important in many applications. Consider intrusion detection in computer networks, for example.
- If the communication behavior of a computer is very different from the normal patterns (e.g., a large number of packages is broadcast in a short time), this behavior may be considered as a global outlier and the corresponding computer is a suspected victim of hacking.
- As another example, in trading transaction auditing systems, transactions that do not follow the regulations are considered as global outliers and should be held for further examination.

2. Contextual Outliers:

- If an individual data instance is anomalous in a specific context or condition (but not otherwise), then it is termed as a contextual outlier.

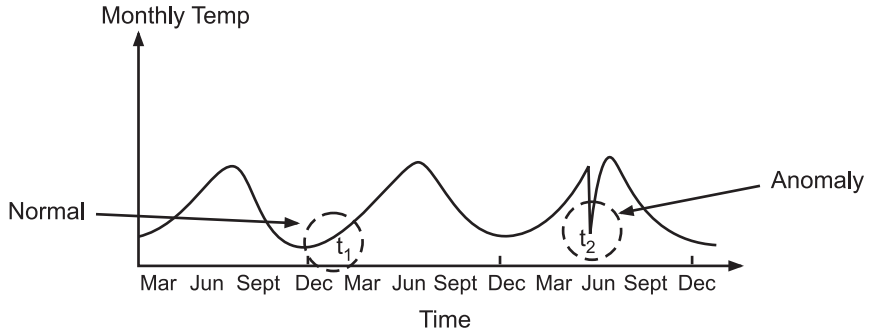


Fig. 2.10

- Attributes of data objects should be divided into two groups:
  - **Contextual Attributes:** Defines the context, e.g., time and location.
  - **Behavioral Attributes:** Characteristics of the object, used in outlier evaluation, e.g., temperature.

3. Collective Outliers:

- If a collection of data points is anomalous with respect to the entire data set, it is termed as a collective outlier.
- In Fig. 2.11, the black objects as a whole form a collective outlier because the density of those objects is much higher than the rest in the data set.
- However, every black object individually is not an outlier with respect to the whole data set.

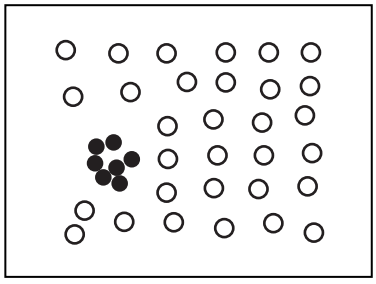


Fig. 2.11: The Black Objects form a Collective Outlier

- Collective outlier detection has many important applications. For example, in intrusion detection, a denial-of-service package from one computer to another is considered normal, and not an outlier at all.
- However, if several computers keep sending denial-of-service packages to each other, they as a whole should be considered as a collective outlier.

- The computers involved may be suspected of being compromised by an attack. As another example, a stock transaction between two parties is considered normal.
- However, a large set of transactions of the same stock among a small party in a short period are collective outliers because they may be evidence of some people manipulating the market.
- Unlike global or contextual outlier detection, in collective outlier detection we have to consider not only the behavior of individual objects, but also that of groups of objects.
- Therefore, to detect collective outliers, we need background knowledge of the relationship among data objects such as distance or similarity measurements between objects.
- In summary, a data set can have multiple types of outliers. Moreover, an object may belong to more than one type of outlier. In business, different outliers may be used in various applications or for different purposes.
- Global outlier detection is the simplest. Context outlier detection requires background information to determine contextual attributes and contexts.
- Collective outlier detection requires background information to model the relationship among objects to find groups of outliers.

2.5.2 Outlier Detection Methods

- The outlier detection methods can be divided into supervised methods, semi-supervised methods and unsupervised methods.

1. Supervised Methods:

- Supervised methods model data normality and abnormality. Domain experts examine and label a sample of the underlying data.
- Outlier detection can then be modeled as a classification problem .The task is to learn a classifier that can recognize outliers. The sample is used for training and testing.
- In some applications, the experts may label just the normal objects, and any other objects not matching the model of normal objects are reported as outliers.
- Other methods model the outliers and treat objects not matching the model of outliers as normal. Although many classification methods can be applied, challenges to supervised outlier detection include the following:
- The two classes (i.e., normal objects versus outliers) are imbalanced. That is, the population of outliers is typically much smaller than that of normal objects.
- Therefore, methods for handling imbalanced classes may be used, such as oversampling (i.e., replicating) outliers to increase their distribution in the training set used to construct the classifier.
- Due to the small population of outliers in data, the sample data examined by domain experts and used in training may not even sufficiently represent the outlier distribution.

- The lack of outlier samples can limit the capability of classifiers built as such. To tackle these problems, some methods "make up" artificial outliers.
- In many outlier detection applications, catching as many outliers as possible (i.e., the sensitivity or recall of outlier detection) is far more important than not mislabeling normal objects as outliers.
- Consequently, when a classification method is used for supervised outlier detection, it has to be interpreted appropriately so as to consider the application interest on recall.
- In summary, supervised methods of outlier detection must be careful in how they train and how they interpret classification rates due to the fact that outliers are rare in comparison to the other data samples.

2. Unsupervised Methods:

- In some application scenarios, objects labeled as "normal" or "outlier" are not available. Thus, an unsupervised learning method has to be used.
- Unsupervised outlier detection methods make an implicit assumption: The normal objects are somewhat "clustered."
- In other words, an unsupervised outlier detection method expects that normal objects follow a pattern far more frequently than outliers. Normal objects do not have to fall into one group sharing high similarity.
- Instead, they can form multiple groups, where each group has distinct features. However, an outlier is expected to occur far away in feature space from any of those groups of normal objects. This assumption may not be true all the time.
- For example, the normal objects do not share any strong patterns. Instead, they are uniformly distributed.
- The collective outliers, however, share high similarity in a small area. Unsupervised methods cannot detect such outliers effectively.
- In some applications, normal objects are diversely distributed, and many such objects do not follow strong patterns.
- For instance, in some intrusion detection and computer virus detection problems, normal activities are very diverse and many do not fall into high-quality clusters.
- In such scenarios, unsupervised methods may have a high false positive rate - they may mislabel many normal objects as outliers (intrusions or viruses in these applications), and let many actual outliers go undetected.
- Due to the high similarity between intrusions and viruses (i.e., they have to attack key resources in the target systems), modeling outliers using supervised methods may be far more effective.
- Many clustering methods can be adapted to act as unsupervised outlier detection methods. The central idea is to find clusters first, and then the data objects not belonging to any cluster are detected as outliers.

- However, such methods suffer from two issues. First, a data object not belonging to any cluster may be noise instead of an outlier. Second, it is often costly to find clusters first and then find outliers.
- It is usually assumed that there are far fewer outliers than normal objects. Having to process a large population of non-target data entries (i.e., the normal objects) before one can touch the real meat (i.e., the outliers) can be unappealing.
- The latest unsupervised outlier detection methods develop various smart ideas to tackle outliers directly without explicitly and completely finding clusters.

3. Semi-supervised Methods:

- In many applications, although obtaining some labeled examples is feasible, the number of such labeled examples is often small.
- We may encounter cases where only a small set of the normal and/or outlier objects are labeled, but most of the data are unlabeled.
- Semi-supervised outlier detection methods were developed to tackle such scenarios. Semi-supervised outlier detection methods can be regarded as applications of semi-supervised learning methods.
- For example, when some labeled normal objects are available, we can use them, together with unlabeled objects that are close by, to train a model for normal objects.
- The model of normal objects then can be used to detect outliers - those objects not fitting the model of normal objects are classified as outliers.
- If only some labeled outliers are available, semi-supervised outlier detection is trickier. A small number of labeled outliers are unlikely to represent all the possible outliers.
- Therefore, building a model for outliers based on only a few labeled outliers is unlikely to be effective.

PRACTICE QUESTIONS

Q.I Multiple Choice Questions:

1. Which is a branch of mathematics that includes the collection, analysis, interpretation, and validation of stored data?
   (a) Statistics (b) Algebra
   (c) Geometry (d) Trigonometry
2. Which data analysis allows the execution of statistical operations using quantitative approaches?
   (a) Statistical (b) Predictive
   (c) Predictive (d) Descriptive

3. Which statistics are used to draw conclusions about a population based on data observed in a sample?
   (a) descriptive (b) inferential
   (c) Both (a) and (b) (d) None of the mentioned
4. The dispersion of data values can be measured,
   (a) range (b) standard deviation
   (c) variance (d) All of the mentioned
5. Which range is calculated by finding the difference between the third quartile and the first quartile?
   (a) interquartile (b) middlequartile
   (c) meanquartile (d) modequartile
6. Which test is a statistical method to determine if two categorical variables have a significant correlation between them?
   (a) t-test (b) chi-square
   (c) z-test (d) ANOVA
7. The difference between the largest and the smallest numbers in a series of numbers defines the,
   (a) variance (b) harmonic mean
   (c) range (d) standard deviation
8. The ways to measure central tendency includes,
   (a) mean (b) median
   (c) mode (d) All of the mentioned
9. A researcher has collected the sample data 3 5 12 3 2. The mean of the sample is 5. The range is,
   (a) 1 (b) 2
   (c) 10 (d) 12
10. Which may indicate an experimental error, or it may be due to variability in the measurement?
    (a) outlier (b) variance
    (c) range (d) standard deviation
11. Which t-test determines whether the sample mean is statistically different from a known or hypothesized population mean?
    (a) two sample (b) one sample
    (c) paired sample (d) None of the mentioned

12. Which is a statistical inference test that lets us to compare multiple groups at the same time?
    (a) ANalysis Of VAriance (ANOVA) (b) t-test
    (c) z-test (d) All of the mentioned
13. Which of the following is a measure of dispersion?
    (a) percentiles (b) interquartile range
    (c) quartiles (d) None of the mentioned
14. Which testing is used to check whether a stated hypothesis is accepted or rejected?
    (a) hypothesis (b) variance
    (c) range (d) Interquartile
15. The numerical value of the standard deviation can never be,
    (a) zero (b) negative
    (c) one (d) ten
16. The value(s) that separates the critical region from the acceptance region is (are) called the, critical value(s)
    (a) critical value(s) (b) ranhe values(s)
    (c) variance values(s) (d) hypothesis values(s)
17. A binary attribute has only one of two states,
    (a) 0 (b) 1
    (c) Both 0 and 1 (d) None of the mentioned
18. If an individual data point can be considered anomalous with respect to the rest of the data, then the datum is termed as a,
    (a) point outlier (b) contextual outlier
    (c) collective outlier (d) All of the mentioned

Answers

1. (a)	2. (a)	3. (b)	4. (d)	5. (a)	6. (b)	7. (c)	8. (d)	9. (c)	10. (a)
11. (b)	12. (a)	13. (b)	14. (a)	15. (b)	16. (a)	17. (c)	18. (a)		

Q.II Fill in the Blanks:

1. _____ is a way to collect and analyze the numerical data in a large amount and finding meaningful insights from it.
2. _____ attributes may also be obtained from the discretization of numeric attributes by splitting the value range into a finite number of categories.
3. _____ statistics are mainly used for presenting, organizing, and summarizing data of a dataset.

4. The measures of _____ tendency provide a single number to represent the whole set of scores of a feature.
5. The _____ measures that are commonly used for computing the dissimilarity of objects described by numeric attributes and include the Euclidean, Manhattan and Minkowski distances.
6. _____ refers to the modal value which is the value in a series of numbers that has the highest frequency.
7. Measures of dispersion or variability indicate the degree to which scores differ around the _____.
8. The point estimation of a population parameter considers only a _____ value of a statistic.
9. _____ outlier detection detects outliers in an unlabelled data set under the assumption that the majority of the instances in the dataset are normal by looking for instances that seem to fit least to the remainder of the dataset.
10. _____ deviation is found by finding the square root of the sum of squared deviation from the mean divided by the number of observations in a given dataset.
11. The measures of _____ determine where value falls in relation to the rest of the values provided in the data distribution.
12. The _____ is mainly used when the population mean and standard deviation are given.
13. The interquartile _____ is calculated by finding the difference between the third quartile and the first quartile.
14. The _____ is the “spread of the data” which measures how far the data is spread.
15. If a collection of data points is anomalous with respect to the entire data set, it is termed as a _____ outlier.

Answers

1. Statistics	2. Ordinal	3. Descriptive	4. central
5. distance	6. Mode	7. average	8. single
9. Unsupervised	10. Standard	11. position	12. z-test
13. range	14. dispersion	15. collective	

Q.III State True or False:

1. Statistical analysis is the science of collecting, exploring and presenting large amounts of data to discover underlying patterns and trends.
2. If an individual data instance is anomalous in a specific context or condition (but not otherwise), then it is termed as a contextual outlier.

3. Statistical data analysis deals with two types of data, namely, continuous data and discrete data.
4. Inferential statistics are mainly used for presenting, organizing, and summarizing data of a dataset.
5. The harmonic mean is obtained by dividing the total number of digits with the sum of the reciprocal of all numbers.
6. In hypothesis testing the two hypotheses are the null hypothesis (denoted by H_0) and the alternative hypothesis (denoted by H_a).
7. Measures of dispersion or variability indicate the degree to which scores differ around the average.
8. Variance is calculated by finding the square of the standard deviation of given data distribution.
9. An outlier may indicate an experimental error, or it may be due to variability in the measurement.
10. Descriptive statistics summarizes the data through numbers and graphs.
11. Z-test test is a statistical method to determine if two categorical variables have a significant correlation between them.
12. The paired sample t-test is also called dependent sample t-test.
13. Data matrix structure stores the n data objects in the form of a relational table
14. The value of dispersion is one when all the data are of the same value.
15. The three most common measures of central tendency are the mean, median, and mode and each of these measures calculates the location of the central point or value.
16. The variance is a measure of variability.

Answers

1. (T)	2. (T)	3. (T)	4. (F)	5. (T)	6. (T)	7. (T)	8. (T)	9. (T)	10. (T)
11. (F)	12. (T)	13. (T)	14. (F)	15. (T)	16. (T)				

Q.IV Answer the following Questions:

(A) Short Answer Questions:

1. Define statistical data analysis?
2. Define descriptive statistics.
3. What is meant by central tendency of data?
4. Define geometric mean.
5. What is meant by mode?
6. Define outlier.

7. What is inferential statistics?
8. What is mean by range?
9. Define standard deviation?
10. Define hypothesis testing.
11. What is binary attribute and ordinal attribute.
12. Define variance.
13. Define interquartile range.
14. Give parameter estimation methods.

(B) Long Answer Questions:

1. What is statistical data analysis? Plain it role in detail.
2. Explain any two methods of dispersion for a series of data values and write the Python code for the same.
3. How is the interquartile range used as a measure of position? Find the interquartile range for the data values: 23, 18, 19, 27, 22, 31, 38, 24, 11, 16, 20.
4. What is meant by dispersion? If the standard deviation of a given dataset is equal to zero, what can we say about the data values included in the given dataset? The frequency table of the monthly salaries of 20 people is shown below. Find the range, standard deviation, and variance of the given data.

Salary (Rs.)	3500	4000	4200	4300
Frequency	5	8	5	2

5. What is meant by parameter estimation? Differentiate between a point estimate and interval estimate.
6. The amount spent by the group of 10 students in the school canteen is as follows:

   110, 117, 129, 197, 190, 100, 100, 178, 255, 790.

   Find the range and the co-efficient of the range.
7. What is mean by descriptive statistics? Explain types of descriptive statistics.
8. What are measures of central tendency? Explain each in detail.
9. Find the mean, median, mode, and range for the following list of values:

   13, 18, 13, 14, 13, 16, 14, 21, 13.
10. How to measuring data similarity and dissimilarity? Explain in detail. Describe data matrix versus dissimilarity matrix.
11. What is outlier? List types of outliers. Also explain methods for outlier detection.
12. Explain t-test and z-test with example.
13. What is dispersion? Explain range and interquartile range. Also compare them.
14. What is standard deviation and variance?
15. Find the mean of the following distribution:

x	4	6	9	10	15
f	5	10	10	7	8

16. The following table indicates the data on the number of patients visiting a hospital in a month. Find the average number of patients visiting the hospital in a day.

Number of Patients	Number of Days Visiting Hospital
0-10	2
10-20	6
20-30	9
30-40	7
40-50	4
50-60	2

❖❖❖

CHAPTER 3

Data Preprocessing

Objectives...

- To learn Concept of Data Preprocessing
- To study Data Quality
- To understand Cleaning of Data
- To learn Data Transformation, Data Reduction and Data Discretization

3.0 INTRODUCTION

- The real-world data is in the form of raw facts and unprocessed. The data is incomplete, unreliable, error-prone and/or deficient in certain behaviors or trends.
- Such data needs to be preprocessed for converting it to a meaningful form. The data preprocessing is required to improve the quality of data.
- Data preprocessing is the method of collecting raw data and translating it into usable/meaningful information.
- Data preparation takes place in usually following two phases for any data science project:

  Phase 1 (Data Preprocessing): It is the task of transforming raw data to be ready to be fed into an algorithm. It is a time-consuming yet important step that cannot be avoided for the accuracy of results in data analysis.

  Phase 2 (Data Wrangling/Data Munging): It is the task of converting data into a feasible format that is suitable for the consumption of the data. It typically follows a set of common steps like extracting data from various data sources, parsing data into predefined data structures and storing the converted data into a data sink for further analysis.
- The various preprocessing steps (See Fig. 3.1) include data cleaning, data integration, data transformation, data reduction, and data discretization.
- All these preprocessing steps are very essential for transforming the raw and error-prone data into a useful and valid format.
- Once, the data preprocessing operations are all completed, the data will be free from all possible data error types.

- Such data will be transformed into a format that will be easy to use for data analysis and data visualization.

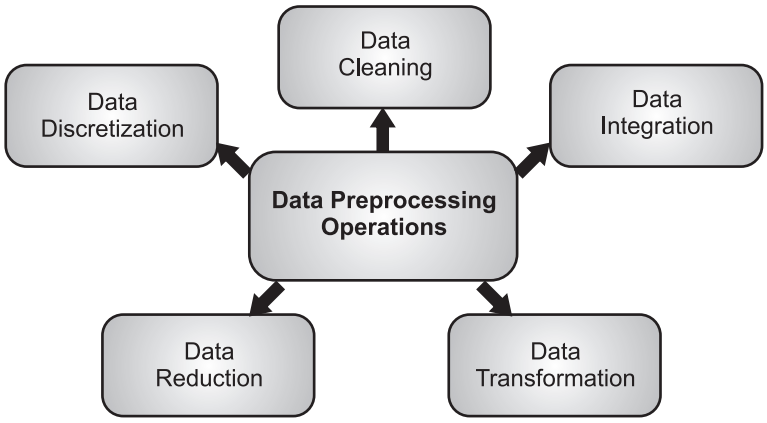


Fig. 3.1: Operations on Data Processing

3.1 DATA OBJECTS AND ATTRIBUTES TYPES

- Data are the basic units of information that are collected through observation. Data are a set of values of qualitative or quantitative variables about one or more persons or objects, while a datum (singular of data) is a single value of a single variable.
- So, data is information that has been translated into a form that is efficient for movement or processing.
- For preprocessing of data as well as for exploratory data analytics, it is important to know the type of data that needs to be dealt with.
- Knowing the type of data helps in choosing the right statistical measure, the appropriate data visualization and so on.
- There are mainly two types of data namely, Categorical data and Numerical data.
    - **Categorical Data** is non-numeric and consists of text that can be coded as numeric. However, these numbers do not represent any fixed mathematical notation or meaning for the text and are simply assigned as labels or codes. Categorical data can be of two types namely, Nominal data is used to label variables without providing any quantitative value and Ordinal data of data is used to label variables that need to follow some order.
    - **Numerical Data:** This type of data is numeric and it usually follows an order of values. These quantitative data represent fixed values and can be of two types namely, Interval data follows numeric scales in which the order and exact differences between the values is considered and Ratio data also follows numeric scales and has an equal and definitive ratio between each data.
- Data is a collection of data objects and their attributes.

3.1.1 Data Attributes

- An attribute is a property or characteristic of an object. A data attribute is a single-value descriptor for a data object. For example, eye color of a person, name of a student, etc.
- Attribute is also known as variable, field, characteristic, or feature. The distribution of data involving one attribute (or variable) is called univariate. A bivariate distribution involves two attributes, and so on.

3.1.2 Data Objects

- A collection of attributes describe an object. Data objects can also be referred to as samples, examples, instances, case, entity, data points or objects.
- If the data objects are stored in a database, they are data tuples. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes (See Table 3.1).

Table 3.1: Branch Table

Attributes

Data Objects

B_ID	Name	Address
B1	ATTRONICS@Delhi	Delhi
B2	ATTRONICS-Gwalior	Gwalior
B3	ATTRONICS_4_Nashik	Nashik
B4	ATTRONICS_Pune	Pune
B5	ATTRONICS&Hydrabad	Hydrbad
B6	ATTRONICS@Jaipur	Jaipur

- Consider the case study of the company named ATTRONICS is described by the relation tables: customer, item, employee, and branch.
- The headers of the tables described here are shown as per followings:
  - customer (Cust_ID, Name, Address, Age, Occupation, Annual_Income, Credit_Information, Category)
  - item (Item_ID, Brand, Category, Type, Price, Place_Made, Supplier, Cost)
  - employee (Emp_ID, Name, Category, Group, Salary, Commission)
  - branch (Br_ID, Name, Address)
  - purchases (Trans_ID, Cust_ID, Emp_ID, Date, Time, Method_Paid, Amount)

- items sold (Trans_ID, Item_ID, Qty)
- works at (emp_ID, Br_ID)

- The relation customer consists of a set of attributes describing the customer information, including a unique customer identity number (Cust_ID), Cust_Name, Address, Age, Occupation, Annual_Income, Credit_Information and Category.
- Similarly, each of the relations Item, Employee and Branch (See Table 3.1) consists of a set of attributes describing the properties of these entities. Tuples/rows in the table are known as data objects.

3.1.3 Types of Data Attributes

- There are broadly four types of attributes namely, Nominal attribute, Binary attribute, Ordinal attribute and Numeric attributes.

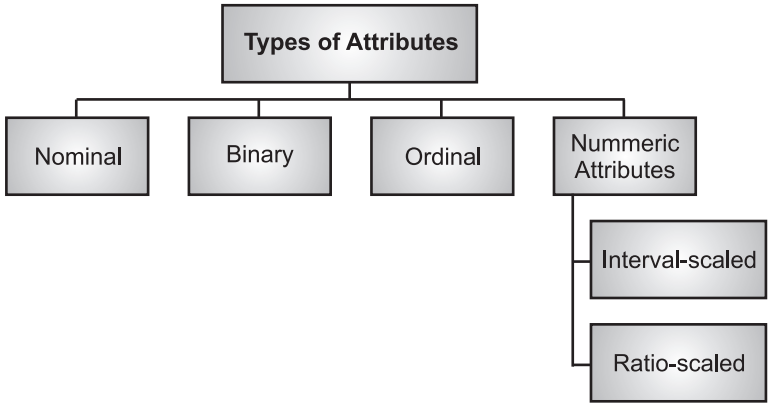


Fig. 3.2: Types of Attributes

- The attributes in Fig. 3.2 are explained below:

1. Nominal Attribute:

- Nominal means “relating to names.” The values of a nominal attribute are symbols or names of things.
- Each value in nominal attribute represents some kind of category, code, or state, and so nominal attributes are also referred to as categorical.
- For example, ID, eye color, zip codes. The values do not have any meaningful order. In computer science, the values are also known as enumerations.
- Branch relation of ATTRONICS company has attributes like B_ID and Name which are come under Nominal type.

2. Binary Attributes:

- A binary attribute is a nominal attribute with only two categories or states namely, 0 or 1 where 0 typically means that the attribute is absent and 1 means that it is present.
- Binary attributes are referred to as Boolean if the two states correspond to true and false.
- Examples: The attribute medical test is binary, where a value of 1 means the result of the test for the patient is positive, while 0 means the result is negative.

3. Ordinal Attributes:

- An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them, but the magnitude between successive values is not known.
- Examples: rankings (e.g., taste of potato chips on a scale from 1- 10), grades, height in {tall, medium, short}.

