DATA SCIENCE

Data-->collection of facts. Ex: name ,Contact No,Address,Aadhar no, "I am Dora ,I love dosa" ,likes,views,photos,comments

Data science --> research (collecting, analyzing and interpreting data to gather)and reporting purposes -->Prediction -->To make decisions -->Development

"Data science is the science of analyzing raw data using statistics and machine learning techniques with the purpose of conclusions about that information"

AI&DS --> To detect and optimize use Tools and automate process.

Data Science process/Role of a Data Scientist

Extracting usable data from data sources --> Using machine learning tools select features --> prepossessing --> Enhancing datacollection -->(Processing, cleansing, and validating the integrity of data to be used for) data analysis --> To find patterns --> Developing prediction systems -->Presenting results -->Propose solutions and strategies to tackle business challenges

Benefits of Data Science in Business

- Improves business predictions
- Interpretation of complex data
- Better decision making
- Improves data security
- Development of user-centric products

Applications of Data Science

- Chat bot
- Product Recommendation
- Fraud and Risk Detection
- Self-Driving Car
- Image Recognition
- Speech to text Convert
- Healthcare
- Search Engines

Applications of Data Science

- Anomaly detection (fraud, disease and crime)
- Classification (background checks; an email server classifying emails as “important”)
- Forecasting (sales, revenue and customer retention)
- Pattern detection (weather patterns, financial market patterns)
- Recognition (facial, voice and text)

Applications of Data Science

- Recommendation (based on learned preferences, recommendation engines can refer you to movies, restaurants and books)
- Regression (predicting food delivery times, predicting home prices based on amenities)
- Optimization (scheduling ride-share pickups and package deliveries)

Job Roles in Data Science

- Data Analyst --> mining huge amounts of data, looking for patterns, relationships, and trends, and coming up with compelling visualizations and reports -->To make business decisions.
- Data Engineer --> data cleansing, data extraction, and data preparation for businesses
- Data Scientist --> compelling business insights through the deployment of various techniques, methodologies, algorithms, Data Science tools

Data science Vs Data Analytics

Criteria	Data Science	Data Analytics
Skills Needed	✔ Data capturing ✔ statistics ✔ problem-solving	✔ Analytical ✔ mathematical ✔ statistical skills
Data Used	All types of data	Mostly structured and numeric data
Life Cycle	✔ Explore ✔ discover ✔ investigate ✔ visualize	✔ Report ✔ predict ✔ prescribe ✔ optimize

Data Science tools

Data Analysis –-> Informatica PowerCenter, Rapidminer, Excel, SAS

Data Visualization –-> Tableau, Qlikview, RAW, Jupyter

Data Warehousing –->Apache Hadoop, Informatica/Talend, Microsoft HD insights

Data Modelling –-> H2O.ai, Datarobot, Azure ML Studio, Mahout

Data Science Tools

- Microsoft Excel (data analytics tool)
- Microsoft Power BI (business intelligence data analytics and data visualization tool)
- MongoDB (database tool)
- Apache Spark (data analytics tool)
- Apache Hadoop (big data tool)
- KNIME (data analytics tool)

Data Science process/Role of a Data Scientist

Extracting usable data from data sources --> Using machine learning tools select features --> prepossessing --> Enhancing datacollection -->(Processing, cleansing, and validating the integrity of data to be used for) data analysis --> To find patterns --> Developing prediction systems -->Presenting results -->Propose solutions and strategies to tackle business challenges

https://youtu.be/7Z7JItvbnt8

EDA fundamentals

Exploratory data analysis (EDA) -->used by data scientists

-->To analyze and investigate data sets and summarize their characteristics --> To discover patterns

EDA provides a better understanding of data set variables and the relationships between them

Objective of Exploratory Data Analysis

To obtain vital insights

- Identifying and removing data outliers
- Identifying trends in time and space
- Uncover patterns related to the target
- Creating hypotheses and testing them through experiments
- Identifying new sources of data

Role of EDA in Data Science

- analysis --> Discover patterns -->To meet the expected goals of the business.
- data science involves building models for prediction.
- EDA ensures that the correct ingredients in patterns and trends are made available for training the model to achieve the correct outcome

Steps in Exploratory Data Analysis (EDA)

1. Data Collection -data from various sources through surveys, social media, and customer reviews to understand and get valuable insights from them.

Business goal -->Types of data(Data collection) -->identify data source--> collecting sufficient and relevant data -->activities begin.

2. Finding all Variables and Understanding Them- available data--> identifying the important variables which affect the outcome and their possible impact.

Steps in Exploratory Data Analysis (EDA)

3. Cleaning the Dataset - It may contain null values and irrelevant information.Preprocessing takes care of all issues, such as identifying null values, outliers, anomaly detection, etc

4. Identify Correlated Variables- Finding a correlation between variables helps to know how a particular variable is related to another-->correlation matrix method

Steps in Exploratory Data Analysis (EDA)

5. Choosing the Right Methods or Tools -- depending on the data, categorical or numerical, the size, type of variables, and the purpose of analysis, different statistical tools are employed

Statistical formulae applied for numerical outputs give fair information

graphical visuals are more appealing and easier to interpret.

Steps in Exploratory Data Analysis (EDA)

6. Visualizing and Analyzing Results -The findings are to be observed -->The Patterns in the data and correlation between variables give good insights for making suitable changes in the data variables.

Importance of EDA in Data Science

- To make data 'clean' and To eliminate or resolve all of the dataset's undesirable qualities.
- Identifying incorrect data points and missing or duplicate values
- Identifying the relationship between the variables
- broader view of the data
- Expanding data on dataset
- Identifying Outliers or abnormal occurrences in a dataset
- Evaluate the dataset's statistical measurements.

EDA Tools

- Python(Plotly, Seaborn, or Matplotlib)
- R
- Matlab

Pandas – for reading the dataset files

Seaborn – for graphical visualization of the data

Numpy – for numerical calculations

Matplotlib – for graphic visualization

Sklearn – for scaling the dataset

Types of EDA

Primary types of EDA:

- Univariate Analysis
- Bi-Variate analysis
- Multivariate Analysis

Report Format:

Scatter plot, chart,map

- Univariate Analysis-->analyze only one variable at a time
--> To describe the data and find patterns that exist within it.

- Bi-Variate analysis-->Two different variables-->to find out the relationship between the two variables

- Multivariate Analysis -->more variables

Advantages of Using EDA

1. Gain Insights Into Underlying Trends and Patterns
2. Improved Understanding of Variables
3. Better Preprocess Data to Save Time
4. Make Data-driven Decisions

Example of Exploratory Data Analysis

Sample Dataset

"Students Performance in Exams"

Gold medalist -- university rankers -- Pass Percentage -- Subject wise toppers -- % of people got Top grades -- full score in any one subject

NEET Exam :

Eligible candidates,Toppers,Rank list,Pass percentage

OVERVIEW

Data Exploration

Data Exploration --> It retrieves data from existing data sources.

Data retrieval - Data harvesting - Data crawling - Data Exploration

Data Visualization

Data Visualization is the process of presenting data in the form of graphs or charts.

EDA

Exploratory data analysis --> To gather information about the possible solutions and the affecting factors --> To find the best possible solution that provides suitable results after processing the data.

A dataset contains many observations about a particular object.

Universities – students' and teachers' data,
Medical researchers – patients' data,
Real estate industries -- storehouse and building datasets

Types of data

Numbers, string, date, image, audio, video

For example, dataset about students in university

A student Reg no(ID), name, address, weight, date of birth, address, email, and gender, Aadhar no, course ,photo

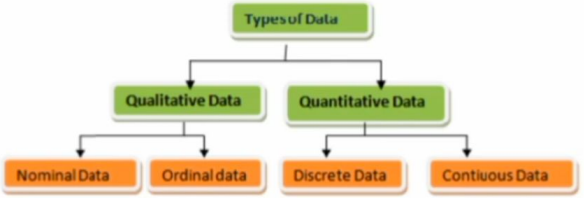
Features that describes a variables

Student Reg no : 80114101007
Name : Siva
Address : 78 st name , Abcd city , taluk name ,dist name ,state name ,pin code.
Date of birth : 06.12.1990
Email : siva123*321@gmail.com
Weight : 56.5 kg
Gender : Male
Aadhaar no : 1234 5678 9980

Photo:

Types of data

- Categorical data or Qualitative data
  1. Nominal data
  2. Ordinal data
- Numerical data or Quantitative data
  1. Discrete Data
  2. Continuous Data



Qualitative data-- Qualitative or Categorical Data is data that can't be measured or counted in the form of numbers.

These types of data are sorted by category, not by number.

customers' tastes, perception of people

Format of Data :: audio, images, symbols, or text.

Ex:

- What language do you speak ?
- Favourite food items?
- Opinion on something (agree, disagree, or neutral)

Gender : Male, Female, Other, or Unknown

Marital Status : Married, Unmarried, Widowed, Divorced, Legally Separated, Domestic Partner, Never Married, or Unknown

Movie genres : Action, Adventure, Comedy, Crime, Drama, Fantasy, Historical, Horror, Mystery, Philosophical, Political, Romance

Blood type : A, B, AB, or O

Qualitative data

1.A binary categorical variable - exactly two values

Ex: either success or failure ,True or False

2. Polytomous variables- more than two possible values

For example

marital status :divorced, legally separated, married, never married, unmarried, widowed, and unknown

categorical dataset follows either nominal or ordinal measurement scales.

Nominal

1. Nominal Data is used to label variables
2. In nominal data, we can't do any numerical tasks-
3. These data don't have any meaningful order
4. Their values are distributed into distinct categories.

Eg: Industry -- "financial," "engineering," "retail."
Nationality - Indian, German, American
Marital status (Single, Widowed, Married)

Ordinal

1. number is present in any one kind of order
2. we can't do any arithmetical tasks on data.
3. used for observation like customer satisfaction, happiness,
   grades in the exam (A, B, C, D)
   Ranking of student in a Exam(First, Second, Third, etc.)
   Star rating

Eg: How do u rate your satisfaction

1.not satisfied
2.Ok
3.satisfied
4.Very satisfied

Quantitative data

1. Quantitative data can be expressed in numerical values.
2. used for statistical manipulation.
3. data can be represented on a graphs and charts

Eg:

weight of a student
Your shoe size
Scores of exam (99, 70, 45, etc.)s

1.Discrete data- countable -- a fixed number Or whole numbers.

Eg : student in a classroom ,how many students are in "A " grade

2. Continuous data - The continuous variable can take fractional numbers as value within a range.

Eg : weight variable , version of software, Market share price

Conclusion

- Types of data
- Features that describes a variables
- Qualitative data
  - 1.Nominal data — Ex: Nationality - Indian, German, American
  - 2.Ordinal data — Ex: How do u rate your satisfaction *****
- Quantitative data
  - 1.Discrete data — Ex: how many students are in "A " grade
  - 2.Continuous data — Ex:Market share price

Can you justify

Dataset -- datasets are collections of data that are stored

Which of the variables are continuous?

