SYLLABUS

Data Exploration and Visualization - (AD3301)

UNIT I EXPLORATORY DATA ANALYSIS

EDA fundamentals - Understanding data science - Significance of EDA - Making sense of data - Comparing EDA with classical and Bayesian analysis - Software tools for EDA - Visual Aids for EDA - Data transformation techniques-merging database, reshaping and pivoting, Transformation techniques - Grouping Datasets - data aggregation - Pivot tables and cross-tabulations. **(Chapter - 1)**

UNIT II VISUALIZING USING MATPLOTLIB

Importing Matplotlib - Simple line plots - Simple scatter plots - visualizing errors - density and contour plots - Histograms - legends - colors - subplots - text and annotation - customization - three dimensional plotting - Geographic Data with Basemap - Visualization with Seaborn. **(Chapter - 2)**

UNIT III UNIVARIATE ANALYSIS

Introduction to Single variable : Distributions and Variables - Numerical Summaries of Level and Spread - Scaling and Standardizing - Inequality - Smoothing Time Series. **(Chapter - 3)**

UNIT IV BIVARIATE ANALYSIS

Relationships between Two Variables - Percentage Tables - Analyzing Contingency Tables - Handling Several Batches - Scatterplots and Resistant Lines - Transformations. **(Chapter - 4)**

UNIT V MULTIVARIATE AND TIME SERIES ANALYSIS

Introducing a Third Variable - Causal Explanations - Three-Variable Contingency Tables and Beyond - Longitudinal Data - Fundamentals of TSA - Characteristics of time series data - Data Cleaning - Time-based indexing - Visualizing - Grouping - Resampling. **(Chapter - 5)**

TABLE OF CONTENTS

UNIT I

UNIT II

UNIT III

UNIT IV

UNIT V

UNIT I

1 Exploratory Data Analysis

Syllabus

EDA fundamentals - Understanding data science - Significance of EDA – Making sense of data - Comparing EDA with classical and Bayesian analysis - Software tools for EDA - Visual Aids for EDA - Data transformation techniques-merging database, reshaping and pivoting, Transformation techniques - Grouping Datasets - data aggregation - Pivot tables and cross-tabulations.

Contents

1.1 Data and EDA Fundamentals

1.1.1 Data and Information

- Data (Singular Datum), is collection of different facts, figures, objects, symbols and events that are capable of providing informative pieces usually formatted in a particular manner. Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things.
- Data can be qualitative or quantitative. **Qualitative data** is descriptive information (it describes something). **Quantitative data** is numerical information (numbers).
- **Quantitative data** can be discrete or continuous. **Discrete data** can only take certain values (like whole numbers). **Continuous data** can take any value (within a range). **Discrete data** is counted, **continuous data** is measured.

Examples of Data

Qualitative

- My friend's favorite holiday destination.
- The most common names in India.
- How people describe the smell of a new deodorant.

Quantitative

- Height (Continuous).
- Weight (Continuous).
- Leaves on a tree (Discrete).
- Customers in a shop (Discrete).

Information

- Information is defined as classified or organized data that has some meaningful value for the user. Information is also the processed data used to make decisions and take action. Processed data must meet the following criteria for it to be of any significant use in decision-making :
  - **Accuracy** - The information must be accurate.
  - **Completeness** - The information must be complete.
  - **Timeliness** - The information must be available when it's needed.

1.1.2 Data Collection Methods

- Data collection is defined as a method of **collecting, analyzing data** for the purpose of validation and research using some techniques. Data collection is done to analyze a problem and learn about its outcome and future trends. When there is a need to arrive at a solution for a question, data collection methods help to make assumptions about the result in the future.
- Data collection methods can be classified as,
  - **Primary :** This is original, first-hand data collected by the data researchers. This process is the initial information gathering step, performed before anyone carries out any further or related research. Primary data results are highly accurate provided the researcher collects the information. However, there is a downside, as first-hand research is potentially time-consuming and expensive.
  - **Secondary :** Secondary data is second-hand data collected by other parties and already having undergone statistical analysis. This data is either information that the researcher has tasked other people to collect or information the researcher has looked up. Simply put, it is second-hand information. Although it is easier and cheaper to obtain than primary information, secondary information raises concerns regarding accuracy and authenticity. Quantitative data makes up a majority of secondary data.

Types of data collection :

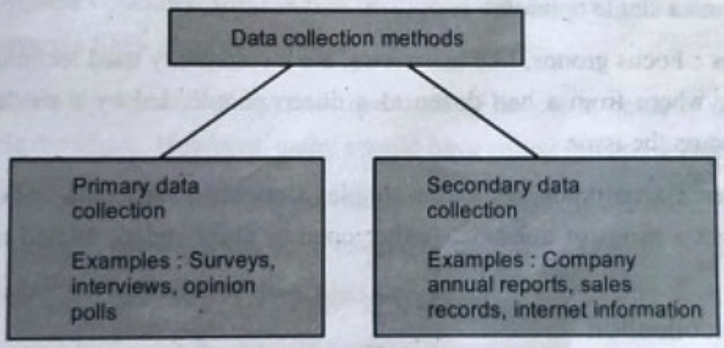

Fig. 1.1.1 : Data collection methods

- Below are widely used data collection methods,

1. Interviews
2. Questionnaires and surveys
3. Observations
4. Documents and records
5. Focus groups
6. Oral histories.

- Some of the above methods are quantitative, dealing with something that can be counted. Others are qualitative, meaning that they consider factors other than numerical values. In general, questionnaires, surveys and documents and records are quantitative, while interviews, focus groups, observations and oral histories are qualitative. There can also be crossover between the two methods.

Primary data collection

- **Interviews :** The researcher asks questions of a large sampling of people, either by direct interviews or means of mass communication such as by phone or mail. This method is by far the most common means of data gathering.
- **Projective Technique :** Projective data gathering is an indirect interview, used when potential respondents know why they are being asked questions and hesitate to answer. For instance, someone may be reluctant to answer questions about their phone service if a cell phone carrier representative poses the questions. With projective data gathering, the interviewees get an incomplete question and they must fill in the rest, using their opinions, feelings and attitudes.
- **Delphi Technique :** The Oracle at Delphi, according to Greek mythology, was the high priestess of Apollo's temple, who gave advice, prophecies and counsel. In the realm of data collection, researchers use the Delphi technique by gathering information from a panel of experts. Each expert answers questions in their field of specialty and the replies are consolidated into a single opinion.
- **Focus Groups :** Focus groups, like interviews, are a commonly used technique. The group consists of anywhere from a half-dozen to a dozen people, led by a moderator, brought together to discuss the issue.
- **Questionnaires :** Questionnaires are a simple, straightforward data collection method. Respondents get a series of questions, either open or close-ended, related to the matter at hand.

Secondary data collection

- Unlike primary data collection, there are no specific collection methods. Instead, since the information has already been collected, the researcher consults various data sources, such as :
  - Financial Statements
  - Sales Reports
  - Retailer / Distributor / Deal Feedback

- Customer Personal Information (e.g., name, address, age, contact info)
- Business Journals
- Government Records (e.g., census, tax records, Social Security info)
- Trade/Business Magazines
- The Internet.

Data collection tools : Below are widely used tools for data collection.

- **Word Association :** The researcher gives the respondent a set of words and asks them what comes to mind when they hear each word.
- **Sentence Completion :** Researchers use sentence completion to understand what kind of ideas the respondent has. This tool involves giving an incomplete sentence and seeing how the interviewee finishes it.
- **Role-Playing :** Respondents are presented with an imaginary situation and asked how they would act or react if it was real.
- **In-Person Surveys :** The researcher asks questions in person.
- **Online / Web Surveys :** These surveys are easy to accomplish, but some users may be unwilling to answer truthfully, if at all.
- **Mobile Surveys :** These surveys take advantage of the increasing proliferation of mobile technology. Mobile collection surveys rely on mobile devices like tablets or smartphones to conduct surveys via SMS or mobile apps.
- **Phone Surveys :** No researcher can call thousands of people at once, so they need a third party to handle the chore. However, many people have called screening and won't answer.
- **Observation :** Sometimes, the simplest method is the best. Researchers who make direct observations collect data quickly and easily, with little intrusion or third-party bias. Naturally, it is only effective in small-scale situations.

1.1.3 Common Issues / Problems in Data

Inconsistent data

- When working with various data sources, it is conceivable that the same information will have discrepancies between sources. The differences could be in formats, units or occasionally spellings. The introduction of inconsistent data might also occur during firm mergers or relocations. Inconsistencies in data have a tendency to accumulate and reduce

the value of data if they are not continually resolved. Organizations that have heavily focused on data consistency do so because they only want reliable data to support their analytics.

Data downtime

- Data is the driving force behind the decisions and operations of data-driven businesses. However, there may be brief periods when their data is unreliable or not prepared. Customer complaints and subpar analytical outcomes are only two ways that this data unavailability can have a significant impact on businesses. A data engineer spends about 80 % of their time updating, maintaining and guaranteeing the integrity of the data pipeline. In order to ask the next business question, there is a high marginal cost due to the lengthy operational lead time from data capture to insight.
- Schema modifications and migration problems are just two examples of the causes of data downtime. Data pipelines can be difficult due to their size and complexity. Data downtime must be continuously monitored and it must be reduced through automation.

Ambiguous data

- Even with thorough oversight, some errors can still occur in massive databases or data lakes. For data streaming at a fast speed, the issue becomes more overwhelming. Spelling mistakes can go unnoticed, formatting difficulties can occur and column heads might be deceptive. This unclear data might cause a number of problems for reporting and analytics.

Duplicate data

- Streaming data, local databases and cloud data lakes are just a few of the sources of data that modern enterprises must contend with. They might also have application and system silos. These sources are likely to duplicate and overlap each other quite a bit. For instance, duplicate contact information has a substantial impact on customer experience. If certain prospects are ignored while others are engaged repeatedly, marketing campaigns suffer. The likelihood of biased analytical outcomes increases when duplicate data are present. It can also result in ML models with biased training data.

Too much data

- While the data-driven analytics and its advantages are prominent, a data quality problem with excessive data exists. There is a risk of getting lost in an abundance of data when searching for information pertinent to analytical efforts. Data scientists, data analysts and business users devote 80 % of their work to finding and organizing the appropriate data. With an increase in data volume, other problems with data quality become more serious, particularly when dealing with streaming data and big files or databases.

Inaccurate data

- For highly regulated businesses like healthcare, data accuracy is crucial. Given the current experience, it is more important than ever to increase the data quality for COVID-19 and later pandemics. Inaccurate information does not provide a true picture of the situation and cannot be used to plan the best course of action. Personalized customer experiences and marketing strategies underperform if the customer data is inaccurate.
- Data inaccuracies can be attributed to a number of things, including data degradation, human mistake and data drift. Worldwide data decay occurs at a rate of about 3 % per month, which is quite concerning. Data integrity can be compromised while being transferred between different systems and data quality might deteriorate with time.

