DEPARTMENT OF BIOMEDICAL ENGINEERING

24AI671-FUNDAMENTALS OF DATA SCIENCE

UNIT III – EXPLORATORY DATA ANALYSIS AND VISUALIZATION

3.4. VISUALIZATION TOOLS

Visualization tools are software and libraries that help us create charts, graphs, maps and dashboards from data. The choice of **tool** depends on the **skill of the user, size and type of data, level of interactivity required and the way results will be shared**. Tools range from simple spreadsheet software to powerful programming libraries and enterprise business-intelligence (BI) platforms.

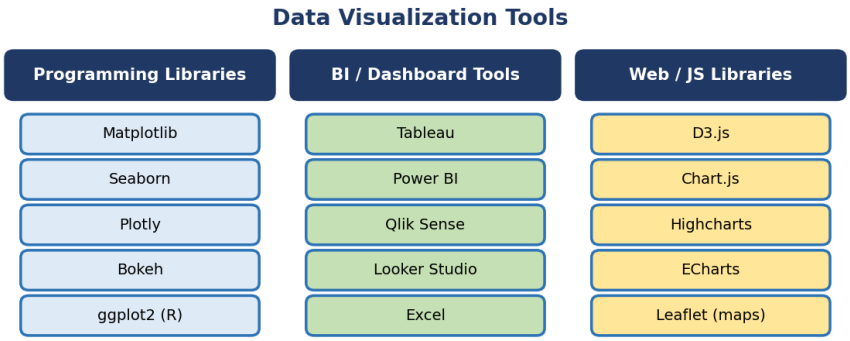

Fig 2.1 Popular data visualization tools

Categories of Visualization Tools

1. Spreadsheet Tools

Microsoft Excel and Google Sheets are used for quick and simple charts from small data. They are easy to learn but have limited chart types and are not suitable for very large data or advanced interactivity.

2. Programming Libraries

These need coding but give **full control and flexibility**, and they are easy to combine with data analysis and machine learning.

Library	Language	Features
Matplotlib	Python	Basic and most widely used 2D plotting library; foundation of many others
Seaborn	Python	High-level statistical plots built on Matplotlib; attractive default styles
Plotly	Python / R / JS	Interactive, web-ready charts with zoom and hover
Bokeh	Python	Interactive visualization for web browsers
ggplot2	R	Grammar-of-graphics based, very popular in statistics
D3.js	JavaScript	Highly customised, web-based interactive graphics

3. Business Intelligence (BI) and Dashboard Tools

These are **drag-and-drop tools** that connect to databases and files and create interactive dashboards **without much coding**. Examples are **Tableau, Microsoft Power BI, Qlik Sense and Google Looker Studio**. They are widely used in industry for reporting and decision-making.

4. Specialised and Web Tools

- **Geographic tools** – QGIS, ArcGIS, Leaflet and Google Maps for map-based visualization.
- **Network and graph tools** – Gephi, NetworkX for relationship and network diagrams.
- **JavaScript chart libraries** – Chart.js, Highcharts and ECharts for charts on websites.

Comparison of Important Tools

Feature	Matplotlib / Seaborn	Tableau	Power BI
Type	Programming library	BI tool	BI tool
Coding needed	Yes (Python)	No / very little	No / little (DAX for calculations)
Ease of learning	Moderate	Easy	Easy, especially for Excel users
Interactivity	Limited (static plots)	High	High
Cost	Free, open source	Paid (free Tableau Public / Desktop trial)	Free Desktop; paid cloud service
Best for	Data analysis, research, custom plots	Advanced visual analytics	Business reports, Microsoft ecosystem

Factors for Selecting a Tool

- Type, size and source of the data.
- Skills of the users – coders or business users.
- Level of interactivity and dashboards required.
- Cost and licensing.
- Ability to share, publish and update results automatically.
- Integration with other tools and databases.

3. MATPLOTLIB

Matplotlib is the most popular and fundamental **Python library for creating static, animated and interactive 2D plots**. It was created by John D. Hunter in 2003. It works closely with **NumPy and Pandas** and is the base on which

libraries like Seaborn are built. It can produce publication-quality figures in many formats (PNG, PDF, SVG).