Which of the variables are discrete? JUSTIFY THE Variable types

Student Reg nno	NAME	GENDER	DOB	EMAIL	WEIGHT(KG)	%	ADDRESS	REMARKS
8421	SAJAYAN	MALE	10.10.2009	sajay@gmail.com	45	90	Boys st,Top city,super TK,chennai DT	GOOD
8422	PRAJESH	MALE	04.02.2008	prajesh04@gmail.com	54	98.05	Boys st,Top city,super TK,chennai DT	EXCELLENT
8423	GUGAN	MALE	05.03.2008	gugan05#03@gmail.com	72	92.16	Boys st,Top city,super TK,chennai DT	GOOD
8424	VETRI	MALE	08.04.2009	vetri@gmail.com	39	85.1	Boys st,Top city,super TK,chennai DT	NOT BAD
8425	SAI PRASATH	MALE	04.06.2008	sai1990@gmail.com	60	75.3	Boys st,Top city,super TK,chennai DT	NOT BAD
8426	ROHIT	MALE	10.02.2010	rohitkarti@gmail.com	48	85.6	Boys st,Top city,super TK,chennai DT	NOT BAD
8427	SAI RAKSHAN	MALE	07.06.2008	saikutty2003@gmail.com	38	91	Boys st,Top city,super TK,chennai DT	OK
8428	DHARSHIT	MALE	06.05.2009	dharsit07@gmail.com	40	92.5	Boys st,Top city,super TK,chennai DT	OK
8429	DHIYA	FEMALE	01.01.2007	dhiyapretty@gmail.com	35	98.8	GIRLS st, pretty city,Pleasant TK,chennai DT	EXCELLENT
8430	DEEKSHI	FEMALE	05.06.2008	deekshisweety@gmail.com	30	90.5	GIRLS st, pretty city,Pleasant TK,chennai DT	OK

Data Analysis Types

Data analysis approach: problem → conclusions(meaningful insights from data)

1.Classical Data Analysis

Problem Definition => Data collection => Model development => data Analysis => Conclusions

2.Exploratory Data Analysis(EDA)

Problem Definition => Data collection => data Analysis => Model development => Conclusions

3.Bayesian Data Analysis

Problem Definition => Data collection => Model development => Prior Distribution => Analysis => Conclusions

Classical data Analysis

- It Analyse quantitative data and produce numeric output
- finding trends in data through statistics and probability
- Classical techniques are generally quantitative

Classical techniques to analyze data

1.)ANOVA 2.)T-tests 3.)Chi-squared tests 4.)F tests

Eg: sensex data

Classical→Male and female ratio, which state has maximum population

Exploratory Data Analysis(EDA)

EDA-It produce graphical output Eg: lineplot, scatter plot, histogram

- maximize insights into dataset.
- extract important variables
- detect outliers and anomalies
- Test assumptions

EDA techniques

1)Scatter plots 2)Histograms 3)Box Plots 4) Residual Plots

Eg: sensex data

Age-group wise migration of people, Number of people in medical field, Population in year wise

Bayesian Data Analysis

Bayesian data analysis -probability distribution knowledge(Evidence) into the analysis → only 2 possibilities of output

Eg:

e-mail spam filtering→ yes or no

share value prediction→ increase or decrease

Coin flips→ head or tail, Result → pass or fail

Eg: sensex data

Corona affected status→ positive or negative

Most affected gender→ Male or Female

Comparing Data Analysis

Comparing Data Analysis		
Classical Data Analysis	**Explotary Data Analysis**	**Bayesian Data Analysis**
quantitative data	Data set	probability distribution
Numeric output	Graphical output	only 2 Possible Output
ANOVA,T-tests ,Chi-squared tests	Scatter plots ,Histograms ,Box Plots	Bayesian Distribution

Software tools for EDA

- Python- Data analysis, Data mining, and Data science
- R programming language- statistical computation and graphical data analysis
- Weka - Data mining package
- KNIME- Data analysis
- MATLAB- Mathematical calculation

Python

Python is adopted for data analysis and data mining by data science experts.

- Fundamental concepts of variables
- string and data types
- Conditionals and functions
- Sequences, collections, and iterations
- Working with files
- Object-oriented programming

NumPy

- Create arrays with NumPy
- copy arrays and divide arrays
- Perform different operations on NumPy arrays
- Understand array selections
- advanced indexing and expanding.Working with multi-dimensional arrays
- Linear algebraic functions and built-in NumPy functions

Eg:

```
import numpy as np
myArray = np.array([11, 18, 32, 64])
print(myArray)
```

Pandas

- Understand and create Data Frame objects
- Sub setting data and indexing data
- Arithmetic functions, and mapping with pandas
- Managing index Building style for visual analysis

Eg

```
import pandas as pd
print("Pandas Version:", pd.__version__)
series = pd.Series([2, 3, 7, 11, 13, 17, 19, 23])
print(series)
```

SciPy

SciPy - SciPy is a scientific library for Python and is open source

- Importing the package
- Using statistical packages from SciPy
- Performing descriptive statistics
- Inference and data analysis

Matplotlib

Matplotlib - Matplotlib is a library for creating static, animated, and interactive visualizations in Python

Matplotlib consists of several plots like line, bar, scatter, histogram etc.

- Loading linear datasets
- Adjusting axes, grids, labels, titles, and legends
- Saving plots

Conclusion

1. Data Analysis Types
2. Comparing EDA
3. Software tools for EDA
4. Python
5. Matplotlib

Visual Aids for EDA

Data analysis approach: problem → conclusions(meaningful insights from data)

DATA EXPLORATION AND VISUALIZATION

Extract knowledge from the data and To present the data

Matplotlib(python)

Matplotlib -making 2D plots for from data in array →It is open-source, cross-platform

pyplot -A set of methods

Installation of Matplotlib

pip install matplotlib

Data Visualization

Data visualization is the presentation qualitative and quantitative of data in graphical format.

Benefits

1.Visualize data (make sense of data)
2.Classify and categorise data
3.Find a relationship among data
4.Understanding distribution and composition of data
5.To find a patterns and trends
6.Detect outliers and anomalies in data
7.To predict future
8.To make Decisions

Data Types

Quantitative data can be measured – eg:customer count ,sales, weight

Qualititative Data can be classified /categorised eg: colors,ranking,satisfaction

Discrete Data→ Quantitative data with a finite number of values eg:500 customer,85kg

Continuous data→ Quantitative data within a range eg: Last year sales

Nominal data→ Qualititative data –no order eg: colors

Ordinal data→ Qualititative data has order eg: very satisfied, satisfied, not satisfied

Determine your visualization goal

1. **Determine your visualization goal**

Inform: convey a single important message or data point that doesn't require much context to understand

- **Compare**: show similarities or differences among values or parts of a whole
- Show **Change:** visualize trends over time or space
- **Organize: show groups, patterns, rank or order**
- **Reveal Relationships:** show correlations among variables or values

2.Choose the best types of charts to achieve that goal

Types of techniques used in visualization

Line chart
Bar chart
Scatter plot
Area plot and stacked plot
Pie chart
Table chart
Polar chart
Histogram
Lollipop chart
Choosing the best chart

Line chart

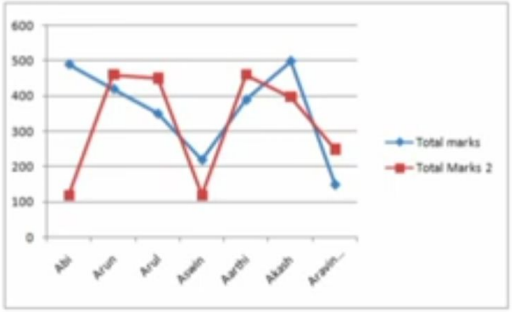
line chart

- ✔ The relationship between two or more continuous variables (That changes continuously over time) Eg: weight of student ,share prices
- ✔ To display the information as a series of the line

Input: x,y axis values

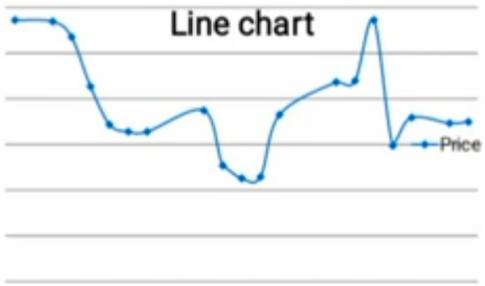
sample line chart:

Student name	Total marks	Total Marks 2
Abi	489	120
Arun	420	460
Arul	350	450
Aswin	220	120
Aarthi	389	460
Akash	498	398
Aravindh	150	250



Line chart

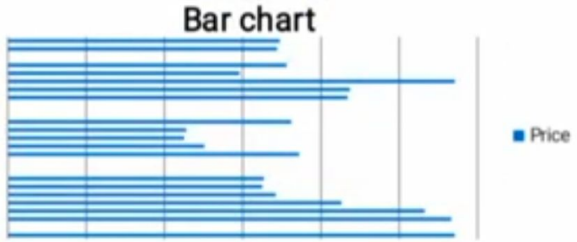
Date	Price	Open	High	Low	Volume	Chg%
25-Sep-23	1,944.75	1,944.40	1,946.60	1,940.25	0	0.02%
24-Sep-23	1,944.40	1,944.95	1,946.05	1,944.10	0	-0.06%
22-Sep-23	1,945.60	1,939.60	1,949.10	1,939.60	0	0.31%
21-Sep-23	1,939.60	1,952.00	1,952.20	1,933.10	0	-1.40%
20-Sep-23	1,967.10	1,953.00	1,968.90	1,948.60	220.98K	0.69%
19-Sep-23	1,953.70	1,955.30	1,958.90	1,950.90	131.10K	0.02%
18-Sep-23	1,953.40	1,945.70	1,955.70	1,943.80	138.03K	0.37%
15-Sep-23	1,946.20	1,932.50	1,952.40	1,931.20	199.77K	0.69%
14-Sep-23	1,932.80	1,930.70	1,934.50	1,921.70	201.74K	0.02%
13-Sep-23	1,932.50	1,935.20	1,938.40	1,927.20	161.00K	-0.13%
12-Sep-23	1,935.10	1,945.60	1,947.50	1,929.90	161.99K	-0.62%
11-Sep-23	1,947.20	1,943.30	1,954.60	1,939.50	131.08K	0.23%
08-Sep-23	1,942.70	1,943.80	1,954.00	1,940.80	138.69K	0.01%
07-Sep-23	1,942.50	1,942.20	1,947.90	1,940.30	117.17K	-0.09%
06-Sep-23	1,944.20	1,951.50	1,954.50	1,940.00	148.78K	-0.43%
05-Sep-23	1,952.60	1,966.70	1,972.60	1,950.60	192.79K	-0.54%
04-Sep-23	1,963.25	1,966.60	1,972.55	1,962.55	0	-0.17%
03-Sep-23	1,966.65	1,966.40	1,967.05	1,965.50	0	-0.02%
01-Sep-23	1,967.10	1,966.40	1,980.20	1,960.70	160.73K	0.06%