Hidden data

- The majority of businesses only utilize a portion of their data, with the remainder sometimes being lost in data silos or discarded in data graveyards. For instance, the customer service team might not receive client data from sales, missing an opportunity to build more precise and comprehensive customer profiles. Missing out on possibilities to develop novel products, enhance services and streamline procedures is caused by hidden data.

Finding relevant data

- Finding relevant data is not so easy. There are several factors that are to be considered to consider while trying to find relevant data, which include -
  - Relevant domain
  - Relevant demographics
  - Relevant time period and so many more factors that one needs to consider while trying to find relevant data.
- Data that is not relevant to the study in any of the factors render it obsolete and one cannot effectively proceed with its analysis. This could lead to incomplete research or analysis, re-collecting data again and again or shutting down the study.

Deciding the data to collect

- Determining what data to collect is one of the most important factors while collecting data and should be one of the first factors while collecting data. One must choose the subjects the data will cover, the sources used to gather it and the required quantity of information. The responses to these queries will depend on the aims or what is expected to achieve utilizing the data. As an illustration, one may choose to gather information on the categories of articles that website visitors between the ages of 20 and 50 most frequently access. One can also decide to compile data on the typical age of all the clients who made a purchase from the business over the previous month.

Dealing with big data

- Big data refers to exceedingly massive data sets with more intricate and diversified structures. These traits typically result in increased challenges while storing, analyzing and using additional methods of extracting results. Big data refers especially to data sets that are quite enormous or intricate that conventional data processing tools are insufficient. The overwhelming amount of data, both unstructured and structured, that a business faces on a daily basis.
- The amount of data produced by healthcare applications, the internet, social networking sites social, sensor networks and many other businesses are rapidly growing as a result of recent technological advancements. Big data refers to the vast volume of data created from numerous sources in a variety of formats at extremely fast rates. Dealing with this kind of data is one of the many challenges of data collection and is a crucial step toward collecting effective data.

1.1.4 Exploratory Data Analysis (EDA)

- **Exploratory Data Analysis (EDA)**, a term defined by John W. Tukey, is a process of examining or understanding the data and extracting insights or main characteristics of the data. EDA is generally classified into two methods, that is, **graphical** analysis and non-**graphical** analysis.
- "EDA" is a critical first step in analyzing the data from an experiment. Any method of looking at data that does not include formal statistical modeling and inference falls under the term exploratory data analysis.
- It is also known as **visual analytics or descriptive statistics**. It is the practice of inspecting and exploring data, before stating hypotheses, fitting predictors and other expected inferential goals. It typically includes the computation of simple summary statistics which capture some property of interest in the data and **visualization**. EDA can be thought of as an assumption-free, purely algorithmic practice.
- Exploratory Data Analysis (EDA) is the crucial process of using summary statistics and graphical representations to perform preliminary investigations on data in order to uncover patterns, detect anomalies, test hypotheses and verify assumptions.
- EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model. EDA is not a mere collection of techniques; EDA is a philosophy that dissects a data set; what is to look for; how

to look; and how to interpret. It is true that EDA heavily uses the collection of techniques that are called as "**statistical graphics**", but it is not identical to statistical graphics.

- Basic purpose of EDA is to spot problems in data (as part of data wrangling) and understand variable properties like,
  - Central trends (mean)
  - Spread (variance)
  - Skew
  - Outliers and anomalies.
- **Below are the most prominent reasons to use EDA**

1. Detection of mistakes.
2. Examine the data distribution.
3. Handling missing values of the dataset(a most common issue with every dataset) .
4. Handling the outliers.
5. Removing duplicate data.
6. Encoding the categorical variables.
7. Normalizing and scaling.
8. Checking of assumptions.
9. Preliminary selection of appropriate models.
10. Determining relationships among the explanatory variables.
11. Assessing the direction and rough size of relationships between explanatory and outcome variables.

1.2 Understanding Data Science

- Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information and make business decisions.
- Data science combines math and statistics, specialized programming, advanced analytics, Artificial Intelligence (AI) and machine learning with specific subject matter expertise to uncover actionable insights hidden in an organization's data. These insights can be used to guide decision making and strategic planning.

- Data science is the field of study that combines domain expertise, programming skills and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio and more to produce Artificial Intelligence (AI) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value.
- The data used for analysis can come from many different sources and presented in various formats.

The data science lifecycle

- Data science's lifecycle consists of five distinct stages, each with its own tasks :
  1. **Capture** - Data acquisition, data entry, signal reception, data extraction - This stage involves gathering raw structured and unstructured data.
  2. **Maintain** - Data warehousing, data cleansing, data staging, data processing, data architecture - This stage covers taking the raw data and putting it in a form that can be used.
  3. **Process** - Data mining, clustering/classification, data modeling, data summarization - Data scientists take the prepared data and examine its patterns, ranges and biases to determine how useful it will be in predictive analysis.
  4. **Analyze** - Exploratory/confirmatory, predictive analysis, regression, text mining, qualitative analysis - This stage involves performing the various analyses on the data.
  5. **Communicate** - Data reporting, data visualization, business intelligence, decision making - In this final step, analysts prepare the analyses in easily readable forms such as charts, graphs and reports.

Data science tools

- Below are various data science tools that can be used in various stages of the data science process.
  - Data Analysis - SAS, Jupyter, R Studio, MATLAB, Excel, RapidMiner
  - Data Warehousing - Informatica / Talend, AWS Redshift
  - Data Visualization - Jupyter, Tableau, Cognos, RAW
  - Machine Learning - Spark MLib, Mahout, Azure ML studio.

Use of data science

1. Data science may detect patterns in seemingly unstructured or unconnected data, allowing conclusions and predictions to be made.
2. Tech businesses that acquire user data can utilize strategies to transform that data into valuable or profitable information.
3. Data science has also made inroads into the transportation industry, such as with driverless cars. It is simple to lower the number of accidents with the use of driverless cars. For example, with driverless cars, training data is supplied to the algorithm and the data is examined using data science approaches, such as the speed limit on the highway, busy streets, etc.
4. Data science applications provide a better level of therapeutic customization through genetics and genomics research.
5. Data science has found its applications in almost every industry.

Applications of data science

1. **Healthcare :** Healthcare companies are using data science to build sophisticated medical instruments to detect and cure diseases. Machine learning models and other data science components are used by hospitals and other healthcare providers to automate X-ray analysis and assist doctors in diagnosing illnesses and planning treatments based on previous patient outcomes.
2. **Gaming :** Video and computer games are now being created with the help of data science and that has taken the gaming experience to the next level.
3. **Image recognition :** Identifying patterns in images and detecting objects in an image is one of the most popular data science applications.
4. **Recommendation systems :** Netflix and Amazon give movie and product recommendations based on what people like to watch, purchase or browse on their platforms.
5. **Logistics :** Data science is used by logistics companies to optimize routes to ensure faster delivery of products and increase operational efficiency.
6. **Fraud detection :** Banking and financial institutions use data science and related algorithms to detect fraudulent transactions.

7. **Internet search :** When one thinks of search, immediately Google comes into mind. However, there are other search engines, such as Yahoo, Duckduckgo, Bing, AOL, Ask and others, that employ data science algorithms to offer the best results for the search query in a matter of seconds. Given that Google handles more than 20 petabytes of data per day. Google would not be the 'Google', widely popular today, if data science did not exist.

8. **Speech recognition :** Speech recognition is dominated by data science techniques. There are excellent applications of these algorithms in daily lives. A virtual speech assistant like Google assistant, Alexa or Siri are few speech recognition applications to name. Its voice recognition technology is operating behind the scenes, attempting to interpret and evaluate the words and delivering useful results from the use. Image recognition may also be seen on social media platforms such as Facebook, Instagram and Twitter. When one submits a picture of their own with someone onone's list, these applications will recognise them and tag them.

9. **Targeted advertising :** From displaying banners on various websites to digital billboards at airports, data science algorithms are utilised to identify almost anything. This is why digital advertisements have a far higher CTR (Call-Through Rate) than traditional marketing. They can be customized based on a user's prior behaviour. That is why one may see adverts for data science training programs while another person sees an advertisement for clothes in the same region at the same time.

10. **Airline route planning :** As a result of data science, it is easier to predict flight delays for the airline industry, which is helping it grow. It also helps to determine whether to land immediately at the destination or to make a stop in between, such as a flight from Delhi to the United States of America or to stop in between and then arrive at the destination.

11. **Virtual reality :** A virtual reality headset incorporates computer expertise, algorithms and data to create the greatest viewing experience possible. The popular game Pokemon GO is a minor step in that direction. The ability to wander about and look at Pokemon on walls, streets and other non-existent surfaces. The makers of this game chose the locations of the Pokemon and gyms using data from ingress, the previous app from the same business.

12. **To evaluate client in retail shopping :** Retailers evaluate client behaviour and purchasing trends in order to provide individualized product suggestions as well as targeted advertising, marketing and promotions. Data science also assists them in managing product inventories and supply chains in order to keep items in stock.

13. **Entertainment :** Da consumer's views, v algorithms are also u of a user.

14. **Finance :** Banks an activities, manage f order to uncover up

15. **Manufacturing :** management and d probable equipmer

1.3 Significa

- Exploratory Data A identify trends in d model is a feasible
- EDA helps data sc or hypothesis test so as to gain insig

Importance of usi

- EDA helps iden EDA helps detec and the relations
- Different fields data primarily i made using the
- It is practically points without collected data are distinctive
- Exploratory da visualize data exploratory an in a data mini

13. **Entertainment :** Data science enables streaming services to follow and evaluate what consumer's views, which aids in the creation of new TV series and films. Data-driven algorithms are also utilised to provide tailored suggestions based on the watching history of a user.

14. **Finance :** Banks and credit card firms mine and analyse data in order to detect fraudulent activities, manage financial risks on loans and credit lines and assess client portfolios in order to uncover upselling possibilities.

15. **Manufacturing :** Data science applications in manufacturing include supply chain management and distribution optimization, as well as predictive maintenance to anticipate probable equipment faults in facilities before they occur.

1.3 Significance of EDA

- Exploratory Data Analysis (EDA) involves using statistics and visualizations to analyze and identify trends in data sets. The primary intent of EDA is to determine whether a predictive model is a feasible analytical tool for business challenges or not.
- EDA helps data scientists gain an understanding of the data set beyond the formal modeling or hypothesis testing task. Exploratory data analysis is essential for any research analysis, so as to gain insights into a data set.

Importance of using EDA for analyzing data sets is,

- EDA helps identify errors in data sets. EDA gives a better understanding of the data set. EDA helps detect outliers or anomalous events. EDA helps to understand data set variables and the relationship among them.
- Different fields of science, economics, engineering and marketing accumulate and store data primarily in electronic databases. Appropriate and well-established decisions should be made using the data collected.
- It is practically impossible to make sense of datasets containing more than a handful of data points without the help of computer programs. To be certain of the insights that the collected data provides and to make further decisions, data mining is performed where there are distinctive analysis processes.
- Exploratory data analysis is key and usually the first exercise in data mining. It allows us to visualize data to understand it as well as to create hypotheses for further analysis. The exploratory analysis centers around creating a synopsis of data or insights for the next steps in a data mining project.