Installation and import (the module **pyplot** gives MATLAB-like plotting commands):

```
pip install matplotlib


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

Anatomy of a Matplotlib Figure

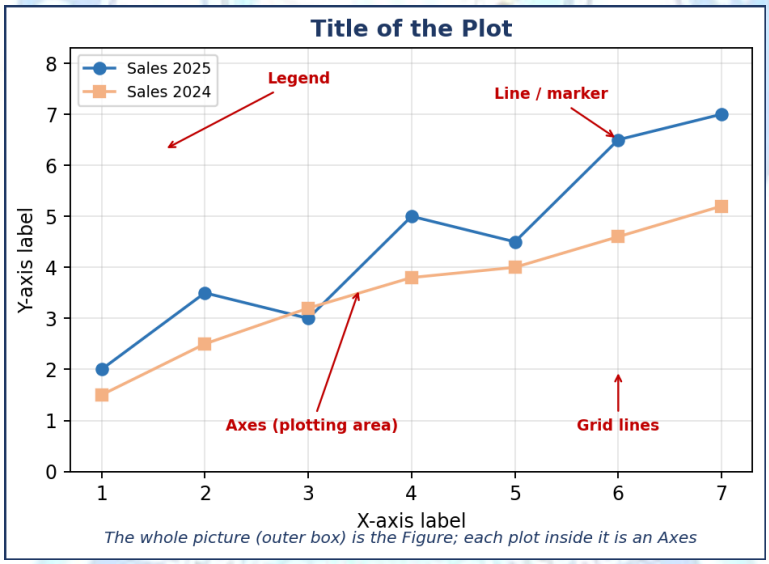

Fig 3.1 Parts of a Matplotlib figure

- **Figure** – the whole window or page that contains everything.
- **Axes** – one individual plot (area with x-axis and y-axis). A figure can contain many axes.
- **Axis, ticks and tick labels** – the number lines and the markings on them.
- **Title, x-label and y-label** – describe the plot and units.
- **Legend** – explains the lines or markers.
- **Grid** – optional lines that help read values.

Basic Plotting Functions

Function	Plot	Use
plt.plot()	Line plot	Trends over time
plt.bar() / plt.barh()	Vertical / horizontal bar chart	Comparing categories
plt.scatter()	Scatter plot	Relationship between two variables
plt.hist()	Histogram	Distribution of one variable
plt.pie()	Pie chart	Parts of a whole
plt.boxplot()	Box plot	Spread and outliers
plt.imshow()	Image / heatmap	Matrices and images

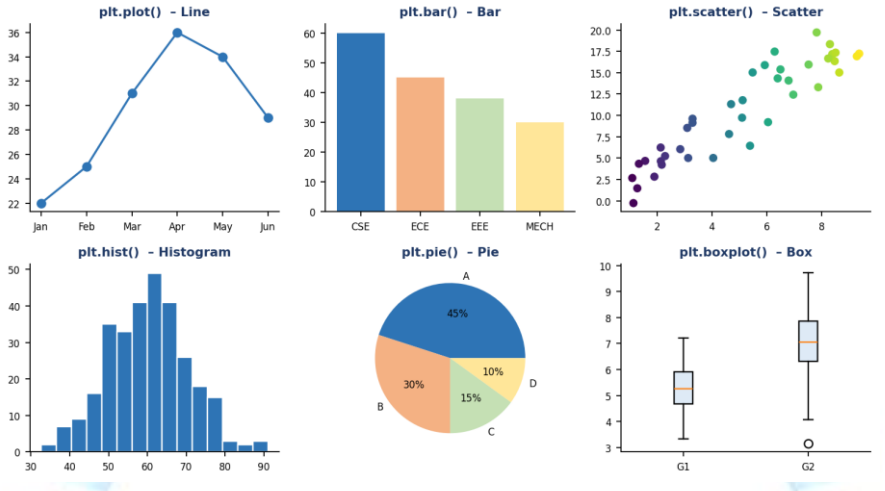

Fig 3.2 Basic Matplotlib plots

Example 1 – Simple Line Plot