Bar chart

Bar - Each rectangle is a representation of one category.. The differences in bar length, in size and color are easier to perceive

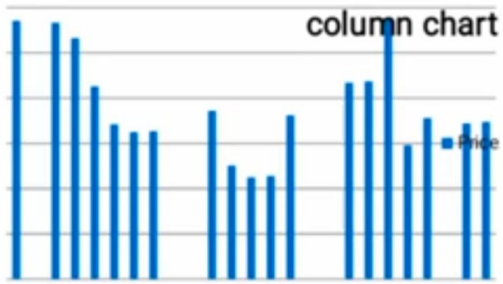
Date	Price	Open	High	Low	Volume	Chg%
25-Sep-23	1,944.75	1,944.40	1,946.60	1,940.25	0	0.02%
24-Sep-23	1,944.40	1,944.95	1,946.05	1,944.10	0	-0.06%
22-Sep-23	1,945.60	1,939.60	1,949.10	1,939.60	0	0.31%
21-Sep-23	1,939.60	1,952.00	1,952.20	1,933.10	0	-1.40%
20-Sep-23	1,967.10	1,953.00	1,968.90	1,948.60	220.98K	0.69%
19-Sep-23	1,953.70	1,955.30	1,958.90	1,950.90	131.10K	0.02%
18-Sep-23	1,953.40	1,945.70	1,955.70	1,943.80	138.03K	0.37%
15-Sep-23	1,946.20	1,932.50	1,952.40	1,931.20	199.77K	0.69%
14-Sep-23	1,932.80	1,930.70	1,934.50	1,921.70	201.74K	0.02%
13-Sep-23	1,932.50	1,935.20	1,938.40	1,927.20	161.00K	-0.13%
12-Sep-23	1,935.10	1,945.60	1,947.50	1,929.90	161.99K	-0.62%
11-Sep-23	1,947.20	1,943.30	1,954.60	1,939.50	131.08K	0.23%
08-Sep-23	1,942.70	1,943.80	1,954.00	1,940.80	138.69K	0.01%
07-Sep-23	1,942.50	1,942.20	1,947.90	1,940.30	117.17K	-0.09%
06-Sep-23	1,944.20	1,951.50	1,954.50	1,940.00	148.78K	-0.43%
05-Sep-23	1,952.60	1,966.70	1,972.60	1,950.60	192.79K	-0.54%
04-Sep-23	1,963.25	1,966.60	1,972.55	1,962.55	0	-0.17%
03-Sep-23	1,966.65	1,966.40	1,967.05	1,965.50	0	-0.02%
01-Sep-23	1,967.10	1,966.40	1,980.20	1,960.70	160.73K	0.06%



Column chart

Bars can be drawn horizontally or vertically (column) to represent categorical variables.

Date	Price	Open	High	Low	Volume	Chg%
25-Sep-23	1,944.75	1,944.40	1,946.60	1,940.25	0	0.02%
24-Sep-23	1,944.40	1,944.95	1,946.05	1,944.10	0	-0.06%
22-Sep-23	1,945.60	1,939.60	1,949.10	1,939.60	0	0.31%
21-Sep-23	1,939.60	1,952.00	1,952.20	1,933.10	0	-1.40%
20-Sep-23	1,967.10	1,953.00	1,968.90	1,948.60	220.98K	0.69%
19-Sep-23	1,953.70	1,955.30	1,958.90	1,950.90	131.10K	0.02%
18-Sep-23	1,953.40	1,945.70	1,955.70	1,943.80	138.03K	0.37%
15-Sep-23	1,946.20	1,932.50	1,952.40	1,931.20	199.77K	0.69%
14-Sep-23	1,932.80	1,930.70	1,934.50	1,921.70	201.74K	0.02%
13-Sep-23	1,932.50	1,935.20	1,938.40	1,927.20	161.00K	-0.13%
12-Sep-23	1,935.10	1,945.60	1,947.50	1,929.90	161.99K	-0.62%
11-Sep-23	1,947.20	1,943.30	1,954.60	1,939.50	131.08K	0.23%
08-Sep-23	1,942.70	1,943.80	1,954.00	1,940.80	138.69K	0.01%
07-Sep-23	1,942.50	1,942.20	1,947.90	1,940.30	117.17K	-0.09%
06-Sep-23	1,944.20	1,951.50	1,954.50	1,940.00	148.78K	-0.43%
05-Sep-23	1,952.60	1,966.70	1,972.60	1,950.60	192.79K	-0.54%
04-Sep-23	1,963.25	1,966.60	1,972.55	1,962.55	0	-0.17%
03-Sep-23	1,966.65	1,966.40	1,967.05	1,965.50	0	-0.02%
01-Sep-23	1,967.10	1,966.40	1,980.20	1,960.70	160.73K	0.06%



Scatter plot

Scatter plots can be constructed in the following two situations:

1. When one continuous variable is dependent on another variable
2. When both continuous variables are independent – regressors

To shows relationship for two different numeric variables.

The scatter plot explains the correlation between two variables

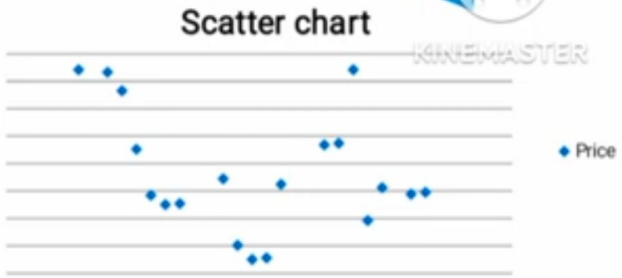
can calculate a **coefficient of correlation** for the given data

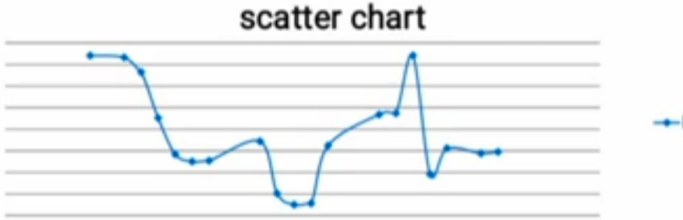
It is a quantitative measure of the association of the random variables

Its value is always less than 1, and it may be positive or negative.

Scatter plot

Date	Price	Open	High	Low	Volume	Chg%
25-Sep-23	1,944.75	1,944.40	1,946.60	1,940.25	0	0.02%
24-Sep-23	1,944.40	1,944.95	1,946.05	1,944.10	0	-0.06%
22-Sep-23	1,945.60	1,939.60	1,949.10	1,939.60	0	0.31%
21-Sep-23	1,939.60	1,952.00	1,952.20	1,933.10	0	-1.46%
20-Sep-23	1,967.10	1,953.00	1,968.90	1,948.60	220.98K	0.69%
19-Sep-23	1,953.70	1,955.30	1,958.90	1,950.90	131.10K	0.02%
18-Sep-23	1,953.40	1,945.70	1,955.70	1,943.80	138.03K	0.37%
15-Sep-23	1,946.20	1,932.50	1,952.40	1,931.20	199.77K	0.69%
14-Sep-23	1,932.80	1,930.70	1,934.50	1,921.70	201.74K	0.02%
13-Sep-23	1,932.50	1,935.20	1,938.40	1,927.20	161.00K	-0.13%
12-Sep-23	1,935.10	1,945.60	1,947.50	1,929.90	161.99K	-0.62%
11-Sep-23	1,947.20	1,943.30	1,954.60	1,939.50	131.08K	0.23%
08-Sep-23	1,942.70	1,943.80	1,954.90	1,940.80	138.69K	0.01%
07-Sep-23	1,942.50	1,942.20	1,947.90	1,940.30	117.17K	-0.09%
06-Sep-23	1,944.20	1,951.50	1,954.50	1,940.00	148.78K	-0.43%
05-Sep-23	1,952.60	1,966.70	1,972.60	1,950.60	192.79K	-0.54%
04-Sep-23	1,963.25	1,966.60	1,972.55	1,962.55	0	-0.17%
03-Sep-23	1,966.65	1,966.40	1,967.05	1,965.50	0	-0.02%
01-Sep-23	1,967.10	1,966.40	1,980.20	1,960.70	160.73K	0.06%



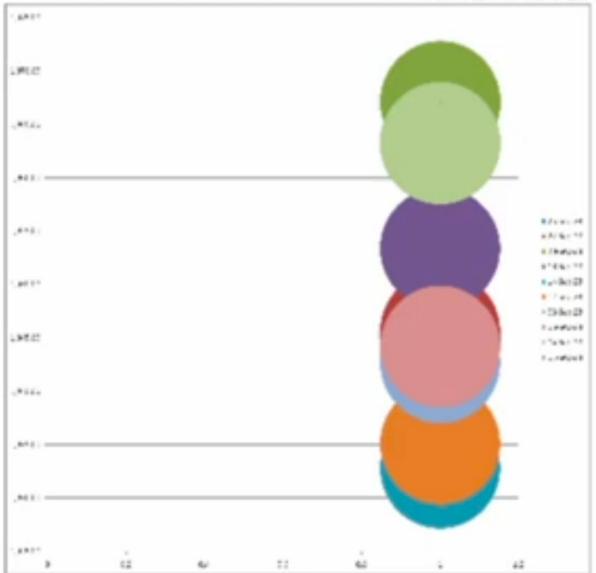


Virtual classroom Lectures

Bubble chart

Bubble chart→scatter plot where each data point on the graph is shown as a bubble.
Each bubble can be displayed with a different color, size, and appearance.

