DA7 - SEABORN, BOKEH AND PLOTLY

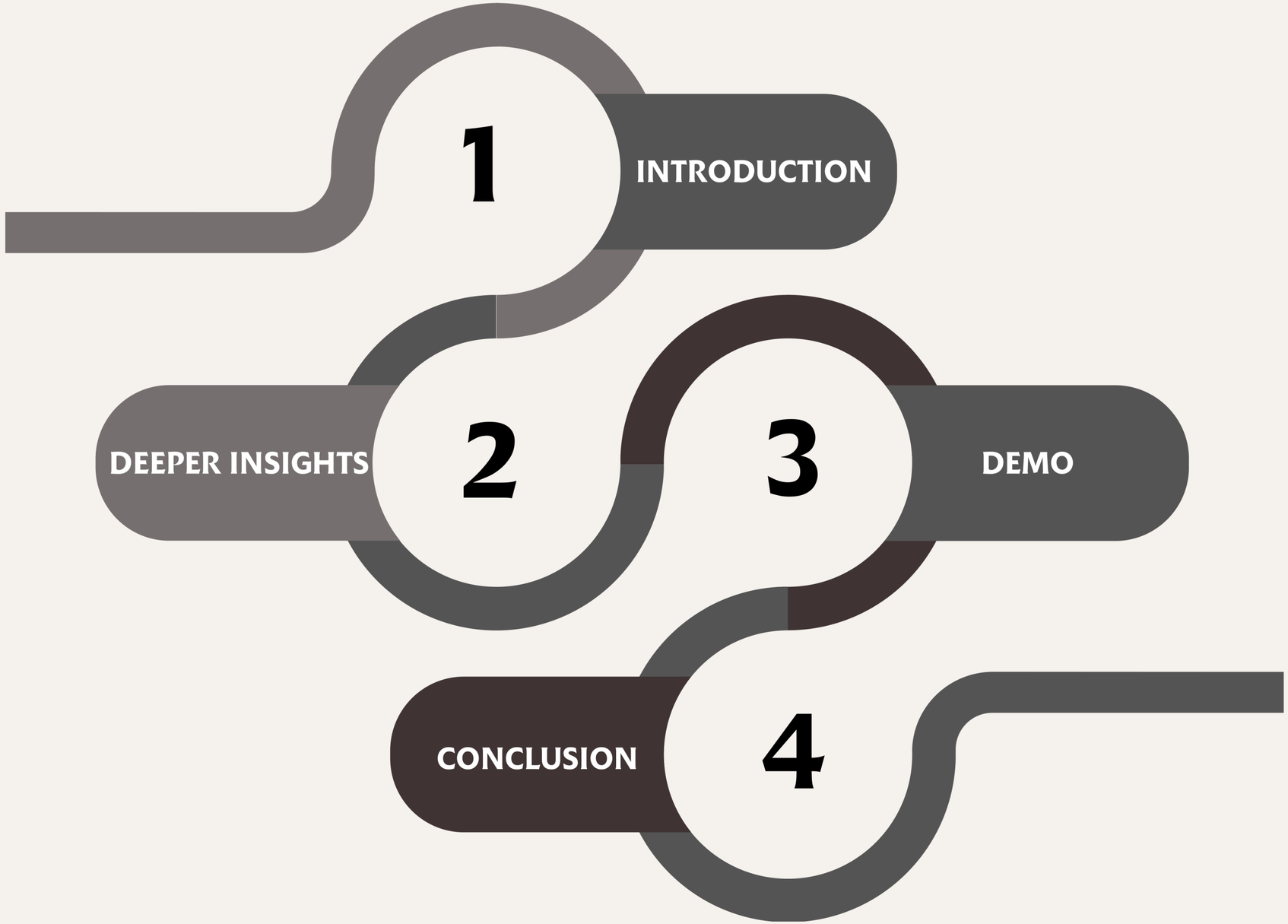
Group KHMT2 - 09

21127012 - Trần Huy Bân

21127143 - Nguyễn Minh Quân

21127240 - Nguyễn Phát Đạt