4. Numeric Attributes:

- A numeric attribute is quantitative; that is, it is a measurable quantity, represented in integer or real values.
- Numeric attributes can be interval-scaled or ratio-scaled.

  (i) Interval-scaled Attributes:

  - Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be positive, 0, or negative.
  - Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the difference between values.
  - Examples: Temperatures in Celsius or Fahrenheit.
  - Purchase relation of ATTRONICS company/organization has attributes like date which are come under Interval-scaled type.

  (ii) Ratio-scaled Attribute:

  - The ratio-scaled attribute is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value.
  - In addition, the values are ordered and we can also compute the difference between values, as well as the mean, median, and mode.
  - Examples: temperature in Kelvin, length, time, counts.
- Table 3.2 show the nature of attribute types, which helps to appropriate use of attribute based on its types.

Table 3.2: Attribute types with its compatibility with operations

Operation	Nominal	Ordinal	Interval	Ratio
Equality	√	√	√	√
Order		√	√	√
Add/Subtract			√	√
Multiply/Divide				√
Mode	√	√	√	√
Median		√	√	√
Arithmetic Mean			√	√
Geometric Mean				√

3.1.4 Discrete versus Continuous Attributes

- There are many ways to organize attribute types. Many machine learning algorithms specially, classification algorithms advocate the attributes categorization as being either discrete or continuous.
- A discrete attribute has a finite or countably infinite set of values, which may or may not be represented as integers.
- The attributes hair color, smoker, medical test, and drink size each have a finite number of values, and so are discrete.
- Discrete attributes may have numeric values, such as 0 and 1 for binary attributes or, the values 0 to 110 for the attribute age.
- An attribute is countably infinite if the set of possible values is infinite but the values can be put in a one-to-one correspondence with natural numbers.
- For example, the attribute customer ID is countably infinite. The number of customers can grow to infinity, but in reality, the actual set of values is countable, (where the values can be put in one-to-one correspondence with the set of integers). Zip codes are another example.
- If an attribute is not discrete, it is continuous. In practice, real values are represented using a finite number of digits. Continuous attributes are typically represented as floating-point variables.
- For example consider the employee relation of ATTRONICS company:

  employee (Emp_ID, Name, Category, Group, Salary, Commission)
- In employee relation EMP_ID, Name, Category and Group are discrete attributes. However salary and commission are continuous attributes.

3.2 DATA QUALITY: WHY PREPROCESS THE DATA?

- Data have quality if they satisfy the requirements of the intended use. Data quality can be defined as, “the ability of a given data set to serve an intended purpose”.
- Data preprocessing is responsible for maintain the quality of data. The phrase "garbage in, garbage out" is particularly applicable to such projects.
- Data-collection methods are often loosely controlled, resulting in out-of-range values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes) and missing values, etc.
- Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running any analysis.
- If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult.
- Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes cleaning, transformation, reduction and data discretization.
- There are many factors comprising data quality, including accuracy, completeness, consistency, timeliness, believability, and interpretability.
- There are many reasons for inaccurate, incomplete, and inconsistent in real-world databases and data warehouses.

1. Inaccuracy:

- Inaccurate data means having incorrect attribute values. Consider the following items_sold relation.

Table 3.3: items_sold Relation

Trans_ID	Item_ID	Qty
T1	IT04	24
T2	IT24	−1012
T3	IT58	78
T4	IT16	100

- In above Table 3.3 of items_sold relation of ATTRONICS company has tuple for transaction T2 with Item_ID IT24 and quantity −1012.
- In this tuple, quantity of item sold seems incorrect which may have typing error or some garbage entry during the auto transmission of data.

- There are many reasons, which may responsible for inaccurate data:
  (i) The data collection instruments used may be faulty.
  (ii) There may have been human or computer errors occurring at data entry.
  (iii) Users may purposely submit incorrect data values for mandatory fields when they do not wish to submit personal information (e.g., by choosing the default value "January 1" displayed for birthday). This is known as disguised missing data.
  (iv) Errors in data transmission can also occur.
  (v) There may be technology limitations such as limited buffer size for coordinating synchronized data transfer and consumption.

2. Inconsistency:

- Incorrect and redundant data may result from inconsistencies in naming conventions or data codes, or inconsistent formats for input fields (e.g., date).
- Duplicate tuples also require data cleaning.

3. Incompleteness:

- Consider the instance of branch relation of ATTRONICS company.

Table 3.4: Branch Relation

Cust_ID	Name	Address	Age	Occupation	Category
C01	Ravindra Singh	Delhi	45	Service	Platinum
C02	Gouri Salunkhe	Gwalior	34	Business of Cloths	Gold
C03	Sunil Joshi	Nashik	23		Silver
C04	Ravi Deshmukh	Pune		Grocery shop	Gold
C05	Rohit Kulkarni	Hyderabad	41	Service	Gold

- In above Table 3.4 occupation of customer C03 is not available. Also age of customer C04 is also missing. Incomplete data can occur for a number of reasons:
  (i) Attributes of interest may not always be available, such as customer information for sales transaction data.
  (ii) Other data may not be included simply because they were not considered important at the time of entry.
  (iii) Relevant data may not be recorded due to a misunderstanding or because of equipment malfunctions.
  (iv) Data that were inconsistent with other recorded data may have been deleted. Furthermore, the recording of the data history or modifications may have been overlooked.
  (v) Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.

4. Timeliness:

- Timeliness also affects data quality. Failure in to follow the schedule of record submission may be occur due many reasons like:

  (i) At the time of record submission numerous corrections and adjustments occurs.
  (ii) Technical error during data uploading.
  (iii) Unavailability of responsible person.

- For a period of time following each month, the data stored in the database are incomplete.
- However, once all of the data are received, it is correct. The fact that the month-end data are not updated in a timely fashion has a negative impact on the data quality.

Problems with Believability and Interpretability:

- Two other factors affecting data quality are believability and interpretability.
- Believability reflects how much the data are trusted by users, while interpretability reflects how easy the data are understood.
- Suppose that a database, at one point, had several errors, all of which have since been corrected.
- The past errors, however, had caused many problems for sales department users, and so they no longer trust the data.
- The data also use many accounting codes, which the sales department does not know how to interpret.
- Even though the database is now accurate, complete, consistent, and timely, users may regard it as of low quality due to poor believability and interpretability.

3.3 DATA MUNGING / WRANGLING OPERATIONS

- Data wrangling is the task of converting data into a feasible format that is suitable for the consumption of the data.
- The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data.
- Data munging process includes operations such as Cleaning Data, Data Transformation, Data Reduction and Data Discretization.
- Data munging or wrangling refers to preparing data for a dedicated purpose - taking the data from its raw state and transforming and mapping into another format, normally for use beyond its original intent and can be used for it more appropriate and valuable for a variety of downstream purposes such as analytics.

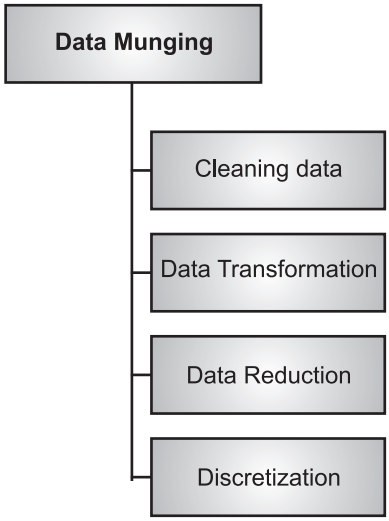


Fig. 3.3: Data Munging Operations

3.3.1 Data Cleaning

- Real-world data tend to be incomplete, noisy, and inconsistent. This dirty data can cause an error while doing data analysis. Data cleaning is done to handle irrelevant or missing data.
- Data cleaning also known as data cleansing or scrubbing. Data is cleaned by filling in the missing values, smoothing any noisy data, identifying and removing outliers, and resolving any inconsistencies.
- Data cleaning is the process of correcting or removing incorrect, incomplete or duplicate data within a dataset.

3.3.1.1 Missing Values

- The raw data that is collected for analyzing usually consists of several types of errors that need to be prepared and processed for data analysis.
- Some values in the data may not be filled up for various reasons and hence are considered missing.
- If in database, some of the tuples have no recorded value for several attributes then it will become difficult to proceed with data.
- For example, in Table 3.4 occupation of customer C03 is not available, also age of customer C04 is missing.
- In some cases, missing data arises because data was never gathered in the first place for some entities. Data analyst needs to take appropriate decision for handling such data.

- In general, there can be three cases of missing data s explained below:
  - **Missing Completely At Random (MCAR),** which occurs due to someone forgetting to fill in the value or have lost the information.
  - **Missing At Random Data (MAR),** which occurs due to someone purposely not filling up the data mainly due to privacy issues.
  - **Missing Not At Random (MNAR),** which occurs as data maybe not available.
- Analyst can take following actions for handling such missing values like:

1. Ignore the Tuple:

- This is usually done when the class label is missing, (assuming the mining task involves classification).
- This method is not very effective, unless the tuple contains several attributes with missing values.
- It is especially poor when the percentage of missing values per attribute varies considerably.
- By ignoring the tuple, we do not make use of the remaining attributes' values in the tuple. Such data could have been useful to the task at hand.

2. Fill in the Missing Value Manually:

- In general, this approach is time consuming and may not be feasible given a large data set with many missing values.

3. Use a Global Constant to Fill in the Missing Value:

- Replace all missing attribute values by the same constant such as a label like "Unknown" or -∞.
- If missing values are replaced by, say, "Unknown," then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common - that of "Unknown." Hence, although this method is simple, it is not foolproof.

4. Use a Measure of Central Tendency for the Attribute (e.g., the Mean or Median) to Fill in the Missing Value.

- For this, a particular column is selected for which the central value (say, median) is found. Then the central value is replaced with all the NaN values of that particular column.
- Instead of the median, the mean or mode value can also be used for the same. Replacing NaN values with either mean, mode or median is considered as a statistical approach of handling the missing values.

- For normal (symmetric) data distributions, the mean can be used, while skewed data distribution should employ the median.

```
#Program for Handling Missing Values
import pandas as pd
import numpy as np
#Creating a DataFrame with Missing Values
dframe = pd.DataFrame(np.random.randn(5, 3), index=['p', 'r',
                        't', 'u','w'], columns=['C0L1', 'C0L2', 'C0L3'])
dframe = dframe.reindex(['p', 'q', 'r', 's', 't', 'u', 'v', 'w'])
print("\n Reindexed Data Values")
print("-------------------------")
print(dframe)
#Method 1 - Filling Every Missing Values with 0
print("\n\n Every Missing Value Replaced with '0':")
print("-------------------------------------------")
print(dframe.fillna(0))
#Method 2 - Dropping Rows Having Missing Values
print("\n\n Dropping Rows with Missing Values:")
print("----------------------------------------")
print(dframe.dropna())
#Method 3 - Replacing missing values with the Median
medianval = dframe['C0L1'].median()
dframe['C0L1'].fillna(medianval, inplace=True)
print("\n\n Missing Values for Column 1 Replaced with
                                                    Median Value:")
print("-------------------------------------------------------------")
print(dframe)
```

- For above program shows the output for handling missing values is displayed next.
- Initially, the dataset with missing values is shown and then the various three methods used for handling or replacing missing values (NaN values) are also displayed

```
 Reindexed Data Values
-------------------------
       C0L1      C0L2      C0L3
p  0.445611  0.934896  1.292210
q       NaN       NaN       NaN
r -0.564345 -0.153329  0.594044
s       NaN       NaN       NaN
t -1.306026 -0.797253 -0.236945
u  0.366105  0.878192  1.876519
v       NaN       NaN       NaN
w -0.958643  0.582960 -1.164970


 Every Missing Value Replaced with '0':
-----------------------------------------
       C0L1      C0L2      C0L3
p  0.445611  0.934896  1.292210
q  0.000000  0.000000  0.000000
r -0.564345 -0.153329  0.594044
s  0.000000  0.000000  0.000000
t -1.306026 -0.797253 -0.236945
u  0.366105  0.878192  1.876519
v  0.000000  0.000000  0.000000
w -0.958643  0.582960 -1.164970


 Dropping Rows with Missing Values:
------------------------------------
       C0L1      C0L2      C0L3
p  0.445611  0.934896  1.292210
r -0.564345 -0.153329  0.594044
t -1.306026 -0.797253 -0.236945
u  0.366105  0.878192  1.876519
w -0.958643  0.582960 -1.164970


 Missing Values for Column 1 Replaced with Med
-----------------------------------------------
       C0L1      C0L2      C0L3
p  0.445611  0.934896  1.292210
q -0.564345       NaN       NaN
r -0.564345 -0.153329  0.594044
s -0.564345       NaN       NaN
t -1.306026 -0.797253 -0.236945
u  0.366105  0.878192  1.876519
v -0.564345       NaN       NaN
w -0.958643  0.582960 -1.164970
```

- One can also use the fillna() function to fill NaN values with the value from the previous row or the next row.

5. **Use the Attribute Mean or Median for all Samples belonging to the same Class as the given Tuple:**

- All the customers who belong to the same class, their missing attribute value can be replaced by mean of that class only.

6. **Use the most probable value to fill in the missing value:**

- This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction. So prediction algorithms can be utilized to find missing values.
- For example income of any customer can be predicted by training a decision tree with the help of remaining customer data and value for missing attribute can be identified.

3.3.1.2 Noisy Data

- The noisy data contains errors or outliers. For example, for stored employee details, all values of the age attribute are within the range 22-45 years whereas one record reflects the age attribute value as 80.
- There are times when the data is not missing, but it is corrupted for some reason. This is, in some ways, a bigger problem than missing data.
- Data corruption may be a result of faulty data collection instruments, data entry problems, or technology limitations.

Table 3.5: items_sold Relation

Trans_ID	Item_ID	Qty
T01	IT04	24
T02	IT24	-1012234
T03	IT58	78
T04	IT16	100

- In items_sold relation of ATTRONICS company, transaction T02 has incorrect value of items quatity, which treated as noisy data.
- There is no single technique to take care of missing data, there is no one way to remove noise, or smooth out the noisiness in the data.
- Following topics explained some causes of noisy data:

Duplicate Entries:

- Duplicate entries in a dataset are big problem, before start analysis it is suppose to identify such duplicity and handle it properly.
- In those cases, we usually want to compact them into one entry, adding an additional column that indicates how many unique entries there were. In other cases, the duplication is purely a result of how the data was generated.

- For example, it might be derived by selecting several columns from a larger dataset, and there are no duplicates if we count the other columns.
- Data duplication can also occur when you are trying to group data from various sources. This is a common issue with organizations that use webpage scraping tools to accumulate data from various websites.
- Duplicate entries consequences many problems like it leads data redundancy, inconsistencies, degrading data quality, which impact on data analysis outcomes.

Multiple Entries for a Single Entity:

- In real world databases, each entity logically corresponds to one row in the dataset, but some entities are repeated multiple times with different data.
- The most common cause of this is that some of the entries are out of date, and only one row is currently correct.
- Another case where there can be multiple entries is if, for some reason, the same entity is occasionally processed twice by whatever gathered the data.

NULLs:

- If value of an attribute is not known then it considered as NULL.

Table 3.6: Customer Relation

Cust_ID	Name	Address	Age	Occupation	Category
C01	Ravindra Singh	Delhi	45	Service	Platinum
C02	Gouri Salunkhe	Gwalior	34	Business of Cloths	Gold
C03	Sunil Joshi	Nashik	23	NULL	Silver
C04	Ravi Deshmukh	Pune	NULL	Grocery shop	Gold
C05	Rohit Kulakarni	Hyderabad	41	Service	Gold
C06	Raj Patil	Nagpur	25	Medical shop	Silver
C07	Pritam Shah	Banglore	32	Service	Gold
C08	Sarwaarth Chawre	Satara	51	LIC agent	Silver
C09	Ishan Basin	Ahmadabad	33	Grocery shop	Gold
C10	Harshal Patil	Bhopal	21	Business	Platinum

- In Table 3.6, occupation of customer C03 and age of customer C04 is not available. NULLs can arise because the data collection process was failed in some way.
- When it comes time to do analytics, NULLs cannot be processed by many algorithms. In these cases, it is often necessary to replace the missing values with some reasonable proxy.

- What we will see most often is that it is guessed from other data fields, or you simply putting the mean of all the non- null values.
- For example mean of age attribute for all Gold category customer is 35. So for customer C04, Null value of age can be replaced with 35.Also in some cases, the NULL values arise because that data was never collected.

Huge Outliers:

- An outlier is a data point that differs significantly from other observations.
- They are extreme values that deviate from other observations on data; they may indicate variability in a measurement, experimental errors or a novelty.
- Most common causes of outliers on a data set:
  1. Data entry errors (human errors).
  2. Measurement errors (instrument errors).
  3. Experimental errors (data extraction or experiment planning/executing errors).
  4. Intentional (dummy outliers made to test detection methods).
  5. Data processing errors (data manipulation or data set unintended mutations).
  6. Sampling errors (extracting or mixing data from wrong or various sources).
  7. Natural (not an error, novelties in data).
- Sometimes, a massive outlier in the data is there because there was truly an unusual event. How to deal with that depends on the context. Sometimes, the outliers should be filtered out of the dataset.
- For example, we are usually interested in predicting page views by humans. A huge spike in recorded traffic is likely to come from a bot attack, rather than any activities of humans.
- In other cases, outliers just mean missing data. Some storage systems don't allow the explicit concept of a NULL value, so there is some predetermined value that signifies missing data. If many entries have identical, seemingly arbitrary values, then this might be what's happening.

Out-of-Date Data:

- In many databases, every row has a timestamp for when it was entered. When an entry is updated, it is not replaced in the dataset; instead, a new row is put in that has an up-to-date timestamp.
- For this reason, many datasets include entries that are no longer accurate and only useful if you are trying to reconstruct the history of the database.

Artificial Entries:

- Many industrial datasets have artificial entries that have been deliberately inserted into the real data.
- This is usually done for purposes of testing the software systems that process the data.

Irregular Spacings:

- Many datasets include measurements taken at regular spacings. For example, you could have the traffic to a website every hour or the temperature of a physical object measured at every inch.
- Most of the algorithms that process data such as this assume that the data points are equally spaced, which presents a major problem when they are irregular.
- If the data is from sensors measuring something such as temperature, then typically we have to use interpolation techniques to generate new values at a set of equally spaced points.
- A special case of irregular spacings happens when two entries have identical timestamps but different numbers. This usually happens because the timestamps are only recorded to finite precision.
- If two measurements happen within the same minute, and time is only recorded up to the minute, then their timestamps will be identical.

Formatting Issues:

- Various formatting issues are explained below:

Formatting Is Irregular between Different Tables/Columns

- This happens a lot, typically because of how the data was stored in the first place.
- It is an especially big issue when joinable/groupable keys are irregularly formatted between different datasets.

Extra Whitespaces:

- A white space is the blank space among the text. An appropriate use of white spaces will increase readability and focus the readers' attention.
- For example, within a text, white spaces split big chunks of text into small paragraphs which makes them easy to understand.
- String with and without blank spaces is not the same. "ABC" != " ABC" these two ABCs are not equal, but the difference is so small that you often don't notice.
- Without the quotes enclosing the string you hardly would ABC != ABC. But the computer programs are incorruptible in the interpretation and if these values are a merging key, we would receive an empty result.
- Blank strings, spaces, and tabs are considered as the empty values represented as NaN. Sometimes it consequences an unexpected results.
- Also, even though the white spaces are almost invisible, pile millions of them into the file and they will take some space and they may overflow the size limit of your database column leading to an error.

Irregular Capitalization and Inconsistent Delimiters:

- Data set may have problems of irregular capitalization of text data also a dataset will have a single delimiter, but sometimes, different tables will use different ones.
- Mostly use delimiters are Commas, Tabs and Pipes (the vertical line "|").

Irregular NULL Format:

- There are a number of different ways that missing entries are encoded into CSV files, and they should all be interpreted as NULLs when the data is read in.
- Some popular examples are the empty string "", "NA," and "NULL." Occasionally, you will see others such as "unavailable" or "unknown" as well.

Invalid Characters:

- Some data files will randomly have invalid bytes in the middle of them.
- Some programs will throw an error if we try to open up anything that isn't valid text. In these cases, we may have to filter out the invalid bytes.

Weird or Incompatible Date and Times:

- Date and times are one of the most frequently mangled types of data field. Some of the date formats we will see are as follows:

  June 1, 2020

  JUN 1, '20

  2020-06-01

- There is an important way that dates and times are different from other formatting issues.
- Most of the time we have two different ways of expressing the same information, and a perfect translation is possible from the one to the other. But with dates and times, the information content itself can be different.
- For example, we might have just the date, or there could also be a time associated with it. If there is a time, does it go out to the minute, hour, second, or something else? What about time zones?

3.3.2 Data Transformation

- Data transformation is the process of converting raw data into a format or structure that would be more suitable for data analysis.
- Data transformation is a data preprocessing technique that transforms or consolidates the data into alternate forms appropriate for mining.
- Data transformation is a process of converting raw data into a single and easy-to-read format to facilitate easy analysis.

- Data transformation is the process of changing the format, structure, or values of data. The choice of data transformation technique depends on how the data will be later used for analysis.
- For example, standardizing salutations or street names, date and time format changing are related with data format transformation.
- Renaming, moving, and combining columns in a database are related with structural transformation of data.
- Transformation of values of data is relevant with transformed the data values into a range of values that are easier to be analyzed. This is done as the values for different information are found to be in a varied range of scales.
- For example, for a company, age values for employee scan be within the range of 20-55 years whereas salary values for employees can be within the range of Rs. 10,000 – Rs. 1,00,000.
- This indicates one column in a dataset can be more weighted compared to others due to the varying range of values. In such cases, applying statistical measures for data analysis across this dataset may lead to unnatural or incorrect results.
- Data transformation is hence required to solve this issue before applying any analysis of data.
- Various data transformation techniques are used during data preprocessing. The choice of data transformation technique depends on how the data will be later used for analysis.
- Some of these important standard data preprocessing techniques are Rescaling, Normalizing, Binarizing, Standardizing, Label and One Hot Encoding.

Benefits of Data Transformations:

1. Data is transformed to make it better-organized. Transformed data may be easier for both humans and computers to use.
2. Properly formatted and validated data improves data quality and protects applications from potential landmines such as null values, unexpected duplicates, incorrect indexing, and incompatible formats.
3. Data transformation facilitates compatibility between applications, systems, and types of data.

- Data used for multiple purposes may need to be transformed in different ways. Many strategies are available for data transformation in Data preprocessing.
- Fig. 3.4 shows some of the strategies for data transformation.

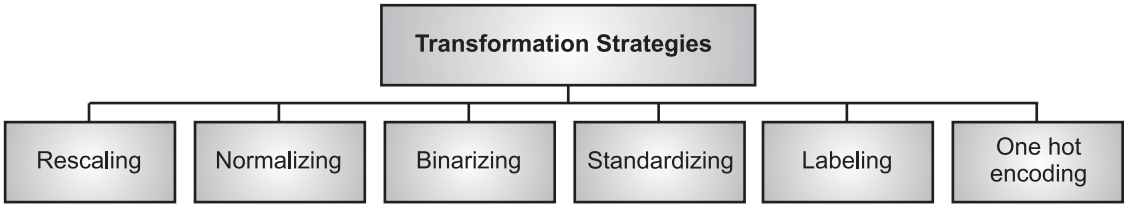


Fig. 3.4: Transformation Strategies

- Some of the strategies for data transformation include the following:

1. Rescaling:

- Rescaling means transforming the data so that it fits within a specific scale, like 0-100 or 0-1. Rescaling of data allows scaling all data values to lie between a specified minimum and maximum value (say, between 0 and 1).
- When the data encompasses attributes with varying scales, many statistical or machine learning techniques prefer rescaling the attributes to fall within a given scale.
- Scaling variables helps to compare different variables on equal footing.