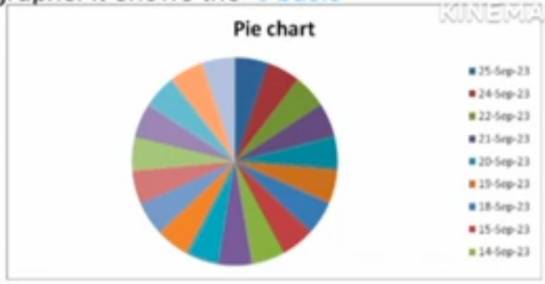
Date	Price	Open	High	Low	Volume	Chg%
25-Sep-23	1,944.75	1,944.40	1,946.60	1,940.25	0	0.02%
24-Sep-23	1,944.40	1,944.95	1,946.05	1,944.10	0	-0.06%
22-Sep-23	1,945.60	1,939.60	1,949.10	1,939.60	0	0.31%
21-Sep-23	1,939.60	1,952.00	1,952.20	1,933.10	0	-1.40%
20-Sep-23	1,967.10	1,953.00	1,968.90	1,948.60	220.98K	0.69%
19-Sep-23	1,953.70	1,955.30	1,958.90	1,950.90	131.10K	0.02%
18-Sep-23	1,953.40	1,945.70	1,955.70	1,943.80	138.03K	0.37%
15-Sep-23	1,946.20	1,932.50	1,952.40	1,931.20	199.77K	0.69%
14-Sep-23	1,932.80	1,930.70	1,934.50	1,921.70	201.74K	0.02%
13-Sep-23	1,932.50	1,935.20	1,938.40	1,927.20	161.00K	-0.13%
12-Sep-23	1,935.10	1,945.60	1,947.50	1,929.90	161.99K	-0.62%
11-Sep-23	1,947.20	1,943.30	1,954.60	1,939.50	131.08K	0.23%
08-Sep-23	1,942.70	1,943.80	1,954.00	1,940.80	138.69K	0.01%
07-Sep-23	1,942.50	1,942.20	1,947.90	1,940.30	117.17K	-0.09%
06-Sep-23	1,944.20	1,951.50	1,954.50	1,940.00	148.78K	-0.43%
05-Sep-23	1,952.60	1,966.70	1,972.60	1,950.60	192.79K	-0.54%
04-Sep-23	1,963.25	1,966.60	1,972.55	1,962.55	0	-0.17%
03-Sep-23	1,966.65	1,966.40	1,967.05	1,965.50	0	-0.02%
01-Sep-23	1,967.10	1,966.40	1,980.20	1,960.70	160.73K	0.06%

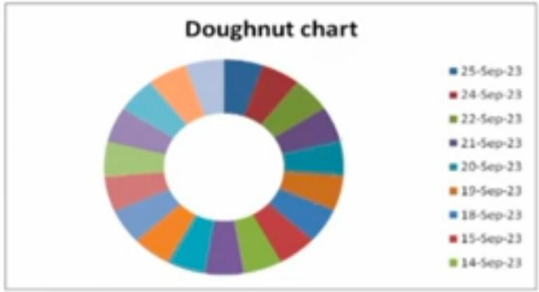


Pie chart

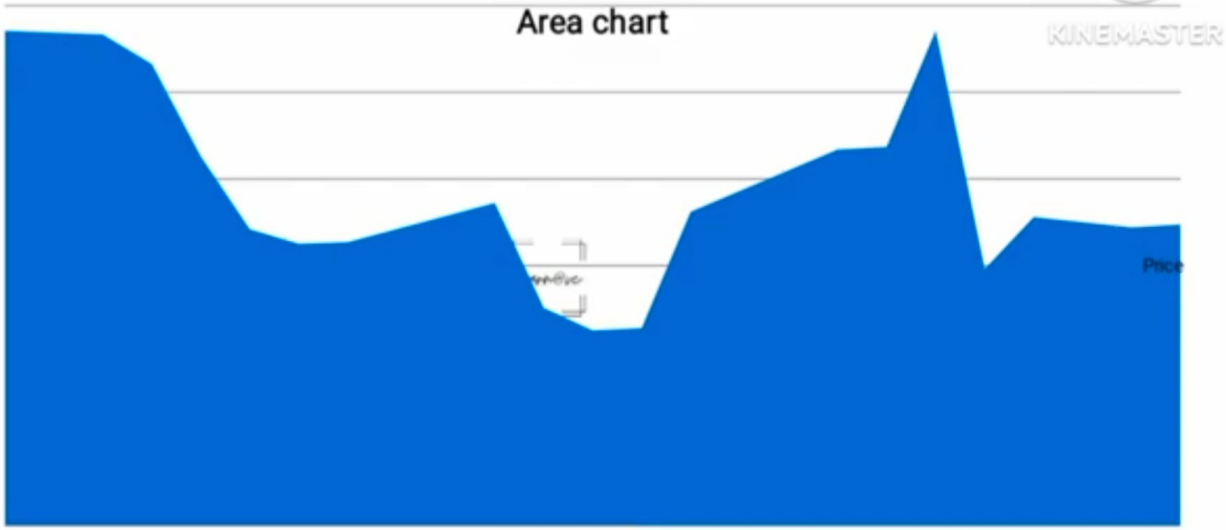
Pie chart- interesting types of data visualization graphs. It shows the % basis

Date	Price	Open	High	Low	Volume	Chg%
25-Sep-23	1,944.75	1,944.40	1,946.60	1,940.25	0	0.02%
24-Sep-23	1,944.40	1,944.95	1,946.05	1,944.10	0	-0.06%
22-Sep-23	1,945.60	1,939.60	1,949.10	1,939.60	0	0.31%
21-Sep-23	1,939.60	1,952.00	1,952.20	1,933.10	0	-1.40%
20-Sep-23	1,967.10	1,953.00	1,968.90	1,948.60	220.98K	0.69%
19-Sep-23	1,953.70	1,955.30	1,958.90	1,950.90	131.10K	0.02%
18-Sep-23	1,953.40	1,945.70	1,955.70	1,943.80	138.03K	0.37%
15-Sep-23	1,946.20	1,932.50	1,952.40	1,931.20	199.77K	0.69%
14-Sep-23	1,932.80	1,930.70	1,934.50	1,921.70	201.74K	0.02%
13-Sep-23	1,932.50	1,935.20	1,938.40	1,927.20	161.00K	-0.13%
12-Sep-23	1,935.10	1,945.60	1,947.50	1,929.90	161.99K	-0.62%
11-Sep-23	1,947.20	1,943.30	1,954.60	1,939.50	131.08K	0.23%
08-Sep-23	1,942.70	1,943.80	1,954.00	1,940.80	138.69K	0.01%
07-Sep-23	1,942.50	1,942.20	1,947.90	1,940.30	117.17K	-0.09%
06-Sep-23	1,944.20	1,951.50	1,954.50	1,940.00	148.78K	-0.43%
05-Sep-23	1,952.60	1,966.70	1,972.60	1,950.60	192.79K	-0.54%
04-Sep-23	1,963.25	1,966.60	1,972.55	1,962.55	0	-0.17%
03-Sep-23	1,966.65	1,966.40	1,967.05	1,965.50	0	-0.02%
01-Sep-23	1,967.10	1,966.40	1,980.20	1,960.70	160.73K	0.06%

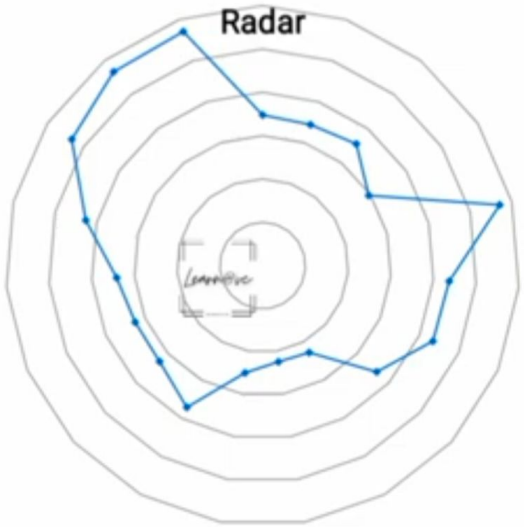




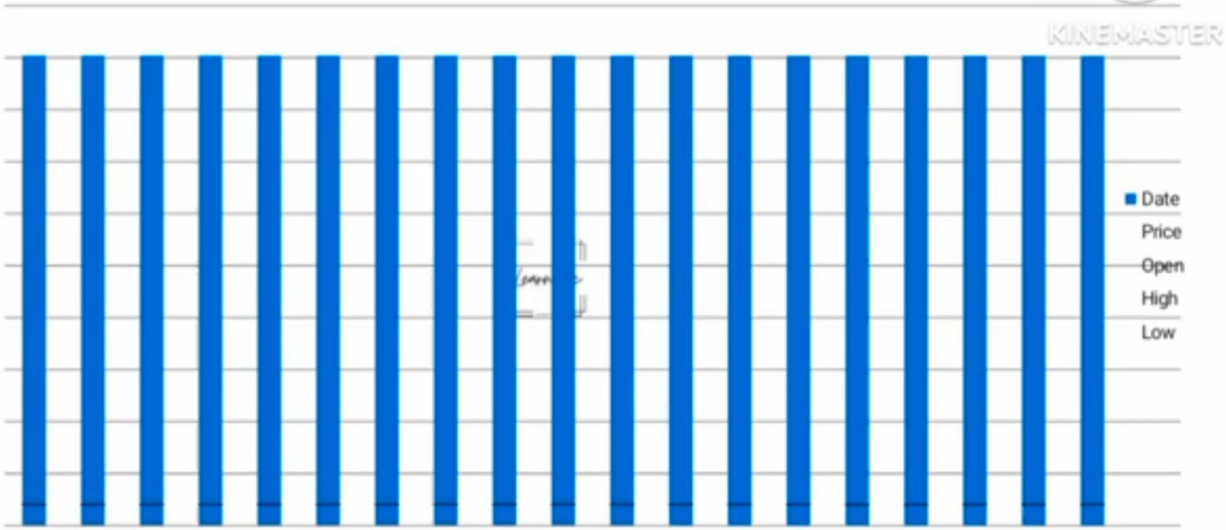
Area chart



Radar chart or spider chart.



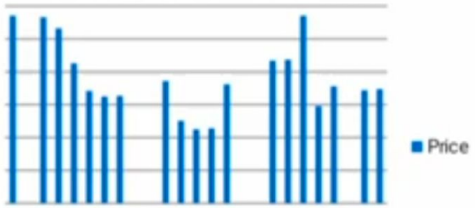
Stock chart



Histogram

Histogram plots are used to depict the distribution of any continuous variable.