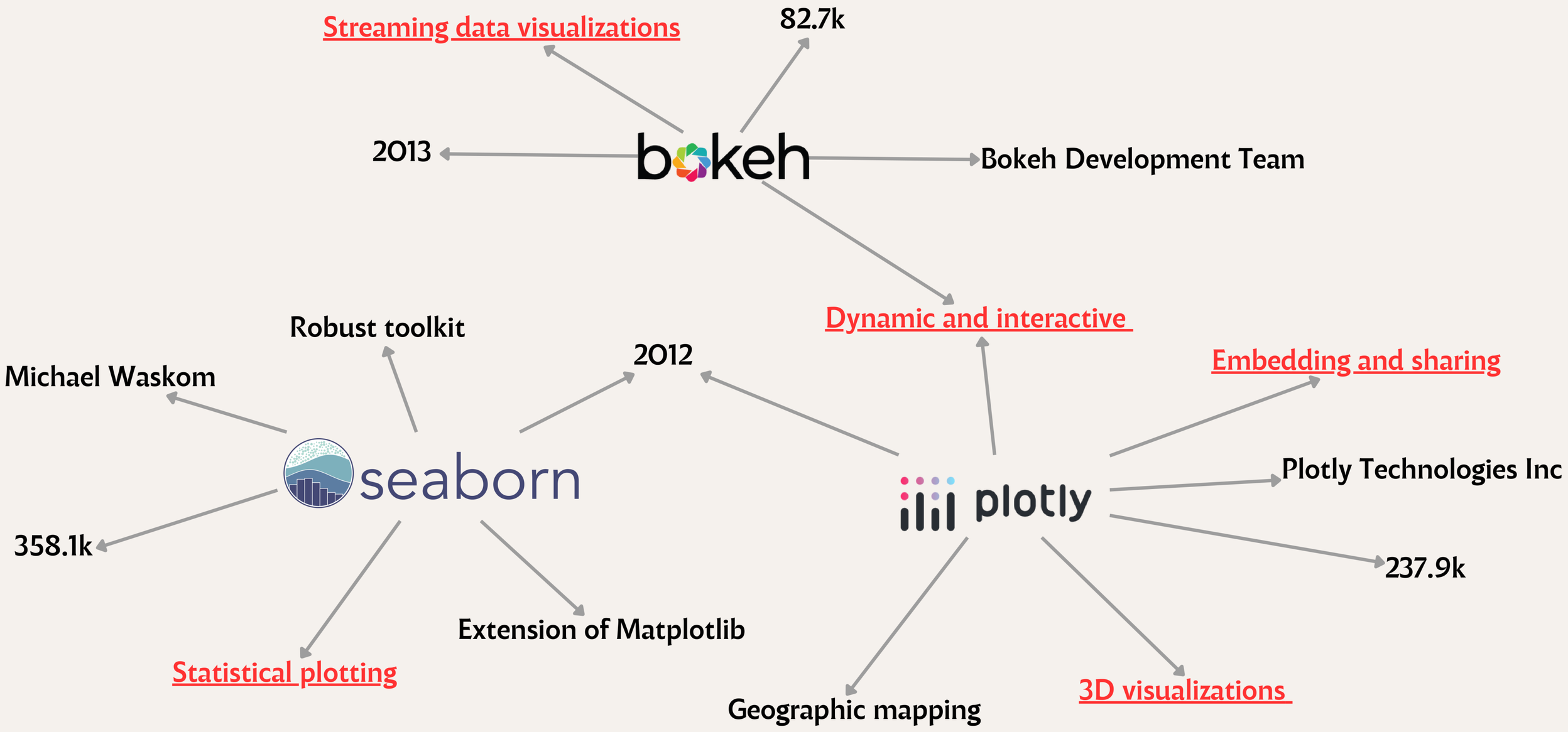
21127507 - Đinh Công Huy Hoàng


1
INTRODUCTION
DEEPER INSIGHTS
2
3
DEMO
CONCLUSION
4

INTRODUCTION

WHICH PROBLEMS COULD BE SOLVED ?