2. Normalizing:

- The measurement unit used can affect the data analysis. For example, changing measurement units from meters to inches for height or from kilograms to pounds for weight, may lead to very different results.
- In general, expressing an attribute in smaller units will lead to a larger range for that attribute and thus tend to give such an attribute greater effect or "weight."
- To help avoid dependence on the choice of measurement units, the data should be normalized.
- Normalization scaled the attribute data so as to fall within a smaller range, such as 0.0 to 1.0 or -1.0 to 1.0
- Normalization ensures that the attributes values which we are using in computations are not affected by trivial variations like height, width, scaling factors, orientations etc.
- Normalizing the data attempts to give all attributes an equal weight.

3. Binarizing:

- It is the process of converting data to either 0 or 1 based on a threshold value.
- All the data values above the threshold value are marked 1 whereas all the data values equal to or below the threshold value are marked as 0.
- Data binarizing is done prior to data analysis in many cases such as, dealing with crisp values for the handling of probabilities and adding new meaningful features in the dataset.

4. Standarizing:

- Standardization also called mean removal. It is the process of transforming attributes having a Gaussian distribution with differing mean and standard deviation values into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.
- In other words, Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation.
- This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.
- Standardization of data is done prior to data analysis in many cases such as, in the case of linear discriminate analysis, linear regression, and logistic regression.

```
#Program for Data Transformation
import pandas as pd
import numpy as np
from sklearn import preprocessing
import scipy.stats as s
#Creating a DataFrame
d1 = {'C0L1':[2,4,8,5],'C0L2':[14,4,9,3],'C0L3':[24,36,-13,10]}
df1 = pd.DataFrame(d1)
print("\n ORIGINAL DATA VALUES")
print("------------------------")
print(df1)
#Method 1: Rescaling Data
print("\n\n Data Scaled Between 0 to 1")
data_scaler = preprocessing.MinMaxScaler(feature_range = (0, 1))
data_scaled = data_scaler.fit_transform(df1)
print("\n Min Max Scaled Data")
print("-----------------------")
print(data_scaled.round(2))
#Method 2: Normalization rescales such that sum of each row is 1.
dn1 = preprocessing.normalize(df1, norm = 'l1')
print("\n L1 Normalized Data")
```

```
print("----------------------")
print(dn1.round(2))
#Method 3: Binarize Data (Make Binary)
data_binarized = preprocessing.Binarizer(threshold=5).transform(df1)
print("\n Binarized data")
print("-----------------")
print(data_binarized)
#Method 4: Standardizing Data
print("\n Standardizing Data")
print("----------------------")
X_train = np.array([[ 2., -1., 1.],[ 0., 0., 2.],[ 0., 2., -1.]])
print("Orginal Data \n", X_train)
print("\n Initial Mean : ", s.tmean(X_train).round(2))
print("Initial Standard Deviation : " ,round(X_train.std(),2))
X_scaled = preprocessing.scale(X_train)
X_scaled.mean(axis=0)
X_scaled.std(axis=0)
print("\n Standardized Data \n", X_scaled.round(2))
print("\n Scaled Mean : ",s.tmean(X_scaled).round(2))
print("Scaled Standard Deviation : ",round(X_scaled.std(),2))
```

- The output of the above program is given next. The original dataset values are at first displayed that have three columns COL1, COL2 and COL3 and four rows.
- The dataset is then rescaled to between 0 and 1 and the transformed rescaled data is displayed next.
- Next, the L1 normalization of data is done and the transformed normalized data is displayed next.
- Again, the data is binarized to change the values of all data to either 0 or 1 and the binarized data is also displayed as output.
- Lastly, for a new dataset, the values are standardized to obtain the mean value of 0 and the standard deviation value of 1. The transformed standardized data is displayed at the last.

```
 ORIGINAL DATA VALUES
------------------------
   COL1  COL2  COL3
0     2    14    24
1     4     4    36
2     8     9   -13
3     5     3    10


 Data Scaled Between 0 to 1

 Min Max Scaled Data
-----------------------
[[0.   1.   0.76]
 [0.33 0.09 1.  ]
 [1.   0.55 0.  ]
 [0.5  0.   0.47]]

 L1 Normalized Data
----------------------
[[ 0.05  0.35  0.6 ]
 [ 0.09  0.09  0.82]
 [ 0.27  0.3  -0.43]
 [ 0.28  0.17  0.56]]

 Binarized data
-----------------
[[0 1 1]
 [0 0 1]
 [1 1 0]
 [0 0 1]]

 Standardizing Data
----------------------
Orginal Data
 [[ 2. -1.  1.]
 [ 0.  0.  2.]
 [ 0.  2. -1.]]

 Initial Mean :  0.56
Initial Standard Deviation :  1.17

 Standardized Data
 [[ 1.41 -1.07  0.27]
 [-0.71 -0.27  1.07]
 [-0.71  1.34 -1.34]]

 Scaled Mean :  0.0
Scaled Standard Deviation :  1.0
```

- The above program illustrates how data transformation techniques – rescaling, normalizing, binarizing, and standardizing data - are applied for a given dataset.

5. Label Encoding:

- The label encoding process is used to convert textual labels into numeric form in order to prepare it to be used in a machine - readable form.
- The labels are assigned a value of 0 to (n–1) where n is the number of distinct values for a particular categorical feature.

- The numeric values are repeated for the same label of that attribute. For instance, let us consider the feature 'gender' having two values - male and female.
- Using label encoding, each gender value will be marked with unique numerical values starting with 0. Thus males will be marked 0, females will be marked 1.

6. One Hot Coding:

- One hot encoding refers to splitting the column which contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains "0" or "1" corresponding to which column it has been placed.
- Many data science algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric.
- Categorical data must be converted to a numerical form before to proceed for data analysis.
- One hot coding is used for categorical variables where no ordinal relationship exists among the variable's values.
- For example consider the variable named "color", It may have value red, green, blue, etc. which have no specific order. In other words different category of color (Red, green, blue etc.) do not have any specific order.
- As a first step, each unique category value is assigned an integer value. For example, "red" is 1, "green" is 2, and "blue" is 3.
- But assigning a numerical value creates a problem because the integer values have a natural ordered relationship between each other.
- But here we do not want or to assign any order to color categories. In this case, a one-hot encoding can be applied to the integer representation.
- This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

  For example,

red	green	blue
1	0	0
0	1	0
0	0	1

- In the "color" variable example, there are 3 categories and therefore 3 binary variables are needed.
- A "1" value is placed in the binary variable for the color and "0" values for the other colors. This encoding method is very useful for encoding categorical variables where order of variable's value not matter.
- Following program shows the Python code for data transformations using the label encoding technique and the one-hot encoding technique.

- For each two technique, two different datasets have been used and accordingly for each categorical attribute, the values are encoded with numerical values.

```
#Program for Data Transformation using Encoding
import pandas as pd
from sklearn import preprocessing
#Create a Dataframe
data=d={'Name':['Ram','Meena','Sita','Richa','Manoj','Shyam','Pratik',
   'Sayali','Nikhil','Prasad'],'Gender':['Male','Female','Female','Female',
                     'Male','Male','Male','Female','Male','Male']}
dframe = pd.DataFrame(data)
print(dframe)
#Method of Label Encoding
print("\n LABEL ENCODING")
print("------------------")
print("\n Gender Encoding - Male : 0, Female - 1")
label_encoder = preprocessing.LabelEncoder()
#Encode labels
dframe['Gender']= label_encoder.fit_transform(dframe['Gender'])
print("Distinct Coded Gender Values : ", dframe['Gender'].unique())
print("\n",dframe)
#Create another Dataframe (DataFrame)
data={'Name':['Maharashtra','Kerala','Haryana','Gujarat','Goa',
                     'Chhattisgarh','Assam','Punjab','Sikkim','Rajasthan']}
dframe = pd.DataFrame(data)
print("\n ONE HOT ENCODING")
print("---------------------")
print("\n", dframe)
leb=preprocessing.LabelEncoder()
p=dframe.apply(leb.fit_transform)
# 1. INSTANTIATE
enc = preprocessing.OneHotEncoder()
# 2. FIT
enc.fit(p)
# 3. Transform
onehotlabels = enc.transform(p).toarray()
print("\n",onehotlabels)
```

- The output of the above program is shown below. In the program, for label encoding, the gender attribute has been coded as – Male: 0, Female: 1.
- Again for one-hot encoding, each vegetable has been marked 1 in a particular column and rest of the other vegetables are marked 0 in the same column.

```
      Name  Gender
0      Ram    Male
1    Meena  Female
2     Sita  Female
3    Richa  Female
4    Manoj    Male
5    Shyam    Male
6   Pratik    Male
7   Sayali  Female
8  Nikhil    Male
9  Prasad    Male
```

```
 LABEL ENCODING
------------------

 Gender Encoding - Male : 0, Female - 1
Distinct Coded Gender Values :  [1 0]

      Name  Gender
0     Ram       1
1   Meena       0
2    Sita       0
3   Richa       0
4   Manoj       1
5   Shyam       1
6  Pratik       1
7  Sayali       0
8  Nikhil       1
9  Prasad       1

 ONE HOT ENCODING
--------------------

            Name
0   Maharashtra
1        Kerala
2       Haryana
3       Gujarat
4           Goa
5  Chhattisgarh
6         Assam
7        Punjab
8        Sikkim
9     Rajasthan

 [[0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]]
```

- The above program illustrates how the two data transformation techniques - label encoding and one hot encoding - are applied for a given dataset for rescaling of data.
- Rescaling the attributes help in making the attributes fall within a given scale. This simplifies the complexity of data which consists of a variety of values and helps in easy and efficient data analysis.

3.3.3 Data Reduction

- When the data is collected from different data sources for analysis, it results in a huge amount of data. It is difficult for a data analyst to deal with this large volume of data.
- It is even difficult to run the complex queries on the huge amount of data as it takes a long time and sometimes it even becomes impossible to track the desired data.
- Data reduction is an essential and important phase in data preprocessing that is carried out to reduce the unimportant or unwanted features from a dataset.
- Data reduction techniques obtain a reduced representation of the data while minimizing the loss of information content.
- Data reduction process reduces the volume of original data and represents it in a much smaller volume. Data reduction techniques ensure the integrity of data while reducing the data.
- Data reduction is a preprocessing technique which helps in obtaining reduced representation of data set (i.e., data set having much smaller volume of data) from the available data set.
- Strategies for data reduction include Dimensionality reduction, Data cube aggregation, Numerosity reduction.

1. Dimensionality Reduction:

- Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration.
- In many machine learning techniques such as classification and clustering, features are studied to obtain the analysis output.
- However, the higher the number of features, the more difficult it is for the training process for output analysis.
- There may be many features that are correlated or redundant. Dimensionality reduction can then be applied to reduce the number of unwanted variables, by obtaining a set of principal variables.
- Dimensionality, in such a case, refers to the number of features available in a dataset. It can be divided into two main components – feature selection (also known as attribute subset selection) and feature extraction.

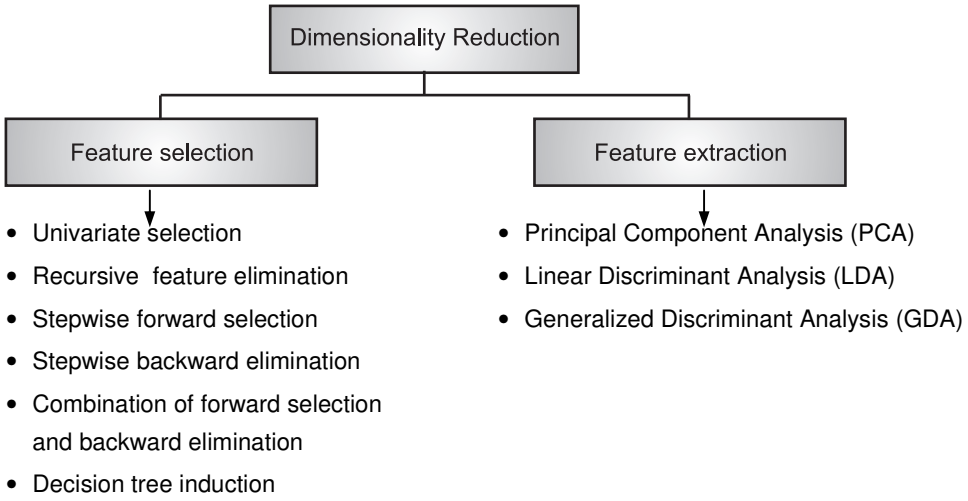


Fig 3.5: Dimensionality Reduction

Feature Selection:

- Feature selection is the process of extracting a subset of features from the original set of all features of a dataset to obtain a smaller subset that can be used to model a given problem.
- Few of the standard techniques used for feature selection are:
  - **(i) Univariate Selection** method works by inspecting each feature and then finding the best feature based on statistical tests. It also analyses the capability of these features in accordance with the response variable.
  - **(ii) Recursive Feature Elimination** method works by performing a greedy search to acquire the best feature subset from a given dataset. This is done in an iterative process by determining the best or the worst feature at each iteration.
  - **(iii) Stepwise Forward Selection** method initially starts with an empty set of attributes which is considered as the minimal set. In each iteration the most relevant attribute is then added to the minimal set until the stopping rule is satisfied. One of the stopping rules is to stop when all remaining variables have a p-value above some threshold.
  - **(iv) Stepwise Backward Elimination** method initially starts with all the sets of attributes that are considered as the initial set. In each iteration the most irrelevant attribute is then removed from the minimal set until the stopping rule is satisfied. One of the stopping rules is to stop when all remaining variables have a significant p-value defined by some significance threshold.
  - **(v) Combination of Forward Selection and Backward Elimination** method is commonly used for attribute subset selection and works by combining both the methods of forward selection and backward elimination.

(vi) **Decision Tree Induction** method uses the concept of decision trees for attribute selection. A decision tree consists of several nodes that have branches. The nodes of a decision tree indicate a test applied on an attribute while the branch indicates the outcome of the test. The decision tree helps in discarding the irrelevant attributes by considering those attributes that are not a part of the tree.

Feature Extraction:

- Feature extraction process is used to reduce the data in a high dimensional space to a lower dimension space.
- While feature selection chooses the most relevant features from among a set of given features, feature extraction creates a new, smaller set of features that consists of the most useful information.
- Few of the methods for dimensionality reduction include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Generalized Discriminant Analysis (GDA).
- These methods are discussed below:
  - (i) **Principal Component Analysis (PCA):** PCA is an unsupervised method of feature extraction that creates linear combinations of the original features. The features are uncorrelated and are ranked in order of variance. The data has to be normalized before performing PCA. PCA has several variations of it such as sparse PCA, kernel PCA, and so on.
  - (ii) **Linear Discriminant Analysis (LDA):** LDA is a supervised method of feature extraction that also creates linear combinations of the original features. However, it can be used for only labeled data and can be thus used only in certain situations. The data has to be normalized before performing LDA.
  - (iii) **Generalized Discriminant Analysis (GDA):** GDA deals with nonlinear discriminant analysis using kernel function operator. Similar to LDA, the objective of GDA is to find a projection for the features into a lower-dimensional space by maximizing the ratio of between-class scatters to within-class scatter. The main idea is to map the input space into a convenient feature space in which variables are nonlinearly related to the input space.
- Feature selection and feature extraction are extensively carried out as data preprocessing techniques for dimensionality reduction.
- This helps in removing redundant features, reducing computation time, as well as in reducing storage space.
- However, dimensionality reduction results in loss of data and should be used with proper understanding to effectively carry out data preprocessing before performing analysis of data.

2. Data Cube Aggregation:

- A data cube (or datacube) is a multi-dimensional ("n-D") array of values. A data cube is generally used to easily interpret data.
- Data cube is especially useful when representing data together with dimensions as certain measures of business requirements.
- Data cube aggregation is a process in which information is gathered and expressed in a summary form for purposes such as statistical analysis.
- Data cubes store multidimensional aggregated information. For example, a sales report has been prepared to analyze the number of sales of mobile phones per brand in each branch for the year 2009 to 2019.
- This can be represented in the form of a data cube as shown in Figure 3. This Figure has three dimensions – time, brand and branch.
- Data cubes provide fast access to pre-computed, summarized data, thereby benefiting online analytical processing as well as data mining.
- They are optimized for analytical purposes so that they can report on millions of records at a time.

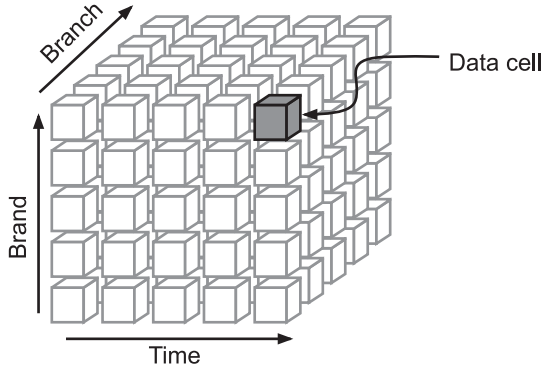


Fig. 3.6: Data Cube

- The cube created at the lowest abstraction level is referred to as the base cuboid. The base cuboid should correspond to an individual entity of interest such as sales or customer.
- In other words, the lowest level should be usable, or useful for the analysis. A cube at the highest level of abstraction is the apex cuboid.
- For the sales data in Fig. 3.7 the apex cuboid would give one total - the total sales for all three years, for all item types, and for all branches.
- Data cubes created for varying levels of abstraction are often referred to as cuboids, so that a data cube may instead refer to a lattice of cuboids. Each higher abstraction level further reduces the resulting data size.

- Consider we have the data of ATTRONICS Company sales per quarter for the year 2008 to the year 2010.
- In case we want to get the annual sale per year then we just have to aggregate the sales per quarter for each year.
- In this way, aggregation provides us with the required data which is much smaller in size and thereby we achieve data reduction even without losing any data.

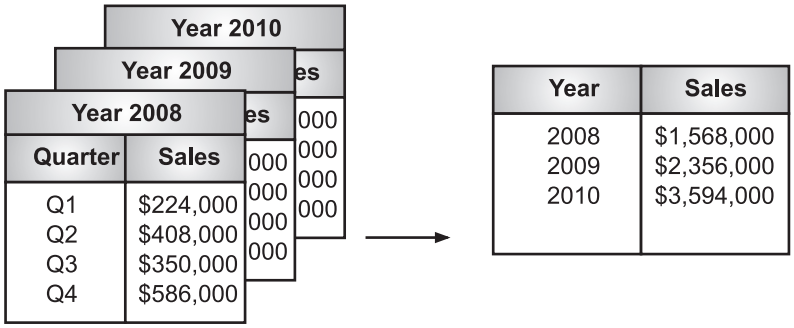


Fig. 3.7: Aggregated Data

- The data cube aggregation is a multidimensional aggregation which eases multidimensional analysis. Like in the image below the data cube represent annual sale for each item for each branch.
- The data cube present pre-computed and summarized data which eases the data mining into fast access.

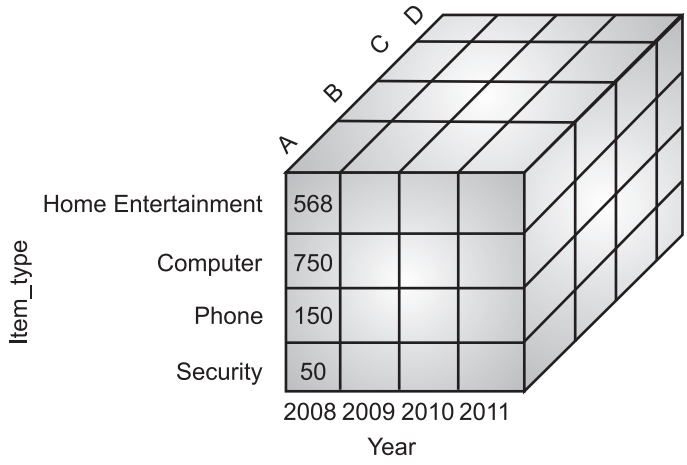


Fig. 3.8: Data Cube Aggregation

3. Numerosity Reduction:

- Numerosity reduction reduces the data volume by choosing alternative smaller forms of data representation.

- Numerosity reduction method is used for converting the data to smaller forms so as to reduce the volume of data.
- Numerosity reduction may be either parametric or non parametric as explained below:
  (i) Parametric methods use a model to represent data in which parameters of the data are stored, rather than the data itself. Examples of parametric models include regression and log-linear models.
  (ii) Non-parametric methods are used for storing reduced representations of the data. Examples of non-parametric models include clustering, histograms, sampling, and data cube aggregation.

3.3.4 Data Discretization

- Data discretization is characterized as a method of translating attribute values of continuous data into a finite set of intervals with minimal information loss.
- Data discretization facilitates the transfer of data by substituting interval marks for the values of numeric data.
- Similar to the values for the 'generation' variable, interval labels such as (0-10, 11-20…) or (0-10, 11-20…) may be substituted (kid, youth, adult, senior).
- The data discretization technique is used to divide the attributes of the continuous nature into data with intervals.
- We replace many constant values of the attributes by labels of small intervals. This means that mining results are shown in a concise and easily understandable way.
- The two approaches for discretization are explained below:
  1. **Top-down Discretization:** If we first consider one or a couple of points (so-called breakpoints or split points) to divide the whole set of attributes and repeat of this method up to the end, then the process is known as top-down discretization also known as splitting.
  2. **Bottom-up Discretization:** If we first consider all the constant values as split-points, some are discarded through a combination of the neighbourhood values in the interval, that process is called bottom-up discretization.
- Some of the data discretization strategies are as per followings:

Discretization by Binning:

- Binning is the famous methods of data discretization. Data binning, bucketing is a data pre-processing method used to minimize the effects of small observation errors.
- The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin.

- This has a smoothing effect on the input data and may also reduce the chances of over fitting in case of small datasets.
- Binning is a top-down splitting technique based on a specified number of bins. These methods are also used as discretization methods for data reduction and concept hierarchy generation.
- Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.
- For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median respectively.
- These techniques can be applied recursively to the resulting partitions to generate concept hierarchies.
- Distributing of values into bins can be done in a number of ways. One such way is called equal width binning in which the data is divided into n intervals of equal size. The width wof the interval is calculated as w = (max_value – min_value) / n.
- Another way of binning is called equal frequency binning in which the data is divided into n groups and each group contains approximately the same number of values as shown in the example below:
  - **Equal Frequency Binning:** Bins have equal frequency.
  - **Equal Width Binning:** Bins have equal width with a range of each bin are defined as [min + w], [min + 2w] …. [min + nw] where, w = (max – min) / (no of bins).

    Equal Frequency:

    Input: [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

    Output:

    [5, 10, 11, 13]

    [15, 35, 50, 55]

    [72, 92, 204, 215]

    Equal Width:

    Input: [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

    Output:

    [10, 11, 13, 15, 35, 50, 55, 72]

    [92]

    [204]

- One of the ways of finding the value of n in case of both equal width binning and equal frequency binning is by plotting a histogram and then trying different intervals to find an optimum value of n.
- Both equal width binning and equal frequency binning are unsupervised binning methods as these methods transform numerical values into categorical counterparts without using any class information.
- Following program shows the Python code for carrying out equal width binning for the price of nine items that are stored in a DataFrame.
- For equal width binning, the minimum and the maximum price values are used to three equal-width bins named Low, Medium and High.
- A histogram is also plotted for the three bins based on the price range.

```
#Program for Equal Width Binning
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
#Create a Dataframe
data={'item':['Apple','Apricot','Pears','Pomegranate'],
                                              'price':[297,216,185,140]}
#print the Dataframe
dframe = pd.DataFrame(data)
print("\n ORIGINAL DATASET")
print(" ----------------")
print(dframe)
#Creating bins
m11=min(dframe["price"])
m12=max(dframe["price"])
bins=np.linspace(m11,m12,4)
names=["low", "medium", "high"]
dframe["price_bin"]=pd.cut(dframe["price"],
                                      bins,labels=names,include_lowest=True)
print("\n BINNED DATASET")
print(" ----------------")
print(dframe)
```

- The output of above program is displayed below. In the output, the original dataset containing item names and price names are displayed.
- Then, the dataset is partitioned into three bins based on price range, which are categorically defined as Low, Medium, and High.

```
 ORIGINAL DATASET
 ----------------
          item  price
0       Apple     297
1      Apricot    216
2        Pears    185
3  Pomegranate    140

 BINNED DATASET
 ----------------
          item  price price_bin
0       Apple     297      high
1      Apricot    216    medium
2        Pears    185       low
3  Pomegranate    140       low
```

Discretization by Histogram Analysis:

- Like binning, histogram analysis is an unsupervised discretization technique because it does not use class information.
- Histogram analysis partitions the values for an attribute into disjoint ranges called buckets.
- In histogram analysis the histogram distributes an attribute's observed value into a disjoint subset, often called buckets or bins.
- A histogram partitions the values of an attribute, A, into disjoint ranges called buckets or bins. Various partitioning rules can be used to define histograms. In an equal-width histogram
- For example Consider the following data are a list of ATTRONICS Company prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted:

  1, 1, 5, 5, 5,5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18,

  18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.
- Fig. 3.7 shows a histogram for the data using singleton buckets.
- To further reduce the data, it is common to have each bucket denote a continuous value range for the given attribute.
- In Fig. 3.9, each bucket represents a different $10 range for price.

