

# Introduction To Data Visualization

With *matplotlib*

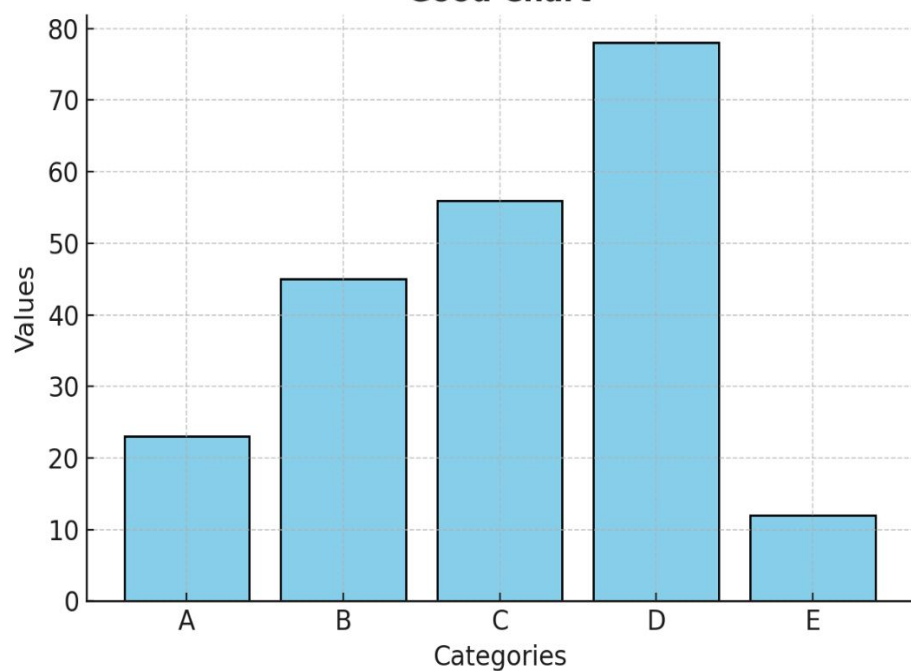
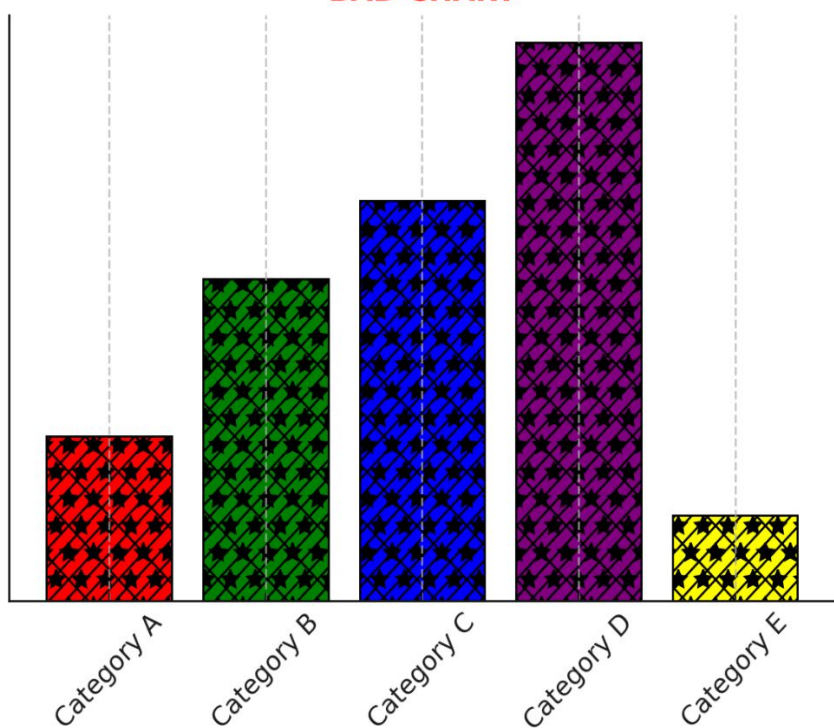


# What is Data Visualization

- Data visualization is the graphical representation of data to help identify patterns, trends, and insights
- In data science, it is an essential tool for exploring data, communicating findings, and making data-driven decisions.
- Effective visualizations simplify complex datasets, making them easier to understand and analyze.



# Why good visualizations matter



# Bar Charts

## Use Cases:

- Comparing categorical data

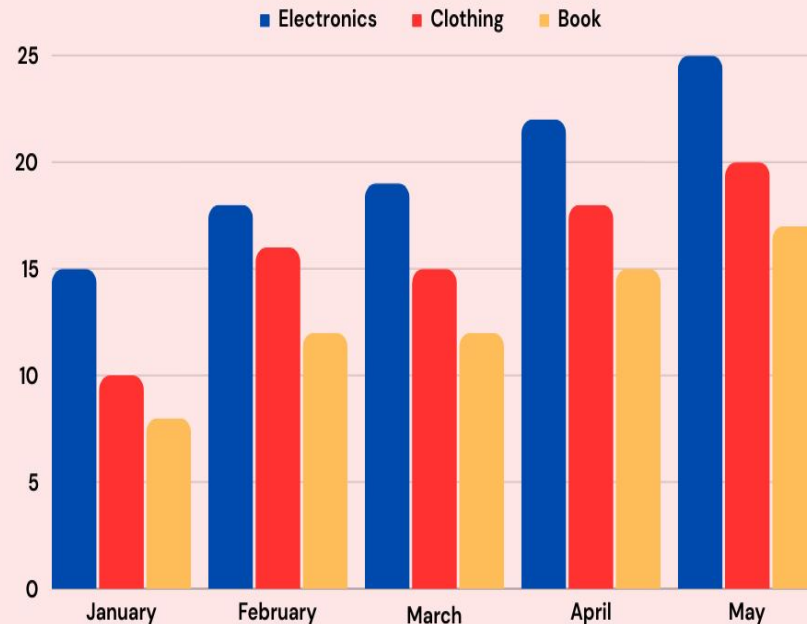
## Data Types:

- Categorical (x-axis) and Numerical (y-axis)

## Benefits:

- Easy to compare values across categories
- Clearly visualizes trends and differences
- Works well with both small and large datasets

## PRODUCT SALES REVENUE



# Scatter Plots

## Use Cases:

- Examining the relationships and correlation between variables

## Data Types:

- Two numerical variables (x-axis and y-axis)

## Benefits:

- Great for spotting trends and correlations
- Helps identify clusters and patterns in data
- Useful for regression analysis





# Pie Charts

## Use Cases:

- Showing distribution and percentage breakdowns

## Data Types:

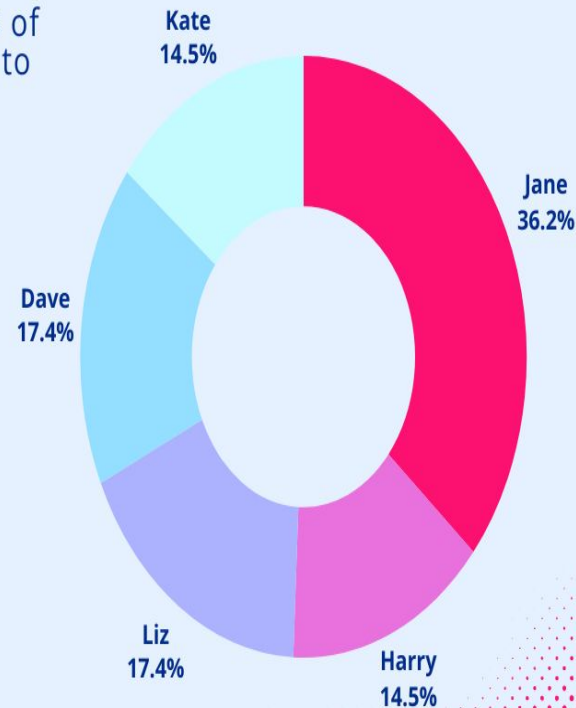
- Categorical (labels) and Numerical (proportions/percentages)

## Benefits:

- Helps reveal the shape and spread of data
- Useful for detecting skewness and outliers
- Ideal for summarizing large datasets

## Project Resources Distribution

Display proportions of resources assigned to team members



# Histograms

(AKA Bar Graph)

## Use Cases:

- Understanding distributions and frequency within ranges

## Data Types:

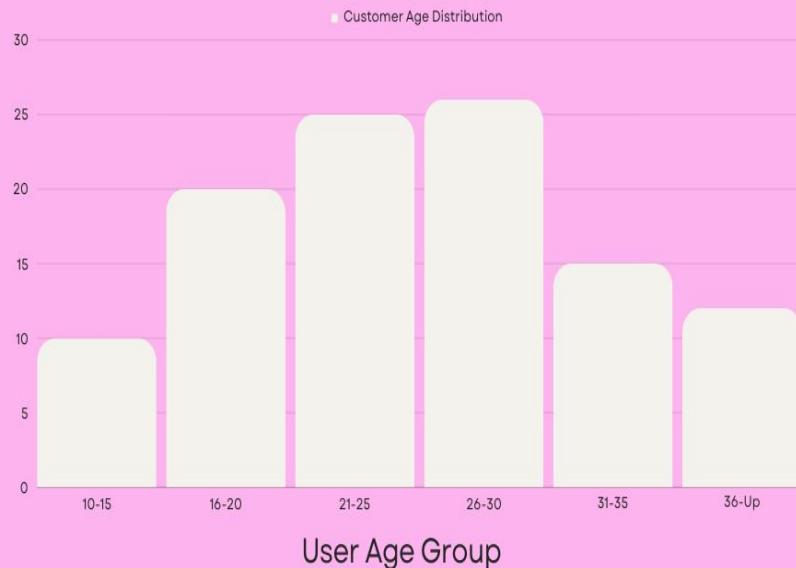
- Numerical (grouped into bins)

## Benefits:

- Simple and intuitive representation of proportions
- Effective for showing part-to-whole relationships
- Best for datasets with a few distinct categories

## Distribution of Customer Ages

January 2024



# Lets Get Plotting

Key matplotlib functions

## `plt.title()`

Key Parameters:

- String of title name

## `plt.xlabel()` and `plt.ylabel()`

Key Parameters:

- String of respective axis name

## `plt.scatter()`

Key Parameters:

- x: Numerical data
- y: Numerical data
- color: Point color

## `plt.bar()`

Key Parameters:

- x: Categorical data
- height: Numerical values
- color: Bar color (default is blue)

## `plt.pie()`

Key Parameters:

- x: Numerical data
- labels: Names
- autopct: Displays percentage values
- colors: Slices color

## `plt.hist()`

Key Parameters:

- x: Numerical data
- bins: Number of intervals (default is 10)
- color: Bar color