- EDA actually reveals ground truth about the content without making any underlying assumptions. This is the fact that data scientists use this process to actually understand what type of modeling and hypotheses can be created. Key components of exploratory data analysis include summarizing data, statistical analysis and visualization of data.
- Python provides expert tools for exploratory analysis, with pandas for summarizing; scipy, along with others, for statistical analysis; and matplotlib and plotly for visualizations.
- Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including machine learning.

Importance of EDA in Data Processing and Modeling

- EDA makes it simple to comprehend the structure of a dataset, making data modeling easier. The primary goal of EDA is to make data 'clean' implying that it should be devoid of redundancies. It aids in identifying incorrect data points so that they may be readily removed and the data cleaned. Furthermore, it aids us in comprehending the relationship between the variables, providing us with a broader view of the data and allowing us to expand on it by leveraging the relationship between the variables. It also aids in the evaluation of the dataset's statistical measurements.
- Outliers or abnormal occurrences in a dataset can have an impact on the accuracy of machine learning models. The dataset might also contain some missing or duplicate values. EDA may be used to eliminate or resolve all of the dataset's undesirable qualities.

Data Cleaning and Preprocessing

- Data preprocessing and cleansing are critical components of EDA. Understanding the variables and the structure of the dataset is the initial stage in this process. The data must then be cleaned. The dataset may contain redundancy such as irregular data, missing values or outliers that may cause the model to overfit or underfit during training. Removing or resolving these redundancies is known as **data cleaning**. The last part is analysing the relationship between the variables.

1.4 Types of Exploratory Data Analysis

- There are four primary types of EDA namely,
  1. **Univariate non-graphical :** This is the simplest form of data analysis, where the data being analyzed consists of just one variable. Since it is a single variable, it does not deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.
  2. **Univariate graphical :** Non-graphical methods do not provide a full picture of the data. Graphical methods are therefore required. Common types of univariate graphics include :
     - Stem-and-leaf plots, which show all data values and the shape of the distribution.
     - Histograms, a bar plot in which each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values.
     - Box plots, which graphically depict the five-number summary of minimum, first quartile, median, third quartile and maximum.
  3. **Multivariate non graphical :** Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.
  4. **Multivariate graphical :** Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.
- Other common types of multivariate graphics include :
  - Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.
  - Multivariate chart, which is a graphical representation of the relationships between factors and a response.
  - Run chart, which is a line graph of data plotted over time.
  - Bubble chart, which is a data visualization that displays multiple circles (bubbles) in a two-dimensional plot.
  - Heat map, which is a graphical representation of data where values are depicted by color.

1.5 Making Sense of Data

- In today's digital era data is a powerful tool. Just having possession of data is not sufficient. One should be able to analyze data in the proper manner to get insight into the data.
- Making sense of data means finding meanings from data, generating inference from data. Data analysis is a process of making meaning. Making sense of data educates owners or readers on the steps and issues that need to be considered in order to successfully complete a data analysis or data mining project.
- Consider the process of creating something out of a set of raw materials. For example, given a bolt of fabric and some notions (thread, zippers, buttons and so forth), different people might end up making very different kinds of creations. One person might make a baby dress, another a dance costume and a thirda diwali lamp. But what they come up with must at least be plausibly created from the raw materials. It would be awfully difficult to make a working car or an edible meal out of a bolt of fabric.

Basic steps to make sense of data can be as follows,

1. **Managing data** - Initially in the process of analyzing qualitative data, first task is to arrange or organize data so that one can begin to make sense of it. That is, one has to do some arrangements for holding data and keep it in a certain format before starting the analysis. One should make a comprehensive list of all the data materials. At one level, this is a fairly mechanical process of gathering all the materials and devising a filing system or other way of organizing them so that one can easily access it.
   - Separating different types of data would help to manage the data. If there is a variety of data one can classify it as per need. This classification will help to access the data easily.
   - Keeping data in chronological order makes data access fast. If applicable data can be arranged in certain order so as to access it quickly.
   - Organizing data in topic wise or document wise helps quick access.
   - While maintaining data one should always give a thought to the fact that how many copies of data are to be stored. It would also be important to keep in details of updations of data.
2. **Getting familiar with data** - Based on data collection and its size, it might take long time to get a complete data set. Over the time user may lose on tiny details about data. Hence it is necessary that once the data set is ready the user should browse through data and get well acquainted with data.

3. **Making sense of data** - Once data is collected and a well organized analysis process can start, that would make sense of data. For analyzing data there is a need to have data in certain presentable or viewable format. Data is best seen using tables, charts, graphs, patterns. This would help to analyze the data set.

1.6 Comparing EDA with Classical and Bayesian Analysis

- There are three widely used approaches for data analysis namely,
  1. Classical data analysis
  2. Exploratory data analysis
  3. Bayesian data analysis.
- These three approaches are similar in that they all begin with a general science/engineering problem and all yield science/engineering conclusions. The difference is the sequence and focus of the intermediate steps.
- Classical data analysis follows below steps,

  Problem -> Data -> Model -> Analysis -> Conclusions
- Exploratory data analysis follows below steps,

  Problem -> Data -> Analysis -> Model -> Conclusions
- Bayesian data analysis follows below steps,

  Problem -> Data -> Model -> Prior Distribution -> Analysis -> Conclusions
- That is the in classical analysis, the data collection is followed by the imposition of a model (normality, linearity, etc.) and the analysis, estimation and testing that follows are focused on the parameters of that model.
- For EDA, the data collection is not followed by a model imposition; rather it is followed immediately by analysis with a goal of inferring what model would be appropriate.
- For a Bayesian analysis, the analyst attempts to incorporate scientific/engineering knowledge/expertise into the analysis by imposing a data - independent distribution on the parameters of the selected model; the analysis thus consists of formally combining both the prior distribution on the parameters and the collected data to jointly make inferences and/or test assumptions about the model parameters. In the real world, data analysts freely mix elements of all of the above three approaches and if required other approaches as well.

Classical Analysis Vs EDA

- Classical analysis approach and EDA can be compared on the basis of below parameters

1. **Model**
   - The classical approach imposes models (both deterministic and probabilistic) on the data. Deterministic models include, for example, regression models and analysis of variance (ANOVA) models. The most common probabilistic model assumes that the errors about the deterministic model are normally distributed - This assumption affects the validity of the ANOVA F tests.
   - The exploratory data analysis approach does not impose deterministic or probabilistic models on the data. On the contrary, the EDA approach allows the data to suggest admissible models that best fit the data.
2. **Focus**
   - The two approaches differ drastically in focus. For classical analysis, the focus is on the model estimating parameters of the model and generating predicted values from the model.
   - For exploratory data analysis, the focus is on the data - Its structure, outliers and models suggested by the data.
3. **Techniques**
   - Classical techniques are generally quantitative in nature. They include ANOVA, t tests, chi-squared tests and F tests. EDA techniques are generally graphical. They include scatter plots, character plots, box plots, histograms, bi-histograms, probability plots, residual plots and mean plots.
4. **Rigor**
   - Classical techniques serve as the probabilistic foundation of science and engineering. The most important characteristic of classical techniques is that they are rigorous, formal and "objective". EDA techniques do not share in that rigor or formality. EDA techniques make up for that lack of rigor by being very suggestive, indicative and insightful about what the appropriate model should be. EDA techniques are subjective and depend on interpretation which may differ from analyst to analyst, although experienced analysts commonly arrive at identical conclusions.

5. **Data treatment**

- Classical estimation techniques have the characteristic of taking all of the data and mapping the data into a few numbers ("estimates"). This is both a virtue and a vice. The virtue is that these few numbers focus on important characteristics (location, variation, etc.) of the population. The vice is that concentrating on these few characteristics can filter out other characteristics (skewness, tail length, autocorrelation, etc.) of the same population. In this sense there is a loss of information due to this "**filtering**" process.
- Whereas, the EDA approach often makes use of (and shows) all of the available data. In this sense there is no corresponding loss of information.

6. **Assumptions**

- The classical approach tests based on classical techniques that are usually very sensitive- that is, if a true shift in location, say, has occurred, such tests frequently have the power to detect such a shift and to conclude that such a shift is "**statistically significant**". But classical tests depend on underlying assumptions (e.g., normality) and hence the validity of the test conclusions becomes dependent on the validity of the underlying assumptions. Further the issue is, the exact underlying assumptions may be unknown to the analyst or if known, untested. Thus the validity of the scientific conclusions becomes intrinsically linked to the validity of the underlying assumptions. In practice, if such assumptions are unknown or untested, the validity of the scientific conclusions becomes suspect. Many EDA techniques make little or no assumptions they present and show the data - all of the data - as is, with fewer encumbering assumptions.

1.7 Software Tools for EDA

Python, R, Excel are some of the popular EDA tools.

1. **R** - An open-source programming language and free software environment for statistical computing and graphics supported by the R foundation for statistical computing. The R language is widely used among statisticians in developing statistical observations and data analysis.
2. **Python** - An interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python and EDA can be used together to identify missing values in a data set, which is important so one can decide how to handle missing values for machine learning.

EDA can be done using python for identifying the missing value in a data set. Other functions that can be performed are the description of data, handling outliers, getting insights through the plots. Its high-level, built-in data structure and dynamic typing and binding make it an attractive tool for EDA. Analyzing a dataset is a hectic task that takes a lot of time. Python provides certain open-source modules that can automate the whole process of EDA and help in saving time.

3. **Excel / Spreadsheet :** A very long running and dependable tool excel remains an indispensable part of analytics industry. Most of the problems faced in analytics projects are solved using this software. It supports all the important features like summarizing data, visualizing data, data wrangling etc. which are powerful enough to inspect data from all possible angles. Microsoft Excel is paid but there are various other spreadsheet tools like open office, google docs, which are certainly worth using.
4. **Weka -** Weka is an easy to learn, machine learning tool, having an intuitive interface to get the job done quickly. It provides options for data pre-processing, classification, regression, clustering, association rules and visualization. Most of the steps while model building can be achieved using Weka. It is built on Java.
5. **Tableau public -** Tableau is a data visualization software. Its a fast visualization software which allows exploring of data, every observation using various possible charts. It uses intelligent algorithms figure out by self about the type of data, best method available etc. For understanding data in real time, tableau can get the job done. In a way, tableau imparts a colorful life to data and allows sharing work with others.