To show the frequency distribution for quantitative data. It works with numerical data



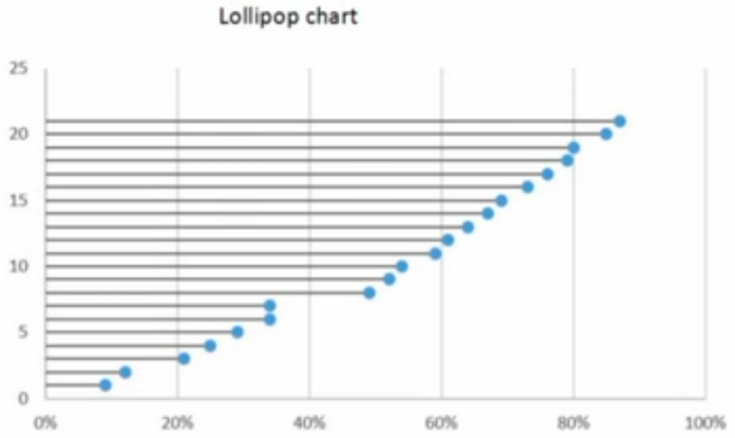
Lollipop chart

A lollipop chart can be used to display ranking in the data

It shows the relationship between a numeric and a categorical variable

Line→ the magnitude

a dot→ the data value



Virtual classroom Lectures

20

Comparison of all charts

S.No	Chart Type	When to choose this chart
1	Column Chart	To compare the multiple values . It shown through vertical bars . It show continuous data over time
2	Line Chart	To show the ups and downs over a period of time, like for months or years.
3	Bar Chart	To compare the values across a few categories.It shown through horizontal bar
4	Area chart	Area chart - same as the line chart. To plot the change over time.This chart is best to use for indicating a change among different sets.
5	Pie or Doughnut chart	To quantify the values and show them as percentage.
6	Scatter Chart	To show similarities between large sets of data . To compare many data points
7	Bubble Chart	It shows to represent the data points in the data series.
8	Stock Chart	To show fluctuations in other data, such as daily rainfall or annual temperatures.
9	Surface Chart	To find the optimum combinations between two sets of data.colors and patterns indicate areas that are in the same range of values.Ensure the categories and the data series are numeric values.
10	Radar Chart	To compare the aggregate values of several data series.
11	Combo Chart	combine two or more chart types to make the data easy to understand, especially when the data is widely varied
12	Histogram	To show the frequency distribution for quantitative data.It works with numerical data

Choosing the best chart

Choosing the best chart base on type of data you have.

continuous variables→ histogram

To compare categories →bar chart (many catogeries),bubble chart(few categories),Pie chart(composition),stacked bar chart(composition,over time),bubble chart (across categories)

To show change over time→ Linechart(Many series over time),Area chart(few series over time), Time line (events in time),Map chart(by location)

To show data organisation(grouping,Ranking)→List,flow chart,Venn diagram,Mind map, Pyramid(hierarchy),Table(many),ordered bar chart (ranking)

To show the relationships(correlations,distributions)→scatter plot,histogram

S.No	Chart Type	When to choose this chart	Examples
1	Column Chart	To compare the multiple values . It shown through vertical bars . It show continuous data over time	Customer survey data,Sales volume,Profit and loss,
2	Line Chart	To show the ups and downs over a period of time, like for months or years.	how many chats or emails your team responds to per month.
3	Bar Chart	To compare the values across a few categories.It shown through horizontal bar	Survey Results,Marketing traffic by month or year.
4	Area chart	Area chart - same as the line chart. To plot the change over time.This chart is best to use for indicating a change among different sets.	Spot and analyze industry trends.Visualize which product categories or products within a category are most popular.
5	Pie or Doughnut chart	To quantify the values and show them as percentage.	Revenue from your most popular products,Percent of total profit from different store locations.
6	Scatter Chart	To show similarities between large sets of data . To compare many data points	Show the frequency of survey responses. Identify outliers in historical data. Visitor numbers and outdoor temperature. Sales growth and tax laws.
7	Bubble Chart	It shows to represent the data points in the data series.	Customer satisfaction scores. Product usage. Customer shopping habits.Top sales by month and location. Customer satisfaction surveys. Store performance tracking. Social media usage by platform.
8	Stock Chart	To show fluctuations in other data, such as daily rainfall or annual temperatures.	weather report,stock report
9	Surface Chart	To find the optimum combinations between two sets of data.colors and patterns indicate areas that are in the same range of values.Ensure the categories and the data series are numeric values.	Organisation past 5 year performance,Manufacturing count of products for 5 years
10	Radar Chart	To compare the aggregate values of several data series.	Competitive Analysis ,Employee Performance
11	Combo Chart	combine two or more chart types to make the data easy to understand, especially when the data is widely varied	profit, expenses, and headcount plotted in the same chart by month.
12	Histogram	To show the frequency distribution for quantitative data.It works with numerical data	Student Mark statement
13	Gantt chart	This chart maps the different tasks completed over a period of time.	Assign tasks to the team and individuals.Set important events, meetings, and announcements.Break projects into tasks.Track the start and end of the tasks.

Choosing the best chart

Purpose	Show correlation	Show deviation	Show distribution	Show composition	Show change	Show ranking
Charts	Scatter plot Correlogram Pairwise plot Jittering with strip plot Counts plot Marginal histogram Scatter plot with a line of best fit Bubble plot with circling	Area chart Diverging bars Diverging texts Diverging dot plot Diverging lollipop plot with markers	Histogram for continuous variable Histogram for categorical variable Categorical plots pyramid plot Distributed dot plot Box plot	Pie chart Treemap Bar char	Time series plot Cross-correlation plot Multiple time series Plotting with different scales Stacked area chart Seasonal plot Calendar heat map Area chart	Ordered bar chart Lollipop chart Dot plot Slope plot

Lets assume the any one company sales Report and show it in graph .Which graph will be suitable for the below questions.

- ✔ Revenue by payment method (2023) for all months- **Line Chart**
- ✔ Customers by country (location based)- **Maps**
- ✔ Last year top 5 products sales - **Horizontal Bar Graphs**
- ✔ *Amount of Sales per Channel and Country (last year)*- **Grouped column chart**
- ✔ Number of sales by product category(last month)- **Pie Charts**
- ✔ Revenue by product category by country(last month)- **scatter plot**
- ✔ Number of sales by payment method per month (2023)- **Area chart**
- ✔ Average sales by category (last month)- bar chart

Which chart will be suitable for this scenario

let's assume a pharmacy in India keeps track of the amount of paracetamol sold every month. paracetamol is a medicine prescribed to patients suffering from fever and pain. To keep track of the months of the year (1 to 12) corresponding to January to December.

1.Sales report by months
2. Customers by state (location based)
3. Average sales
4. Last 5 year tablet sales
5. Show the relations of stock and sales

Conclusion

1. Visual Aids for EDA
2. Types of techniques used in visualization
3. All charts
4. Benefits of data visualization
5. Choosing the best chart
6. Examples

Data transformation techniques

Data transformation →converting, cleansing, and structuring data into a usable format → analysis →decision making processes→ growth of an organization.

" To **convert** the raw data into a suitable format"

changes → format, structure, or values of the data

ETL (extract, Transform for better representation, and load)

removing duplicates, replacing values, renaming axis indexes, discretization(transferring continuous functions into discrete)and binning(group), and detecting and filtering outliers(abnormal).

Data transformation activities

- Data deduplication--> identification of duplicates and their removal.
- Data cleansing --> extracting words and deleting out-of-date, inaccurate, and incomplete information
- Data validation --> formulating rules or algorithms (validating different types of data)
- Format revisioning --> converting from one format to another.
- Data derivation --> creating a set of rules to generate more information

Data transformation activities

- Data aggregation --> searching, extracting, summarizing, and preserving important information
- Data integration--> converting different data types and merging them into a common structure or schema(interoperability)
- Data filtering -->identifying information relevant to any particular user.
- Data joining--> establishing a relationship between two or more tables.

Benefits of data transformation

- better-organized data
- similar format and structure
- data quality
- higher performance and scalability

Challenges

- Increased cost of the operation.
- cleansing process can be time-consuming.
- very slow

Merging DB

A professor teaching a 2 different Elective Paper for the same class .one is Data Science course and an Machine Learning course, and there are students to split into two classes. They gave two different mark statement as dataframes.we would need to concatenate them.

pandas concat() method:

dataframe = pd.concat([dataFrame1, dataFrame2], ignore_index=True)

StudentID	ScoreSE
1	89
3	39
5	50
7	97
9	20
…	…
…	…
27	73
29	92

StudentID	ScoreSE
2	98
4	93
6	44
8	77
10	69
…	…
…	…
28	56
30	27

The output is

StudentID	ScoreSE
1	89
3	39
5	50
7	97
9	20
…	…
…	…
27	73
29	92
2	98
4	93
6	44
8	77
10	69
…	…
…	…
28	56
30	27

pd.concat([dataFrame1, dataFrame2], axis=1)

The output is

	StudentID	Score	StudentID	Score
0	1	89	2	98
1	3	39	4	93
2	5	50	6	44
3	7	97	8	77
4	9	22	10	69
5	11	66	12	56
6	13	31	14	31
7	15	51	16	53
8	17	71	18	78
9	19	91	20	93
10	21	56	22	56
11	23	32	24	77
12	25	52	26	33
13	27	73	28	56
14	29	92	30	27

When we specify axis=1, the concatenation happens on a side-by-side basis.

Reshaping and pivoting

we need to rearrange data in a dataframe

hierarchical indexing

Stacking: Stack rotates from any particular column in the data to the rows.

Unstacking: Unstack rotates from the rows into the column.stacked.unstack()

	Bergen	Oslo	Trondheim	Stavanger	Kristiansand
Rainfall	0	1	2	3	4
Humidity	5	6	7	8	9
Wind	10	11	12	13	14

stacked = dframe1.stack()

```
Rainfall  Bergen         0
          Oslo           1
          Trondheim      2
          Stavanger      3
          Kristiansand   4
Humidity  Bergen         5
          Oslo           6
          Trondheim      7
          Stavanger      8
          Kristiansand   9
Wind      Bergen        10
          Oslo          11
          Trondheim     12
          Stavanger     13
          Kristiansand  14
```

4	
9	
14	

Rainfall	Bergen	0
	Oslo	1
	Trondheim	2
	Stavanger	3
	Kristiansand	4
Humidity	Bergen	5
	Oslo	6
	Trondheim	7
	Stavanger	8
	Kristiansand	9
Wind	Bergen	10
	Oslo	11
	Trondheim	12
	Stavanger	13
	Kristiansand	14

Data transformation activities

- Data deduplication--> identification of duplicates and their removal.
- Data cleansing --> extracting words and deleting out-of-date, inaccurate, and incomplete information
- Data validation --> formulating rules or algorithms (validating different types of data)
- Format revisioning --> converting from one format to another.
- Data derivation --> creating a set of rules to generate more information
- Data aggregation --> searching, extracting, summarizing, and preserving important information
- Data integration--> converting different data types and merging them into a common structure or schema(interoperability)
- Data filtering -->identifying information relevant to any particular user.
- Data joining--> establishing a relationship between two or more tables.

Performing data deduplication

Dataframe contains duplicate rows. Removing them is essential to enhance the quality of the dataset

Input

--> frame3.duplicated()

	column 1	column 2
0	Looping	10
1	Looping	10
2	Looping	22
3	Functions	23
4	Functions	23
5	Functions	24
6	Functions	24

```
0    False
1     True
2    False
3    False
4     True
5    False
6     True
dtype: bool
```

The rows that say True are duplicated data.