```
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May']

sales  = [22, 25, 31, 36, 34]
```

```
plt.plot(months, sales, color='blue', marker='o', linestyle='-', label='2025')
plt.title('Monthly Sales')          # title
plt.xlabel('Month')                 # x-axis label
plt.ylabel('Sales (lakh)')          # y-axis label
plt.legend()                        # show legend
plt.grid(True)                      # show grid
plt.show()                          # display the figure
```

Example 2 – Multiple Plots Using Subplots

```
fig, ax = plt.subplots(1, 2, figsize=(10, 4))   # 1 row, 2 columns


ax[0].bar(['CSE', 'ECE', 'EEE'], [60, 45, 38], color='green')
ax[0].set_title('Students per Branch')


ax[1].scatter(x, y, c='red')
ax[1].set_title('X vs Y')


plt.tight_layout()
plt.savefig('chart.png', dpi=300)               # save the figure
plt.show()
```

Customising Plots

- **Colour, line style and marker:** color='red', linestyle='--', marker='o', linewidth=2.

- **Figure size:** plt.figure(figsize=(8, 5)) – width and height in inches.
- **Axis limits:** plt.xlim(0, 10), plt.ylim(0, 100).
- **Text and annotation:** plt.text(), plt.annotate() to highlight points.
- **Style sheets:** plt.style.use('ggplot') or 'seaborn-v0_8' to change the overall look.
- **Saving:** plt.savefig('file.png') – formats PNG, JPG, PDF, SVG.

Advantages and Limitations

Advantages	Limitations
Free and open source; very flexible	Needs more code for attractive plots
Full control over every element	Default style is plain
Works with NumPy, Pandas, SciPy	Mainly static; limited interactivity
Publication-quality output in many formats	Complex statistical plots need more effort

4. SEABORN

Seaborn is a Python data visualization library **built on top of Matplotlib** and closely integrated with **Pandas DataFrames**. It provides a **high-level interface for drawing attractive and informative statistical graphics** with very little code. It automatically handles grouping, aggregation, colour palettes and statistical estimation such as confidence intervals and regression lines.

```
pip install seaborn


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

Seaborn vs Matplotlib

Point	Matplotlib	Seaborn
Level	Low-level; full control	High-level; built on Matplotlib
Code needed	More lines of code	Few lines for complex plots
Appearance	Plain default style	Attractive default themes and palettes
Data input	Lists, arrays	Works directly with Pandas DataFrames
Statistics	Manual calculation	Built-in statistical estimation (KDE, regression, confidence interval)
Best for	Custom and basic plots	Statistical and exploratory data analysis

Types of Seaborn Plots

1. Distribution Plots (one variable)

- **histplot()** – histogram, with optional KDE (smooth density curve).
- **kdeplot()** – smooth estimate of the distribution.
- **boxplot() and violinplot()** – show median, quartiles, spread and outliers.

2. Relational Plots (two variables)

- **scatterplot()** – relationship between two numeric variables; colour (hue) and size can show extra variables.
- **lineplot()** – trend over time, with confidence interval.

3. Categorical Plots

- **barplot()** – average value per category with error bars.
- **countplot()** – count of observations in each category.

- **boxplot(), violinplot(), stripplot(), swarmplot()** – distribution within each category.

4. Matrix Plots

- **heatmap()** – colour-coded matrix; most often used for a correlation matrix.
- **clustermap()** – heatmap with hierarchical clustering.

5. Multi-Plot Grids

- **pairplot()** – scatter plots of every pair of variables plus histograms; a quick overview of the whole data set.
- **FacetGrid / catplot / relplot** – the same plot repeated for different subsets of the data.
- **jointplot()** – scatter plot of two variables with their histograms on the margins.
- **regplot() / lmplot()** – scatter plot with a fitted regression line.