1.8 Visual Aids for EDA

1. Univariate Plots (Used for univariate data - the data containing one variable)

- Univariate plots show the frequency or the distribution shape of a variable. Below are visual tools used to analyze univariate data.

i. Histograms

- Histograms are two-dimensional plots in which the x-axis divides into a range of numerical bins or time intervals. The y-axis shows the frequency values, which are counts of occurrences of values for each bin. Bar graphs have gaps between the bars to indicate that they compare distinct groups, but there are no gaps in histograms. Hence, they convey if the distribution is left/positively skew (most of the data falls to the right side), right/negatively skewed (most of the data falls to the left side), bi-modal (graphs having two distinct peaks), normal (perfectly symmetrical without skew) or uniform (almost all the bins have similar frequency).

- **Example :** Below is the waiting time of the customer at the cash counter of the grocessary shop during peak hours, which was observed by the cashier.

Customer Waiting Time (in mins)
2.30
5.00
3.55
2.50
5.10
4.21
3.33
4.10
2.55
5.07
3.45
4.10
5.12

- For this data a histogram can be created using five bins with five different frequencies, as seen in the chart below. On the Y-axis, it's the average number of customers falling in that particular category. On the X-axis, there is a range of waiting times. For example, the 1st bin range is 2.30 mins to 2.86 mins. It can be noted that the count is three for that category from the table and as seen in the below graph.

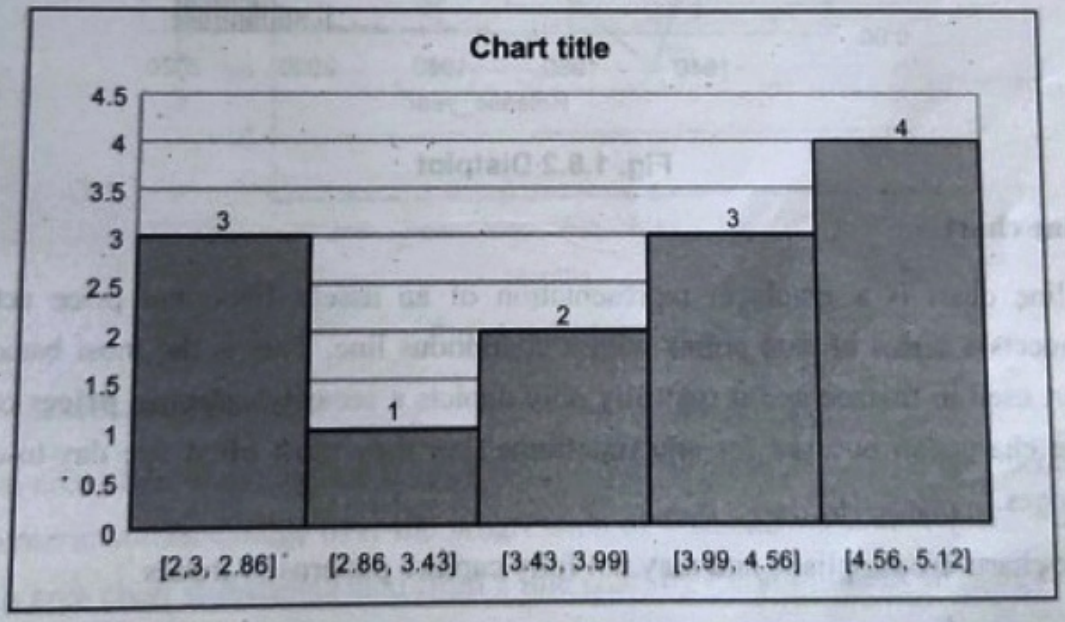

Fig. 1.8.1 Histogram

- It is a random distribution, which is a type of distribution that has several peaks and lacks an apparent pattern.

ii. Distplot

- Distplot is also known as the **second histogram** because it is a slight improved version of the histogram. Distplot gives us a KDE(Kernel Density Estimation) over histogram which explains PDF(Probability Density Function) which means what is the probability of each value occurring in this column. Below is the distplot of distribution of when news channel is most active with new additions over the year.

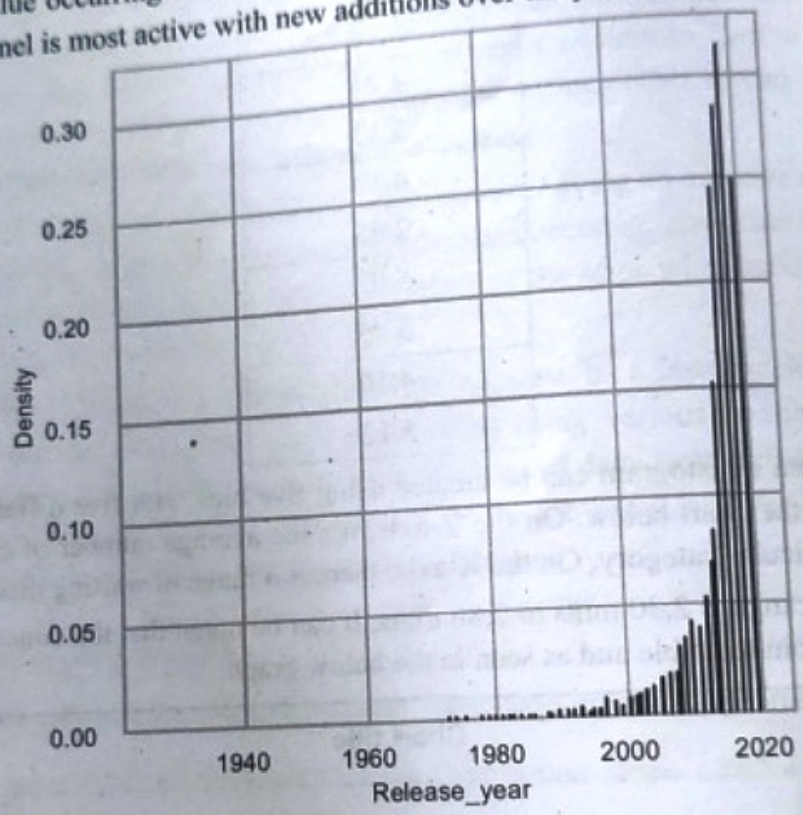

Fig. 1.8.2 Distplot

iii. Line chart

- A line chart is a graphical representation of an asset's historical price action that connects a series of data points with a continuous line. This is the most basic type of chart used in finance and it typically only depicts a security's **closing prices** over time. Line charts can be used for any timeframe, but they most often use day-to-day price changes.
- Line charts are simplistic and may not fully capture patterns or trends.

- There are different types of line charts. They are :
  - **Line chart** - It shows the trend over time (years, months, days) or other categories. It is used when the order of time or types is important.
  - **Line chart with markers** - It is similar to the line chart, but it will highlight data points with markers.
  - **Stacked line chart** - This is a line chart where lines of the data points do not get overlapped because they will be cumulative at each point.
  - **Stacked line chart with markers** - This is similar to the stacked line chart but will highlight data points with markers.
  - **100 % stacked line chart** - It shows the percentage contribution to a whole-time or category.
  - **100 % stacked line chart with markers** - This is similar to a 100 % stacked line chart, but markers will highlight data points.
- Below line chart shows number of houses sold in particular months.

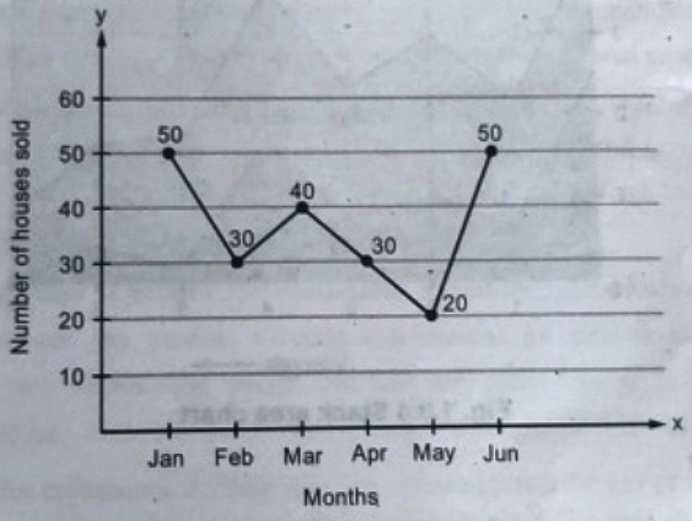

Fig. 1.8.3 Lineplot

iv. Stacked area plot/chart

- An area chart combines the **line chart** and **bar chart** to show how one or more group's numeric values change over the progression of a second variable, typically that of time. An area chart is distinguished from a line chart by the addition of shading between lines and a baseline, like in a bar chart.

- **Stacked area chart is plotted in the form of several area series stacked on top of one another.** The height of each series is determined by the value in each data point. A typical use case for stacked area charts is analyzing how each of several variables, whose their totals vary, on the same graphic.
- A stacked area graph is useful for comparing multiple variables changing over an interval.
- **Example :** Below stack area chart shows data plotting of three data sets.

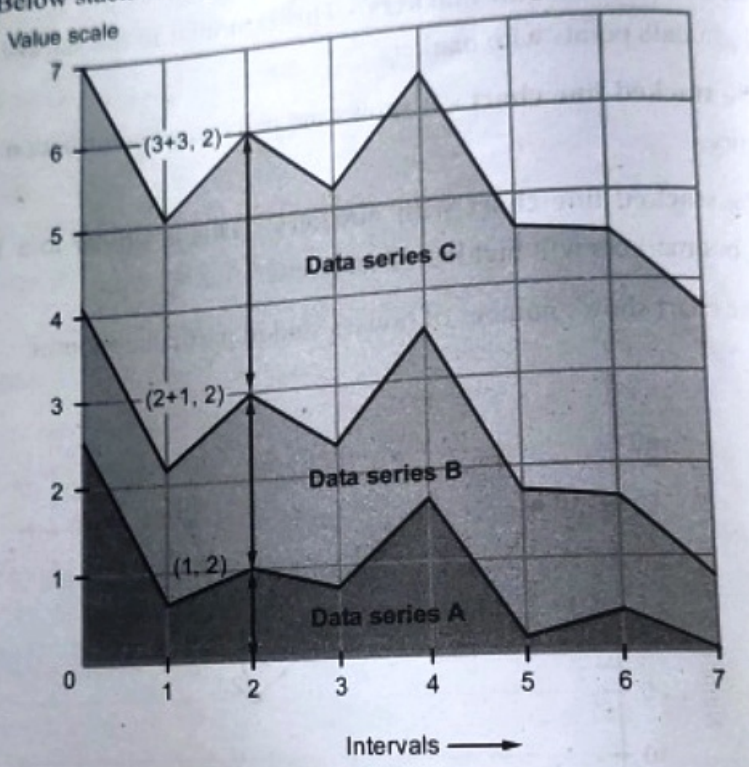

Fig. 1.8.4 Stack area chart

v. Table chart

- **A table chart is** a means of arranging data in rows and columns. **The use of tables is pervasive throughout all communication, research and data analysis. Tables appear in print media, handwritten notes, computer software, architectural ornamentation, traffic signs and many other places.**
- Tables are meant to be read, so they are ideal **when there is data that cannot easily be presented visually or when the data requires more specific attention.**

vi. Probability distribution plots

- Probability distributions are mathematical functions that deseribe all the possible values that a random variable can assume within a given range. They help model random phenomena, allowing us in order to estimate the probability of a particular event. This type of distribution is helpful to know the likely outcomes and the spread of potential values.
- For a single random variable, probability distributions can be divided into two types :

Types :

- **Discrete probability distributions for discrete variables :** Also known as probability mass functions, the random variable can assume a discrete number of values like the number of reviews; it can be 100 or 101, but nothing in between. It returns probabilities; hence the output is between 0 and 1. There are a variety of discrete probability distributions that can be used to model different types of data.
- **Binomial distribution :** There are two possible outcomes in this distribution - success or failure and multiple trials are carried out. The probability of success and failure is the same for all trials. The sum of all probabilities must equal one.

