WorkShop
Data Science

www.svnapro.com

ESTD 2001

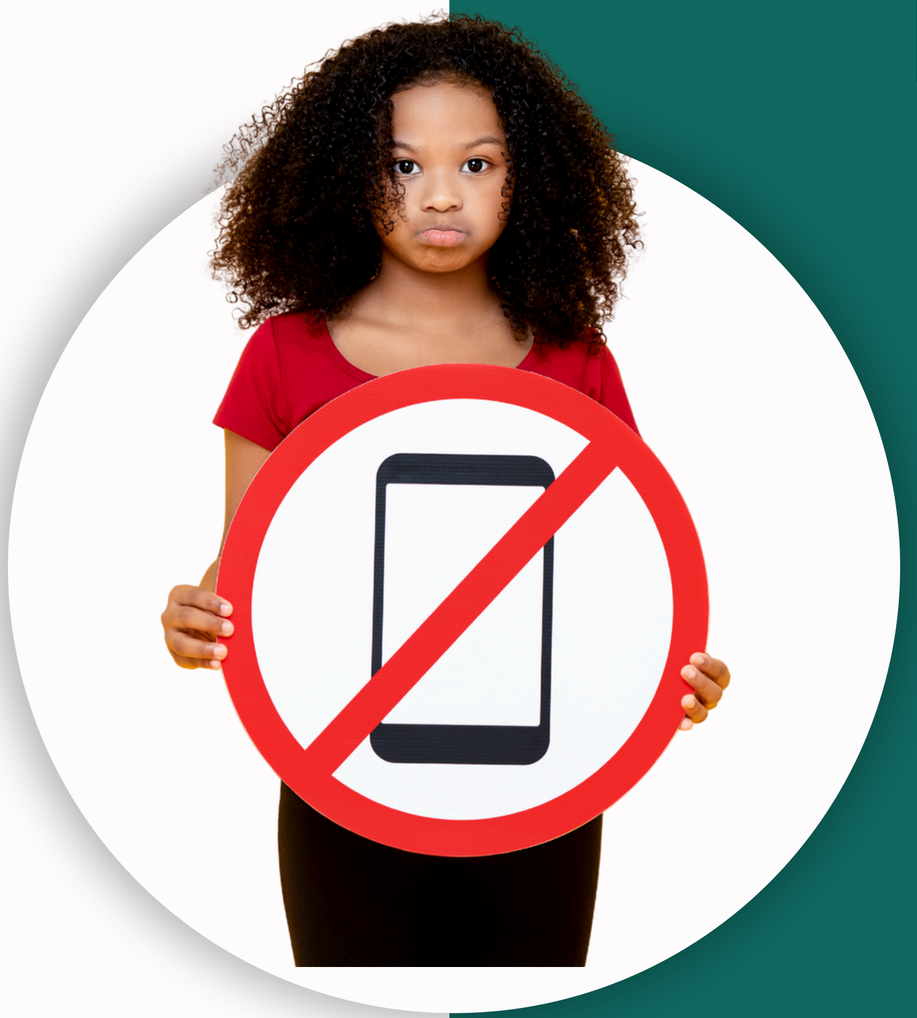
SAI SPURTHI
Institute of Technology
(Approved BY AICTE, Affiliated To JNTU Hyderabad)

svnapro

info@svnapro.com
+917989975085

Ground Rules

- **No Mobile.**
- **Download all files.**
- **Work along for an interactive session.**
- **Raise your hand🙋 if you have any doubts.**



About Me...

I'm Venu Prakasg. I'm a solutions specialist, a freelancer, and a trainer who loves to help people solve problems and build their skills. I graduated from PES University and I'm excited to share what I've learned with all of you today.

📞+91 6302643093 - 📨venuprakash17@gmail.com

Lets start . . . .

Numpy

www.svnapro.com

Introduction to NumPy

- NumPy = Numerical Python
- Core library for scientific computing
- Provides ndarray (N-dimensional array)
- Backbone of Pandas, SciPy, TensorFlow, PyTorch

Use Cases: Data science, ML, simulations, image processing.

svnapro

Why NumPy?