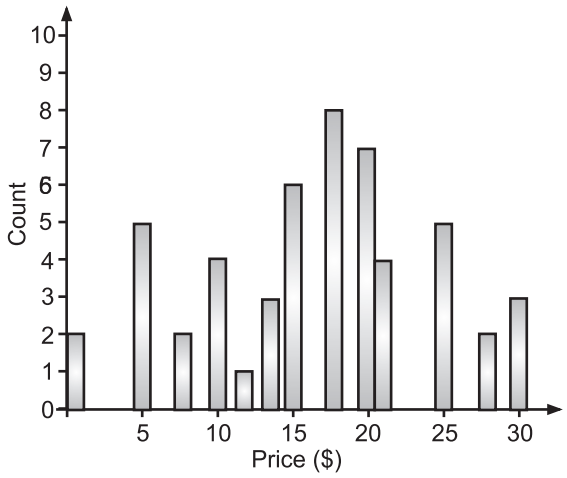


Fig. 3.9: A Histogram for Price using Singleton Buckets-each Bucket represents one Price-Value/Frequency Pair

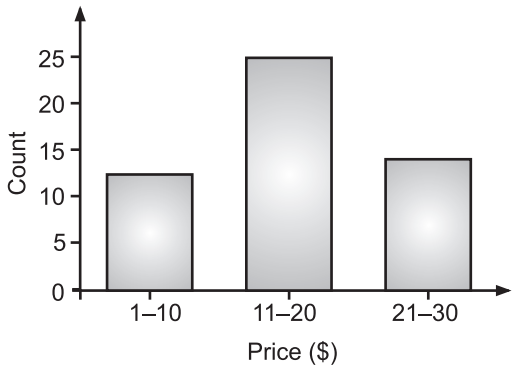


Fig. 3.10: An Equal-Width Histogram for Price, where Values are Aggregated so that each Bucket has a Uniform Width of $10

- There are several partitioning rules, including the following:
  - **Equal-width:** In an equal-width histogram, the width of each bucket range is uniform (e.g., the width of $10 for the buckets in Fig. 3.10).
  - **Equal-frequency (or Equal-depth):** In an equal-frequency histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (i.e., each bucket contains roughly the same number of contiguous data samples).
- For example, the values are partitioned into equal-size partitions or ranges (e.g., earlier in Fig. 3.10 for price, where each bucket has a width of $10).
- With an equal-frequency histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples.

- The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a pre-specified number of concept levels has been reached.
- A minimum interval size can also be used per level to control the recursive procedure. This specifies the minimum width of a partition, or the minimum number of values for each partition at each level.
- Histograms can also be partitioned based on cluster analysis of the data distribution, as described next.

Discretization by Cluster, Decision Tree and Correlation Analysis:

- Clustering, decision tree analysis and correlation analysis can be used for data discretization.
- Cluster analysis is a popular data discretization method. Cluster analysis method discretizes a numerical attribute by partitioning its value into clusters.
- A clustering algorithm can be applied to discretize a numeric attribute, A, by partitioning the values of A into clusters or groups.
- Clustering takes the distribution of A into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results.
- Clustering can be used to generate a concept hierarchy for A by following either a top-down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy.
- In the former, each initial cluster or partition may be further decomposed into several subclusters, forming a lower level of the hierarchy.
- In the latter, clusters are formed by repeatedly grouping neighboring clusters in order to form higher-level concepts.
- Techniques to generate decision trees for classification can be applied to discretization. Such techniques employ a top-down splitting approach.
- Unlike the other methods mentioned so far, decision tree approaches to discretization are supervised, that is, they make use of class label information.
- For example, we may have a data set of patient symptoms (the attributes) where each patient has an associated diagnosis class label.
- Class distribution information is used in the calculation and determination of split-points (data values for partitioning an attribute range).
- Intuitively, the main idea is to select split-points so that a given resulting partition contains as many tuples of the same class as possible.
- Entropy is the most commonly used measure for this purpose. To discretize a numeric attribute, A, the method selects the value of A that has the minimum entropy as a plit-

point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization.

- Such discretization forms a concept hierarchy for A. Because decision tree–based discretization uses class information, it is more likely that the interval boundaries (split-points) are defined to occur in places that may help improve classification accuracy.
- Measures of correlation can be used for discretization. ChiMerge is a χ2 based discretization method which uses bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively.
- As with decision tree analysis, ChiMerge is supervised in that it uses class information. The basic notion is that for accurate discretization, the relative class frequencies should be fairly consistent within an interval.
- Therefore, if two adjacent intervals have a very similar distribution of classes, then the intervals can be merged. Otherwise, they should remain separate.
- ChiMerge proceeds as follows:
  - Initially, each distinct value of a numeric attribute A is considered to be one interval. χ^2 tests are performed for every pair of adjacent intervals.
  - Adjacent intervals with the least χ2 values are merged together, because low χ^2 values for a pair indicate similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met.

PRACTICE QUESTIONS

Q.I Multiple Choice Questions:

1. Which is an essential step that needs to be considered before any analysis of data for more reliable and valid output?
   (a) preprocessing (b) cleaning
   (c) statistical analysis (d) None of the mentioned
2. The real-world data having the following issues,
   (a) Incomplete data (values of some attributes in the data are missing)
   (b) Noisy data (contains errors or outliers)
   (c) Inconsistent data (containing discrepancies in codes or names).
   (d) All of the mentioned
3. Some values in the data may not be filled up for various reasons and hence are considered as,
   (a) preprocessing value (b) missing values
   (c) absolute values (d) None of the mentioned

4. Which is a property of an object?
   (a) attribute value (b) parameter
   (c) argument (d) value
5. Data objects can also be referred to as,
   (a) samples (b) instances
   (c) data points (d) All of the mentioned
6. The values of a nominal attribute are,
   (a) strings (b) arrays
   (c) symbols (d) None of the mentioned
7. The factors comprising data quality including,
   (a) accuracy (b) completeness
   (c) consistency (d) All of the mentioned
8. Which involves processing the data in various formats such merging, grouping, concatenating etc. for the purpose of analyzing or getting them ready to be used with another set of data?
   (a) Data wrangling (b) Data munging
   (c) Both (a) and (b) (d) Data preprocessing
9. Data is cleaned by,
   (a) filling in the missing values (b) removing outliers
   (c) resolving any inconsistencies (d) All of the mentioned
10. Which method of outlier detection initially calculates the interquartile range (IQR) for the given data points?
    (a) Interquartile range (b) Variance range
    (c) Standard deviation range (d) None of the mentioned
11. Which of data allows scaling all data values to lie between a specified minimum and maximum value (say, between 0 and 1)?
    (a) cleaning (b) scaling
    (c) preprocessing (d) None of the mentioned
12. Which is used to convert textual labels into numeric form in order to prepare it to be used in a machine-readable form?
    (a) label encoding (b) Normalizing
    (c) Binarizing (d) Standardizing
13. Which is the process of changing the format, structure, or values of data?
    (a) Data cleaning (b) Data reduction
    (c) Data transformation (d) Data preprocessing

14. Which coding is used for categorical variables where no ordinal relationship exists among the variable's values?
    (a) One hot (b) label
    (c) cleaning (d) None of the mentioned
15. The process of reduction the unimportant or unwanted features from a dataset called as,
    (a) data cleaning (b) data discretization
    (c) data preprocessing (d) data reduction
16. Which store multidimensional aggregated information?
    (a) data normalization (b) data cubes
    (c) data binarizing (d) data standardizing
17. Which of the following is not data transformation technique?
    (a) Rescaling (b) Rescaling
    (c) Standardizing (d) One hot coding
18. Which method is used to partition the range of continuous attributes into intervals?
    (a) data cleaning (b) data discretization
    (c) data preprocessing (d) data reduction

Answers

1. (a)	2. (d)	3. (b)	4. (a)	5. (d)	6. (c)	7. (d)	8. (c)	9. (d)	10. (a)
11. (b)	12. (a)	13. (c)	14. (a)	15. (d)	16. (b)	17. (d)	18. (b)		

Q.II Fill in the Blanks:

1. Data _____ is the task of transforming raw data into a form so as to be ready to be fed into an algorithm.
2. A data _____ is a single-value descriptor for a data object.
3. Data _____ is the task of converting data into a feasible format that is suitable for the consumption of the data.
4. Data _____ is done to handle irrelevant or missing data.
5. Data _____ is carried out in data preprocessing to reduce the unimportant or unwanted features from a dataset.
6. Data _____ is carried out in data preprocessing to partition the range of continuous attributes into intervals.

7. A data _____ is a structured representation of data.
8. A collection of attributes describe a data _____.
9. Interval-scaled attributes are measured on a scale of _____ units.
10. Data _____ can be defined as the ability of a given data set to serve an intended purpose.
11. Filling up the missing values in data is known as the imputation of _____ data.
12. _____ is mainly applied to improve the accuracy of a predictive model by reducing the noise from data.
13. _____ may occur due to several reasons such as measurement error, data entry error, experimental error, intentional inclusion of outliers, sampling error or natural occurrence of outliers.
14. The outliers can be pictorially represented in the form of a _____.
15. _____ rescales data in such a way that each row of observation equals to a length of 1 (called a unit norm in linear algebra).
16. The _____ are assigned a value of 0 to (n-1) where n is the number of distinct values for a particular categorical feature.
17. _____ reduction can then be applied to reduce the number of unwanted variables, by obtaining a set of principal variables.
18. A _____ in a data cube is a part of the cube that represents a point of interest.
19. _____ discretization process also called splitting, works by first finding one or a few split points to divide the entire attribute range.

Answers

1. preprocessing	2. attribute	3. wrangling	4. cleaning
5. reduction	6. discretization	7. frame	8. object
9. equal-size	10. quality	11. missing	12. Binning
13. Outliers	14. histogram	15. Normalizing	16. labels
17. Dimensionality	18. cell	19. Top-down	

Q.III State True or False:

1. Data preprocessing involves data cleaning, data integration, data transformation, data reduction, and data discretization.
2. Nominal type of data is used to label variables that need to follow some order.
3. An outlier is a data point that is aloof or far away from other related datapoints.
4. Extra spaces are responsible for noisy data.

5. Label encoding is a data transformation technique.
6. Multiplication operation can be performed on ratio attributes.
7. Filling up the missing values in data is known as the imputation of missing data.
8. Normalizing also helps in easy identification of outliers or invalid values for numerical data.
9. Standard deviation method of outlier detection initially calculates the mean and standard deviation of the data points.
10. A white space is the blank space among the text.
11. The real-world data needed to be used for data analysis are often incomplete, unreliable, error-prone, and/or deficient in certain behaviors or trends.
12. Binarizing is the process of converting data to either 0 or 1 based on a threshold value.
13. Bottom-up discretization process also called as merging, works by first considering all the continuous values as potential split-points.
14. As data transformation preprocessing affects the manner in which outcomes of the final data will result, data analysts dedicate time and effort to resolve the issues of nurturing the raw, incomplete and inconsistent data to convert it to a usable state for analysis.
15. In one hot coding each column contains "0" or "1" corresponding to which column it has been placed.
16. Feature extraction process is used to reduce the data in a high dimensional space to a lower dimension space.
17. Numerosity reduction is used for converting the data to smaller forms so as to reduce the volume of data.
18. Data preprocessing transformation is the process of converting raw data into a format or structure that would be more suitable for data analysis.
19. Data reduction process reduces the volume of original data and represents it in a much smaller volume.
20. Linear Discriminant Analysis (LDA) is a supervised method of feature extraction that also creates linear combinations of the original features.
21. A data cube provides a multidimensional view of data and allows the pre-computation and fast access of summarized data.

Answers

1. (T)	2. (F)	3. (T)	4. (T)	5. (T)	6. (T)	7. (T)	8. (F)	9. (T)	10. (T)
11. (T)	12. (T)	13. (T)	14. (T)	15. (F)	16. (T)	17. (T)	18. (T)	19. (F)	20. (T)
21. (T)									

Q.IV Answer the following Questions:

(A) Short Answer Questions:

1. What is data?
2. Define data attributes.
3. What is nominal attribute?
4. Give purpose of preprocessing.
5. Which operations are not possible with ordinal attributes?
6. Define data object.
7. What is missing values?
8. Define data cleaning?
9. Define data discretization.
10. Differentiate discrete and continuous attributes, (any two points).
11. Why data reduction is important operation of data preprocessing?
12. What is data transformation?
13. What is one hot coding?
14. What are different methods for feature extraction.
15. Define rescaling?
16. What is binarizing?
17. Define data wrangling.
18. What is data filling?
19. What is meant by data quality?
20. What is data cube?

(B) Long Answer Questions:

1. Explain the need for data preprocessing before applying any analysis of data.
2. Explain different types of data attributes with example.
3. What are the various types of data available? Give an example of each.
4. What is the role of data cleaning? Explain in detail, any two standard data cleaning methods.
5. Differentiate between equal-width binning and equal-frequency binning.
6. What is data quality? Which factors are affected data quality.
7. Explain the following data transformation techniques:
   (i) Rescaling data
   (ii) Normalizing data

(iii) Binarizing data
(iv) Standardizing data
(v) Label encoding
(vi) One hot encoding.

8. What is meant by dimensionality reduction? Differentiate between feature selection and feature extraction.
9. What is meant by discretization and concept hierarchy? Differentiate between top-down discretization and bottom-up discretization
10. Suppose a group of 12 sales price records has been sorted as follows:

    5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215.

    Partition them into three bins by each of the following methods:
    (i) equal-frequency (equal-depth) Binning
    (ii) equal-width Binning.
11. Explain data cube aggregation in detail.
12. What do you mean by discretization? explain discretization by Histogram analysis.
13. Explain concept hierarchy generation for nominal data.

❖❖❖

CHAPTER 4

Data Visualization

Objectives...

- To learn Concept of Data Visualization
- To study Visual Encoding and Visualization Libraries
- To understand Basic Data Visualization Tools
- To learn Specialized Data Visualization Tools

4.0 INTRODUCTION

- Today, we are live in a data-driven world. On each day, the bulk amount of data is produced which when analyzed can produce valuable information.
- However, it is not simple and easy to interpret what information does data produce simply by observing loads of data values. Data is usually processed and analyzed for being converted to meaningful information.
- Once the information is ready or produced, it is preferred to be presented in a graphical format rather than textual format as it is said that the human brain processes visual content better than plain textual information. This is where the role of data visualization in data analytics or data science comes into play.
- Visualization is a process that transforms the representation of real raw data into meaningful information/insights in a visual representation.
- Data visualization is the graphical representation of information and data. Data visualization representation can be considered as a mapping between the original data (usually numerical) and graphic elements (for example, lines or points in a chart).
- The mapping determines how the attributes of these elements vary according to the data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.
- Data visualization, which is the graphical representation of data that can make information easy to analyze and understand.

Advantages of Visualization:

1. Visualization makes it easier for humans to detect trends, patterns, correlations, and outliers in a group of data.

2. Data visualization makes humans understand the big picture of big data using a small, impactful visualizations.
3. A simple data visualization built with credible data with good analytical modeling can help businesses/organizations make quick business decisions.

4.1 INTRODUCTION TO EXPLORATORY DATA ANALYSIS

- Exploratory Data Analysis (EDA) is a process of examining or understanding the data and extracting insights of the data.
- EDA is an important step in any Data Science project. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers) and form hypotheses based on the understanding of the dataset.
- EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data easy and better.
- The initial analysis of data supplied or extracted, to understand the trends, underlying limitations, quality, patterns, and relationships between various entities within the data set, using descriptive statistics and visualization tools is called Exploratory Data Analysis (EDA).
- EDA refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.
- EDA is used to understand and summarize the contents of a dataset. EDA typically relies heavily on visualizing the data to assess patterns and identify data characteristics.
- EDA usually involves a combination of the following methods:
    - **Univariate visualization** of and summary statistics for each field in the raw dataset.
    - **Bivariate visualization and summary statistics** for assessing the relationship between each variable in the dataset and the target variable of interest.
    - **Multivariate visualizations** to understand interactions between different fields in the data.
    - **Dimensionality reduction** to understand the fields in the data that account for the most variance between observations and allow for the processing of a reduced volume of data
    - **Clustering** of similar observations in the dataset into differentiated groupings, which by collapsing the data into a few small data points, patterns of behavior can be more easily identified.

4.2 DATA VISUALIZATION AND VISUAL ENCODING

- Visualization is the graphical representation of data that can make information easy to analyze and understand.
- Data visualization has the power of illustrating complex data relationships and patterns with the help of simple designs consisting of lines, shapes, and colors.
- Visual encoding is used to map data into visual structures, thereby building an image on the screen.

4.2.1 Data Visualization

- Data visualization is the presentation of data in graphical format. Data visualization is a generic term used which describes any attempt to help understanding of data by providing visual representation.
- Visualization of data makes it much easier to analyze and understand the textual and numeric data.
- Apart from saving time, increased used of data for decision making further adds to the importance and need of data visualization.
- Data visualization software also plays an important role in big data and in the decision making of the analytics world.
- Data visualization is a very important part of data analysis. We can use it to explore our data. If we understand our data well, we will have a better chance to find some insights.
- Finally, when we find any insights, we can use visualizations again to be able to share our findings with other people.
- Data visualization can help in:

1. **Identify Outliers in Data:** Outliers may occur due to several reasons such as measurement error, data entry error, experimental inclusion of outliers, sampling error or natural occurrence of outliers. For data visualization analysis, outliers should be excluded from the dataset as much as possible as these outliers may mislead the analysis process resulting in abruptly different, incorrect results and longer training time. With the help of data visualization, outliers in data can be easily detected so as to be removed for further analysis
2. **Enhanced Collaboration:** Advanced visualization tools make it easier for teams to collaboratively go through the reports for instant decision-making.
3. **Business Analysis Made Easy:** Business analysts can deal with various important decision-making such as sales prediction, product promotion, and customer behavior through the use of correct data visualization techniques.

4. **Improve Response Time:** Data visualization gives a quick glance of the entire data and, in turn, allows analysts or scientists to quickly identify issues, thereby improving response time. This is in contrast to huge chunks of information that may be displayed in textual or tabular format covering multiple lines or records.
5. **Greater Simplicity:** Data, when displayed graphically in a concise format, allows analysts or scientists to examine the picture easily. The data to be concentrated on gets simplified as analysts or scientists interact only with relevant data.
6. **Easier Visualization of Patterns:** Data presented graphically permits analysts to effortlessly understand the content for identifying new patterns and trends that are otherwise almost impossible to analyze. Trend analysis or time-series analysis are in huge demand in the market for a continuous study of trends in the stock market, companies or business sectors.

4.2.2 Visual Encoding

- Encoding in data visualization means translating the data into a visual element on a chart or map through position, shape, size, symbols and color.
- The visual encoding is the way in which data is mapped into visual structures, upon which we build the images on a screen.
- Visual encoding is the approach/technique used to map data into visual structures, thus building an image on the screen.
- The visualization tool can be more effective due to the easy perception of information conveyed by the former visualization graph than the latter.
- The attribute values signify important data characteristics such as numerical data, categorical data, or ordinal data.
- Spatiotemporal data contains special attributes such as geographical location (spatial dimension) and/or time (temporal dimension).

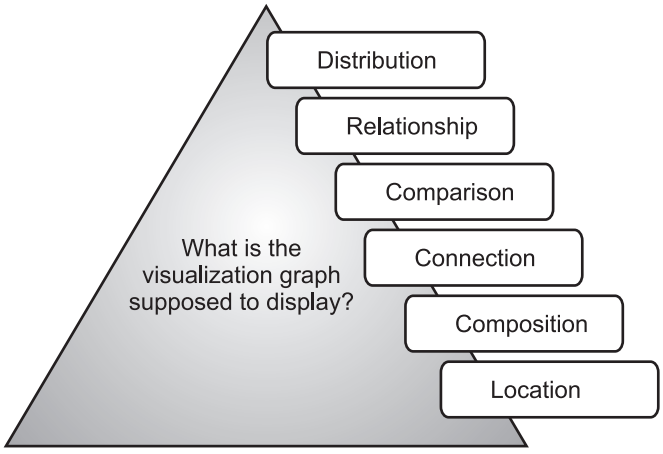


Fig 4.1: Concepts of a Visualization Graph

- The Fig. 4.1 shows the several concepts that a visualization graph may like to convey based on which a particular visualization tool is used.
- While simple data comparisons can be made with a bar chart and column chart, data composition can be expressed with the help of a pie chart or stacked column chart.
- The use of an appropriate visualization graph is a challenging task and should be considered an important factor for data analysis in data science.

Table 4.1: Role of data visualization and its corresponding visualization tool

Role of Data Visualization Possible Illustrative Data	Visualization Graph
Distribution	• Scatter chart • 3D Area chart • Histogram
Relationship	• Bubble chart • Scatter chart
Comparison	• Bar chart • Line chart • Column chart • Area chart
Composition	• Pie chart • Waterfall chart • Stacked column chart • Stacked area chart
Location	• Bubble map • Choropleth map • Connection map
Connection	• Matrix chart • Node-link diagram • Word cloud • Alluvial diagram • Tube map

- Table 4.1 gives a basic idea of which visualization graph can be used to show the accurate role of data provided in a dataset.

- Mapping of the data is based on the visual cues (also called retinal variables) such as location, size, color value, color hue, color saturation, transparency, shape, structure, orientation, and so on.
- To represent data that involves three or more variables, these retinal variables play a major role. For example:
  1. **Shape,** such as circle, oval, diamond and rectangle, may signify different types of data and is easily recognized by the eye for the distinguished look.
  2. **Size** is used for quantitative data as a smaller size indicates less value while bigger size indicates more value.
  3. **Color** saturation decides the intensity of color and can be used to differentiate visual elements from their surroundings by displaying different scales of values.
  4. **Colorhue** plays an important role in data visualization as for instance, the red color signifies something alarming, the blue color signifies something serene and peaceful, while the yellow color signifies something bright and attractive.
  5. **Orientation,** such as vertical, horizontal and slanted, help in signifying data trends such as an upward trend or a downward trend.
  6. **Texture** show differentiation among data and is mainly used for data comparisons.
  7. **Length** decides the proportion of data and is a good visualization parameter for comparing data of varying values.
  8. **Angles** provide a sense of proportion and this characteristic can help data Science Fundamentals and Practical Approaches analysts or data scientists make better data comparisons.
- Based on what type of data, the visualization tools should be effectively chosen to represent data in the visualization graph.
- While on one hand, varying shapes can be used to represent nominal data, on the other hand, various shadings of a particular color can be used for mapping data that has a particular ranking or order (as in case of ordinal data).
- The following softwares are used for data visualization:

Software	Description	Features
Tableau	It is based on Visual Analytics Platform. Easily connects, Visualizes and shares data with an effective seamless experience from desktop to mobile.	Database integration. Drag and drop interface. Email integration. Email notifications. Dashboard creation.