- **Success probability :** So, let's say that there is a success probability of 0.8 of manufacturing a perfect car engine part. What is the probability of having seven successes in 10 attempts ? The probability of success is 0.8 and failure is 0.2. The number of trials is ten and the number of successes is 7.
- **Probability density functions for continuous variables :** Also known as **probability density functions**, the random variable can assume an infinite number of values between any two values; like weight can take any value like 45.3, 45.36. 45.369 or 45.3698 and so on.
- Probabilities for continuous distributions are measured over ranges of values rather than single points. A probability indicates the likelihood that a value will fall within an interval. The entire area under the distribution curve equals 1. For instance, the proportion of the area under the curve that falls within a range of values along the X-axis is the likelihood that a value will fall within that range.

Probability Density Function

- Suppose there is a dataset of adult's heights in a town and the data follows a norm distribution. The mean equals 5.5 and the standard deviation is 1. The shaded p shows the probability that a randomly picked person's height will be smaller than 4 and is approximately equal to 0.15 or 15 %.

vii. Run sequence plots

- A run chart, also known as a **run-sequence plot**, displays observed data in a tim sequence. So, often, the data displayed represents some aspect of a business proce output or performance. It is, therefore, a form of a line chart. They are often analyzed order to locate anomalies in data that suggest shifts in a process over time. Changes location and scale and outliers can easily be detected.

2. Bivariate Plots (Used for bivariate data - the data containing two variables)

- Bivariate plots display the relationship between two variables in exploratory data analysis.

i. Bar graphs

- Bar charts can be used to compare nominal or ordinal data. They are helpful for recognizing trends.

ii. Scatter plots

- Scatter plots are commonly used in statistical analysis in order to visualize numerical relationships. So, they are used in order to determine whether two measures are correlated by plotting them on the x and y-axis. They are suitable for recognizing trends.

iii. Box plots

- These charts show the distribution of values along an axis. Rectangular boxes are used in order to bucket the data, giving us an idea of how the data points are spread out. These boxes are also called **quartiles** which represent a quarter of a data set. Boxes can be drawn vertically or horizontally. One can also easily spot the outliers, which are usually treated as abnormal values and affect the data set's overall observation due to their very high or low values.
- Box plots are suitable for identifying outliers.

iv. Correlation plots (Heat maps)

- For instance, correlation heat maps show the interrelationship between variables - areas as shaded as per the data's values. So, Color differences can easily spot similar and different values and make sense of the data variation. They are usually helpful when there is a large amount of data. They are used during A/B testing to see which parts of a web page are accessed by users on a website. The number of reviews generated every hour or to analyze a cricket match to understand where a batsman is scoring the bulk of his runs or where the bowler is pitching his ball.

v. Cluster map

- One can also use a cluster map to understand the relationship between two categorical variables. A cluster map basically plots a dendrogram that shows the categories of similar behavior together.

3. Special purpose plots

i. Pair plots

- Pair plots are a simple way in order to visualize relationships between multiple variables. So, It produces a matrix of relationships between variables in the data for a direct examination of the data.
- This plot shows how registered and casual users are using bike rentals. It also shows the effect of temperature, humidity and wind speed on bike rentals. This gives an overview of the correlation between multiple variables.

ii. Contour plots

- The contour plot can be used for representing a 3D surface in a 2D format. Contour plots are generally used for continuous variables rather than categorical data.
- The contour maps are inspired by seismic data analysis. They can explain where the data density is high, explore deep learning error functions or gradient analysis.

iii. Density plots

- A density plot is a smoothed, continuous version of a histogram estimated from the data. The most common form of estimation is the kernel density plot. In this method, a continuous curve (the kernel) is drawn at every individual data point. All of these curves are then combined to make a single smooth density estimation.

- So, The y-axis in a density plot is the probability density function for the kernel density estimation and not a probability. The difference is that the probability density is the probability per unit on the x-axis.
- While comparing the distributions of one variable across multiple categories, histograms have issues with readability. Density plots are useful in this scenario.

iv. Polar / Spider / Radar charts

- Polar charts are **circular charts that use values and angles to show information as polar coordinates**. Polar charts are useful for showing scientific data. One can specify a default measure.
- A polar chart is a graphical way of displaying multivariate data of three or more quantitative variables represented on axes starting from the same point. It helps demonstrate a dominant variable.
- **Polar chart** is a common variation of circular graphs. It is useful when relationships between data points can be visualized most easily in terms of radians and angles.
- In polar charts, a series is represented by a closed curve that connects points in the polar coordinate system. Each data point is determined by the distance from the pole (the radial coordinate) and the angle from the fixed direction (the angular coordinate).

v. Lollipop chart

- The lollipop chart is **a composite chart with bars and circles**. It is a variant of the bar chart with a circle at the end, to highlight the data value. Like a bar chart, a lollipop chart is used to compare categorical data. For this kind of composite chart, there are more visual elements to convey information.
- Lollipop Chart (LC) is a handy variation of a bar chart where the bar is replaced with a line and a dot at the end. Just like bar graphs, lollipop plots are used to make **comparisons** between different items or categories. They are also used **for ranking** or for showing **trends over time**. Only one numerical variable is compared per item or category. They are not suitable for relationships, distribution or composition analysis.
- Lollipop charts are two-dimensional with two axes : One axis shows categories or a time series, the other axis shows numerical values.
- LCs are preferred to bar charts when one has to display a large number of similar high values. In that case with a standard bar plot, one may get a cluttered chart and experience an optical effect called a **Moiré pattern** (The Moiré effect is a visual perception that occurs when viewing a set of lines or bars that is superimposed on another set of lines or bars, where the sets differ in relative size, angle or spacing.).

ction for the kernel density
probability density is the

tiple categories, histograms
enario.

to show information as
fic data. One can specify a

e data of three or more
ne same point. It helps

seful when relationships
dians and angles.

nects points in the polar
nce from the pole (the
ular coordinate).

It is a variant of the bar
a bar chart, a lollipop
nposite chart, there are

bar is replaced with a
ots are used to make
o used for ranking or
compared per item or
mposition analysis.

ws categories or a time

umber of similar high
a cluttered chart and
iré effect is a visual
t is superimposed on
gle or spacing.).

vi. Lag plots

- A relationship between an observation and the previous observation is beneficial in time series modeling. Previous observations in a time series are lags, with the observation at one previous time step. It is known as lag1, the observation at two previous steps lag 2 and so on.
- A lag plot is a useful type of plot in order to explore each observation's relationship and a lag of that observation and is displayed as a scatter plot. If the points cluster along a diagonal line from the bottom-left to the plot's top-right, it suggests a positive correlation relationship. If the points cluster along a diagonal line from the top-left to the bottom-right, it means a negative correlation relationship.
- Lag plots can help compare observations simultaneously in the last week or last month or the previous year by using corresponding lag values.
- The plot here shows the count of bike rentals compared to the previous day's count and it displays a relatively strong positive correlation.

vii. Auto-correlation plots

- The correlation between observations and their lag values in a time series name autocorrelation. Correlation coefficients are plotted on an autocorrelation plot.
- A correlation coefficient is a correlation value between observations and their lag 1 values and results in a number between – 1 and + 1. A value close to zero suggests a weak correlation, whereas a value closer to – 1 or 1 indicates a strong correlation. It helps better understand how this relationship changes over the lag. It shows the lag on the x-axis and the correlation on the y-axis.

viii. Lognormal plots

- A normal distribution can be converted to a lognormal distribution using logarithmic mathematics. The lognormal distribution plots the log of random variables from a normal distribution curve. It displays the Probability Density Function (PDF) and is of particular interest when the variable must be positive as log values are always positive.
- Many examples follow lognormal distribution like the concentration of elements and their radioactivity in the Earth's crust, latent periods of infectious diseases, the distribution of particles, chemicals and organisms in the environment, the length of comments posted on social media website discussion forums or fluctuations in the stock markets.

ix. Violin plot

- A violin plot is a hybrid of a box plot and a kernel density plot, which shows peaks in the data. It is used to visualize the distribution of numerical data. Unlike a box plot that can only show summary statistics, violin plots depict summary statistics and the density of each variable.

x. Joint plot

- While the pair plot provides a visual insight into all possible correlations, the joint plot provides bivariate plots with univariate marginal distributions.

xi. Pie chart

- The pie chart is also the same as the countplot, only gives additional information about the percentage presence of each category in data which means category is getting how much weightage in data.

4. Multivariate Visualization (Used for multivariate data - The data containing more than two variables)

- When dealing with multiple variables, it is tempting to make three dimensional plots. Such data can be viewed with scatter plots of the relation between each variable.
- Combined charts also are handy ways to visualize data, since each chart is simple to understand on its own.
- For very large dimensionality, one can reduce the dimensionality using principal component analysis or other techniques and then make a plot of the reduced variables. This is particularly important for high dimensionality data and has applications in deep learning such as visualizing natural language or images.

Text data

- For example with text data, one could create a world cloud, where the size of each word is based on its frequency in the text. To remove the words which add noise to the dataset, the documents can be grouped using topic modeling and only the important words can be displayed.

Image data

- When doing image classification, it is common to use decomposition and remove the dimensionality of the data.
- Instead of blindly using decomposition, a data scientist could plot the result :
  - By looking at the contrast (black and white) in the images, one can see there is an importance with the locations of the eyes, nose and mouth, along with the head shape.

1.9 Data Transformation Techniques

- Data transformation techniques refer to all the actions that help to transform the raw data into a clean and ready-to-use dataset.
- There are different types of data transformation techniques that offer a unique way of transforming the data and there is a chance that there will not be a need for all these techniques on every projects. Below are basic data transformation techniques that can be used in analysis project or data pipelines.

1. Data smoothing

- Smoothing is a technique where an algorithm is applied in order to remove noise from the dataset when trying to identify a trend. Noise can have a bad effect on the data and by eliminating or reducing it one can extract better insights or identify patterns that would not be seen otherwise.
- There are three algorithm types that help with data smoothing :
  - **Clustering :** Where one can group similar values together to form a cluster while labeling any value out of the cluster as an outlier.
  - **Binning :** Using an algorithm for binning will help to split the data into bins and smooth the data value within each bin.
  - **Regression :** Regression algorithms are used to identify the relation between two dependent attributes and help to predict an attribute based on the value of the other.