- Python lists → slow & memory-heavy.
- NumPy arrays → fast, efficient, vectorized.
- Homogeneous data types → optimized for math.

Example: Handle millions of stock prices in seconds instead of minutes.

svnapro

Creating Arrays

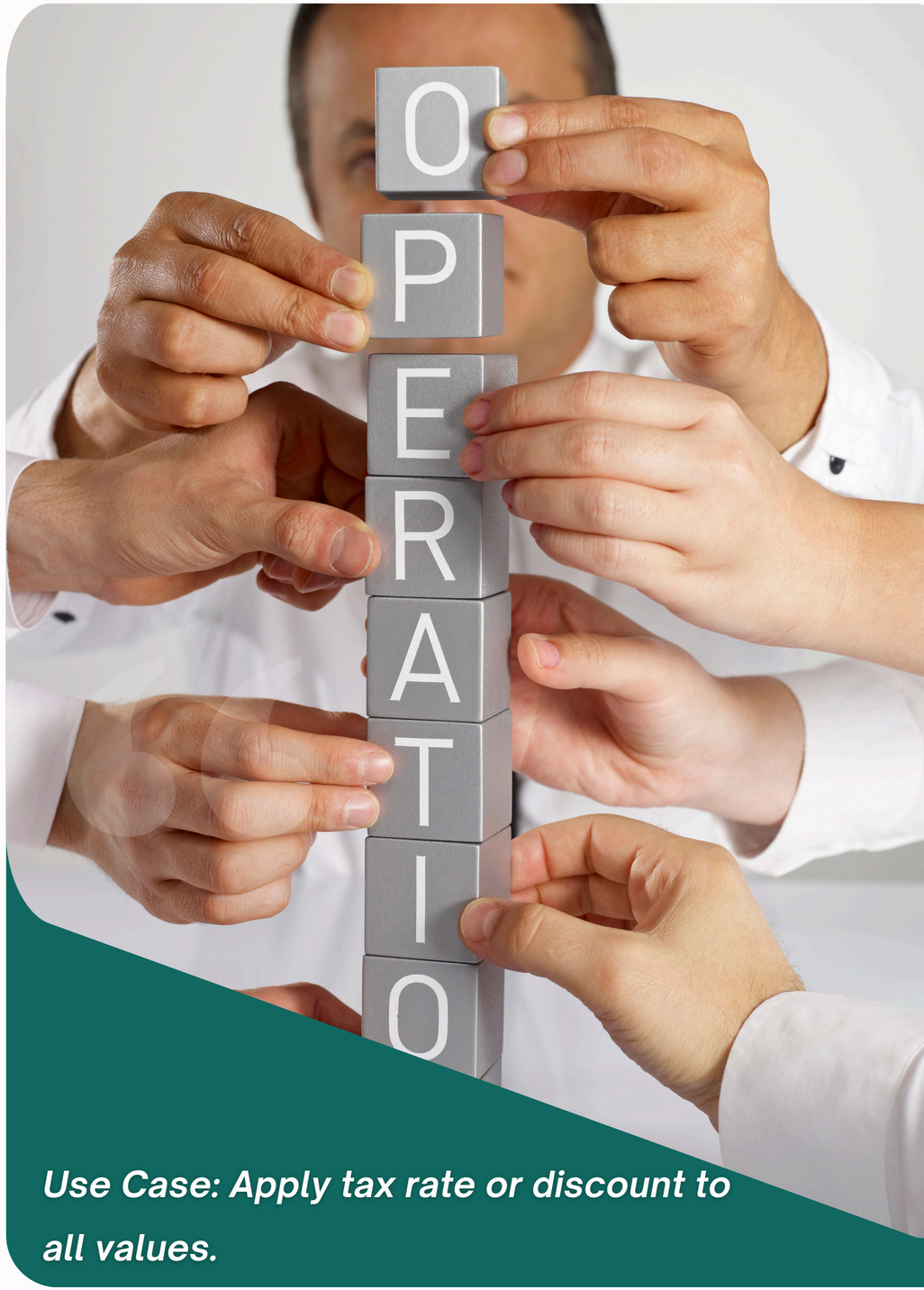
- From lists: np.array([1,2,3])
- Special arrays:
- np.zeros((2,3)) → matrix of zeros
- np.ones((3,3)) → matrix of ones
- np.arange(0,10,2) → range with step
- np.linspace(0,1,5) → evenly spaced values

Use Case: Initialize weight matrices in ML, simulation grids.

svnapro

Array Operations & Broadcasting

- Element-wise ops: + - * / **
- Aggregate functions: mean(), sum(), std(), max()
- Broadcasting: Apply scalar ops to whole arrays.