- Data Exploration and Understanding
- Identifying Outliers and Anomalies
- Performance Monitoring
- Feature Engineering
- Communication and Collaboration
- Scalable Visualization


Streaming data visualizations
82.7k
2013
bokeh
Bokeh Development Team
Dynamic and interactive
Robust toolkit
2012
Embedding and sharing
Michael Waskom
seaborn
plotly
Plotly Technologies Inc
358.1k
237.9k
Extension of Matplotlib
Statistical plotting
Geographic mapping
3D visualizations

DEEPER INSIGHTS

seaborn

MAIN COMPONENTS

Figure and Axes

- **The figure** is the entire canvas or window for the plot.
- **The axes** are the subplots or regions where the data is plotted.

Color palette and themes

- **Allow** users to **customize** plot colors, enhancing the visual appeal of charts.

Plot type

- **Offer** a **diverse range** of plot types, catering to various visualization needs and data representations.

Data mapping

- **Represent** different aspects of the data in the plot.

MAIN FUNCTIONALITIES

Statistical plots

Offers a variety of statistical relationships visualizations

Flexible Aesthetic

Provides several built-in themes and color palettes

High-level interface

Simple and intuitive API

Multi-plots

Organize and display data efficiently across multiple plots.

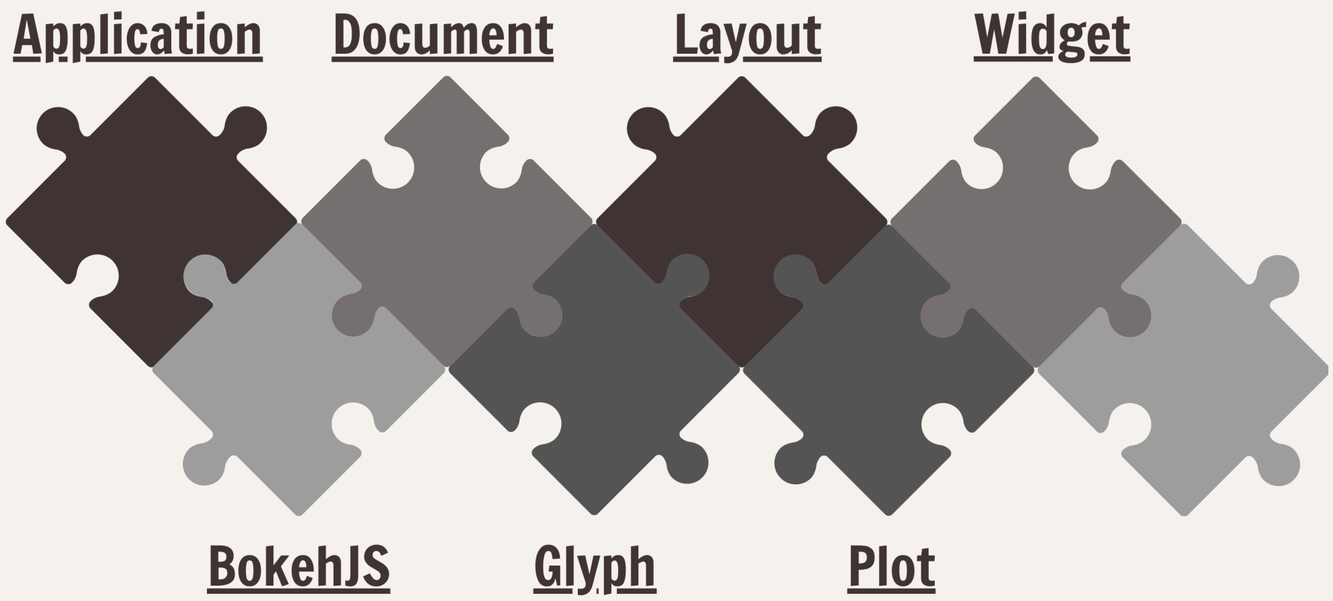
bokeh

Refer to a **set of instructions** for generating Bokeh documents.