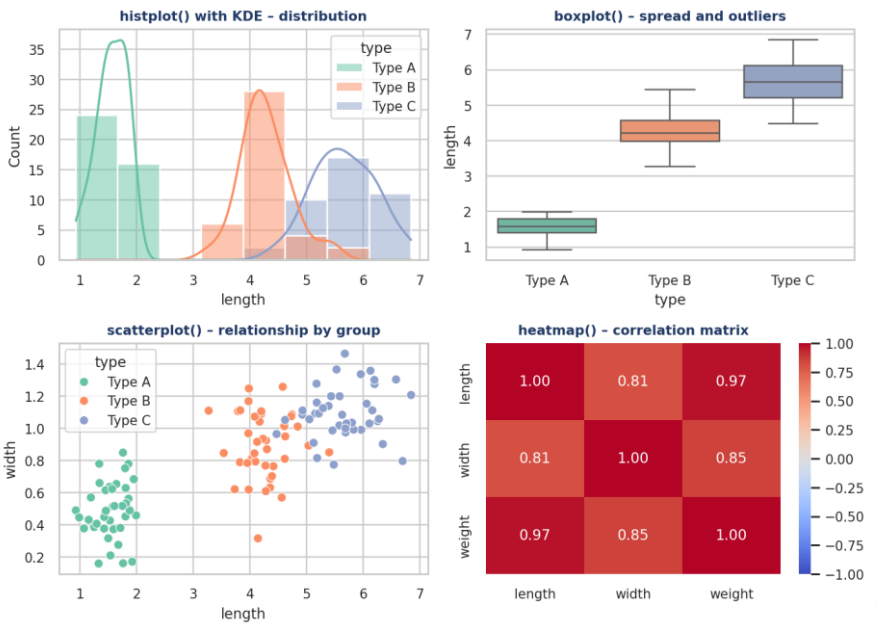

Fig 4.1 Common Seaborn plots

Example Code

```
df = sns.load_dataset('tips')          # built-in sample data set

sns.set_theme(style='whitegrid')       # set theme
```

```
sns.histplot(data=df, x='total_bill', kde=True)          # distribution

sns.boxplot(data=df, x='day', y='total_bill')            # box plot by category

sns.scatterplot(data=df, x='total_bill', y='tip', hue='sex')   # relation + colour

sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')  # correlation

sns.pairplot(df, hue='time')          # all pair relationships



plt.title('Tips Analysis')

plt.show()
```

Themes and Colour Palettes

- **Themes:** darkgrid, whitegrid, dark, white and ticks, set by sns.set_theme(style='...').
- **Colour palettes:** sns.color_palette() – sequential ('Blues'), diverging ('coolwarm', 'RdBu') and categorical ('Set2', 'tab10', 'colorblind').
- **Context:** sns.set_context('paper' / 'notebook' / 'talk' / 'poster') adjusts size of labels and lines for the output medium.

Advantages and Limitations

Advantages	Limitations
Beautiful plots with very little code	Less customisation than pure Matplotlib
Directly supports Pandas DataFrames	Static plots; limited interactivity
Built-in statistical functions	Slow on very large data sets
Excellent for exploratory data analysis (EDA)	Needs Matplotlib for final fine-tuning

5. TABLEAU AND POWER BI

Tableau and **Microsoft Power BI** are the two leading **Business Intelligence (BI) and data visualization tools**. They allow users to connect to many data sources and create **interactive charts, reports and dashboards by drag and drop**, with little or no programming. They are widely used in industry for business analytics and decision-making.

Tableau

Tableau is a powerful visual analytics tool (now owned by Salesforce) known for its **beautiful, highly interactive visualizations and ease of use**. The user simply drags fields onto the workspace and Tableau automatically builds the chart.