-->frame4 = frame3.drop_duplicates()

	column 1	column 2
0	Looping	10
2	Looping	22
3	Functions	23
5	Functions	24

Replacing values

To find and replace some values inside a dataframe

```
import numpy as np
replaceFrame = pd.DataFrame({'column 1': [200., 3000., -786.,
3000., 234., 444., -786., 332., 3332. ], 'column 2': range(9)})
replaceFrame.replace(to_replace =-786, value= np.nan)
```

	column 1	column 2
0	200.0	0
1	3000.0	1
2	NaN	2
3	3000.0	3
4	234.0	4
5	444.0	5
6	NaN	6
7	332.0	7
8	3332.0	8

```
replaceFrame = pd.DataFrame({'column 1': [200., 3000., -786.,
3000., 234., 444., -786., 332., 3332. ], 'column 2': range(9)})
replaceFrame.replace(to_replace =[-786, 0], value= [np.nan, 2])
```

There are two replacements. All -786 values will be replaced by NaN and all 0 values will be replaced by 2.

Handling missing data

Whenever there are missing values, a NaN value is used, which indicates that there is no value specified for that particular index.

Reasons:

- incomplete values in the dataset.
- some values are not matched when we join.
- Data collection errors
- Reindexing of data can result in incomplete data.
- dataframe is

showing sales of different fruits from different stores-No missing values

	store1	store2	store3
apple	15	16	17
banana	18	19	20
kiwi	21	22	23
grapes	24	25	26
mango	27	28	29

Add some missing values

Let's add some missing values to our dataframe:

```
dfx['store4'] = np.nan
dfx.loc['watermelon'] = np.arange(15, 19)
dfx.loc['oranges'] = np.nan
dfx['store5'] = np.nan
dfx['store4']['apple'] = 20.
dfx
```

	store1	store2	store3	store4	store5
apple	15.0	16.0	17.0	20.0	NaN
banana	18.0	19.0	20.0	NaN	NaN
kiwi	21.0	22.0	23.0	NaN	NaN
grapes	24.0	25.0	26.0	NaN	NaN
mango	27.0	28.0	29.0	NaN	NaN
watermelon	15.0	16.0	17.0	18.0	NaN
oranges	NaN	NaN	NaN	NaN	NaN

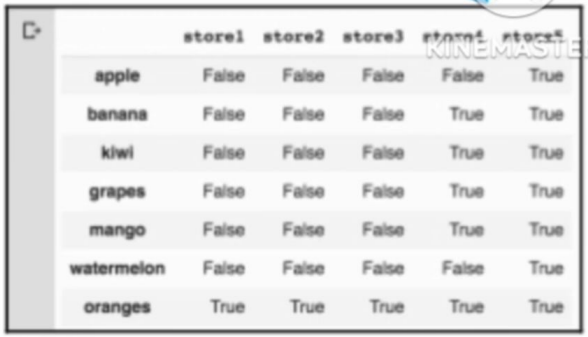
characteristics of missing values

- An entire row can contain NaN values.
- An entire column can contain NaN values.
- Some values in both a row and a column can be NaN.

Based on these characteristics, let's examine NaN values

dfx.isnull()

True values indicate
the values that are NaN.



	store1	store2	store3	store4	store5
apple	False	False	False	False	True
banana	False	False	False	True	True
kiwi	False	False	False	True	True
grapes	False	False	False	True	True
mango	False	False	False	True	True
watermelon	False	False	False	False	True
oranges	True	True	True	True	True

To count the number of NaN values in each store

dfx.isnull().sum()

store1 1

store2 1

store3 1

store4 5

store5 7

To find the total number of missing values

dfx.isnull().sum().sum() 15

Dropping missing values

To handle missing values is to simply remove them from our dataset.

To determine null values:

1.isnull()

2.notnull()

dfx.store4[dfx.store4.notnull()]

apple 20.0

watermelon 18.0

Name: store4, dtype: float64

	store1	store2	store3	store4	store5
apple	15.0	16.0	17.0	20.0	NaN
banana	18.0	19.0	20.0	NaN	NaN
kiwi	21.0	22.0	23.0	NaN	NaN
grapes	24.0	25.0	26.0	NaN	NaN
mango	27.0	28.0	29.0	NaN	NaN
watermelon	15.0	16.0	17.0	18.0	NaN
oranges	NaN	NaN	NaN	NaN	NaN

use the dropna() method to remove the rows:

dfx.store4.dropna()

dropna() method just returns a copy of the dataframe by dropping the rows with NaN.dropna() is applied to the entire dataframe, then it will drop all the rows from the dataframe.

dfx.dropna() –> The output of the preceding code is an empty dataframe.

Dropping by rows &Dropping by columns

Dropping by rows

use the how=all argument to drop only those rows ent

```
dfx.dropna(how='all')
```

orange rows are removed because those entire rows contained NaN

Dropping by columns

axis=1 to indicate a check for NaN by columns.

```
dfx.dropna(how='all', axis=1)
```

store5 is dropped from the dataframe.

	store1	store2	store3	store4	store5
apple	15.0	16.0	17.0	20.0	NaN
banana	18.0	19.0	20.0	NaN	NaN
kiwi	21.0	22.0	23.0	NaN	NaN
grapes	24.0	25.0	26.0	NaN	NaN
mango	27.0	28.0	29.0	NaN	NaN
watermelon	15.0	16.0	17.0	18.0	NaN

	store1	store2	store3	store4
apple	15.0	16.0	17.0	20.0
banana	18.0	19.0	20.0	NaN
kiwi	21.0	22.0	23.0	NaN
grapes	24.0	25.0	26.0	NaN
mango	27.0	28.0	29.0	NaN
watermelon	15.0	16.0	17.0	18.0
oranges	NaN	NaN	NaN	NaN

Virtual class

dfx.dropna(thresh=5, axis=1)

store4 column is now dropped because it has more than five NaN values.

	store1	store2	store3
apple	15.0	16.0	17.0
banana	18.0	19.0	20.0
kiwi	21.0	22.0	23.0
grapes	24.0	25.0	26.0
mango	27.0	28.0	29.0
watermelon	15.0	16.0	17.0
oranges	NaN	NaN	NaN

Filling missing values

fillna() method to replace NaN values with any particular values.

```
filledDf = dfx.fillna(0)
```

	store1	store2	store3	store4	store5
apple	15.0	16.0	17.0	20.0	0.0
banana	18.0	19.0	20.0	0.0	0.0
kiwi	21.0	22.0	23.0	0.0	0.0
grapes	24.0	25.0	26.0	0.0	0.0
mango	27.0	28.0	29.0	0.0	0.0
watermelon	15.0	16.0	17.0	18.0	0.0
oranges	0.0	0.0	0.0	0.0	0.0

all the NaN values are replaced by 0.It affect several statistics including mean, sum, and median.

compute the mean

```
dfx.mean()

 store1 20.0
store2 21.0
store3 22.0
store4 19.0
store5 NaN
dtype: float64
```

compute the mean from the filled dataframe

```
  filledDf.mean()
store1 17.142857
store2 18.000000
store3 18.857143
store4 5.428571
store5 0.000000
dtype: float64
```

filling with 0 might not be the optimal solution.

Backward and forward filling

NaN values can be filled based on the last known values.

dfx.store4.fillna(method='ffill')

apple 20.0
banana 20.0
kiwi 20.0
grapes 20.0
mango 20.0
watermelon 18.0
oranges 18.0
Name: store4, dtype: float64

dfx.store4.fillna(method='bfill')

apple 20.0
banana 18.0
kiwi 18.0
grapes 18.0
mango 18.0
watermelon 18.0
oranges NaN
Name: store4, dtype: float64

Interpolating missing values

It performs a linear interpolation of missing values

```
ser3 = pd.Series([100, np.nan, np.nan, np.nan, 292])
ser3.interpolate()
```

```
0 100.0
1 148.0
2 196.0
3 244.0
4 292.0
dtype: float64
```

first and the last values are 100 and 292 respectively

Renaming axis indexes

to transform the index terms to capital letters:

```
dframe1.index = dframe1.index.map(str.upper)
```

	Bergen	Oslo	Trondheim	Stavanger	Kristiansand
RAINFALL	0	1	2	3	4
HUMIDITY	5	6	7	8	9
WIND	10	11	12	13	14

```
dframe1.rename(index=str.title, columns=str.upper)
```

	BERGEN	OSLO	TRONDHEIM	STAVANGER	KRISTIANSAND
Rainfall	0	1	2	3	4
Humidity	5	6	7	8	9
Wind	10	11	12	13	14

Discretization and binning

when working with continuous datasets, we need to convert them into discrete or interval forms

Each interval is referred to as a bin

Eg:heights of a group of students

height = [120, 122, 125, 127, 121, 123, 137, 131, 161, 145, 141, 132]

To convert that dataset into intervals of 118 to 125, 126 to 135, 136 to 160, and finally 160 and higheruse the cut().

bins = [118, 125, 135, 160, 200]

category = pd.cut(height, bins)

category [(118, 125], (118, 125], (118, 125], (125, 135], (118, 125], ..., (125, 135], (160, 200], (135, 160], (135, 160], (125, 135]] Length: 12 Categories (4, interval[int64]): [(118, 125] < (125, 135] < (135, 160] < (160, 200]]

Outlier detection and filtering

Outliers are data points that differs substantially from the rest of the data
-->presence of such outliers can cause serious issues in statistical analysis.

Eg:The regular transaction is below 10,000 and suddenly he makes a transaction of 1,00,000 this is an outlier.

Outlier Detection

```
import numpy as np
 data =[1, 2, 3, 4,5,6,7, 8, 9,10,1000]
 mean = np.mean(data)
std = np.std(data)
 threshold = 3
outliers = []
for x in data:
    z_score = (x - mean) / std
    if abs(z_score) > threshold:
        outliers.append(x)
print("Mean: ",mean)
print("\nStandard deviation: ",std)
print("\nOutliers  : ", outliers)
```

Output:

```
Mean: 95.9090909090909
        Standard deviation:285.9117646933
        Outliers  : [1000]
```

Grouping Datasets

categorizing a dataset into multiple categories or groups is often essential.

The Pandas **groupby()** function lets you split data into groups based on some criteria

pandas- groupby() performs two functions:

1.It splits the data into groups based on some criteria.

