5- BIG DATA VISUALISATION

INTRODUCTION TO DATA VISUALIZATION

- Data visualization is graphical representation of data that makes it easy to communicate the info to human. Eg- monthly sales of retail store.
- It helps to: understand trends, patterns and outliers in data.
  communicate data insights clearly and quickly
  better decision and clarity
  Highlight the findings
- Advantages: Easy to understand complex data, Saves time, Helps in decision making, Makes storytelling with data more effective.
- Importance: Helps analyze massive & complex datasets quickly.
  Supports real time decision making.
  Reveals hidden pattern in large datasets.
  Identifies area for improvement.
  Clarify factors which influence human behaviour.
- Challenges: 1) Nature of Big Data (5 V's) - High volume, variety & velocity of data.

2) Information overload - more data = more difficult to extract insights
Overcome by: filtering and aggregation, segmentation, prioritize key metrics (display only most imp data points).

3) Poor Data Quality - Data from various sources often varies in accuracy
Overcome by: Data cleaning, Standardization, Validation (some rules for accuracy)

4) Complex Data Relationships - Big data has many variables & intricate relations
Overcome by: Dimensionality Reduction, Hierarchical views (big to detail), Interactive Visualization (users can explore data dynamically).

5) Visual Noise and Over-plotting – Displaying too many data points can lead to cluttered visuals, making it hard to spot trends / outliers.
Overcome by: Clustering, Design Best Practices.

CONVENTIONAL DATA VISUALIZATION TOOLS (elaborated further)

Microsoft Excel - Basic charts and graphs; widely used.
Tableau - Drag and drop features; interactive dashboards with many inputs
Power BI - Microsoft analytics platform.
QlikView - Fast integration of data from many sources; real time analysis.

TECHNIQUES OF VISUAL DATA REPRESENTATION

TYPES OF DATA VISUALIZATION

1) Bar chart - compare values across different categories

eg.
```
import matplotlib.pyplot as plt
fruits = ['Apple', 'Banana', 'Cherry']
prices = ['50', '60', '70']

plt.bar(fruits, prices, color='skyblue')
plt.title('fruit prices by fruits')
plt.xlabel('fruits')
plt.ylabel('price')
plt.show()
```

2) Line chart - show trends overtime

eg.
```
import matplotlib.pyplot as plt
days = ['Mon', 'Tue', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun']
visitors = ["120", "135", "175", "110", 149, 170, 180]

plt.plot(days, visitors, marker='o', color='green')
plt.title("Website visitors over a week")
plt.xlabel('Day')
plt.ylabel('NO. of visitors')
plt.grid(True)
plt.show()
```

3) Pie chart - show proportions of a whole

```
companies = ['Apple', 'Samsung', 'Xiaomi']
shares = [40, 35, 25]
plt.pie(shares, labels=companies, autopct='%1.1f%%', startangle=140)
plt.title("Smartphone Market Share")
plt.show()
```

4) Scatter plot - show relationship between two variables

```
height = [150, 160, 170, 180, 190]
weight = [60, 50, 70, 80, 90]
plt.scatter(height, weight, color='purple')
plt.title('Height vs Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.show()
```

5) Histogram - show distribution of single variable

```
scores = [55, 67, 69, 82, 85, 53, 59, 92, 68, 88, 70, 73, 90]
plt.hist(scores, bins=5, color='orange', edgecolor='black')
plt.title('Distribution of Exam Scores')
plt.xlabel('Score')
plt.ylabel('No. of students')
plt.show()
```

for seaborn

```
data = [ ]
sns.histplot(data)
plt.title
plt.xlabel
ylabel
show()
```

6) Box plot - show distribution, median, quartiles and outliers

```
classA = [55, 67, 78, 90]
classB = [69, 73, 59, 70]

plt.boxplot([classA, classB, labels = ['ClassA', 'ClassB'])
plt.title('Exam score Distribution')
plt.ylabel('Score')
plt.show()
```

7) Heat Map - show data density or intensity using colors.