Use Case: Apply tax rate or discount to all values.

svnapro

Real-World Applications

- Machine Learning → tensor ops, preprocessing
- Finance → stock simulations, portfolio optimization
- Healthcare → MRI/CT scan image arrays
- Engineering → signal/image processing.

NumPy = The Engine ⚡ behind all Python Data Science tools.

Lets start . . . .

Pandas

www.svnapro.com

Pandas

- Pandas = Python Data Analysis Library 🐼
- Built on NumPy, integrates with Matplotlib & Seaborn
- Optimized for structured/tabular data (rows + columns)
- Think of it as: Excel + SQL + Python combined
- Widely used in Data Science, AI, Business Analytics

Example: Analyze employee salaries, sales transactions, or customer demographics.

svnapro

Why Use Pandas?

- Easier than raw Python for data handling
- Can handle large datasets efficiently
- High-level functions for filtering, grouping, cleaning, visualization
- Compatible with CSV, Excel, JSON, SQL databases

💡 Without Pandas: you'd manually loop over lists/dicts.

💡 With Pandas: one-liners do the job.

svnapro

Core Data Structures

- Series → 1D labeled array (like a single column in Excel)
- DataFrame → 2D labeled table (like a full spreadsheet)

Series = Age column; DataFrame = Employee table.