2.It applies a function to each group independently

dataset--> multiple numerical as well as categorical records

```
data = {'Gender':['m','f','f','m','f','m','m'],'Height':[172,171,169,173,170,175,178]}
df_sample = pd.DataFrame(data)
df_sample
```

	Gender	Height
0	m	172
1	f	171
2	f	169
3	m	173
4	f	170
5	m	175
6	m	178

Grouping Datasets

```
f_filter = df_sample['Gender']=='f'
print(df_sample[f_filter])
m_filter = df_sample['Gender']=='m'
print(df_sample[m_filter])
```

	Gender	Height
0	m	172
1	m	173
2	m	175
3	m	178

	Gender	Height
0	f	171
1	f	169
2	f	170

To perform average

```
f_avg = df_sample[f_filter]['Height'].mean()
m_avg = df_sample[m_filter]['Height'].mean()
print(f_avg,m_avg)
```

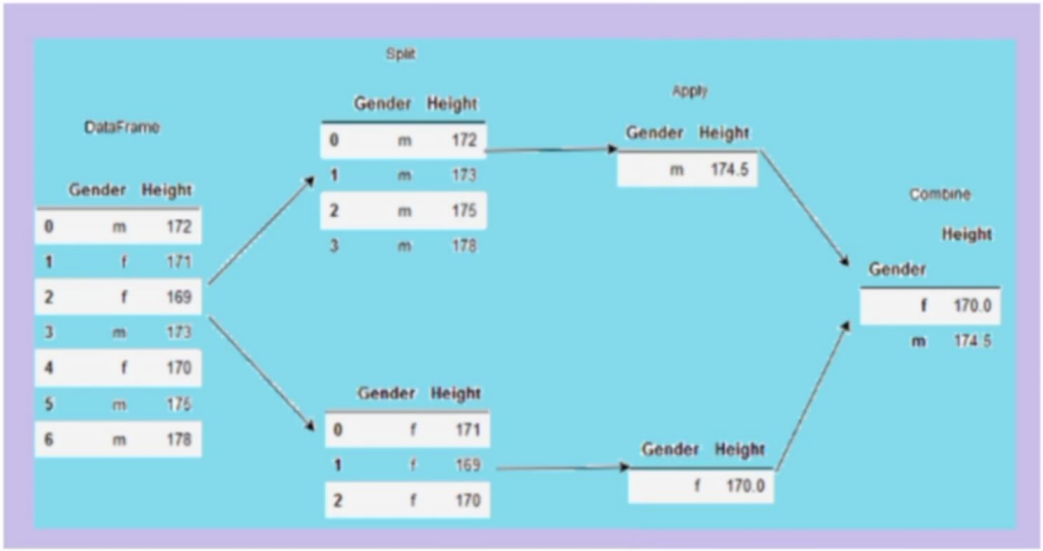
170.0 174.5

```
df_output = pd.DataFrame({'Gender':['f','m'],'Height':[f_avg,m_avg]})
df_output
```

	Gender	Height
0	f	170.0
1	m	174.5

```
df_sample.groupby('Gender').mean()
```

Gender	Height
f	170.0
m	174.5


Split
Apply
Combine
DataFrame
Gender Height
0 m 172
1 f 171
2 f 169
3 m 173
4 f 170
5 m 175
6 m 178
Gender Height
0 m 172
1 m 173
2 m 175
3 m 178
Gender Height
0 f 171
1 f 169
2 f 170
Gender Height
m 174.5
Gender Height
f 170.0
Gender Height
f 170.0
m 174.5

aggregate functions in the Pandas

- count() – Number of non-null observations
- sum() – Sum of values
- mean() – Mean of values
- median() – Arithmetic median of values
- min() – Minimum
- max() – Maximum
- mode() – Mode
- std() – Standard deviation
- var() – Variance

df.aggregate() or df.agg()

df.groupby("Product_Category")[["Quantity"]].aggregate([min, max,sum,'mean'])

	Quantity			
	min	max	sum	mean
Product_Category				
Entertainment	1	100	98107	49.851118
Fashion	1	100	101024	51.255200
Healthcare	1	100	99418	50.905274
Home	1	100	104453	50.705340
Office	1	100	102387	50.913476

Quiz

1.Write a python code to print the follwing data

	pyhton	Data science	MachineLearning	AI
0	91	46	81	96
1	85	100	73	65
2	79	68	88	85

2. Write a python code to calculate the sum, min, and max of each column in the above dataset by using agg() function

Answer 1

```
# import module
import pandas as pd

# Creating our dataset
df = pd.DataFrame([[91, 46, 81, 96],
                   [85, 100, 73, 65],
                   [79, 68, 88, 85]],
                   columns=['pyhton', 'Data science', 'MachineLearning', 'AI'])

# display dataset
print(df)
```

Answer 2

```
import pandas as pd
df = pd.DataFrame([[91, 46, 81, 96],
                   [85, 100, 73, 65],
                   [79, 68, 88, 85]],
                   columns=['pyhton', 'Data science',
                            'MachineLearning', 'AI'])
print(df)
df.agg(['sum', 'min', 'max'])
```

	pyhton	Data science	MachineLearning	AI
sum	255	214	242	246
min	79	46	73	65
max	91	100	88	96

Pivot tables and cross-tabulations

Pivoting and cross-tabulation (cross-table) are two data manipulation techniques commonly used in data analysis to summarize and transform data.

Pivoting is the process of reshaping or transforming a dataset from a "long" format to a "wide" format.

Cross-tabulation is a way of summarizing and displaying the frequency or count of occurrences of one or more categorical variables in a tabular format.

In both cases, these techniques help in summarizing and analyzing data, making it easier to draw insights and conclusions.

Pivoting

Pivoting:

1.Pivoting is the process of reshaping or transforming a dataset from a "long" format to a "wide" format. It's typically used to reorganize data to make it more readable and to perform analysis on it more easily.

2.In a long-format dataset, data is often stored in a tabular format with multiple rows and columns. Pivoting converts this data into a format where one or more columns are used as row indexes, one or more columns are used as column headers, and another column contains the values you want to aggregate.

3.Pivoting is frequently used to create summary tables and perform operations like aggregations (e.g., calculating means, sums) based on specific columns

Pivoting Example:

Suppose you have a dataset with columns for "Date," "Product," and "Sales." Pivoting might involve transforming this data into a table where dates are along the rows, product names are along the columns, and the cells contain the total sales for each product on each date.

Cross-Tabulation

Cross-Tabulation (Cross-Table):

1.Cross-tabulation is a way of summarizing and displaying the count of occurrences of one or more categorical variables in a tabular format.

2To explore relationships between two or more categorical variables in your dataset. Each cell in the table shows the frequency with which a combination of values occurs.

3.To find dependencies between variables, often in the context of hypothesis testing and exploratory data analysis.

It's used in statistics, social sciences, and market research to analyze survey data, for instance.

Cross-Tabulation Example:

You have survey data with columns for "Gender" and "Smoking Status." A cross-tabulation could show how many males and females are smokers or non-smokers, providing insights into the relationship between these two categorical variables.

Sample code

```
import pandas as pd
# Sample data
data = {
   'StudentID': [1, 2, 3, 1, 2, 3, 1, 2, 3, 4, 4, 4],
   'Subject': ['Math', 'Math', 'Math', 'python', 'python', 'python', 'AI', 'AI', 'AI', 'Math', 'python', 'AI'],
   'Score': [95, 88, 72, 78, 90, 85, 60, 75, 82, 92, 87, 70]
}
# Create a DataFrame
df = pd.DataFrame(data)
print(df)
```

	StudentID	Subject	Score
0	1	Math	95
1	2	Math	88
2	3	Math	72
3	1	python	78
4	2	python	90
5	3	python	85
6	1	AI	60
7	2	AI	75
8	3	AI	82
9	4	Math	92
10	4	python	87
11	4	AI	70

```
# Pivot Table: Calculate the average score for each student and subject
pivot_table = pd.pivot_table(df, values='Score', index='StudentID', columns='Subject', aggfunc='mean')
print("Pivot Table:")
print(pivot_table)
```

```
Pivot Table:
Subject      AI  Math  python
StudentID
1            60    95      78
2            75    88      90
3            82    72      85
4            70    92      87
```

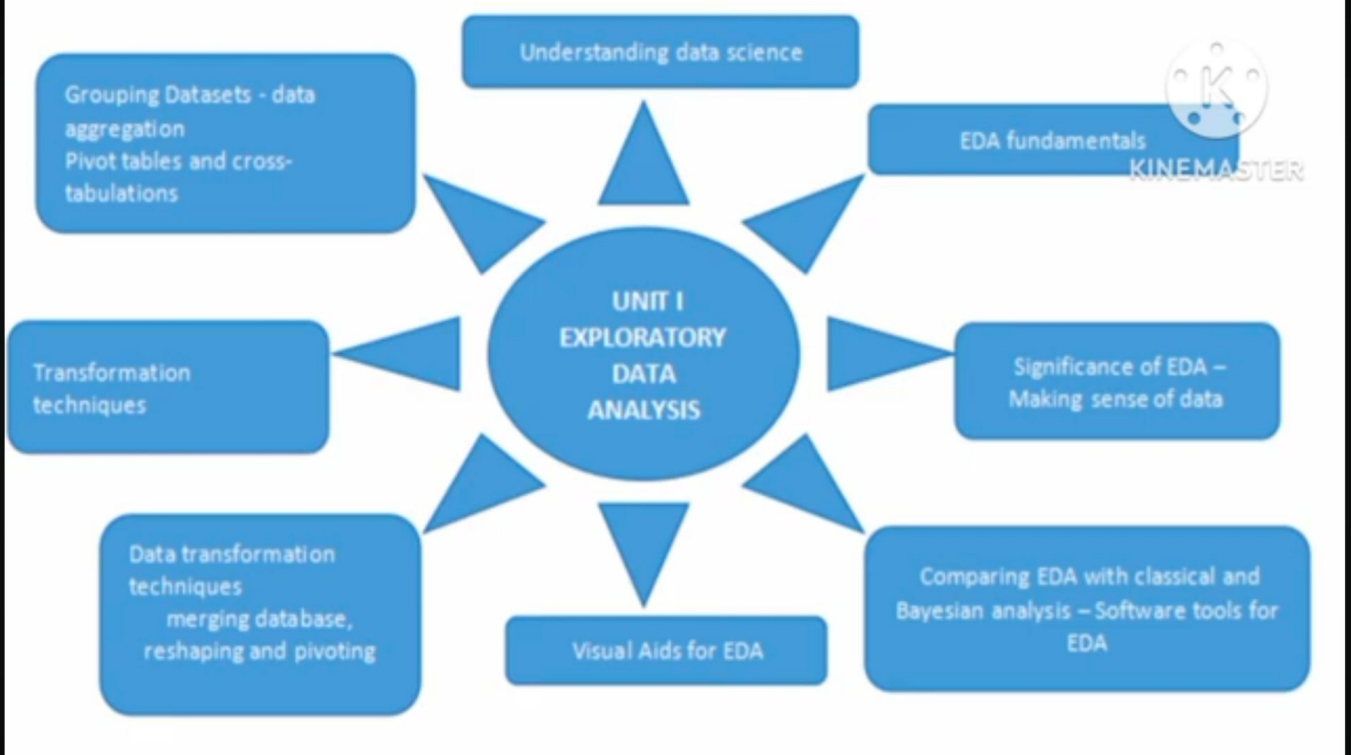
Cross table

```
# Cross-Tabulation: Count the number of students in each grade range for each subject
cross_table = pd.crosstab(df['Subject'], pd.cut(df['Score'], [0, 60, 70, 80, 90, 100], labels=['F', 'D', 'C', 'B', 'A']))
print("\nCross-Tabulation:")
print(cross_table)
```

```
Cross-Tabulation:
Score     F  D  C  B  A
Subject
AI        1  1  1  1  0
Math      0  0  1  1  2
python    0  0  1  3  0
```

Pivoting Vs Cross-Tabulation

Aspect	Pivoting	Cross-Tabulation
Purpose	Reshape and transform data from long to wide format.	Summarize and display the frequency (count) of occurrences of categorical variables.
Data Transformation	Involves selecting one or more columns as row indices, another column for column headers, and another column for values that need to be aggregated.	Groups and counts occurrences of categories for two or more categorical variables, creating a table showing frequency of category combinations.
Aggregation	Typically aggregates data, e.g., calculating means, sums, or other statistics.	Does not aggregate data; it counts and displays frequency.
Typical Use Cases	Often used for reshaping time series data or summarizing data by categories.	Used for examining relationships between categorical variables in hypothesis testing, social sciences, market research, and other contexts.