		Flexible data analysis methods. Mobile friendly. Manageable permission access.
Qlikview	QlikView connects directly to the data source, regardless of where it is stored. Allows users to create default and custom data connectors and templates, depending on one's need.	Personalized data search. Script building. Role-based access.
Sisense	Uses agile analysis software with a variety of data visualization options. Can create dashboards and graphics with drag and drop user interface.	Consolidates, stores, and accumulates data. Interactive browser-based dashboards. Complex business queries without programming or SQL writing. Removes limitations to data size. Integrates with web portals. Easy setup and use data exporting range. Interactive dashboards
Looker	It provides an efficient business intelligence platform for users to utilize SQL for organizing unstructured data.	Integrates with big data platform and databases. Ease of use. Strong collaboration features. Handy and compact visualization. Mobile, tablet, and desktop friendly. Excellent customer support, technical support available through chat in seconds.

Zoho Analytics	Uses a variety of tools, such as pivot tables, KPI widgets and tabular view components. Can generate reports with valuable business insights.	Connect to any data source. Enable performing deep analysis. Insightful reports. Robust security. Integration and app development.
Domo	Generates real-time data in a single dashboard. Can generate various creative data displays such as multipart widgets and trend.	Free trial. Socialization. Dashboard creation.
Microsoft Power BI	Comes with unlimited access to on-site and in-cloud data that gives a centralized data access hub.	Unlimited connective options. Affordability Web publishing.
IBM Watson Analytics	Can type in various questions which the intelligence software can interpret and answer accordingly.	File Upload Public forum support On-site data.
Plotly	Provides a vast variety of colorful designs for data visualization. Can use the chart studio to create web-based reporting templates.	2D and 3D chart options. Open source coding designer. Input dashboards. Authentication. Snapshot engine. Embedding. Big Data for Python. Image storage.
SAP Analytic Cloud	Can generate focused reports and collaborative tools for online discussion. Provides import and export features for spreadsheets and visuals.	Easy forecasting set up important events. Cloudbased protection.

4.3 DATA VISUALIZATION LIBRARIES

- The need convey information with visualizations has increased with the availability of data sources.
- Data visualization is the process of understanding the data in more detail using some plots and graphs. There are many libraries in Python that help us to do the same.
- Python has become one of the most popular languages to be used for data analytics today.
- The Python Package has rich built-in libraries for practically every data visualization that is needed.
- Each library of visualization has its own specification. Using the particular libraries for specific task helps the user to complete the task in more easy and accurate way. Some liberates work better than the others.
- Python Libraries for data visualization that are commonly used are matplotlib, seaborn, ggplot, plotly, bokeh, pygal, geoplotlib, gleam, missingno, leather and so on.

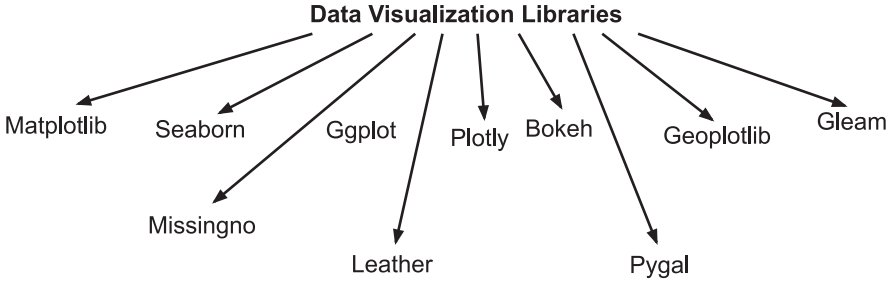


Fig. 4.2: Data Visualization Libraries in Python

- Let us discussed these Python libraries used for data visualization:

matplotlib Library:

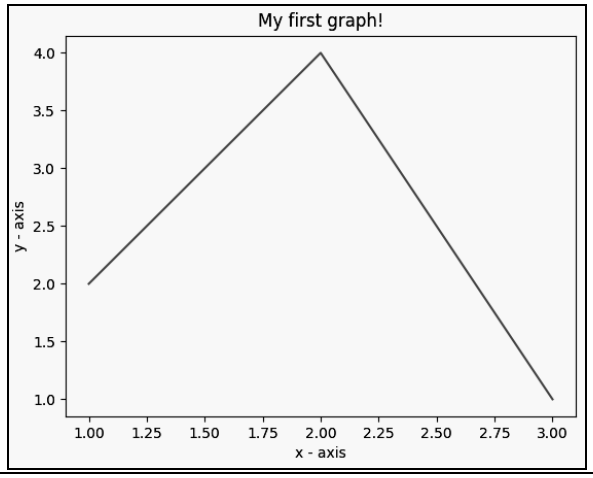
- The matplotlib is the most common standard Python library used for plotting 2D data visualizations. Matplotlib Python library is used to generate simple yet powerful visualizations.
- The matplotlib library is mainly used for creating plots that can be zoomed in on a section of the plot and pan around the plot using the toolbar in the plot window.
- It is the first data visualization library to be developed in Python, and later many other libraries were built on top of it for various other ways of visualizations.
- The versatility of Matplotlib can be used to make visualization types such as Scatter plots, Bar charts and Histograms, Line plots, Pie charts, Stem plots, Contour plots, Quiver plots, Spectrograms and so on.

- The matplotlib library allows easy use of labels, axes titles, grids, legends, and other graphic requirements with customizable values and text.
- Matplotlib library in Python is built on NumPy arrays.

Example:

```
# importing the required module
import matplotlib.pyplot as plt
  # x axis values
x = [1,2,3]
# corresponding y axis values
y = [2,4,1]
  # plotting the points
plt.plot(x, y)
  # naming the x axis
plt.xlabel('x - axis')
# naming the y axis
plt.ylabel('y - axis')
  # giving a title to my graph
plt.title('My first graph!')
  # function to show the plot
plt.show()
```

Output:



seaborn Library:

- The seaborn library in Python couples the power of the matplotlib library to create artistic charts with very few lines of code.

- The seaborn library follows creative styles and rich color palettes, that allows to create visualization plots to be more attractive and modern.
- The seaborn is a popular data visualization library that is built on top of Matplotlib. The seaborn library puts visualization at the core of understanding any data.
- Today's visualization graph is mainly plotted in seaborn rather than matplotlib, primarily because of the seaborn library's rich color palettes and graphic styles that is much more stylish and sophisticated than matplotlib.
- As seaborn is considered to be a higher-level library, there are certain special visualization tools such as, violin plots, heat maps and time series plots that can be created using this library.
- Seaborn is very helpful to explore and understand data in a better way. It provides a high level of a crossing point for sketching attractive and informative algebraic graphics.

Example: To get the built-in dataset names in seaborn:

```
import pandas
import matplotlib
import scipy
import seaborn as sns
print(sns.get_dataset_names())
```

Output:

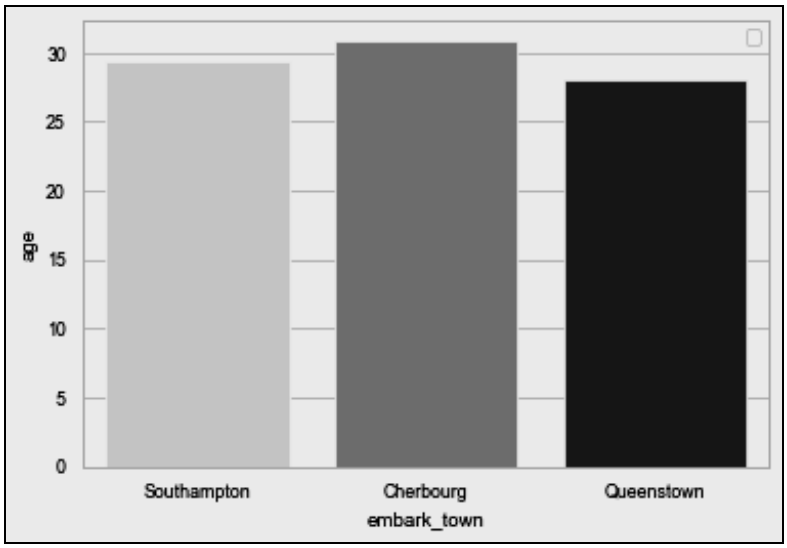
```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'exercise', 'flights', 'fmri', 'gammas', 'geyser',
'iris', 'mpg', 'penguins', 'planets', 'tips', 'titanic']
```

Example: The barplot plot below shows the survivors of the titanic crash based on category.

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid'))
# load dataset
titanic = sns.load_dataset('titanic')
# create plot
sns.barplot(x = 'embark_town', y = 'age', data = titanic,
                                        palette = 'BuGn',ci=None)
plt.legend()
```

```
plt.show()
print(titanic.columns)
```

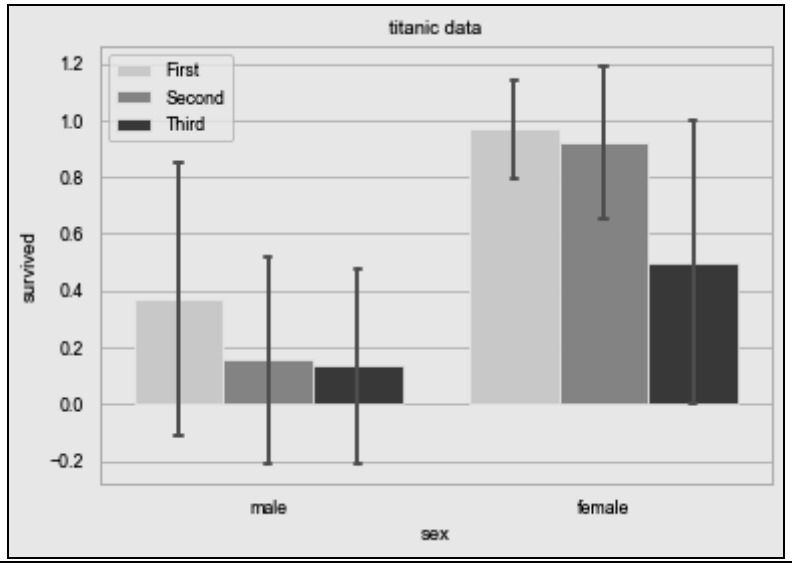
Output:



Example:

```
import matplotlib.pyplot as plt
import seaborn as sns
# load dataset
titanic = sns.load_dataset('titanic')
# create plot
sns.barplot(x = 'sex', y = 'survived', hue = 'class', data = titanic,
            palette = 'BuGn', order = ['male', 'female'],
            capsize = 0.02,saturation = 10,
            errcolor = 'gray', errwidth = 2,
            ci = 'sd'
            )
plt.legend()
plt.title('titanic data')
plt.show()
```

Output:



ggplot Library:

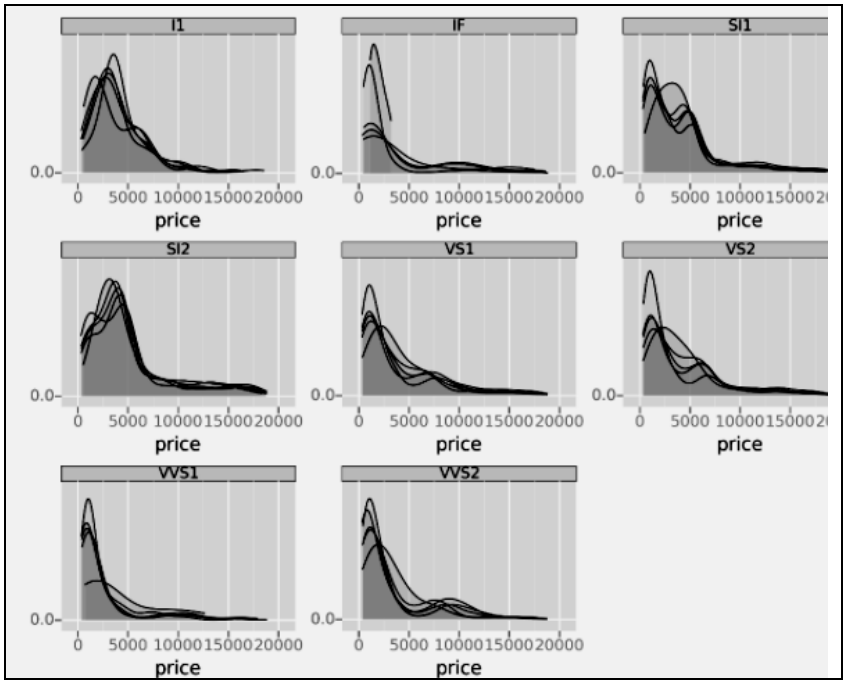
- The ggplot library of Python is based on the ggplot2 library which is an R plotting system and concepts are based on the Grammar of Graphics.
- The ggplot library creates a layer of components for creating plots which makes it different from matplotlib based on the operations of plotting graph.
- The ggplot Library is integrated with pandas and is mainly used for creating very simple graphics.
- This library sacrifices the complexity of plotting complex graphs as its primary focus is on building basic graphs that are often required for analyzing simple data distribution.
- The ggplot is not designed to develop a high level of customized graphics. It has a simpler method of plotting with a lack of complexity.
- It is integrated with Pandas. Therefore, it's best to store data in a data frame while using ggplot.

Example:

```
from plotnine.data import economics
from plotnine import ggplot, aes, geom_line
(
    ggplot(economics) #what data to use
            + aes (x = " price ") #what variables to use
```

```
        +geom_line() #Geometric object to use for creating graphs.
    )
```

Output:



Bokeh Library:

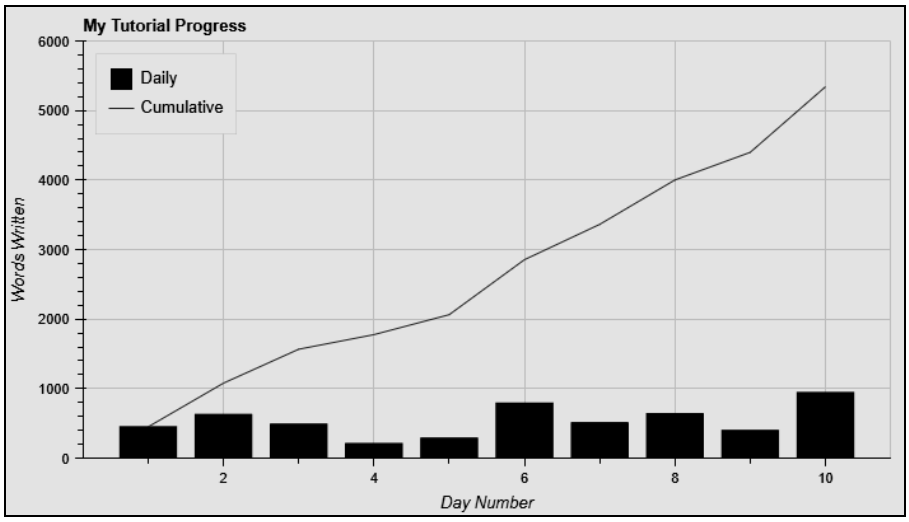
- The Bokeh library in Python is native to Python and is mainly used to create interactive, web-ready plots, which can be easily output as HTML documents, JSON objects, or interactive web applications. Like ggplot,
- The Bokeh library has an added advantage of managing real-time data and streaming. This library can be used for creating common charts such as histograms,bar plots, and box plots.
- It can also handle very minute points of a graph such as handling a dot of a scatter plot.
- The Bokeh library includes methods for creating common charts such as bar plots, box plots and histograms.
- Using Bokeh, it is easy for a user to control and define every element of the chart without using any default values and designs. There are three varying interfaces supported by Bokeh for being used by different types of users.
- There are three interfaces with different levels of control to put up different user types in the Bokeh library.

- The highest level of control is used to create charts rapidly. This library includes different methods of generating and plotting standard charts such as bar plots, histograms and box plots.
- The lowest level focuses on developers and software engineers as this interface provides full support for controlling and customizing each and every component of a graph to deal with complex graphics.
- This level has no pre - set defaults, and users have to define each element of the chart or plot.
- The middle level of control has the specifications same as the Matplotlib library. This level allows the users to control the basic development of blocks of every chart and plot.

Example:

```
import numpy as np
# Bokeh libraries
from bokeh.io import output_notebook
from bokeh.plotting import figure, show
# My word count data
day_num = np.linspace(1, 10, 10)
daily_words = [450, 628, 488, 210, 287, 791, 508, 639, 397, 943]
cumulative_words = np.cumsum(daily_words)
# Output the visualization directly in the notebook
output_notebook()
# Create a figure with a datetime type x-axis
fig = figure(title='My Tutorial Progress',
      plot_height=400, plot_width=700,
      x_axis_label='Day Number', y_axis_label='Words Written',
      x_minor_ticks=2, y_range=(0, 6000),
      toolbar_location=None)
# The daily words will be represented as vertical bars (columns)
fig.vbar(x=day_num, bottom=0, top=daily_words,
      color='green', width=0.75,
      legend='Daily')
# The cumulative sum will be a trend line
fig.line(x=day_num, y=cumulative_words,
         color='gray', line_width=1,
         legend='Cumulative')
# Put the legend in the upper left corner
fig.legend.location = 'top_left'
# Let's check it out
show(fig)
```

Output:



plotly Library:

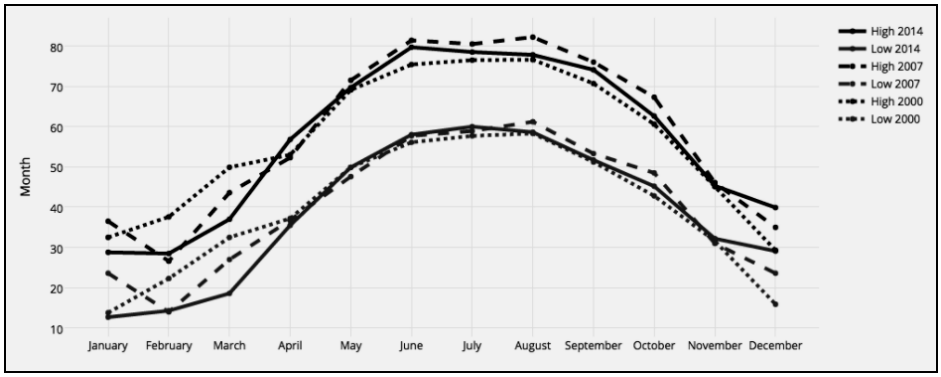
- The plotly library in Python is an online platform for data visualization and it can be used in making interactive plots that are not possible using other Python libraries.
- Few such plots include dendrograms, contour plots, and 3D charts. Other than these graphics, some basic visualization graphs such as area charts, bar charts, box plots, histograms, polar charts, and bubble charts can also be created using the plotly library.
- One interesting fact about plotly is that the graphs are not saved as images but rather serialized as JSON, because of which the graphs can be opened and viewed with other applications such as R, Julia, and MATLAB.
- Plotly library of Python is developed on the top of Plotly JavaScript library.

Example:

```
import plotly.graph_objects as ply
# Add data
months = [ ' Jan ', ' Feb ', ' Mar ', ' Apr ', ' May ', ' June ',
             ' July ',' Aug ', ' Sep ', ' Oct ', ' Nov ', ' Dec ']
high_2010 = [ 31.5, 36.6, 48.9, 52.0, 68.1, 74.4, 75.5, 75.6, 69.7,
                                                    59.6, 44.1, 28.3 ]
low_2010 = [ 12.8, 21.3, 31.5, 36.2, 50.9, 55.1, 56.7, 57.3, 50.2,
                                                    41.8, 30.6, 14.9 ]
high_2015 = [ 35.5, 25.6, 42.6, 51.3, 70.5, 80.4, 81.5, 81.2, 75.0, 66.,
                                                  45.1, 34.0 ]
```

```
low_2015 = [ 22.6, 13.0, 26.0, 35.8, 46.6, 56.7, 57.9, 60.2, 52.3, 47.5,
                                                        30.0, 22.6 ]
high_2020 = [ 27.8, 27.5, 36.0, 55.8, 68.7, 78.7, 77.5, 76.8, 73.1, 61.,
                                                      44.3, 38.9 ]
low_2020 = [ 11.7, 13.3, 17.6, 34.5, 48.9, 57.0, 59.0, 57.6, 50.7, 44.2,
                                                      31.2, 28.1 ]
fig = go.Figure ()
# Create and style traces
fig.add_trace ( ply.Scatter( x = month, y = high_2020, name='High 2020',
                    line = dict ( color = ' firebrick ', width = 4 ) ) )
fig.add_trace ( ply.Scatter( x = month, y = low_2020, name = ' Low 2020,
              line = dict ( color = ' royalblue ', width = 4 ) ))
fig.add_trace ( ply.Scatter( x = month, y = high_2015, name = 'High 201,
 line = dict(color = 'firebrick', width = 4, dash = 'dash' ) # here in t
 his code dash options also involve 'dash', 'dot', and 'dashdot' ) )
fig.add_trace (ply.Scatter ( x = month, y = low_2015, name = 'Low 2015',
       line = dict ( color = 'royalblue', width = 4, dash = 'dash' ) ) )
fig.add_trace (ply.Scatter ( x = month, y=high_2010, name='High 2010',
                    line = dict(color='firebrick', width=4, dash='dot')))
fig.add_trace (ply.Scatter ( x = months, y = low_2010, name='Low 2010',
        line = dict ( color = 'royalblue', width = 4, dash = 'dot') ) )
 # Editing the layout of the graph
 fig.update_layout ( title = 'Average High and Low Temperatures in NYC',
       xaxis_title = ' Months ',yaxis_title = '
                            Temperatures ( degrees F ) ' )
 fig.show()
```

Output:



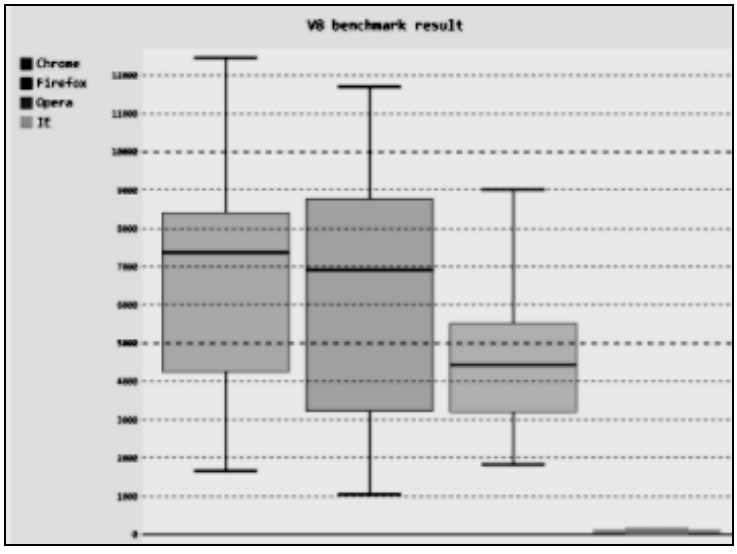
pygal Library:

- The pygal library in Python creates interactive plots that can be embedded in the web browser. It also has the ability to output charts as Scalable Vector Graphics (SVGs).
- All the chart types created using pygal are packaged into a method that makes it easy to create an artistic chart in a few lines of code.
- For instance, to create a bar chart, simply import the pygal library and then create a variable to assign the value of pygal.Bar().
- The graph created can finally be saved in .svg extension to get astylized CSS formatting.
- In the pygal library, it is easy to draw an attractive chart in just a few code lines because it has methods for all different chart types, and it also has built-in styles.

Example:

```
import pygal
box_plot = pygal.Box()
box_plot.title = ' V8 benchmark result '
box_plot.add ( ' Chrome ', [ 6394, 8211, 7519, 7217, 12463, 1659,
                                                    2122, 8606 ] )
box_plot.add ( ' Firefox ', [ 7472, 8098, 11699, 2650, 6360, 1043,
                                                    3796, 9449 ] )
box_plot.add ( ' Opera ', [ 3471, 2932, 4202, 5228, 5811, 1827, 9012,
                                                     4668 ] )
box_plot.add ( ' IE ', [ 42, 40, 58, 78, 143, 135, 33, 101 ] )
```

Output:



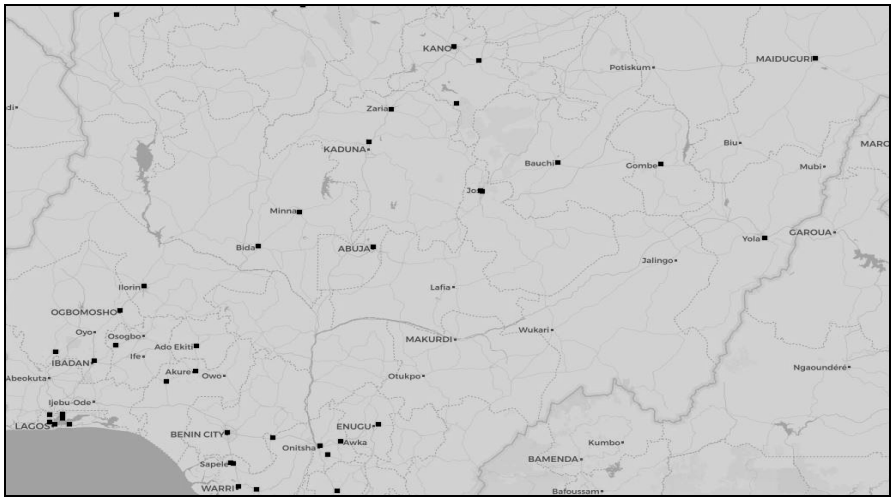
Geoplotlib Library:

- The geoplotlib in Python is a toolbox for designing maps and plotting geographical data.
- Few of the map-types that can be created are heatmaps, dot-density maps, and choropleths.
- In order to use geoplotlib, one has to also install Pyglet, an object oriented programming interface.
- This library is mainly used for drawing maps as no other Python libraries are meant for creating graphics for maps.

Example:

```
import geoplotlib
from geoplotlib.utils import read_csv
data = read_csv("C:\\Users\\Omotayo\\Desktop\\nigeria_cities.csv")
#replace path with your file path
geoplotlib.dot(data,point_size=3)
geoplotlib.show()
```

Output:



Note: This is a dot density map of cities in Nigeria.

gleam Library:

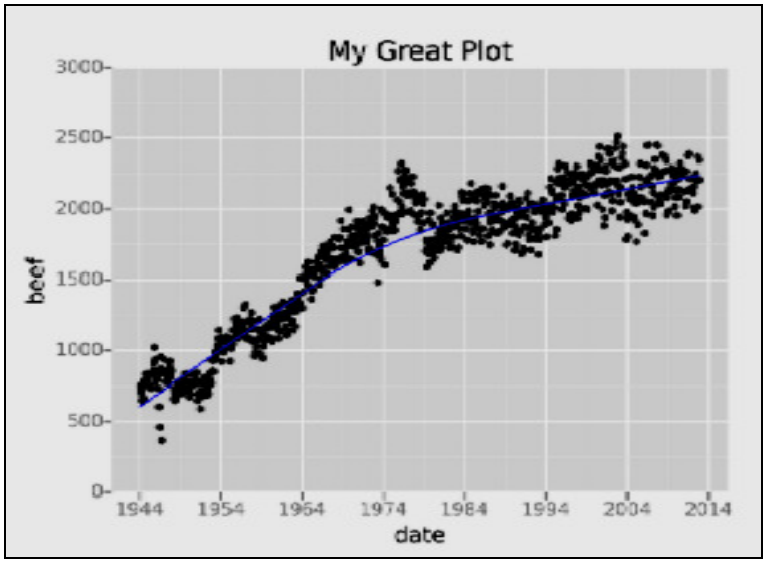
- The gleam Python library lets us to build interactive web visualizations of data without knowledge of HTML or JS.
- We can choose a number of inputs your users can control, then use any Python graphing library to create plots based on those inputs.

- The gleam puts it all together creates a web interface that lets anyone play with your data in real time. It converts analyses into interactive web apps using Python scripts.
- It creates a web interface that lets anyone play with the data in real-time. One interesting capability of this library is that fields can be created on top of the graphic and users can filter and sort data by choosing appropriate field.
- Gleam uses the wtforms package to provide form inputs.

Example:

```
from wtforms import fields
from ggplot import *
from gleam import Page, panels
class ScatterInput ( panels.Inputs ) :
    title = fields.StringField ( label = " Title of plot : " )
    yvar = fields.SelectField ( label = " Y axis " ,
                          choices = [ ( " beef " , " Beef " ) ,
                               ( " pork " , " Pork " ) ] )
 smoother = fields.BooleanField ( label = " Smoothing Curve " )
```

Output:



missingno Library:

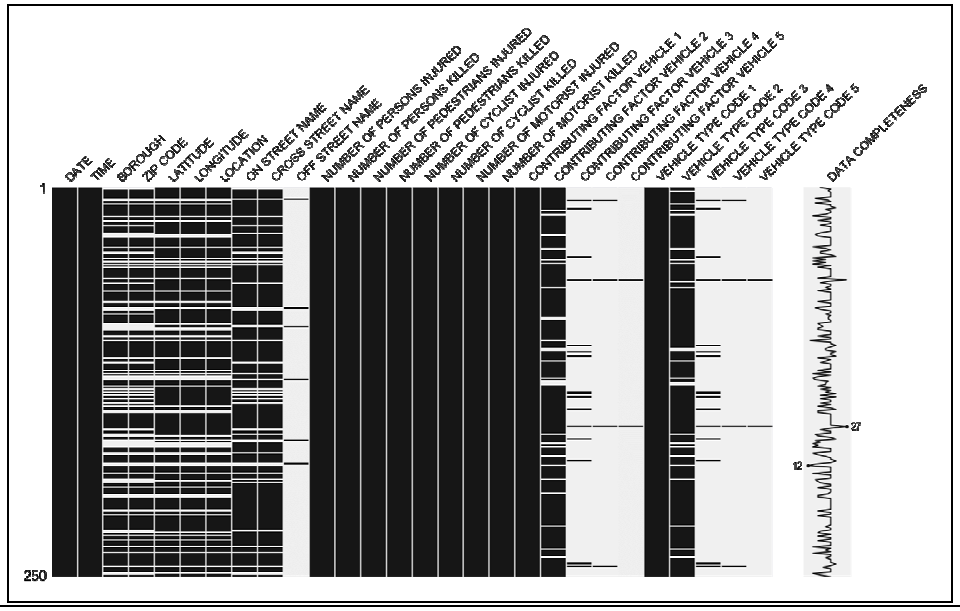
- The missingno library in Python can deal with missing data and can quickly measure the wholeness of a dataset with a visual summary, instead of managing through a table.

- The data can be filtered and arranged based on completion or spot correlations with a dendrogram or heatmap.
- The missingno is a library of the Python programming language used to deal with the dataset having missing values or messy values.
- This library provides a small toolset that is easy - to - use and flexible with missing data visualizations. It has utilities that help the user to get a rapidly visual summary of the completeness dataset.

Example:

```
import missingno as mgno
%matplotlib inline
mgno.matrix(collisions.samples(250))
```

Output:



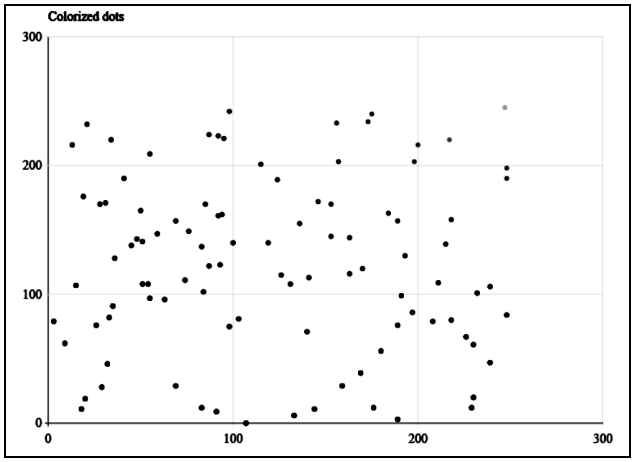
Leather Library:

- Leather library in Python used to create charts for those who need charts immediately and do not care whether the chart is perfect.
- This library works with every type of data set. This library creates the output chats of data as SVGs so that the users can measure the charts with the best quality.
- The leather library is a new library, and still, some of its documentations are in progress.
- The charts created using this library are basic but of good quality, which is roughly made.

Example:

```
import random
import leather
dot_data = [(random.randint(0, 250), random.randint(0,
                                        250)) for i in range(100)]
def colorizer(d):
    return 'rgb(%i, %i, %i)' % (d.x, d.y, 150)
chart = leather.Chart('Colorized dots')
chart.add_dots(dot_data, fill_color=colorizer)
chart.to_svg('examples/charts/colorized_dots.svg')
```

Output:



4.4 BASIC DATA VISUALIZATION TOOLS

- In this section we will discuss the basic data visualization tools such as Histograms, Bar charts/graphs, Scatter plots, Line chart, Area plots, Pie charts, Donut charts and so on.

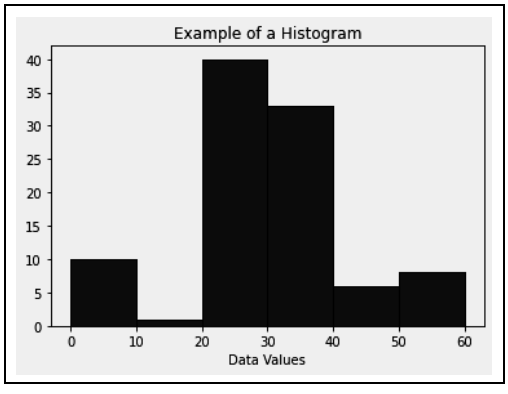
Histogram:

- A histogram is a graphical display of data using bars of different heights. A histogram shows an accurate representation of the distribution of numeric data.
- A histogram is a way to represent the distribution of numerical data elements (mainly statistical) in an approximate manner. A histogram uses a "bin" or a "bucket" for a set or range of values to be distributed.
- A histogram is discrete and need not be a contiguous one. Based on the bins and the values of the data, it can be skewed either to the left or to the right of the visualization.

- To create a histogram, first, we divide the entire range of values into a series of intervals, and second, we count how many values fall into each interval.
- The intervals are also called bins. The bins are consecutive and non-overlapping intervals of a variable.
- They must be adjacent and are often of equal size. To make a histogram with matplotlib, we can use the plt.hist() function.
- The first argument is the numeric data, the second argument is the number of bins. The default value for the bins argument is 10.
- Following python program shows the designing a histogram for displaying the frequency distribution of the given continuous data.
- The histogram consists of seven bins or intervals. The data to be plotted is stored in an array that consists of six numerical values.
- The weights indicate the frequency of the data values that are provided in the array. The bars are displayed in green color and the edges of the bar are displayed in red color.

```
#Program for a histogram
import matplotlib.pyplot as plt
#Creating an array of numerical data
data = [1,11,21,31,41,51]
#Plotting the histogram
plt.hist(data, bins=[0,10,20,30,40,50,60], weights=[10,1,40,33,6,8],
                                        edgecolor="black", color="red")
plt.title("Example of a Histogram")
plt.xlabel("Data Values")
plt.show()
```

Output:

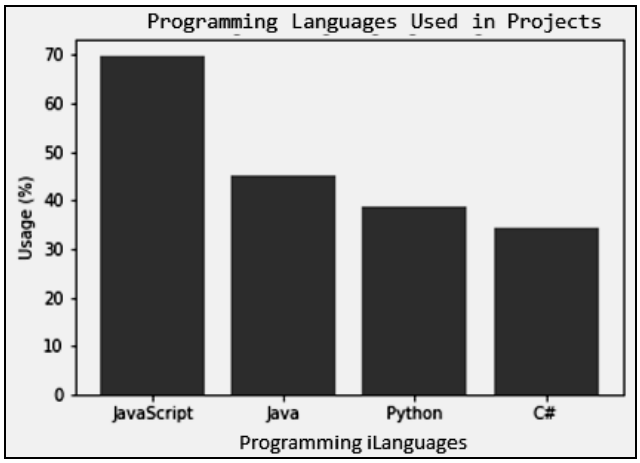


Bar Chart/Graphs:

- Bar charts are used for comparing the quantities of different categories or groups. Values of a category are represented with the help of bars and they can be configured with vertical or horizontal bars, with the length or height of each bar representing the value.
- The major difference between a bar chart and a histogram is that there are gaps between bars in a bar chart but in a histogram, the bars are placed adjacent to each other.
- While the histogram displays the frequency of numerical data, a bar chart uses bars to compare different categories of data.
- Thus, if it quantitative data, histograms should be used, whereas if it is qualitative data, the bar chart can be used.
- A bar chart is a visual representation of values in horizontal or vertical rectangular bars.
- The height is proportional to the values being represented. A bar chart shown in two axes, one axis shows the element and the other axis shows the value of the element (could be time, company, unit, nation, etc.).
- Bar chart represents categorical data with rectangular bars. Each bar has a height corresponds to the value it represents. It's useful when we want to compare a given numeric value on different categories.
- The bar chart is considered as an effective visualization tool for identifying trends and patterns in data or for comparing items between different groups.
- To make a bar chart with Maplotlib, we'll need the plt.bar() function.

```
#Program for a bar graph/chart
import matplotlib.pyplot as plt
import numpy as np
#Creating an array of categorical data
data = (Java script', 'Java', 'Python', 'C#' )
p = [1,2,4,6,8,10]
y = np.arange(len(data))
#Plotting the bar graph
plt.bar(y, p, align='center', alpha=0.5, edgecolor='black')
plt.xticks(y, data)
plt.xlabel('Programming Languages')
plt.ylabel('No. of Usage(%)')
plt.title('Programming Languages Used in Projects')
plt.show()
```

Output:



- From the example, we can see how the usage of the different programming languages compares. Note that some programmers can use many languages, so here the percentages are not summed up to 100%.
- If we change function bar() to barh(), then the bar chart will be displayed horizontally.

Line Plot:

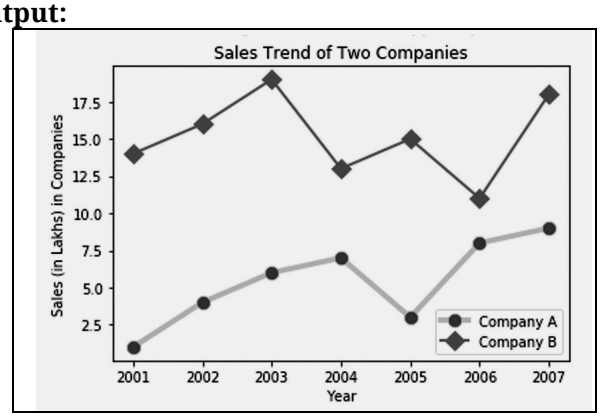
- The line chart is a two-dimensional plotting of values connected following the order. In line chart the values are displayed (or scattered) in an ordered manner and connected.
- A line graph/plot is most frequently used to show trends and analyze how the data has changed over time.
- A line plot displays information as a series of data points called “markers” connected by straight lines.
- In line plot, we need the measurement points to be ordered (typically by their x-axis values).
- This type of plot is often used to visualize a trend in data over intervals of time - a time series. To make a line plot with matplotlib we call plt.plot().
- The first argument is used for the data on the horizontal axis, and the second is used for the data on the vertical axis. This function generates your plot, but it doesn’t display it.
- To display the plot, we need to call the plt.show() function. This is nice because we might want to add some additional customizations to our plot before we display it. For example, we might want to add labels to the axis and title for the plot.
- Line graphs are similar to data plots as in both the cases individual data values are plotted as points in the graph.

- The major difference lies in the connection between points via lines that are provided in a line graph which is not so in case of scatter plots.
- The line graph is more often used when there is a need to study the changein value between two data points in the graph.

- Following Program illustrates the Python code for designing a line chart for displaying the distribution of data along x and y axes.

```
#Program for a Line Chart
import matplotlib.pyplot as plt
#x axis values
x = [2001, 2002, 2003, 2004, 2005, 2006, 2007]
#corresponding y1 and y2 axis values
y1 = [1, 4, 6, 7, 3, 8, 9]
y2 = [14, 16, 19, 13, 15, 11, 18]
# multiple line plot
plt.plot(x, y1, marker='o', markerfacecolor='blue', markersize=10,
                          color='green', linewidth=4, label='Company A')
plt.plot(x, y2, marker='D', markersize=10, color='red',
                                        linewidth=2, label='Company B')
# naming the x axis
plt.xlabel('Year')
# naming the y axis
plt.ylabel('Sales (in Lakhs) in Companies')
# giving a title to graph
plt.title('Sales Trend of Two Companies')
#display legend
plt.legend(loc='lower right')
plt.show()
```

Output:



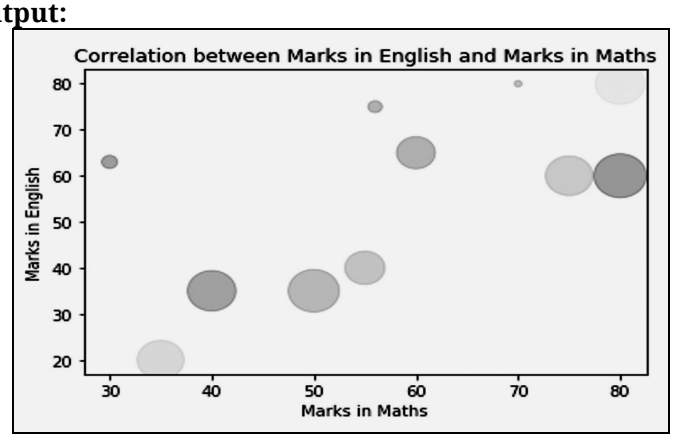
Scatter Plot:

- A scatter plot is a two-dimensional chart showing the comparison of two variables scattered across two axes.
- The scatter plot is also known as the XY chart as two variables are scattered across X and Y axes.
- Scatter plot shows all individual data points. Here, they aren't connected with lines. Each data point has the value of the x-axis value and the value from the y-axis values.
- This type of plot can be used to display trends or correlations. It can be used to study the relationship between two variables. In data science, it shows how 2 variables compare.
- The pattern of the plotted values indicates the pattern of correlation between two variables.
- A scatter plot displays or plots the values of two sets of data placed on two dimensions (usually denoted by X and Y).
- Each dot indicates one observation of data that is placed on the scatter plot by plotting it against the X (horizontal) axis and the Y(vertical) axis.
- To make a scatter plot with Matplotlib, we can use the plt.scatter()function. Again, the first argument is used for the data on the horizontal axis, and the second - for the vertical axis.
- The scatter plot is drawn using the matplotlib library and the scatter() function of the library is used to design the circular dots based on data provided.
- The following program output shows the title of the bar graph is given as Correlation between Marks in English and Marks in Maths.
- The show() function ultimately displays the scatterplot as output.

```
#Program for a Scatter Plot
import matplotlib.pyplot as plt
#Storing values of Two Variables on X and Y Axes
X=[60,55,50,56,30,70,40,35,80,80,75]
Y=[65,40,35,75,63,80,35,20,80,60,60]
#The scattered dots are of different colors and sizes
rng = np.random.RandomState(0)
colors = rng.rand(11)
sizes = 1000 * rng.rand(11)
#Displaying the scatter plot
plt.scatter(X, Y, c=colors, s=sizes, alpha=0.3, marker='o')
plt.xlabel('Marks in Maths')
```

```
plt.ylabel('Marks in English')
plt.title('Correlation between Marks in English and Marks in Maths')
plt.show()
```

Output:



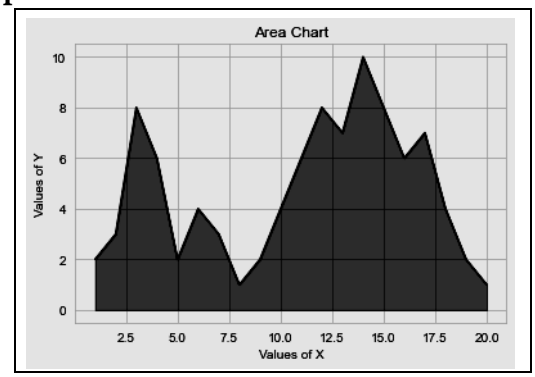
Area Plot/Chart:

- Area charts are used to plot data trends over a while to show how a value is changing.
- The area charts can be rendered for a data element in a row or a column of a data table such as the Pandas data frame.
- An area plot is created as a line chart with an additional filling up of the area with a color between the X-axis.
- An area plot represents the change in data value throughout the X-axis. If more than one data value is considered at the same time, the plotted diagram is called a stacked area chart.
- In Python, an area chart can be created using the fill_between() or stackplot() function of the matplotlib library.
- Following Program illustrates the Python code for designing an area chart for displaying the distribution of data along x and y axes.
- The program shows how grids can be applied to the background and how the area can be displayed based on the plotted values with a chosen color and transparency level.
- The seaborn library is used in the program to create a gridded structure for the plot. The show() function ultimately displays the area plot as output.

```
#Program for Designing an Area Chart
import matplotlib.pyplot as plt
import seaborn as sns
# create data
x=range(1, 21)
y=[2,3,8,6,2,4,3,1,2,4,6,8,7,10,8,6,7,4,2,1]
```

```
#for applying grids in the chart
sns.set_style("whitegrid")
# Change the color and its transparency
plt.fill_between( x, y, color="blue", alpha=0.5)
#for adding a line to the area
plt.plot(x, y, color="black", alpha=0.8)
plt.title("Area Chart")
plt.xlabel('Values of X')
plt.ylabel('Values of Y')
plt.show()
```

Output:



Pie Chart:

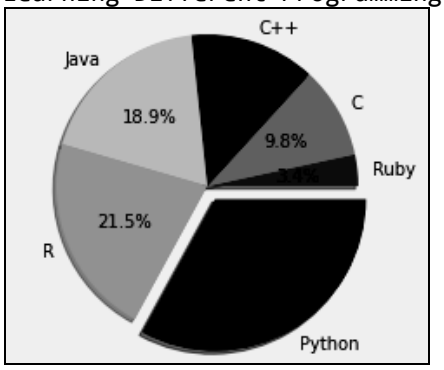
- A pie chart shows the proportion or percentage of a data element in a circular format. The circular chart is split into various pies based on the value/percentage of the data element to highlight.
- The pies represent the "part-of-the-whole" data. The overall sum of pies corresponds to the 100% value of the data being visualized.
- Pie charts are a very effective tool to show the values of one type of data.
- Pie chart is a circular statistical graph which decides slices to illustrate numerical proportion.
- It is a circular plot, divided into slices to show numerical proportion. They are widely used in the business world.
- However, many experts recommend avoiding them. The main reason is that it's difficult to compare the sections of a given pie chart.
- Also, it's difficult to compare data across multiple pie charts. In many cases, they can be replaced by a bar chart.
- In the following program, the Python code for designing a pie chart for displaying the distribution of data based on the proportion. The number of slices is based on the total number of data values provided.

- The pie chart is drawn using the matplotlib library and the pie() function of the library is used to design the sliced area based on data values provided for each label (22 for scala, 64 for C, and more.).
- The title of the bar graph is given as the Percentage ofStudents Learning Different Programming Languages. The show() function ultimately displays the pie chart as output.