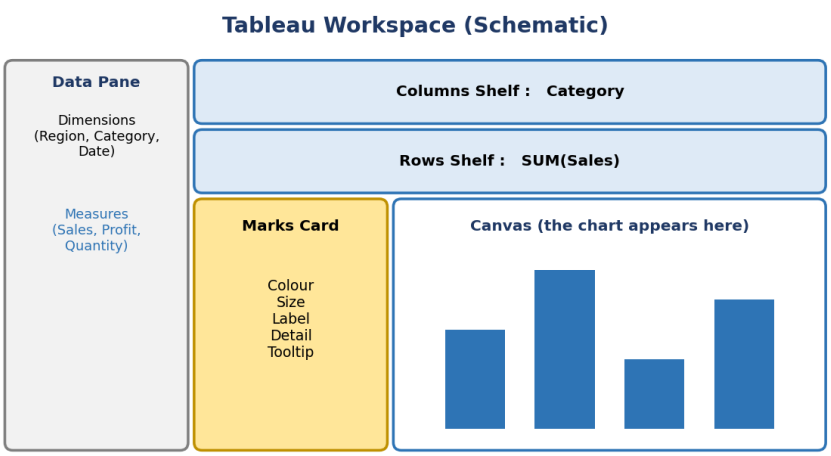

Fig 5.1 Tableau workspace

Tableau Products

- **Tableau Desktop** – used to connect to data and create visualizations and dashboards.
- **Tableau Server / Cloud (Online)** – used to publish, share and manage dashboards securely in an organisation.
- **Tableau Public** – free version where work is published openly on the web.
- **Tableau Prep** – used to clean and prepare data.
- **Tableau Mobile** – view dashboards on phones and tablets.

Important Concepts in Tableau

- **Dimensions** – descriptive, categorical fields (region, product, date). **Measures** – numeric values that can be calculated (sales, profit, quantity).
- **Columns and Rows shelves** – fields dragged here decide the x-axis and y-axis of the chart.
- **Marks card** – controls colour, size, label, detail and tooltip of the data points.
- **Worksheet** – a single chart or view. **Dashboard** – a collection of worksheets arranged on one page. **Story** – a sequence of sheets that tells a data story.
- **Filters** – limit the data shown. **Parameters** – let users change a value and update the view. **Calculated fields** – new fields made using formulas.
- **Data connection types** – Live (always up to date) and Extract (a fast, saved copy of the data).
- **Chart types:** bar, line, pie, scatter, heat map, tree map, bubble, Gantt, maps, dual-axis and many more.

Microsoft Power BI

Power BI is Microsoft's business analytics service. It gives **interactive visualizations with self-service business intelligence** and integrates easily with Excel, Azure, SQL Server and other Microsoft products. It is very popular because of its low cost and easy learning.

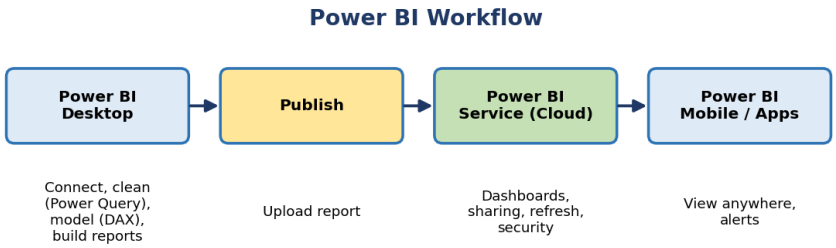

Fig 5.2 Power BI workflow

Power BI Components

- **Power BI Desktop** – free Windows application to connect, transform, model and create reports.

- **Power BI Service (Cloud)** – online platform to publish, share, create dashboards, schedule refresh and manage security.
- **Power BI Mobile** – apps for viewing reports on mobile devices.
- **Power Query** – tool for extracting, cleaning and transforming data (ETL).
- **Power Pivot / Data Model** – creates relationships between tables.
- **DAX (Data Analysis Expressions)** – formula language for calculated columns and measures, e.g. Total Sales = SUM(Sales[Amount]).
- **Report vs Dashboard** – a report has multiple pages of visuals; a dashboard is a single page of pinned visuals (tiles) from one or more reports.
- **Visuals:** bar, column, line, pie, donut, map, card, gauge, matrix, slicer and custom visuals from the marketplace.

Comparison of Tableau and Power BI

Feature	Tableau	Power BI
Developer	Tableau (Salesforce)	Microsoft
Strength	Advanced, highly attractive visual analytics	Easy to use; integrates with Excel and Microsoft tools
Cost	Higher	Lower; Desktop is free
Data handling	Handles very large data and complex visualization	Good; best with Microsoft data sources
Formula language	Calculated fields, LOD expressions	DAX, Power Query (M language)
Learning curve	Easy to start; advanced features need practice	Easy, especially for Excel users
Users	Data analysts, data scientists	Business users, analysts, managers

Steps to Create a Visualization (Common Workflow)

- Connect to the data source (Excel, CSV, SQL database, cloud).
- Clean and transform the data (Tableau Prep / Power Query).
- Drag fields to create charts.
- Add filters, calculated fields or measures.
- Combine charts into a dashboard.
- Publish and share on the Server / Cloud and set refresh schedule.