2. Attribution construction

- Attribution construction is one of the most common techniques in data transformation pipelines. Attribution construction or feature construction is the process of creating new features from a set of the existing features/attributes in the dataset.
- Imagine working in marketing and trying to analyze the performance of a campaign. One may have all the impressions that the campaign generated the total cost for the given time frame. Instead of trying to compare these two metrics across all of the campaigns, one can construct another metric to calculate the cost per million impressions or CPM.

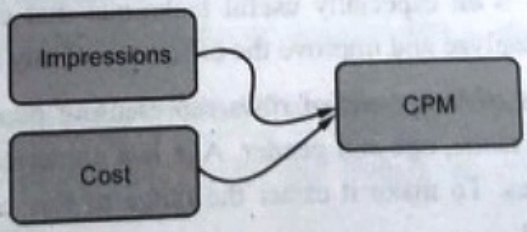

Fig. 1.9.1 Attribution construction

- This will make data exploration and analysis process a lot easier, as one can compare the campaign performance on a single metric rather than two separate metrics.

3. Data generalization

- Data generalization refers to the process of transforming low-level attributes into high-level ones by using the concept of hierarchy. Data generalization is applied to categorical data where they have a finite but large number of distinct values.
- This is something that everyone is already doing without noticing and it helps to get a clearer picture of the data. Suppose there are four categorical attributes in the database,

1. City
2. Street
3. Country
4. State / province.

- One can define a hierarchy between these attributes by specifying the total ordering among them at the schema level, for example :

street < city < state/province < country.

4. Data aggregation

- Data aggregation is possibly one of the most popular techniques in data transformation. When data aggregation is applied to the raw data, it is essentially a process of storing and presenting data in a summary format.
- This is ideal when one wants to perform statistical analysis over the data as one might want to aggregate the data over a specific time period and provide statistics such as average, sum, minimum and maximum.
- For instance, raw data can be aggregated over a given time period to provide statistics such as average, minimum, maximum, sum and count. After the data is aggregated and written as a report, one can analyse the aggregated data to gain insights about particular resources or resource groups. There are two types of data aggregation : Time aggregation and spatial aggregation.

5. Data discretization

- Data discretization refers to the process of transforming continuous data into a set of data intervals. This is an especially useful technique that can help to make the data easier to study and analyze and improve the efficiency of any applied algorithm.
- Imagine having tens of thousands of rows representing people in a survey providing their first name, last name, age and gender. Age is a numerical attribute that can have a lot of different values. To make it easier the range of this continuous attribute can be divided into intervals.

- Mapping this attribute to a higher-level concept, like youth, middle-aged and senior, can help a lot with the efficiency of the task and improve the speed of the algorithms applied.
- There are a wide variety of discretization methods starting with naive methods such as equal-width and equal-frequency to much more sophisticated methods such as MDLP.

6. Data normalization

- Data normalization is the process of scaling the data to a much smaller range, without losing information in order to help minimize or exclude duplicated data and improve algorithm efficiency and data extraction performance.
- There are three methods to normalize an attribute :
  - **Min-max normalization :** Where one performs a linear transformation on the original data.
  - **Z-score normalization :** In z-score normalization (or zero-mean normalization) one is normalizing the value for attribute A using the mean and standard deviation.
  - **Decimal scaling :** Where one can normalize the value of attribute A by moving the decimal point in the value.
- Normalization methods are frequently used when there are values that skew the dataset and it is hard to extract valuable insights.

7. Integration

- Data integration is a crucial step in data pre-processing that involves combining data residing in different sources and providing users with a unified view of these data. It includes multiple databases, data cubes or flat files and works by merging the data from various data sources. There are mainly two major approaches for data integration : Tight coupling approach and loose coupling approach.

8. Manipulation

- Data manipulation is the process of changing or altering data to make it more readable and organised. Data manipulation tools help identify patterns in the data and transform it into a usable form to generate insights on financial data, customer behaviour etc.

1.10 Merging Database (Using Pandas Library)

- Pandas is a software library written for the Python programming language for data manipulation and analysis.

- The Series and DataFrame objects in pandas are powerful tools for exploring and analyzing data. Part of their power comes from a multifaceted approach to combining separate datasets. With pandas, one can **merge**, **join** and **concatenate** datasets, allowing to unify and better to understand the data being analyzed.
- The **Pandas DataFrame** is a structure that contains **two-dimensional data** and its corresponding **labels**. Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). DataFrames are widely used in data science, machine learning, scientific computing and many other data-intensive fields.
- DataFrames are similar to SQL tables or the spreadsheets that one works with in Excel or Calc. In many cases, DataFrames are faster, easier to use and more powerful than tables or spreadsheets because they are an integral part of the Python and NumPy ecosystems.
- A Pandas Series is like a column in a table. It is a one-dimensional array holding data of any type.

1.10.1 Pandas Merge(): Combining Data on Common Columns or Indices

- Merge()can be used when functionality similar to a database's **join** operations is required. It is the most flexible operation that can be applied to data.
- When one wants to combine data objects based on one or more keys, similar to what is done in a relational database, merge() is the tool one can use. More specifically, merge() is most useful when one wants to combine rows that share data.
- One can achieve both **many-to-one** and **many-to-many** joins with merge(). In a many-to-one join, one of the datasets will have many rows in the merge column that repeat the same values. For example, the values could be 1, 1, 3, 5 and 5. At the same time, the merge column in the other dataset won't have repeated values. Take 1, 3 and 5 as an example.
- As it can be seen, in a many-to-many join, both of the merge columns will have repeated values. These merges are more complex and result in the cartesian product of the joined rows.
- This means that, after the merge, there will be every combination of rows that share the same value in the key column.
- What makes merge() so flexible is the sheer number of options for defining the behavior of the merge.

- Formerge() two arguments are required,
  1. The left DataFrame
  2. The right DataFrame.
- After that, it requires other optional arguments mentioned below, to define how the datasets are merged.
  - **how** defines what kind of merge to make. It defaults to 'inner', but other possible options include 'outer', 'left' and 'right'.
  - **on** tells merge() which columns or indices, also called **key columns or key indices**, one wants to join on. This is optional. If it isn't specified and left_index and right_index (covered below) are False, then columns from the two DataFrames that share names will be used as join keys. If on is used, then the specified column or index must be present in both objects.
  - **left_on** and **right_on** specify a column or index that's present only in the left or right object that is being merged. Both default to **None**.
  - **left_index** and **right_index** both default to False, but if it uses the index of the left or right object to be merged, then one can set the relevant argument to True.
  - **suffixes** is a tuple of strings to append to identical column names that aren't merge keys. This allows us to keep track of the origins of columns with the same name.
- These are some of the most important parameters to pass to merge().

Using merge()

- Before getting into the details of how to use merge(), one should first understand the various forms of joins :
  - Inner
  - Outer
  - Left
  - Right

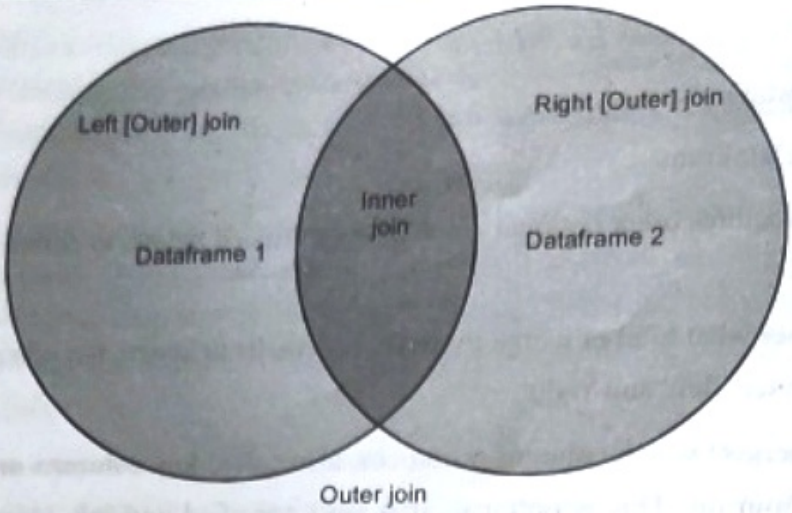

Fig. 1.10.1 Merge and Joins

- In this image, the two circles are the two datasets and the labels point to which part or parts of the datasets is expected to be seen.

1.10.2 Pandas .join() : Combining Data on a Column or Index

- While merge() is a **module function**, .join() is an **instance method** that lives on the DataFrame. This enables you to specify only one DataFrame, which will join the DataFrame call .join() on.
- Under the hood, .join() uses merge(), but it provides a more efficient way to join DataFrames than a fully specified merge() call.

Using .join()

- By default, .join() will attempt to do a left join on indices. If one wants to join on columns similar to merge(), then one needs to set the columns as indices.
- Like merge(), .join() has a few parameters that give more flexibility in the joins operations. However, with .join(), the list of parameters is relatively short,
  - **Other** is the only required parameter. It defines the other DataFrame to join. One can also specify a list of DataFrames here, allowing to combine a number of datasets in a single .join() call.
  - **On** specifies an optional column or index name for the left DataFrame to join the other DataFrame's index. If it's set to none, which is the default, then one will get an index-on-index join.
  - **How** has the same options as how from merge(). The difference is that it's index-based unless the columns are also specified.

- **lsuffix** and **rsuffix** are similar to suffixes in merge(). They specify a suffix to add to any overlapping columns but have no effect when passing a list of other DataFrames.
- **Sort** can be enabled to sort the resulting DataFrame by the join key.