```
#Program for Designing a Pie Chart
# -*- coding: utf-8 -*
#Program for Designing a Box Plot
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
# Labeled Data to plot
labels = 'Ruby', 'C', 'C++', 'Java', 'R', 'Python'
sizes = [22, 64, 88, 123, 140, 215]
colors = ['red', 'gray', 'blue', 'pink', 'orange', 'black']
explode = (0, 0, 0, 0, 0, 0.1) # explode a slice if required
#Create a donut chart
plt.pie(sizes, explode=explode, labels=labels,
 colors=colors, autopct='%1.1f%%', shadow=True)
#draw a circle at the center of pie to make it look like a donut
circle = plt.Circle((0,0),0.75,color='black', fc='white',
linewidth=1.25)
Figure = plt.gcf()
fig.gca().add_artist(circle)
# Set aspect ratio to be equal so that pie is drawn as a circle.
plt.axis('equal')
plt.title("Percentage of Students Learning Different
                                        Programming Languages")
plt.show()
```

Output:

Percentage of Students Learning Different Programming Languages



Donut Chart:

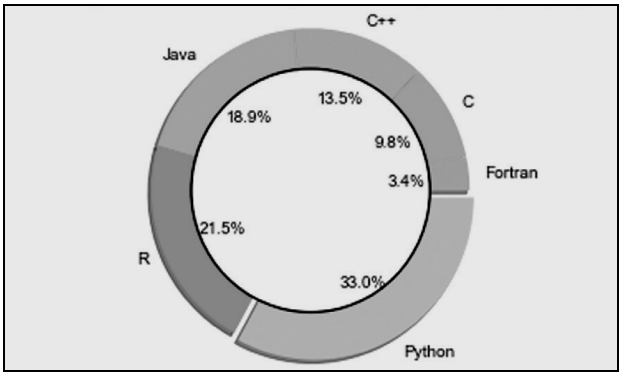
- A doughnut (or a donut) chart is an extension of a pie chart. The center part of the doughnut chart is empty to showcase additional data/metrics or expanded compositions of a pie or showcase another data element.
- The donut charts are considered more space-efficient than the pie chart as the blank inner space in a donut chart can be used to display percentage or any other information related to the data series.
- Since, slices are not provided in a pie shape, an analyst can focus on the arc lengths rather than the slice sizes.
- A donut chart is similar to a pie chart with the main difference in that an area of the center is cut out to give the look of a doughnut.
- The following program illustrates the Python code for designing a donut chart for displaying the distribution of data based on the proportion.
- The number of slices is based on the total number of data values provided. The data can be either numerical or categorical in nature.
- In Python, a donut chart can be created using the pie() function of the matplotlib library.
- The labeled data values and their corresponding distribution in numbers can be stored in an array as shown in the program.
- Each slice color of the pie can also be controlled by mentioning the color names for each slice in the code.
- To give the look of a donut that has a big hole at the center, the Circle() function is used.
- The title of the bar graph is given as the Percentage of Students Learning Different Programming Languages.

```
#Program for Designing a Donut Chart
import matplotlib.pyplot as plt
# Labeled Data to plot
labels = 'Fortran', 'C', 'C++', 'Java', 'R', 'Python'
sizes = [22, 64, 88, 123, 140, 215]
colors = ['red', 'green', 'skyblue', 'pink', 'orange', 'brown']
explode = (0, 0, 0, 0, 0, 0.1) # explode a slice if required
#Create a donut chart
plt.pie(sizes, explode=explode, labels=labels,
                    colors=colors, autopct='%1.1f%%', shadow=True)
```

```
#draw a circle at the center of pie to make it look like a donut
circle = plt.Circle((0,0),0.75,color='black', fc='white',
                                                    linewidth=1.25)
Figure = plt.gcf()
fig.gca().add_artist(circle)
# Set aspect ratio to be equal so that pie is drawn as a circle.
plt.axis('equal')
plt.title("Percentage of Students Learning
                                        Different Programming Languages")
plt.show()
```

Output:

```
Percentage of Students Learning Different Programming Languages
```



Basic Visualization Rules:

- The basic visualization rules help us make nice and informative plots instead of confusing ones.

1. To choose the appropriate plot type. If there are various options, we can try to compare them, and choose the one that fits our modehe best.
2. When we choose your type of plot, one of the most important things is to label your axis. If we don't do this, the plot is not informative enough. When there are no axis labels, we can try to look at the code to see what data is used and understand the plot.
3. We can add a title to make our plot more informative.
4. Add labels for different categories when needed.
5. Optionally we can add a text or an arrow at interesting data points.
6. In some cases we can use some sizes and colo**rs** of the data to make the plot more informative.

4.5 SPECIALIZED DATA VISUALIZATION TOOLS

- This section illustrate the specialized data visualization tools such as Boxplots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots.
- Plots allow distributing two or more data sets over a 2D or even 3D space to show the relationship between these sets and the parameters on the plot.
- Plots also vary: scatter and bubble plots are the most traditional. Though when it comes to big data, analysts use box plots that enable to visualize the relationship between large volumes of different data.

Box Plot:

- Box plot is a commonly used chart for business, professional aspects and extensively in data science-related visualizations.
- A box plot is used to show the distribution of two or more data elements in a summarized manner.
- A box plot is a graph that gives us a good indication of how the values in the data are spread out.
- It is also called the box-and-whisker plot which shows the distribution of values based on the five-number summary namely, minimum, first quartile, median, third quartile and maximum.
- The minimum and the maximum are just the min and max values from our data.
- The median is the value that separates the higher half of a data from the lower half. It's calculated by the steps like order the values and find the middle one.
- In a case when our count of values is even, we actually have 2 middle numbers, so the median here is calculated by summing these 2 numbers and divide the sum by 2. For example, if we have the numbers 1, 2, 5, 6, 8, 9, your median will be (5 + 6) / 2 = 5,5.
- The first quartile is the median of the data values to the left of the median in our ordered values. For example, if we have the numbers 1, 3, 4, 7, 8, 8, 9, the first quartile is the median from the 1, 3, 4 values, so it's 3.
- The third quartile is the median of the data values to the right of the median in our ordered values. For example, if we use these numbers 1, 3, 4, 7, 8, 8, 9 again, the third quartile is the median from the 8, 8, 9 values, so it's 8.
- We also want to mention one more statistic here. That is the IQR (Interquartile Range). The IQR approximates the amount of spread in the middle 50% of the data. The formula is the third quartile - the first quartile.
- This type of plot can also show outliers. An outlier is a data value that lies outside the overall pattern.
- They are visualized as circles. When we have outliers, the minimum and the maximum are visualized as the min and the max values from the values which aren't outliers. There are many ways to identify what is an outlier.

- A commonly used rule says that a value is an outlier if it's less than the first quartile - 1.5 * IQR or high than the third quartile + 1.5 * IQR.

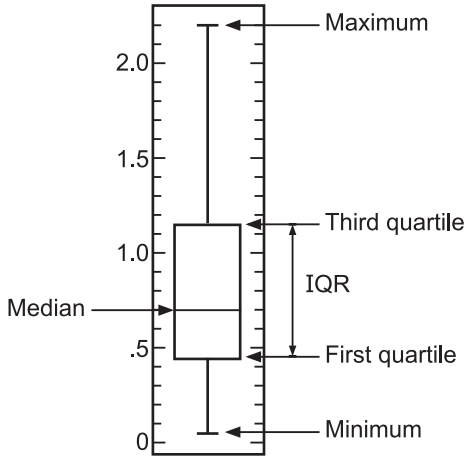


Fig. 4.3

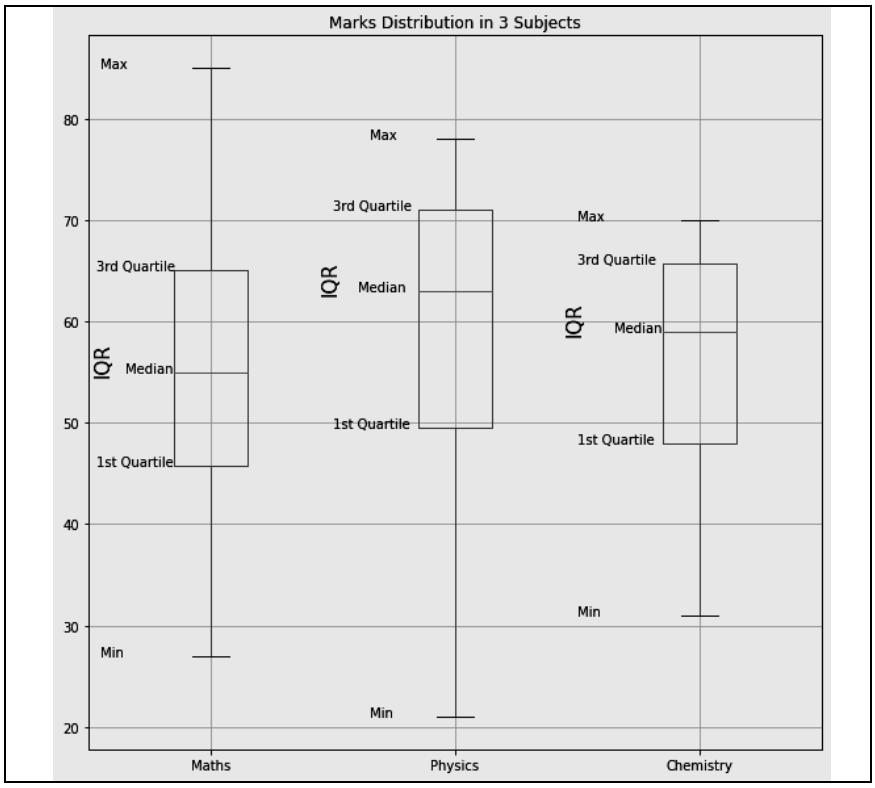
- We need is the function plt.boxplot() to create box plot. The first argument is the data points.

```
#Program for Designing a Box Plot
import pandas as pd
import matplotlib.pyplot as plt
#Create a Dataframe
d={'Name':['Ash','Sam','Riha','Morgan','Ria','Tina','Raj','Rahul',
                    'Don','Ann',' Ajay','Akbar'],
                    'Maths':[60,47,55,74,30,55,85,63,42,27,71,50],
                    'Physics': [57,42,60,70,21,66,78,74,52,40,67,77],
                    'Chemistry':[65,62,48,50,31,48,60,68,32,70,70,58]}
#print the Dataframe
df = pd.DataFrame(d)
print(df)
#Boxplot Representation of the Maths Column
df.boxplot(column=["Maths", "Physics", "Chemistry"],grid=True, figsize=(
10,10))
plt.title("Marks Distribution in 3 Subjects")
plt.text(x=0.53, y=df["Maths"].quantile(0.75), s="3rd Quartile")
plt.text(x=0.65, y=df["Maths"].median(), s="Median")
```

```
plt.text(x=0.53, y=df["Maths"].quantile(0.25), s="1st Quartile")
#Data Plotting and Visualization
plt.text(x=0.55, y=df["Maths"].min(), s="Min")
plt.text(x=0.55, y=df["Maths"].max(), s="Max")
plt.text(x=0.52, y=df["Maths"].quantile(0.50), s="IQR",
                                                rotation=90, size=15)
#Boxplot Representation of the Physics Column
plt.text(x=1.5, y=df["Physics"].quantile(0.75), s="3rd Quartile")
plt.text(x=1.6, y=df["Physics"].median(), s="Median")
plt.text(x=1.5, y=df["Physics"].quantile(0.25), s="1st Quartile")
plt.text(x=1.65, y=df["Physics"].min(), s="Min")
plt.text(x=1.65, y=df["Physics"].max(), s="Max")
plt.text(x=1.45, y=df["Physics"].quantile(0.50), s="IQR",
                                                rotation=90, size=15)
#Boxplot Representation of the Chemistry Column
plt.text(x=2.5, y=df["Chemistry"].quantile(0.75), s="3rd Quartile")
plt.text(x=2.65, y=df["Chemistry"].median(), s="Median")
plt.text(x=2.5, y=df["Chemistry"].quantile(0.25), s="1st Quartile")
plt.text(x=2.5, y=df["Chemistry"].min(), s="Min")
plt.text(x=2.5, y=df["Chemistry"].max(), s="Max")
plt.text(x=2.45, y=df["Chemistry"].quantile(0.50), s="IQR",
                                                 rotation=90, size=15)
```

Output:

```
In [36]: runfile('C:/Users/HP/.spyder-py3/temp1.py', wdir='C:/Users/HP/.spyder-py3')
      Name  Maths  Physics  Chemistry
0      Ash     60       57         65
1      Sam     47       42         62
2     Riha     55       60         48
3   Morgan     74       70         50
4      Ria     30       21         31
5     Tina     55       66         48
6      Raj     85       78         60
7    Rahul     63       74         68
8      Don     42       52         32
9      Ann     27       40         70
10    Ajay     71       67         70
11   Akbar     50       77         58
```

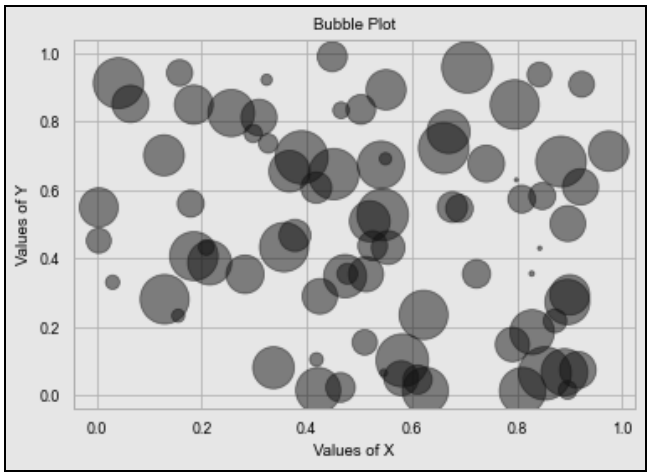


Bubble Plots:

- A bubble chart is a variation of a scatter chart in which the data points are replaced with bubbles and an additional dimension of the data is represented in the size of the bubbles.
- A bubble plot is a scatter plot where a third dimension is added: the value of an additional numeric variable is represented through the size of the dots.
- We need three numerical variables as input: one is represented by the X axis, one by the Y axis, and one by the dot size.
- This plot always:
  1. Include a legend if more than one category of data is being visualized.
  2. Ensure that smaller dots are visible when overlapping with larger dots.
  3. Either by placing smaller dots above larger dots or by making the larger dots transparent.
- **Note:** This plot Don't use a bubble chart if there are an excessive number of values that result in the dots appearing illegible.
- Bubble charts are typically used to compare and show the relationships between categorised circles, by the use of positioning and proportions.

```
#Program for Designing a Bubble Plot
import matplotlib.pyplot as plt
import numpy as np
# create data
x = np.random.rand(80)
y = np.random.rand(80)
z = np.random.rand(80)
# use the scatter function to create a bubble plot
plt.scatter(x, y, s=z*1000, c="red", alpha=0.5)
plt.title("Bubble Plot")
plt.xlabel("Values of X")
plt.ylabel("Values of Y")
plt.show()
```

Output:



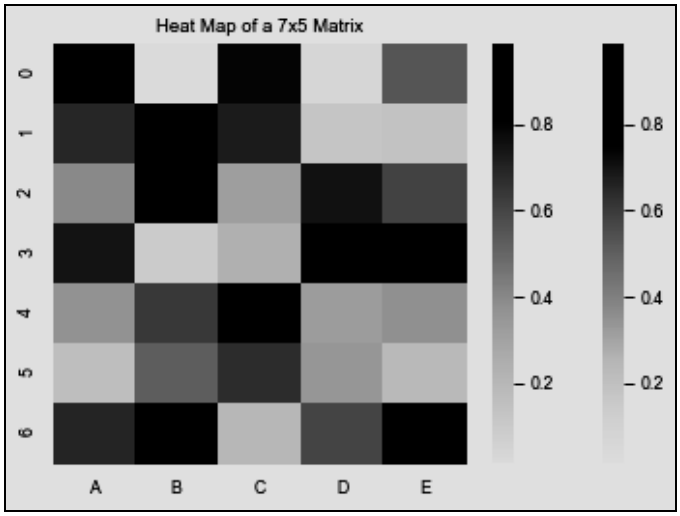
Heat Map:

- A heat map is a tool to show the magnitude of data elements using colors.
- The intensity (or hue) of the colors is shown in a two-dimensional manner, showing how close the two elements are correlated.
- A heat map is data analysis software that uses color the way a bar graph uses height and width: as a data visualization tool.
- Heat maps visualize data through variations in coloring. When applied to a tabular format.
- Heat maps are useful for cross-examining multivariate data, through placing variables in the rows and columns and coloring the cells within the table.

- A heat map represents data in a two dimensional format in which each data value is represented by a color in the matrix.
- Since colors play a major role in displaying a heat map, many different color schemes can be used for illustrating a heat map.
- By definition, heat map visualization or heat map data visualization is a method of graphically representing numerical data where the value of each data point is indicated using colors.
- The most commonly used color scheme used in heat map visualization is the warm-to-cool color scheme, with the warm colors representing high-value data points and the cool colors representing low-value data points.
- In the world of online businesses, website heat maps are used to visualize visitor behavior data so that business owners, marketers and UX designers can identify the best-performing sections of a webpage based on visitor interaction and the sections that are performing sub-par and need optimization.
- Heatmaps were first developed in the 1800s, originating in the 2D display of data in a matrix.
- Heatmaps help measure a website's performance, simplify numerical data, understand visitors' thinking, identify friction areas by identifying dead clicks and redundant links, and ultimately make changes that positively impact conversion rates.
- Netflix is perhaps one of the best examples of a digital business that uses heatmaps to gain insights on user behavior and improve user experiences.

```
#Program for Designing a Heat Map
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Create a 7x7 matrix dataset
df = pd.DataFrame(np.random.random((7,5)),
                                        columns=["A","B","C","D","E"])
# plot heat map using a color palette
sns.heatmap(df, cmap="YlGnBu")
sns.heatmap(df, cmap="Blues")
plt.title("Heat Map of a 7x5 Matrix")
```

Output:



- Heat maps rely on colors to express the variation in data – the darker shades of color indicate more quantity while the lighter shades of color indicate less quantity.

Dendrogram:

- A dendrogram is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering.
- The main use of a dendrogram is to work out the best way to allocate objects to clusters. A dendrogram is a diagram representing a tree.
- A dendrogram is mainly used for the visual representation of hierarchical clustering to illustrate the arrangement of various clusters formed using data analysis.
- A dendrogram can also be used in phylogenetics to illustrate the evolutionary relationships among the biological taxa.
- In computational biology, a dendrogram can be used to illustrate the group of samples or genes based on similarity.
- The dendrogram below shows the hierarchical clustering of six observations shown on the scatterplot to the left.

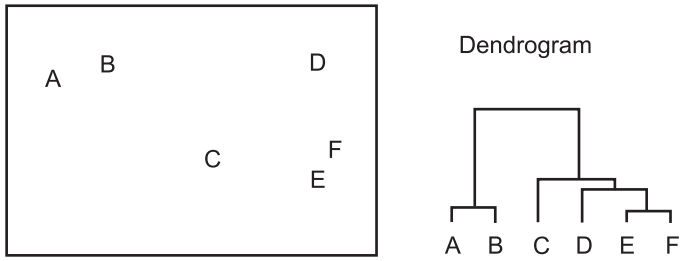


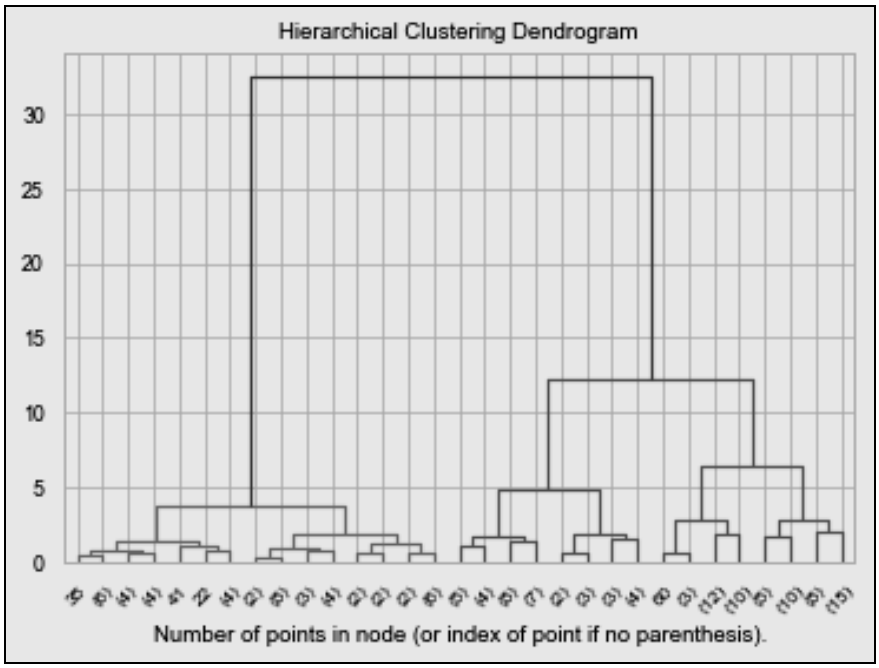
Fig. 4.4

- The key to interpreting a dendrogram is to focus on the height at which any two objects are joined together.
- In the example above, we can see that E and F are most similar, as the height of the link that joins them together is the smallest. The next two most similar objects are A and B.
- In the dendrogram above, the height of the dendrogram indicates the order in which the clusters were joined.
- A more informative dendrogram can be created where the heights reflect the distance between the clusters as is shown below. In this case, the dendrogram shows us that the big difference between clusters is between the cluster of A and B versus that of C, D, E, and F.
- The following Python code is for designing a dendrogram uses the mtcars dataset to calculate the distance between the samples based on the ward featureThe dendrogram is created using the dendrogram() function of the scipy library.
- The text labels are provided to the right and the topmost threshold line of the dendrogram is displayed in black color.

```
#Program for Designing a Dendrogram
import numpy as np
from matplotlib import pyplot as plt
from scipy.cluster.hierarchy import dendrogram
from sklearn.datasets import load_iris
from sklearn.cluster import AgglomerativeClustering
def plot_dendrogram(model, **kwargs):
    # Create linkage matrix and then plot the dendrogram
    # create the counts of samples under each node
    counts = np.zeros(model.children_.shape[0])
    n_samples = len(model.labels_)
    for i, merge in enumerate(model.children_):
        current_count = 0
        for child_idx in merge:
            if child_idx < n_samples:
                current_count += 1  # leaf node
            else:
                current_count += counts[child_idx - n_samples]
        counts[i] = current_count
    linkage_matrix = np.column_stack([model.children_, model.distances_,
                                      counts]).astype(float)
```

```
        # Plot the corresponding dendrogram
        dendrogram(linkage_matrix, **kwargs)
    iris = load_iris()
    X = iris.data
    # setting distance_threshold=0 ensures we compute the full tree.
    model = AgglomerativeClustering(distance_threshold=0, n_clusters=None)
    model = model.fit(X)
    plt.title('Hierarchical Clustering Dendrogram')
    # plot the top four levels of the dendrogram
    plot_dendrogram(model, truncate_mode='level', p=4)
    plt.xlabel("Number of points in node (or index of point
                                                    if no parenthesis).")
    plt.show()
```

Output:



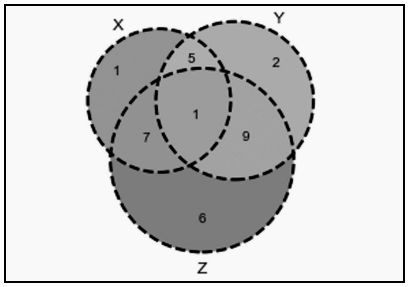
Venn Diagram:

- A Venn diagram (also called primary diagram, set diagram or logic diagram) is a diagram that shows all possible logical relationships between a finite collections of different sets.
- Each set is represented by a circle. The circle size sometimes represents the importance of the group but not always.

- The groups are usually overlapping: the size of the overlap represents the intersection between both groups.
- A Venn diagram is a diagram that visually displays all the possible logical relationships between collections of sets. Each set is typically represented with a circle.
- The Venn diagram is most commonly used to:
  1. **Visually organize** information to quickly grasp the relationship between datasets and identify differences or commonalities.
  2. **Compare and contrast** two or more choices to identify the overlapping elements and clearly see how they fare against each other. It's useful when you're trying to make an important decision, such as making an investment or buying a new product or service.
  3. **Find correlations and predict probabilities** when comparing datasets.
- Venn diagram helps to bring data together in a visual way, allowing to analyze findings more efficiently and identify all possible logical relationships between a collection of sets.
- The following Python code is for designing a Venn diagram. For three sets labeled as X, Y, and Z. The function used in Python to plot the Venn diagram is either venn2() or venn3() found in the matplotlib library. The venn2() function is used if there are two sets to be considered whereas the venn3() function is used if there are three sets to be considered.