Organize data and models
Enable the **creation of interactive** visualizations and applications

Multiple nested rows and columns of plots and widgets for a **cohesive structure**.

Diverse **user interface components** such as sliders, drop-down menus, or buttons.



Client-side JavaScript library for rendering visuals and managing interactions.

API objects for drawing vectorized graphics representing data, such as lines, rectangles, squares, wedges, and circles.

Containers encompassing **renderers, glyphs, and annotations**, forming complete visualizations.

MAIN FUNCTIONALITIES

Interactive dashboards

Excels in creating interactive and responsive dashboards

Web-based visualizations

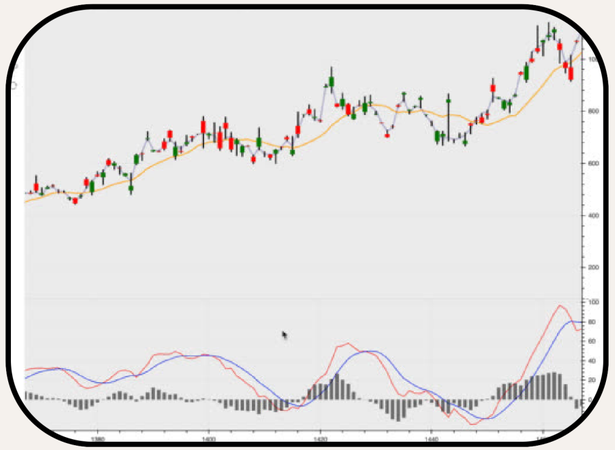
Can be easily shared and accessed through web browsers

Streaming Data Support

Useful for building applications with dynamic and real-time updates

Integration with Jupyter Notebooks

Enhances the workflow within this environment



plotly

MAIN COMPONENTS

Figure

- **Container** for plots, holding traces and layout details like titles, axes, and annotations.

Trace

- Represent a single set of data along with the **graphical representation** of that data.
- Can be **customized** independently.

Layout

- Visual **properties** of the plot
- Layout settings are applied to the entire plot and can be **customized**

MAIN COMPONENTS

Graph Object (GO)

- **Objects** represent the different types of plots, such as scatter plots, line plots, bar charts, histograms, heatmaps, and more.

Express

- High-level **interface** for Plotly that provides a **simplified API** for creating a wide range of interactive visualizations.
- Built on top of **Plotly's Graph Objects**.

Annotations

- **Text** or **shapes** to provide additional information or highlight specific data points.