Example Program - 1

```
#using merge function
import pandas as pd
# First DataFrame
left = pd.DataFrame(

  {
    "key": ["K0", "K1", "K2", "K3"],

    "A": ["A0", "A1", "A2", "A3"],

    "B": ["B0", "B1", "B2", "B3"],

  }
)

# Second DataFrame
right = pd.DataFrame(

  {

    "key": ["K0", "K1", "K2", "K3"],

    "C": ["C0", "C1", "C2", "C3"],

    "D": ["D0", "D1", "D2", "D3"],

  }
)
result = pd.merge(left, right, on="key")
print(result)
```

Example Program - 1 Output

key	A	B	C	D
0 K0	A0	B0	C0	D0
1 K1	A1	B1	C1	D1
2 K2	A2	B2	C2	D2
3 K3	A3	B3	C3	D3

1.10.3 Pandas concat() : Combining Data across Rows or Columns

- Concatenation is a bit different from the merging techniques. With merging, one can expect the resulting dataset to have rows from the parent datasets mixed in together, often based on some commonality. Depending on the type of merge, one might also lose rows that don't have matches in the other dataset.
- With concatenation, the datasets are just stitched together along an **axis** either the **row axis** or **column axis**. Visually, a concatenation with no parameters along rows would look as shown below,

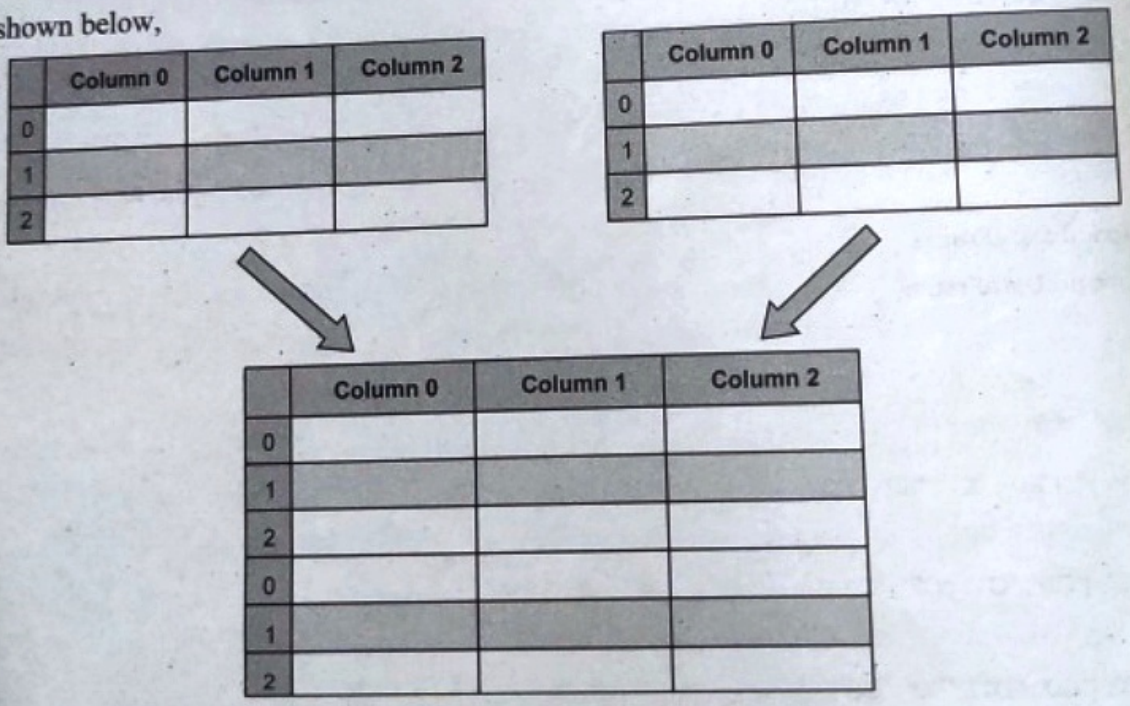

Fig. 1.10.2 Concatenation operation

Using concat()

- As it can be seen, concatenation is a simpler way to combine datasets. It's often used to form a single, larger set to do additional operations on.
- While concatenating datasets, one can specify the axis along which one wants to concatenate.

- By default, a concatenation results in **a set union**, where all data is preserved with the same operations like merge() and .join() as an outer join, with the required join parameter.
- If this parameter is used, then the default is outer, but there is an inner option available, which will perform an inner join or **set intersection**.
- As with the case of other inner joins, some data loss can occur when an inner join with concat() is carried out. Only here the axis labels match preserves rows or columns.
- Below are some of the other parameters that concat() takes,
  - **Objs** takes any sequence typically a list of Series or DataFrame objects to be concatenated. One can also provide a dictionary. In this case, the keys will be used to construct a hierarchical index.
  - **Axis** represents the axis that concatenate along. The default value is 0, which concatenates along the index or row axis. Alternatively, a value of 1 will concatenate vertically, along columns. One can also use the string values "index" or "columns".
  - **Join** is similar to the how parameter in the other techniques, but it only accepts the values inner or outer. The default value is outer, which preserves data, while inner would eliminate data that doesn't have a match in the other dataset.
  - **Ignore_index** takes a **Boolean** True or False value. It defaults to False. If True, then the new combined dataset won't preserve the original index values in the axis specified in the axis parameter. This gives entirely new index values.
  - **Keys** allows to construct a hierarchical index. One common use case is to have a new index while preserving the original indices so that one can tell which rows, for example, come from which original dataset.
  - **Copy** specifies whether one wants to copy the source data. The default value is True. If the value is set to False, then pandas won't make copies of the source data.

Example Program - 2

```
#Using concat function
import pandas as pd
# First DataFrame
df1 = pd.DataFrame({'id': ['T01', 'T02', 'T03', 'T04'],
                    'Name': ['Jiya', 'Riya', 'Maya', 'Piya']})
# Second DataFrame
df2 = pd.DataFrame({'id': ['R05', 'R06', 'R07', 'R08'],
                    'Name': ['Raam', 'Raaj', 'Naaz', 'Saaj']})
```

```
frames = [df1, df2]
result = pd.concat(frames)
print(result)
```

Example Program - 2 Output

```
   id  Name
0 T01  Jiya
1 T02  Riya
2 T03  Maya
3 T04  Piya
0 R05  Raam
1 R06  Raaj
2 R07  Naaz
3 R08  Saaj
```

1.11 Reshaping and Pivoting

1.11.1 Joining and Splitting Data - melt(), split(), pivot()

- In Pandas data reshaping means the transformation of the structure of a table or vector (i.e DataFrame or Series) to make it suitable for further analysis. Some of Pandas reshaping capabilities do not readily exist in other environments (e.g. SQL or bare bone R).
- Pandas has two methods namely, **melt()** and **pivot()**, to reshape the data.

melt()

- This method flattens/melts tabular data such that the specified columns and their respective values are transformed into key-value pairs. 'keys' are the column names of the dataset before transformation and 'values' are the values in the respective columns. After transformation, 'keys' are stored in a column named 'variable' and 'values' are stored in another column named 'value', by default.
- The columns of the data frame are transformed into key-value pairs.
- Unpivot a DataFrame from wide format to long format, optionally leaving identifier variables set.
- This function is useful to massage a DataFrame into a format where one or more columns are identifier variables (id_vars), while all other columns, considered measured variables (value_vars), are "unpivoted" to the row axis, leaving just two non-identifier columns, 'variable' and 'value'.

Syntax

- pandas.melt(frame, id_vars=None, value_vars=None, var_name=None, value_name='value', col_level=None)

Parameters :

Name	Description	Type	Required / Optional
frame	DataFrame	DataFrame - Contains list, numbers, strings	Required
id_vars	Column(s) to use as identifier variables.	tuple, list or ndarray	Optional
value_vars	Column(s) to unpivot. If not specified, uses all columns that are not set as id_vars.	tuple, list or ndarray	Optional
var_name	Name to use for the 'variable' column. If None it uses frame.columns.name or 'variable'.	scalar	Required
value_name	Name to use for the 'value' column.	scalar, default 'value'	Required
col_level	If columns are a MultiIndex then use this level to melt.	int or string	Optional

Returns : Unpivoted DataFrame.

Example Program - 3

```
#using melt() function
import pandas as pd
df1 = {"Name": ["Ravi", "Aniket", "Ana"], "ID": [1, 2, 3], "Role": ["CEO", "Editor", "Author"]}
df = pd.DataFrame(df1)
print(df)
df_melted = pd.melt(df, id_vars=["ID"], value_vars=["Name", "Role"])
print(df_melted)
```

Example Program - 3 Output

	Name	ID	Role
0	Ravi	1	CEO
1	Aniket	2	Editor
2	Ana	3	Author

	ID	variable	value
0	1	Name	Ravi
1	2	Name	Aniket
2	3	Name	Ana
3	1	Role	CEO
4	2	Role	Editor
5	3	Role	Author

pivot()

- This method does the reverse of what melt() did. It transforms the key-value pairs into columns.
- It returns a reshaped DataFrame organized by given index / column values.
- Reshape data (produce a "pivot" table) based on column values. Uses unique values from specified index / columns to form axes of the resulting DataFrame. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns.
- pandas.pivot(data, index=None, columns=None, values=None)[source]

Parameters :

Name	Description	Type	Required / Optional
Data	DataFrame	DataFrame - Contains list, numbers, strings	Required
Index	Column to use to make a new frame's index. If None, uses existing index.	String or object	Optional
Columns	Column to use to make new frame's columns.	String or object	Required
Values	Column(s) to use for populating new frame's values. If not specified, all remaining columns will be used and the result will have hierarchically indexed columns.	String, object or a list of the previous	Optional

Returns : Returns reshaped DataFrame.

Raises : ValueError : When there are any index, column combinations with multiple values.

Example Program - 4

```
import pandas as pd

d1 = {"Name": ["Ravi", "Aniket", "Ana"], "ID": [1, 2, 3], "Role": ["CEO", "Editor", "Author"]}

df = pd.DataFrame(d1)

# print(df)

df_melted = pd.melt(df, id_vars=["ID"], value_vars=["Name", "Role"], var_name="Attribute",
value_name="Value")
print(df_melted)
# unmelting using pivot()
df_unmelted = df_melted.pivot(index='ID', columns='Attribute')
print(df_unmelted)
```

Program Output - 4

```
   ID  Attribute  Value
0   1       Name    Ravi
1   2       Name  Aniket
2   3       Name     Ana
3   1       Role     CEO
4   2       Role  Editor
5   3       Role  Author


        Value
Attribute  Name    Role
ID
1          Ravi     CEO
2        Aniket  Editor
3           Ana  Author
```

1.11.2 Transformation Techniques

1. Grouping data sets

- **Pandas groupby** is used for grouping the data according to the categories and applying a function to the categories. It also helps to aggregate data efficiently.
- Pandas **dataframe.groupby()** function is used to split the data into groups based on some criteria. pandas objects can be split on any of their axes. The abstract definition of grouping is to provide a mapping of labels to group names.
- groupby() is a very powerful function with a lot of variations. It makes the task of splitting the dataframe over some criteria really easy and efficient.
- Any **groupby** operation involves one of the following operations on the original object. They are,
  - **Splitting** the object
  - **Applying** a function
  - **Combining** the results.

Syntax

- DataFrame.groupby(by=None, axis=0, level=None, as_index=True, sort=True, group_keys=True, squeeze=False, **kwargs)

Parameters :

- **By :** mapping, function, str or iterable
- **Axis :** int, default 0
- **Level :** If the axis is a MultiIndex (hierarchical), group by a particular level or levels
- **As_index :** For aggregated output, return an object with group labels as the index. Only relevant for DataFrame input. as_index = False is effectively "SQL-style" grouped output
- **Sort :** Sort group keys. Get better performance by turning this off. Note this does not influence the order of observations within each group. groupby preserves the order of rows within each group.
- **Group_keys :** When calling apply, add group keys to index to identify pieces
- **Squeeze :** Reduce the dimensionality of the return type if possible, otherwise return consistent type

Returns : GroupBy object

- In many situations, the data is split into sets and some functionality can be applied on each subset. In the applied functionality, one can perform the following operations.
  - **Aggregation** - Computing a summary statistic
  - **Transformation** - Perform some group-specific operation
  - **Filtration** - Discarding the data with some condition.