```
#Program for a Venn Diagram
from matplotlib import pyplot as plt
from matplotlib_venn import venn3, venn3_circles
# Create the Venn Diagram
v = venn3(subsets=(1, 2, 5, 6, 7, 9, 1, 3, 4, 5, 6, 8,3, 5, 6, 7, 8,10),
                                        set_labels = ('X', 'Y', 'Z'))
c = venn3_circles(subsets=(1, 2, 5, 6, 7, 9, 1, 3, 4, 5, 6, 8,3, 5, 6,7,
                                        8, 10), linestyle='dashed')
# Add the title and annotation
plt.show()
```

Output:



Treemap:

- A treemap is a visualization that displays hierarchically organized data as a set of nested rectangles, and the sizes and colors of rectangles are proportional to the values of the data points they represent.
- Treemaps are a data-visualization technique for large, hierarchical data sets. They capture two types of information in the data namely, the value of individual data points and the structure of the hierarchy.
- A treemap is a visual tool that can be used to break down the relationships between multiple variables in the data.
- **Definition:** Treemaps are visualizations for hierarchical data. They are made of a series of nested rectangles of sizes proportional to the corresponding data value. A large rectangle represents a branch of a data tree, and it is subdivided into smaller rectangles that represent the size of each node within that branch.
- Data, organized as branches and sub-branches, is represented using rectangles, the dimensions and plot colors of which are calculated w.r.t the quantitative variables associated with each rectangle-each rectangle represents two numerical values.
- We can drill down within the data to, theoretically, an unlimited number of levels. This makes the at-a-glance distinguishing between categories and data values easy.
- Fig. 4.5 (a) provides the hierarchical data consisting of three levels – the first level has only one cluster P, the second level consists of two clusters Q and R, while the third lowermost level consists of five clusters, S, T, U, V, and W.
- The data values of each cluster are proportionally divided among each other. The corresponding tree map chart for the given hierarchical data is displayed in Fig. 4.5 (b).
- The tree map chart consists of eight nodes or rectangular structures divided proportionately based on the data values provided in the hierarchical structure.

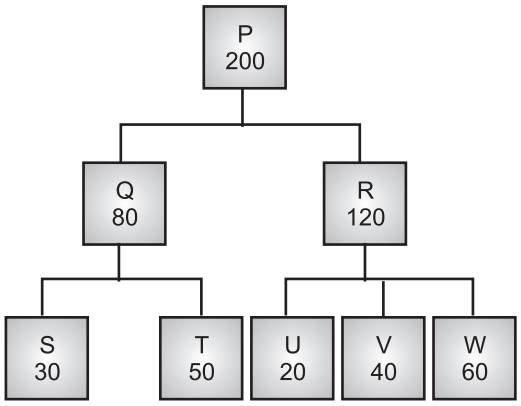


Fig. 4.5 (a)

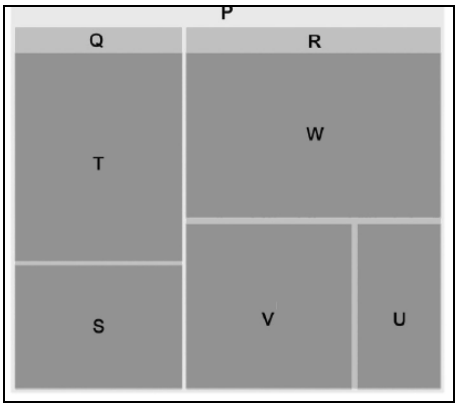
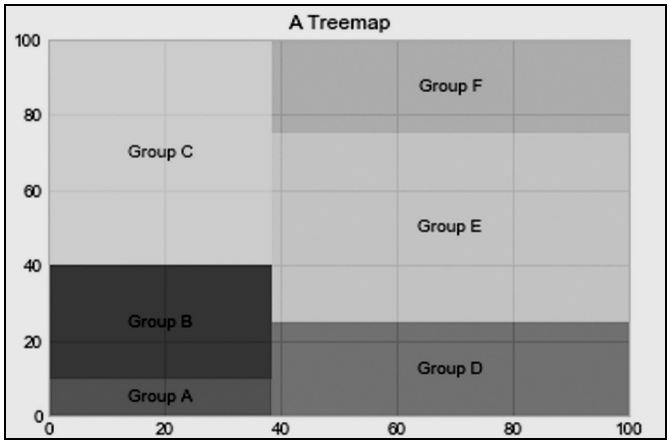


Fig. 4.5 (b)

```
#Program for a Treemap
import matplotlib.pyplot as plt
import squarify
# Plotting a Treemap
squarify.plot(sizes=[5,15,30,20,40,20], label=["Group A", "Group B",
      "Group C", "Group D", "Group E", "Group F"], color=["red","pink",
            "orange", "green","blue","yellow"], alpha=0.6)
plt.axis('on')
plt.title('A Treemap')
# Show the graph
plt.show()
```

Output:



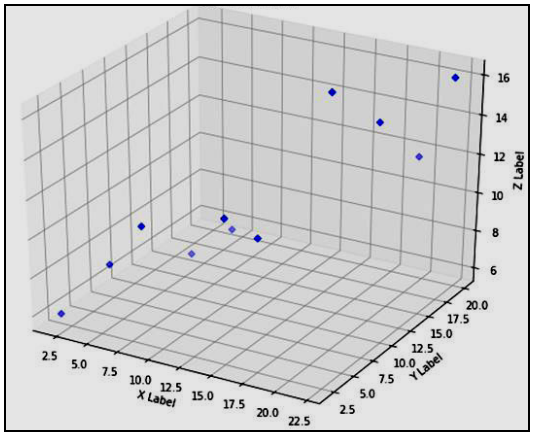
3D Scatter Plots:

- It's becoming increasingly common to visualize 3D data by adding a third dimension to a scatter plot.
- The 3D scatter plots are used to plot data points on three axes in the attempt to show the relationship between three variables.
- Each row in the data table is represented by a marker whose position depends on its values in the columns set on the X, Y, and Z axes.
- A fourth variable can be set to correspond to the color or size of the markers, thus adding yet another dimension to the plot.
- The relationship between different variables is called correlation. If the markers are close to making a straight line in any direction in the three-dimensional space of the 3D scatter plot, the correlation between the corresponding variables is high.

- If the markers are equally distributed in the 3D scatter plot, the correlation is low, or zero. However, even though a correlation may seem to be present, this might not always be the case.
- The variables could be related to some fourth variable, thus explaining their variation, or pure coincidence might cause an apparent correlation.
- 3D scatter plot is one such visualization tool that can represent various data series in one graph with the 3D effect.
- The three-dimensional axes can be created by mentioning the projection='3d' for the add_subplot()function. The 3D plot is a collection of scatter plots created from the sets of (x,y,z) dataset.

```
#Program for a 3D Scatter Plot
import matplotlib.pyplot as plt
# size of the plotted figure
Figure = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection='3d')
#Assign values for the 3D Plot
x =[2,4,6,8,10,12,14,16,18,20,22]
y =[2,5,6,9,11,7,8,14,17,19,20]
z =[6,8,10,8,9,11,10,16,14,12,16]
#Plot the 3D Scatter Plot
ax.scatter(x, y, z, c='r', marker='D')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
plt.show()
```

Output:



4.6 ADVANCED DATA VISUALIZATION TOOL : WORD CLOUDS

- There are more advanced and complex visualization tools that are used in data analytics namely, word clouds, waffle charts and seaborn plots.
- A word cloud (or *tag cloud*) is a word visualization that displays the most used words in a text from small to large, according to how often each appears.
- Word clouds (also known as text clouds or tag clouds) work in a simple way: the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud.
- A wordcloud or tag cloud is a visual representation of textual data that presents a list of words.
- A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.
- Wordclouds are more often used for aesthetic purposes to depict categorical data.
- Python uses the wordcloud library developed by Andreas Mueller to display the wordcloud visualization tool.
- There are different ways by which Word Clouds can be created but the most widely used type is by using the Frequency of Words in our corpus. And thus, we will be creating our Word Cloud by using the Frequency type.
- The following is the example of word cloud.

Fig. 4.6

Limitation using a word cloud:

1. **When your data isn't optimized for context.** Simply dumping text into a word cloud generator isn't going to give you the deep insights you want. Instead, an optimized data set will give you accurate insights.

2. **When another visualization method would work better.** It's easy to think "Word Clouds are neat!" and overuse them - even when a different visualization should be used instead. We need to make sure you understand the right use case for a word cloud visualization.

4.7 VISUALIZATION OF GEOSPATIAL DATA

- Geospatial data also referred to as spatial data, GIS data or geodata.
- Geospatial data consists of numeric data that denotes a geographic coordinate system (latitude, longitude, and elevation) of a geographical location of a physical object such as as small as a building or a street, or as big as a city, a state, or a country.
- This geodata gives us information about the location, size, area and shape of a physical object.
- Maps are the primary focus of geospatial visualizations. They range from depicting a street, town or park or subdivisions to showing the boundaries of a country, continent, or the whole planet.
- They act as a container for extra data. This allows you to create context using shapes and color to change the visual focus.
- Geospatial visualizations highlight the physical connection between data points. This makes them susceptible to a few common pitfalls that may introduce error:
  1. **Scaling:** Changes in the size of the map can affect how the viewer interprets the data.
  2. **Auto-correlation:** A view may create an association between data points appearing close on a map, even for unrelated data.
- Python is highly efficient to deal with geospatial data and for this many libraries have been developed in Python to deal with mainly GIS data.
- The standard Python libraries commonly used for geospatial data are explained below:
  1. The **shapely Python library** is mainly used to create geometric objects such as a square, polygon, or even a point. Basic geometric calculations such as finding area or intersection can also be handled by this library.
  2. The **geopandas Python library** is more powerful than the shapely library as well as the fiona library as it can not only create geometric objects but also read/write vector file formats and handle projection conversions.
  3. The **gdal (Geospatial Data Abstraction Library)** is originally written in C and is used often to deal with geospatial data. As a library, it presents a single abstract data model to the calling application for all supported formats.

4. The **fiona library** in Python is mainly used to read/write vector file formats such as shapefiles, or handle projection conversions.
5. The **rasterio library** in Python is mainly used to handle raster data and handles transformations of coordinate reference systems. This library uses the matplotlib library to plot data for analysis.

- Geographic data visualization is a constructive practice that integrates interactive visualization into traditional maps, allowing the ability to explore different layers of a map, zoom in and out, change the visual appearance of the map, and relate a variety of factors to the geographic area.
- This section will discuss the three tools used in Python for creating geospatial data namely, Choropleth map, Bubble map and Connection map.

Choropleth Map:

- These maps contain areas that are shaded or patterned in proportion to the statistical variable being displayed on the map.
- A choropleth map is a map containing partitioned geographical regions or areas. The areas are divided based on colors, shades, or patterns in relation to a data variable.
- It is Filled maps for showing ratio and rate data in defined areas.
- The chloropleth maps are great for intuitively visualizing geographic clusters or concentrations of data.
- However, a chloropleth map could be misleading if the size of a region overshadows its color.
- Big regions naturally attract attention, so large areas might get undue importance in a chloropleth map while small regions are overlooked.
- There are mainly two elements required to build a choropleth map:
  - A shape file that gives the boundaries of every zone to be represented on the map.
  - A data frame that gives the values of each zone.
- Choropleth maps display divided geographical areas or regions that are colored, shaded or patterned in relation to a data variable.
- This provides a way to visualize values over a geographical area, which can show variation or patterns across the displayed location. The data variable uses color progression to represent itself in each region of the map.
- The Fig. 4.7 shows an example of a choropleth map that is created in Python by using the folium library which shows the population density of each region.

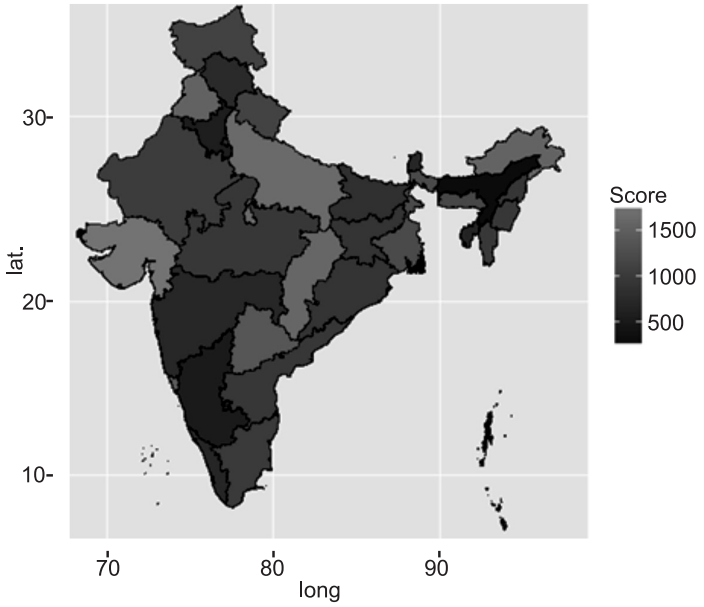


Fig. 4.7: An example of a Choropleth Map shows Population Density of each Region in India

Connection Map:

- Connection maps are drawn by connecting points placed on a map by straight or curved lines.
- A connection map shows the connections between several positions on a map. Connection Map is used to display network combined with geographical data.
- A connection map is a map that shows the connection between numerous positions on a map. Two positions of a map are connected via lines and the positions are marked by circles.
- The connection map that is created in Python by using the basemap library. Each line in a map usually indicates the shortest route between the two positions.
- A connection Map helps to plot connections between two data points on a map as shown in Fig. 4.8.

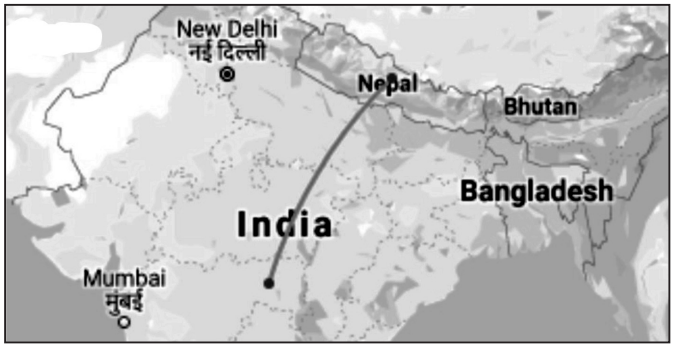


Fig. 4.8: An example of Connection Map

Bubble Map:

- A bubble map uses circles of different size to represent a numeric value on a territory. It displays one bubble per geographic coordinate, or one bubble per region.
- Bubble map uses range of colored bubbles of different sizes in visualization of data. A bubble map shows circular markers for points, lines, and polygon features of different size.
- A bubble map uses circles of different size to represent a numeric value on a territory. It displays one bubble per geographic coordinate, or one bubble per region (in this case the bubble is usually displayed in the baricentre of the region).
- A bubble map is a map containing markers. The markers are given by bubbles of varying sizes that indicate a numeric value. The bubbles can be added to a map by using the basemap library in Python.
- A bubble map uses size as an encoding variable. The size of the circle represents a numeric value on a Geographic area.

Fig. 4.9: Bubble Map

- We often use Choropleth maps to display areas, and in that case, we use a colour encoding. The choropleth maps have an inherent bias problem with large areas.
- In contrast, bubble maps use circles to represent a numeric value of an area or region.

4.8 DATA VISUALIZATION TYPES

- All data visualization graphs discussed in the previous sections can be clustered together into several groups based on the type of data it is dealt with and/or the purpose for which the graphical data is represented.

Sr. No.	Categorial Data	Description	Data Visualization Graphs
1.	Temporal data	Data visualizations are linear and one dimensional.	Scatter plot Line chart Time series sequence

Contd...

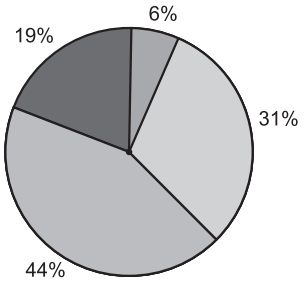
2.	Hierarchical data	Data visualizations having ordered groups within larger groups, each group or cluster of information flowing from a single point of origin.	Dendrogram Ring charts Tree map Sunburst diagram
3.	Network data	Data visualizations showing the relationships among data within a network.	Matrix chart Node-link diagram Word cloud Alluvial diagram Tube map
4.	Multidimensional data	Data visualizations having multiple dimensions.	Scatter plot Pie chart Venn diagram Stacked bar graph Histogram
5.	Multivariate data	Data visualizations having more than two variables to be studied under a single observation.	3D Scatter plot Scatterplot matrix HyperSlice Hyperbox Parallel coordinate Radial coordinate
6.	Geospatial data	Data visualizations relate to real-life physical locations, overlaying familiar maps with different data points.	Bubble map Choropleth map Connection map Heat map

PRACTICE QUESTIONS

Q.I Multiple Choice Questions:

1. Which is a graphical representation of quantitative information and data by using visual elements like graphs, charts, and maps.?
   (a) Data visualization (b) Data preprocessing
   (c) Data reduction (d) Data discretization
2. Data visualization can help in,
   (a) identifying outliers in data
   (b) displaying data in a concise format
   (c) providing easier visualization of patterns
   (d) All of the mentioned

3. Which plot can be used to study the relationship between two variables?
   (a) scatter (b) line
   (c) histogram (d) bar
4. Which chart is created by displaying the markers or the dotted points connected via straight line segments?
   (a) scatter (b) line
   (c) pie (d) bar
5. Which plot is a common statistical graphic used in case of measures of central tendency and dispersion?
   (a) box (b) scatter
   (c) histogram (d) heat map
6. Which map represents data in a two-dimensional format in which each data value is represented by a color in the matrix?
   (a) tree map (b) heat map
   (c) denrogram (d) boxplot
7. Which is specialized visualization tool that plot data in various ways to display informative graphical outputs for data analysis?
   (a) plotly (b) treemaps
   (c) wordcloud (d) scattered plots
8. Data can be visualized using?
   (a) graphs (b) charts
   (c) maps (d) All of the mentioned
9. The best visualization tool that can be used in case of hierarchical clustering is,
   (a) Scatter plot (b) Dendogram
   (c) Histogram (d) Heat map
10. Describe the following type of data visualization.



   (a) It is a histogram used to represent categorical data
   (b) It is a donut chart used to represent numerical data
   (c) It is a pie chart used to represent proportions
   (d) It is a radial chart used for geospatial mapping

11. Which is a Python library used to create 2D graphs and plots?
    (a) NumPy (b) Matplotlib
    (c) SciPy (d) Pandas
12. The visualization tool that contains partitioned geographical regions or areas is called as,
    (a) Choropleth map (b) Connection map
    (c) World map (d) Geospatial map
13. Which data consists of numeric data that denotes a geographic coordinate system (latitude, longitude, and elevation) of a geographical location of a physical object?
    (a) geographical (b) spatial
    (c) geodata (d) All of the mentioned
14. Which charts display data as a cluster of circles?
    (a) bubble (b) bar
    (c) scatter (d) line
15. Which Python library is mainly used to create geometric objects such as a square, polygon, or even a point?
    (a) gdal (b) shapely
    (c) geopandas (d) fiona
16. What is true about data visualization?
    (a) Helps users in analyzing a large amount of data in a simpler way
    (b) Used to communicate information clearly and efficiently to users by the usage of information graphics such as plots and charts
    (c) Makes complex data more accessible, understandable and usable
    (d) All of the mentioned

Answers

1. (a)	2. (c)	3. (d)	4. (b)	5. (c)	6. (a)	7. (c)	8. (d)	9. (b)	10. (a)
11. (c)	12. (a)	13. (d)	14. (b)	15. (d)	16. (c)	17. (a)	18. (b)	19. (c)	20. (d)
21. (b)	22. (c)	23. (d)	24. (b)	25. (d)	26. (c)	27. (a)	28. (d)	29. (c)	30. (a)
31. (b)	32. (a)								

Q.II Fill in the Blanks:

1. _____ is the graphical representation of data that can make information easy to analyze and understand.
2. The _____ data gives us information about the location, size, area, and shape of a physical object.
3. The bubbles in bubble chart can be added to a map by using the _____ library in Python.

4. _____ encoding is the approach used to map data into visual structures, thereby building an image on the screen.
5. _____ are a common statistical graphic used in case of measures of central tendency and dispersion.
6. A _____ is a visual representation of textual data that presents a list of words.
7. _____ is a low level graph plotting library in python that serves as a visualization utility.
8. A _____ chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally.
9. A _____ is mainly used for the visual representation of hierarchical clustering to illustrate the arrangement of various clusters formed using data analysis.
10. The _____ scatter plot is frequently used for comparing the three characteristics of a given dataset.
11. An _____ plot or area chart is created as a line chart with an additional filling up of the area with a color between the X-axis.
12. The _____ library in Python is an online platform for data visualization and it can be used in making interactive plots that are not possible using other Python libraries.

Answers

1. Visualization	2. spatial	3. basemap	4. Visual
5. Boxplots	6. wordcloud	7. Matplotlib	8. bar
9. dendrogram	10. 3D	11. area	12. plotly

Q.III State True or False:

1. Data visualization is the presentation of data in a pictorial or graphical format.
2. The geoplotlib in Python is a toolbox for designing maps and plotting geographical data.
3. A histogram displays data distribution by creating several bars over a continuous interval.
4. Data visualization has the power of illustrating simple data relationships and patterns with the help of simple designs consisting of lines, shapes and colors.
5. A tag cloud is a visual representation of textual data that presents a list of words.
6. The treemap visualization tool is mainly used for displaying hierarchical data that can be structured in the form of a tree.
7. A connection map shows the connections between several positions on a map.
8. A bar chart is created by displaying the markers or the dotted points connected via straight line segments.

9. The pygal library in Python creates interactive plots that can be embedded in the web browser.
10. Geospatial data consists of numeric data that denotes a geographic coordinate system of a geographical location of a physical object.
11. In a bubble plot, a third dimension is added to indicate the value of an additional variable which is represented by the size of the dots.
12. Python has excellent combination of libraries like Pandas, NumPy, SciPy and Matplotlib for data visualization which help in creating in nearly all types of visualizations charts/plots.
13. Pie chart is a chart where various components of a data set are presented in the form of a pie which represents their proportion in the entire data set.
14. Box plot is a convenient way of visually displaying the data distribution through their quartiles.

Answers

1. (T)	2. (T)	3. (T)	4. (F)	5. (T)	6. (T)	7. (T)	8. (F)	9. (T)	10. (T)
11. (T)	12. (T)	13. (T)	14. (T)						

Q.IV Answer the following Questions:

(A) Short Answer Questions:

1. Define data visualization.
2. What is the purpose of data visualization?
3. What is geographical data?
4. Define Exploratory Data Analysis (EDA).
5. What is visual encoding?
6. What is tag cloud?
7. List data visualization libraries in Python.
8. Define box plot?
9. What is the use of bubble plot?
10. Define dendrogram.
11. What is donut charts?
12. Define area chart.

(B) Long Answer Questions:

1. What is data visualization? Why the data visualization important for data analysis? Explain in detail.
2. Explain any five data visualization libraries that are commonly used in Python.
3. How to crate line chart? Explain with example.
4. Write short note on: Data visualization types.

5. What is the various statistical information that a box plot can display? Draw a diagram to represent a box plot containing various statistical information
6. What are the three dimensions used in a bubble plot? Write the Python code to create a simple bubble plot.
7. Mention any two cases when a dendogram can be used. Write the Python code to create a simple dendogram.
8. Write the Python code to create a simple Venn diagram for the following two sets:
   X = {1,3,5,7,9} Y = {2,3,4,5,6}
9. Use Python code to design a 3D scatter plot for the given x, y, and z values.
   X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
   Y = [25, 34, 67, 88, 23, 69, 12, 45, 33, 61]
   Z = [10, 15, 12, 22, 25, 30, 18, 26, 15, 22]
10. What are the roles of choropleth map and connection map? Which Python library function is needed to be used for these two maps?
11. How to visualize geospatial data? Explain in detail.
12. Write short note on: Wordcloud.
13. What is Exploratory Data Analysis (EDA)? Describe in detail.
14. What is Venn digram? How to create it? Explain with example.
15. Differentiate between:
    (i) Histogram and Dendrogram
    (ii) Pie chart and Bar chart
    (iii) Connection map and Choropleth map
    (iv) Box plot and Scatter plot.

❖❖❖