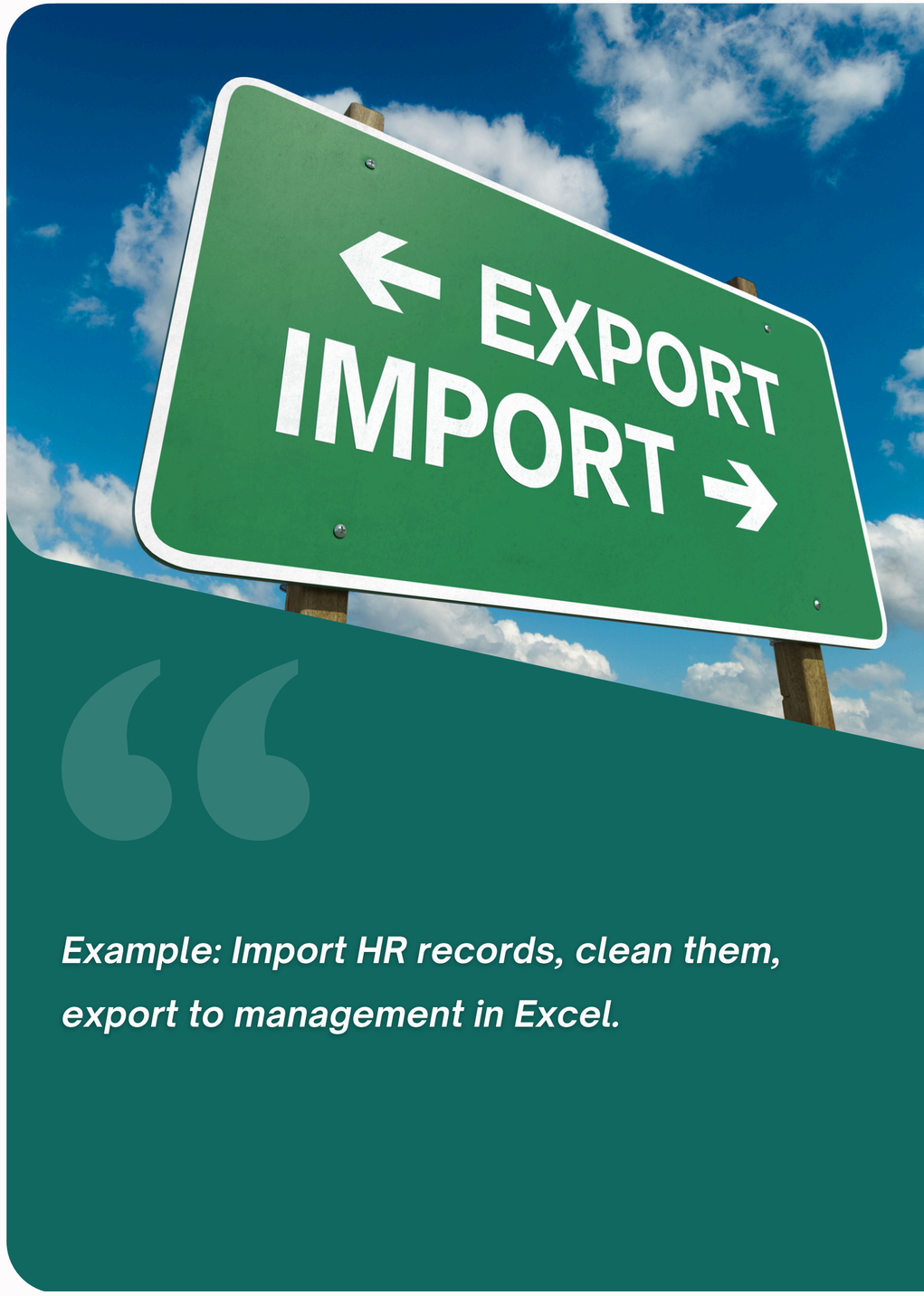
Importing & Exporting Data

Import

- CSV → pd.read_csv("data.csv")
- Excel → pd.read_excel("data.xlsx")
- JSON → pd.read_json("data.json")
- SQL → pd.read_sql(query, connection)

Export

- df.to_csv("clean.csv", index=False)



Example: Import HR records, clean them, export to management in Excel.

Inspecting Data

- Quick look at dataset:
- df.head() → first 5 rows
- df.tail() → last 5 rows
- df.info() → column data types & nulls
- df.describe() → summary statistics
- df.shape → rows × columns

Example: Inspect sales dataset → see number of customers, missing values, avg sales.

svnapro

Selecting & Filtering

- Select Columns → df['Name']
- Select Rows →
- df.iloc[0] (by position), df.loc[2,'Salary'] (by label)
- Filtering → df[df['Age'] > 30]

Example: Extract employees from “IT” earning > 70k.

Modifying & Cleaning

- Add new columns → df['AnnualSalary'] = df['Salary'] * 12
- Handle missing values:
- df.dropna() → remove rows
- df.fillna(0) → replace with constant
- df.fillna(df['col'].mean()) → fill with average

Example: Fill missing sales with avg sales, flag senior employees.

Grouping & Aggregation

- Summarize using groupby()
- Aggregations: mean, sum, min, max, count
- Sorting: df.sort_values(by='Salary')

Example: Avg salary per department, max bonus per region.

Lets start . . . .