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

Understanding data science

Extracting usable data from data sources --> Using machine learning tools select features --> prepossessing --> Enhancing datacollection -->(Processing, cleansing, and validating the integrity of data to be used for) data analysis

--> To find patterns --> Developing prediction systems -
->Presenting results

-->Propose solutions and strategies to tackle business challenges

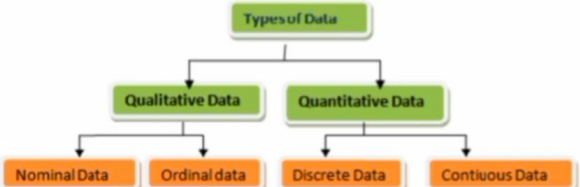
EDA fundamentals

- EDA ensures that the correct ingredients in patterns and trends are made available for training the model to achieve the correct outcome

obtain vital insights , Identifying and removing data outliers , Identifying trends, Uncover patterns related to the target, Creating hypotheses and testing them through experiments

Making sense of data

- Qualitative data
  - 1.Nominal data — Ex: Nationality - Indian, German, American
  - 2.Ordinal data — Ex: How do u rate your satisfaction *****
- Quantitative data
  - 1.Discrete data — Ex: how many students are in "A " grade
  - 2.Continuous data — Ex:Market share price



Comparing Data Analysis

Comparing Data Analysis		
Classical Data Analysis	**Explotary Data Analysis**	**Bayesian Data Analysis**
quantitative data	Data set	probability distribution
Numeric output	Graphical output	only 2 Possible Output
ANOVA,T-tests ,Chi-squared tests	Scatter plots ,Histograms ,Box Plots	Bayesian Distribution

Eg: sensex data
Classical
Male and female ratio, which state has maximum population

Eg: sensex data
EDA
Age-group wise migration of people,
Number of people in medical field,
Population in year wise

Eg: sensex data
Bayesian
Corona affected status→ positive or negative

Software tools for EDA

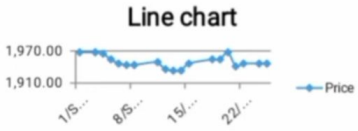
- Python- Data analysis, Data mining, and Data science
- R programming language- statistical computation and graphical data analysis
- Weka - Data mining package
- KNIME- Data analysis
- MATLAB- Mathematical calculation

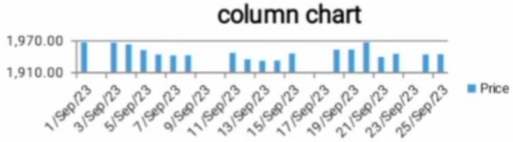
Visual Aids for EDA

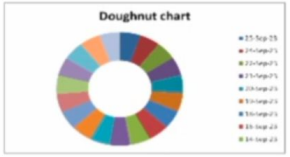
S.No	Chart Type	When to choose this chart
1	Column Chart	To compare the multiple values . It shown through vertical bars . It show continuous data over time
2	Line Chart	To show the ups and downs over a period of time, like for months or years.
3	Bar Chart	To compare the values across a few categories.It shown through horizontal bar
4	Area chart	Area chart - same as the line chart. To plot the change over time.This chart is best to use for indicating a change among different sets.
5	Pie or Doughnut chart	To quantify the values and show them as percentage.
6	Scatter Chart	To show similarities between large sets of data . To compare many data points
7	Bubble Chart	It shows to represent the data points in the data series.
8	Stock Chart	To show fluctuations in other data, such as daily rainfall or annual temperatures.
9	Surface Chart	To find the optimum combinations between two sets of data.colors and patterns indicate areas that are in the same range of values.Ensure the categories and the data series are numeric values.
10	Radar Chart	To compare the aggregate values of several data series.
11	Combo Chart	combine two or more chart types to make the data easy to understand, especially when the data is widely varied
12	Histogram	To show the frequency distribution for quantitative data.It works with numerical data

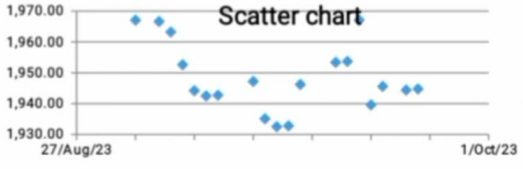
S.No	Chart Type	When to choose this chart	Examples
1	Column Chart	To compare the multiple values . It shown through vertical bars . It show continuous data over time	Customer survey data,Sales volume,Profit and loss,
2	Line Chart	To show the ups and downs over a period of time, like for months or years.	how many chats or emails your team responds to per month.
3	Bar Chart	To compare the values across a few categories.It shown through horizontal bar	Survey Results,Marketing traffic by month or year.
4	Area chart	Area chart - same as the line chart. To plot the change over time.This chart is best to use for indicating a change among different sets.	Spot and analyze industry trends.Visualize which product categories or products within a category are most popular.
5	Pie or Doughnut chart	To quantify the values and show them as percentage.	Revenue from your most popular products,Percent of total profit from different store locations.
6	Scatter Chart	To show similarities between large sets of data . To compare many data points	Show the frequency of survey responses. Identify outliers in historical data. Visitor numbers and outdoor temperature. Sales growth and tax laws.
7	Bubble Chart	It shows to represent the data points in the data series.	Customer satisfaction scores. Product usage. Customer shopping habits.Top sales by month and location. Customer satisfaction surveys. Store performance tracking. Social media usage by platform.
8	Stock Chart	To show fluctuations in other data, such as daily rainfall or annual temperatures.	weather report,stock report
9	Surface Chart	To find the optimum combinations between two sets of data.colors and patterns indicate areas that are in the same range of values.Ensure the categories and the data series are numeric values.	Organisation past 5 year performance,Manufacturing count of products for 5 years
10	Radar Chart	To compare the aggregate values of several data series.	Competitive Analysis ,Employee Performance
11	Combo Chart	combine two or more chart types to make the data easy to understand, especially when the data is widely varied	profit, expenses, and headcount plotted on the same chart by month.
12	Histogram	To show the frequency distribution for quantitative data.It works with numerical data	Student Mark statement
13	Gantt chart	This chart maps the different tasks completed over a period of time.	Assign tasks to the team and individuals.Set important events, meetings, and announcements.Break projects into tasks.Track the start and end of the tasks.

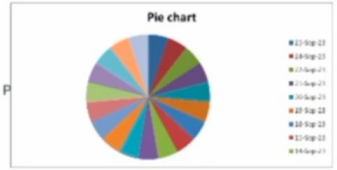
Purpose	Show correlation	Show deviation	Show distribution	Show composition	Show change	Show ranking
Charts	Scatter plot Correlogram Pairwise plot Jittering with strip plot Counts plot Marginal histogram Scatter plot with a line of best fit Bubble plot with circling	Area chart Diverging bars Diverging texts Diverging dot plot Diverging lollipop plot with markers	Histogram for continuous variable Histogram for categorical variable Categorical plots pyramid plot Distributed dot plot Box plot	Pie chart Treemap Bar char	Time series plot Cross-correlation plot Multiple time series Plotting with different scales Stacked area chart Seasonal plot Calendar heat map Area chart	Ordered bar chart Lollipop chart Dot plot Slope plot

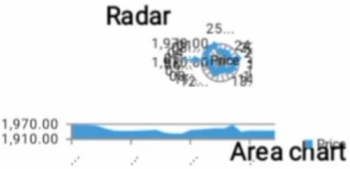












Merging DB

pandas concat() method:

dataframe = pd.concat([dataFrame1, dataFrame2], ignore_index=True)

pd.concat([dataFrame1, dataFrame2], axis=1)

Stacking Reshaping

stacked = dframe1.stack()

data deduplication & Replacing values


```
frame3.duplicated()
frame4 = frame3.drop_duplicates()
```

Replacing values

```
replaceFrame.replace(to_replace =-786, value= np.nan)
replaceFrame.replace(to_replace =[-786, 0], value= [np.nan, 2])
```

null() & notnull()

```
dfx.isnull().sum()
dfx.store4[dfx.store4.notnull()]
```

dropna() & fillna()

```
dfx.store4.dropna()   dfx.dropna(how='all', axis=1)
  filledDf = dfx.fillna(0)
```

filling methods

dfx.mean()

ffill()& bfill()

```
dfx.store4.fillna(method='ffill')
dfx.store4.fillna(method='bfill')
```

Rename

```
dframe1.rename(index=str.title, columns=str.upper)
```

bins

```
bins = [118, 125, 135, 160, 200]
category = pd.cut(height, bins)
```

Grouping & data aggregation


```
df_sample.groupby('Gender').mean()
```

```
df.groupby("Product_Category")[["Quantity"]].aggregate([min, max, sum,'mean'])
```

Pivot tables and cross-tabulations


```
pivot_table = pd.pivot_table(df, values='Score', index='StudentID', columns='Subject', aggfunc='mean')
```

```
cross_table = pd.crosstab(df['Subject'], pd.cut(df['Score'], [0, 60, 70, 80, 90, 100], labels=['F', 'D', 'C', 'B', 'A']))
```

Aspect	Pivoting	Cross-Tabulation
Purpose	Reshape and transform data from long to wide format.	Summarize and display the frequency/count of occurrences of categorical variables.
Data Transformation	Involves selecting one or more columns as row indices, another column for column headers, and another column for values that need to be aggregated.	Groups and counts occurrences of categories for two or more categorical variables, creating a table showing frequency of category combinations.
Aggregation	Typically aggregates data, e.g., calculating means, sums, or other statistics.	Does not aggregate data; it counts and displays frequency.
Typical Use Cases	Often used for reshaping time series data or summarizing data by categories.	Used for examining relationships between categorical variables in hypothesis testing, social sciences, market research, and other contexts.