2. Split data into groups

- Pandas objects can be split into any of their objects. There are multiple ways to split an object like -
  - obj.groupby('key')
  - obj.groupby(['key1','key2'])
  - obj.groupby(key,axis=1)

3. Data aggregations

- An aggregated function returns a single aggregated value for each group. Once the **group by** object is created, several aggregation operations can be performed on the grouped data.

Applying multiple aggregation functions at once

- With grouped series, one can also pass a **list** or **dict of functions** to do aggregation with and generate DataFrame as output

4. Transformations

- Transformation on a group or a column returns an object that is indexed the same size as that of that is being grouped. Thus, the transform should return a result that is the same size as that of a group chunk.

5. Filtration

- Filtration filters the data on a defined criteria and returns the subset of data. The **filter()** function is used to filter the data.

Example Program - 5

```
#grouping data
import pandas as pd
punepl_data = {'Team': ['Warriors', 'Fighters', 'Supers', 'Fighters', 'Warriors',
 'Fighters', 'Fighters', 'Supers', 'Supers', 'Fighters', 'Warriors', 'Supers'],
 'Rank': [1, 2, 2, 3, 3,4 ,1 ,1,2 , 4,1,2],
```

```
  'Year': [2004,2005,2004,2005,2004,2005,2006,2007,2006,2004,2005,2007],
  'Points':[876,789,863,673,741,812,756,788,694,701,804,690]}
df = pd.DataFrame(punepl_data)
print (df)
print (df.groupby('Team').groups)

#iterating through groups
grouped = df.groupby('Year')
for name,group in grouped:
  print (name)
  print (group)
#applying grouping and then applying aggregate function
grouped = df.groupby('Year')
print (grouped['Points'].agg(np.mean))
#applying multiple aggregate functions
grouped = df.groupby('Team')
print (grouped['Points'].agg([np.sum, np.mean, np.std]))
#applying filtration
# filter condition, return the teams which have participated four or more times in IPL
print (df.groupby('Team').filter(lambda x: len(x)>=4))
```

Program Output - 5

	Team	Rank	Year	Points
0	Warriors	1	2004	8761
1	Fighters	2	2005	7892
2	Supers	2	2004	8633
3	Fighters	3	2005	6734
4	Warriors	3	2004	7415
5	Fighters	4	2005	8126
6	Fighters	1	2006	7567
7	Supers	1	2007	7888
8	Supers	2	2006	6949
9	Fighters	4	2004	7010
10	Warriors	1	2005	8014
11	Supers	2	2007	6902

```
{'Fighters': [1, 3, 5, 6, 9], 'Supers': [2, 7, 8, 11], 'Warriors': [0, 4, 10]}
```

2004

	Team	Rank	Year	Points
0	Warriors	1	2004	8761
2	Supers	2	2004	8633
4	Warriors	3	2004	7415
9	Fighters	4	2004	7010

2005

	Team	Rank	Year	Points
1	Fighters	2	2005	7892
3	Fighters	3	2005	6734
5	Fighters	4	2005	8126
10	Warriors	1	2005	8014

2006

	Team	Rank	Year	Points
6	Fighters	1	2006	7567
8	Supers	2	2006	6949

2007

	Team	Rank	Year	Points
7	Supers	1	2007	7888
11	Supers	2	2007	6902

Year

2004 7954.75

2005 7691.50

2006 7258.00

2007 7395.00

Name : Points, dtype: float64

	sum	mean	std
Team			
Fighters	37329	7465.800000	585.456403
Supers	30372	7593.000000	828.822860
Warriors	24190	8063.333333	674.354753

	Team	Rank	Year	Points
1	Fighters	2	2005	7892
2	Supers	2	2004	8633
3	Fighters	3	2005	6734
5	Fighters	4	2005	8126
6	Fighters	1	2006	7567
7	Supers	1	2007	7888
8	Supers	2	2006	6949
9	Fighters	4	2004	7010
11	Supers	2	2007	6902

6. Pivot-tables and cross tabulations

- A pivot table is a table of statistics that helps summarize the data of a larger table by "pivoting" that data. Microsoft Excel popularized the pivot table, where they are known as **PivotTables**.
- *Pivot table in pandas is a tool to summarize one or more numeric variable based on two other categorical variables.* Pandas gives access to creating pivot tables in Python using the .pivot_table() function.
- **pandas.pivot_table(data, values=None, index=None, columns=None, aggfunc='mean', fill_value=None, margins=False, dropna=True, margins_name='All')** create a spreadsheet-style pivot table as a DataFrame.
- Levels in the pivot table will be stored in MultiIndex objects (hierarchical indexes) on the index and columns of the result DataFrame.

Parameters :

- **Data :** DataFrame
- **Values :** Column to aggregate, optional
- **Index :** Column, Grouper, array or list of the previous
- **Columns :** Column, Grouper, array or list of the previous
- **Aggfunc :** Function, list of functions, dict, default numpy.mean.

  -> If list of functions passed, the resulting pivot table will have hierarchical columns whose top level are the function names.

  -> If dict is passed, the key is column to aggregate and value is function or list of functions.

- **fill_value[scalar, default None]** : Value to replace missing values with **margins[boolean, default False]** : Add all row / columns (e.g. for subtotal / grand totals)
- **dropna[boolean, default True]** : Do not include columns whose entries are all NaN **margins_name[string, default 'All']** : Name of the row / column that will contain the totals when margins is True.
- **Returns :** DataFrame

Example Program - 5

```
import pandas as pd
import numpy as np

df = pd.DataFrame({'First Name': ['Mina', 'Sima', 'Rima', 'Sita', 'Rita', ],
          'Last Name': ['Das', 'Wani', 'Kapur', 'Pande', 'Kumar'],
          'Type': ['Full-time Employee', 'Intern', 'Full-time Employee',
               'Part-time Employee', 'Full-time Employee'],
          'Department': ['Administration', 'Technical', 'Administration',
                    'Technical', 'Management'],
          'YoE': [2, 3, 5, 7, 6],
          'Salary': [20000, 5000, 10000, 10000, 20000]})
print (df)
#creating pivot table
output = pd.pivot_table(data=df,
               index=['Type'],
               columns=['Department'],
               values='Salary',
               aggfunc='mean')
print (output)

# Pivot table with multiple aggfuncs
output = pd.pivot_table(data=df, index=['Type'],
               values='Salary',
               aggfunc=['sum', 'mean', 'count'])
print (output)
```

```
# Calculate row and column totals (margins)
output = pd.pivot_table(data=df, index=['Type'],
                        values='Salary',
                        aggfunc=['sum', 'mean', 'count'],
                        margins=True,
                        margins_name='Grand Total')
print (output)
```

Program Output - 6

	First Name	Last Name	Type	Department	YoE	Salary
0	Mina	Das	Full-time Employee	Administration	2	20000
1	Sima	Wani	Intern	Technical	3	5000
2	Rima	Kapur	Full-time Employee	Administration	5	10000
3	Sita	Pande	Part-time Employee	Technical	7	10000
4	Rita	Kumar	Full-time Employee	Management	6	20000

Department	Administration	Management	Technical
Type			
Full-time Employee	15000.0	20000.0	NaN
Intern	NaN	NaN	5000.0
Part-time Employee	NaN	NaN	10000.0

	sum	mean	count
	Salary	Salary	Salary
Type			
Full-time Employee	50000	16666.666667	3
Intern	5000	5000.000000	1
Part-time Employee	10000	10000.000000	1

	sum	mean	count
	Salary	Salary	Salary
Type			
Full-time Employee	50000	16666.666667	3
Intern	5000	5000.000000	1
Part-time Employee	10000	10000.000000	1
Grand Total	65000	13000.000000	5

- **The Pandas crosstab function is one of the many ways in which Pandas allows to customize data**. On the surface, it appears to be quite similar to the Pandas pivot table function.

Review Questions with Answers

1. *Explain various data collection methods.* **(Refer section 1.1)**
2. *What is EDA ? What is significance of EDA ?* **(Refer section 1.1)**
3. *Define data science.* **(Refer section 1.2)**
4. *What do you mean by making sense of data ?* **(Refer section 1.5)**
5. *Discuss various visual aids for EDA.* **(Refer section 1.8)**

1.12 Two Marks Questions and Answers

Q.1 List widely used data collection tools.

Ans. : Below are some of the widely used tools for data collection,

- **Word Association :** The researcher gives the respondent a set of words and asks them what comes to mind when they hear each word.
- **Sentence Completion :** Researchers use sentence completion to understand what kind of ideas the respondent has. This tool involves giving an incomplete sentence and seeing how the interviewee finishes it.
- **Role-Playing :** Respondents are presented with an imaginary situation and asked how they would act or react if it was real.
- **In-Person Surveys :** The researcher asks questions in person.
- **Online / Web Surveys :** These surveys are easy to accomplish, but some users may be unwilling to answer truthfully, if at all.
- **Mobile Surveys :** These surveys take advantage of the increasing proliferation of mobile technology. Mobile collection surveys rely on mobile devices like tablets or smartphones to conduct surveys via SMS or mobile apps.

Q.2 Define exploratory data analysis.

Ans. : Exploratory Data Analysis (EDA) is defined by John W. Tukey, is a process of examining or understanding the data and extracting insights or main characteristics of the data. EDA is generally classified into two methods, that is, **graphical** analysis and **non-graphical** analysis.

It is also known as **visual analytics or descriptive statistics**. It is the practice of inspecting and exploring data, before stating hypotheses, fitting predictors and other expected inferential goals. It typically includes the computation of simple summary statistics which capture some property of interest in the data and **visualization**. EDA can be thought of as an assumption-free, purely algorithmic practice.

Q.3 Explain multivariate graphical methods for data analysis.

Ans. : Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.

Common types of multivariate graphics include,

- Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.
- Multivariate chart, which is a graphical representation of the relationships between factors and a response.
- Run chart, which is a line graph of data plotted over time.

Q.4 Write a note on box plots and bar plots.

Ans. : Box plots show the distribution of values along an axis. Rectangular boxes are used in order to bucket the data, giving us an idea of how the data points are spread out. These boxes are also called **quartiles** which represent a quarter of a data set. Boxes can be drawn vertically or horizontally. One can also easily spot the outliers, which are usually treated as abnormal values and affect the data set's overall observation due to their very high or low values. Box plots are suitable for identifying outliers.

Bar charts can be used to compare nominal or ordinal data. They are helpful for recognizing trends.

Q.5 Write a Python program to demonstrate use merge() function.

Ans. : Refer example program 1.

❑❑❑