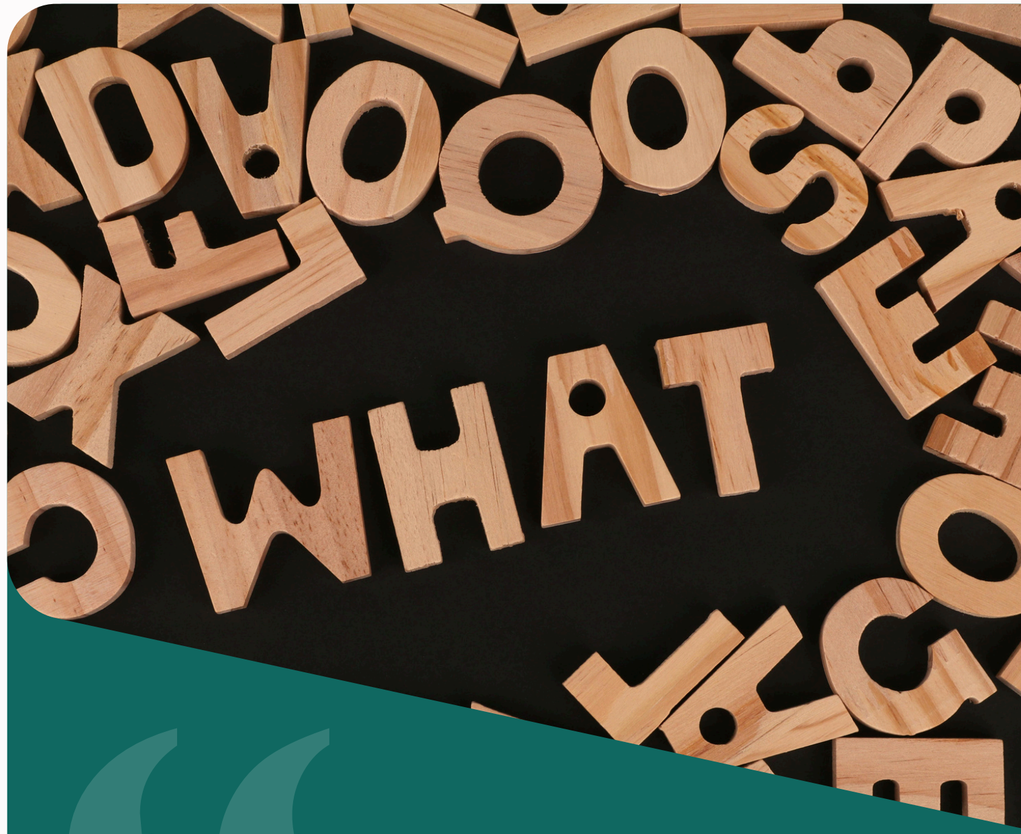
Matplotlib

www.svnapro.com

svnapro

Matplotlib - What is Matplotlib?

- Foundation of data visualization in Python
- Creates static, animated, and interactive plots
- Works with NumPy arrays & Pandas DataFrames
- Gives full control of every element in the plot



💡 *Why it matters: Almost every visualization library in Python builds on Matplotlib (Seaborn, Plotly, etc.).*

svnapro

Basic Plot Types

- Line plot (plt.plot()) → show trends/changes over time
- Bar chart (plt.bar()) → compare categories
- Histogram (plt.hist()) → distribution of values
- Scatter plot (plt.scatter()) → relationship between variables