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

data = np.random.rand(5,5)
sns.heatmap(data, annot=True, cmap='coolwarm')
plt.title('Sample Heatmap')
plt.show()
```

4 main types of visualization techniques.

1) Comparative Plots - comparing data points
   Column & Bar chart, Line chart, Area chart, Bubble & Pie chart
2) Statistical plot - analysis
   Histograms, Scatterplot, Boxplot, Radarchart, Treemap, Waterfallchart.
3) Topology plot - geometric structures to show relationships
   Linear, Graph, Tree topology.
4) Spatial plots - use logical space view
   Choropleth Map, Point map, Raster Surface, Heat map, Word cloud

VISUALIZATION TOOLS

1) Tableau
Type - Desktop, server, and online versions available
Features - Drag & Drop interface, interactive dashboards, strong mapping capabilities, supports many data sources (CSV, Google Analytics
Use case - Widely used for business analytics and reporting.
Pros - Free public version, many tutorials, powerful mapping.
Cons - Paid versions are expensive, public version lacks privacy for data.

2) Microsoft PowerBI
Type - Cloud based & desktop
features - Deep integration with microsoft products, real time dashboards, natural language queries, AI features.
Use case - Business intelligence and sharing insights across teams.
Pros - User friendly, integrates with office/Azure, real time updates
Cons - Limited customization, complex pricing, can lag with huge datasets.

—

~~And~~ many more.

PROPRIETY DATA VISUALIZATION TOOLS - means paid version
OPEN-SOURCE DATA VISUALIZATION TOOLS - free version sns, plt, etc
ANALYTICAL TECHNIQUES USED IN BIG DATA - same techniques
DATA VISUALIZATION USING TABLEAU - same as above

INTRODUCTION TO: CANDELA, D3.js, GOOGLE CHART API

1) candela - open source suite / virtualization library build on top of Vega, D3.js and WebGL
- features: Offers standardized API for use in JS, Python & R.
  Supports variety of chart types: bar, line, scatter, box, heatmap, etc
  Integrates advanced visualization like LineUp, Upset & Onset.
  research
  Used in web apps, Jupyter nb, and RStudio.
- Use case: Ideal for developers and data scientists who want reusable, scalable and interactive visualizations across different platforms.

2) D3.js - (Data Driven Document) is powerful JS library for creating dynamic, interactive and customizable data visualizations for the web.
- Eg. Animated bar charts, Interactive line graphs, Dynamic n/w diagrams

- Features : Uses HTML, SVG & CSS to render visualization in browser.
Enables creation of wide range of visuals
Highly flexible and customizable, allowing for animations, transitions and user interactions.
Large, active community & extensive documentation

- Use case : Best for developers who need complete control over the look & behaviour of their visualizations.

3) Google Chart API - It is a free, web-based toolset for creating interactive charts and visualizations that can be embedded in web pages.

- Eg. : Pie chart showing budget distribution, Geo chart showing users by location, Bar chart for sales comparison.

- Features : Supports wide range of chart types.
Easy to use with simple JS code and Google's online documentation
Highly interactive and customizable.
Integrates well with other Google products & works in any browser.

- Use case : Suitable for anyone needing quick, interactive charts for websites or reports, without complex setup or coding.

PYQ] Data visualization wrt 1D, 2D & 3D.

1D involves single variable or feature. Typically used for showing the distribution or frequency of that variable.

Eg. Line chart - daily temp over a month. Ideal for showing trends, patterns in single variable.
Histogram - Distribution of students test score.

2D involves 2 variables, often represented on x & y axis of flat plane.

Eg. Scatter plot - Height vs weight
Bar chart - Sales by Product
Heat map - Website activity by hour & day.
Ideal for data analysis

3D involves 3 variables, represented on x, y & z axes, adding depth to the visualization. Displays more complex relationships & patterns.

Eg. 3D Scatter plot - relation betn age, income & spending
Surface plot - elevation data for geographic region
3D Bar chart - 3 values

Ideal for exploring complex dataset & spatial relationships.