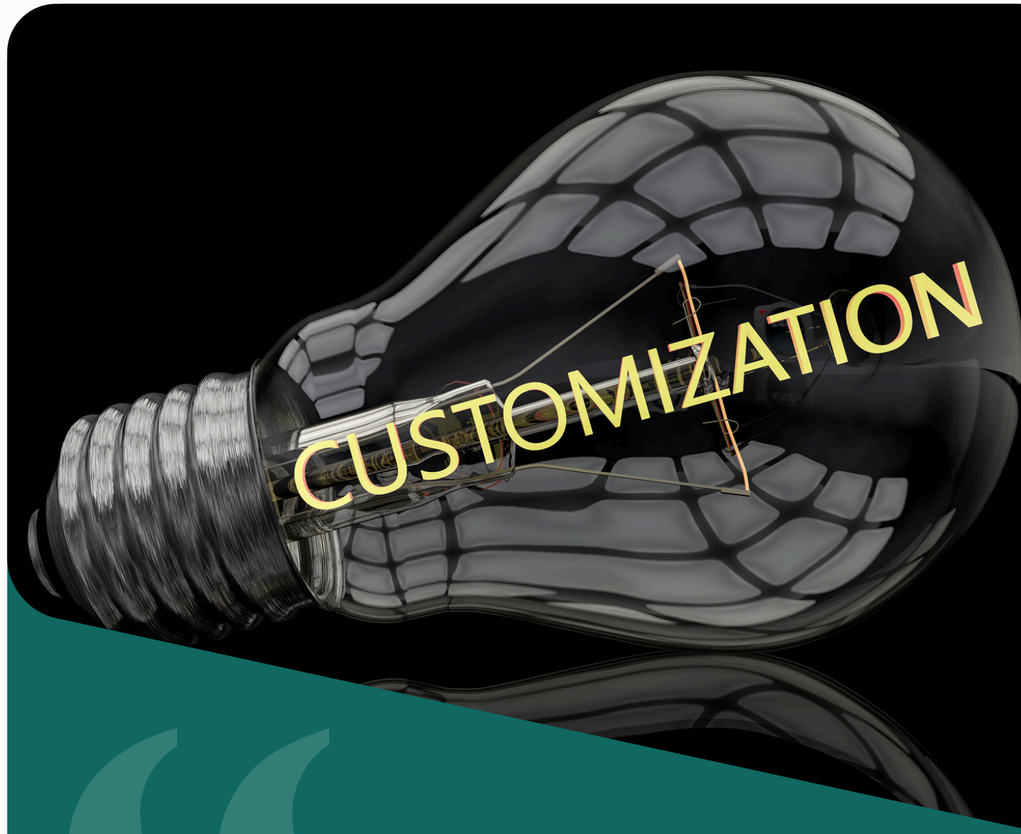
💡 *Examples:*

- *Line → Sales over months*
- *Bar → Revenue by department*
- *Histogram → Age distribution of customers*
- *Scatter → Age vs Salary correlation*

svnapro

Matplotlib - Customizing Plots

- Titles & labels: plt.title(), plt.xlabel(), plt.ylabel()
- Legends: explain colors/markers (plt.legend())
- Colors, markers, line styles: 'r--' (red dashed line), marker='o'
- Gridlines & axis limits: plt.grid(True), plt.xlim(), plt.ylim()



💡 *Why: Customization makes plots clearer for presentations & reports.*

svnapro

Layouts & Advanced Features

- Subplots: Show multiple plots in one figure (plt.subplot(), plt.subplots())
- Pie charts: Visualize proportions (plt.pie())
- Annotations: Highlight key points (plt.annotate())
- Saving plots: plt.savefig("plot.png")

💡 *Example: Create a dashboard with sales trend, department revenue, and customer age distribution in one figure.*

svnapro

Matplotlib - Summary

- Matplotlib = Power + Flexibility
- Best when you need fine control over plots
- Great for: Research papers, customized reports, dashboards
- Often paired with Seaborn for faster, prettier visuals

Lets start . . . .

Seaborn

www.svnapro.com

svnapro

Seaborn - What is Seaborn?

- High-level statistical visualization library built on Matplotlib
- Provides beautiful default styles
- Works directly with Pandas DataFrames
- Focuses on statistical insights: distributions, categories, correlations

💡 *Why it matters: Makes EDA (Exploratory Data Analysis) faster & prettier.*

svnapro

Visualizing Distributions & Relationships

- Histplot / KDE: Show distribution of a variable
- Scatterplot: Relationship between two variables
- Jointplot: Combines scatter + histograms
- Pairplot: Grid of pairwise relationships across multiple variables

💡 *Examples:*

- *Histplot → Salary distribution*
- *Scatterplot → Age vs Salary correlation*
- *Pairplot → Explore multiple HR features (Age, Salary, Bonus)*

svnapro

Visualizing Categories

- Boxplot: Shows spread, median, and outliers
- Violinplot: Boxplot + density estimate
- Countplot: Frequency of categories
- Barplot: Shows average/aggregated values with error bars

💡 *Examples:*

- *Boxplot → Compare salaries across departments*
- *Countplot → Number of employees in each department*
- *Barplot → Average revenue per region*

svnapro

Advanced Statistical Visuals

- Heatmap: Show correlations between variables
- FacetGrid: Create multiple plots split by category
- Lineplot: Trends over time (with confidence intervals)

💡 *Examples:*

• Heatmap → Correlation of Salary, Age, Bonus

• FacetGrid → Age distribution per department

• Lineplot → Monthly sales trend with

svnapro

Seaborn - Summary

- Seaborn = Quick, Beautiful, Insightful
- Best for: Exploratory data analysis
- Statistical insights
- Fast plots with Pandas DataFrames
- Use Seaborn for EDA, switch to Matplotlib for custom dashboards/reports

Thank You!

www.svnapro.com

info@svnapro.com
+91 79899 75085

svnapro