MAIN FUNCTIONALITIES

Various Plots

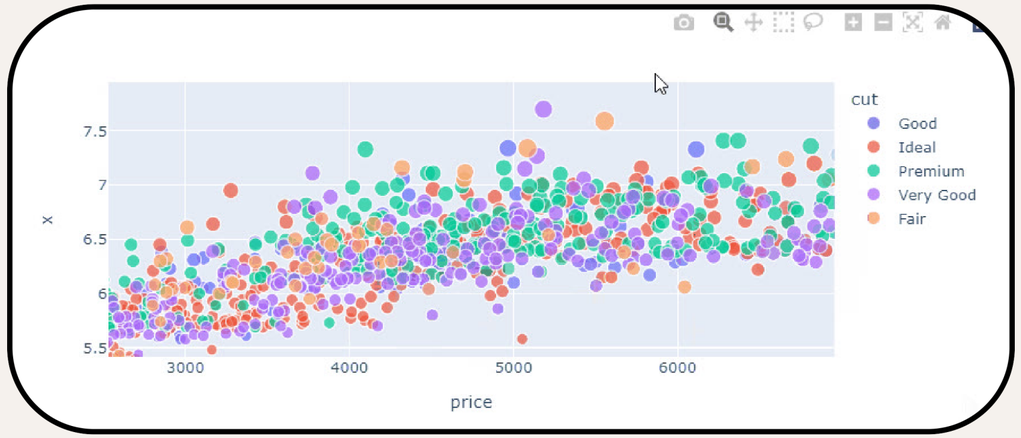
More than 40 type of plots

Customization

Offers extensive customization options

Interactivity

Display tooltips, zooming and panning, toggling visibility



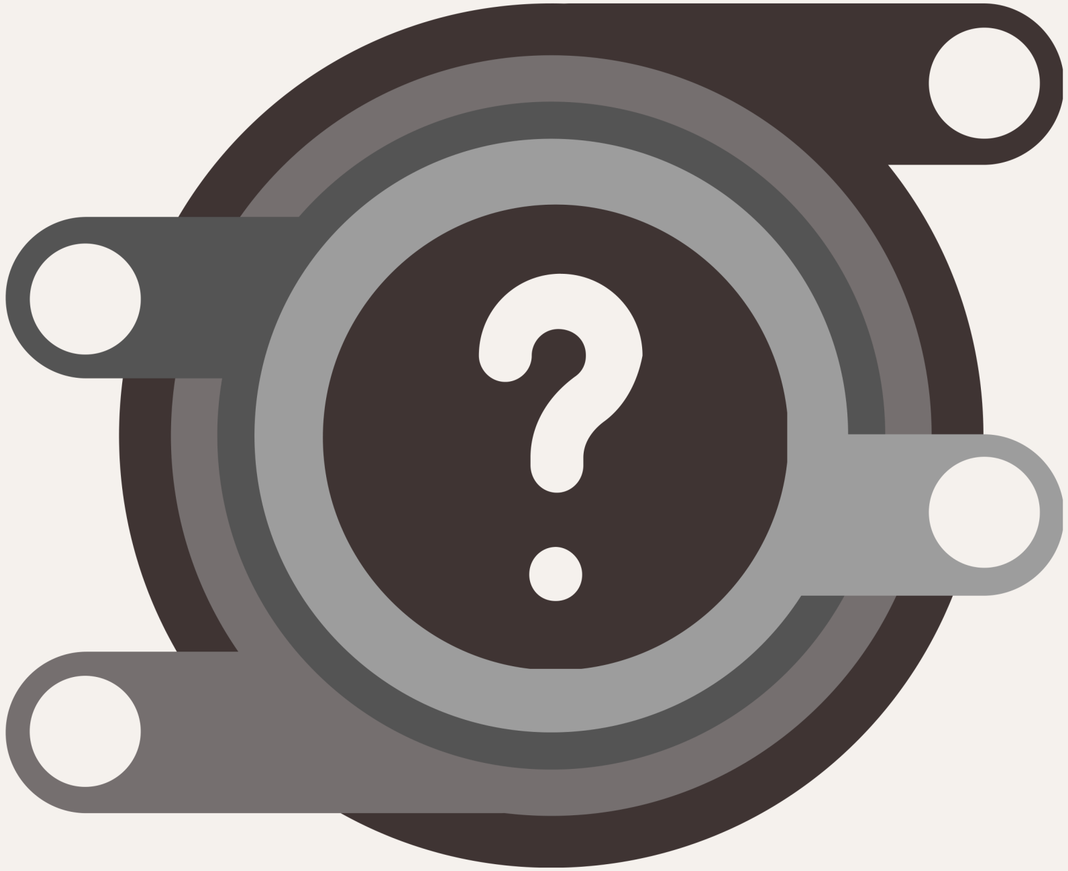
PROBLEMS

OVERSIMPLIFICATION

Presenting complex data in a manner that makes it loses important nuances, insights, or context.

DATA HANDLING

Processing, cleaning, transforming, and preparing data for effective visualization



MISSING DATA

Each tool may deal with missing data differently.

TOO MUCH INFORMATION

Overwhelming viewers with an excessive amount of data or details

DEMONSTRATION

COMPARISON

Matplotlib	Seaborn	Plotly	Bokeh	GGplot	Altair	Pygal
- Graphical representation types supported. - Compatible with NumPy arrays and the broader SciPy stack. - Offers diverse plots for trend analysis and correlations.	- Maps and aggregates data for visuals. - Enhances data exploration and understanding. - User-friendly interface for creating informative graphics.	- Robust API works in local and web browser modes. - Interactive, open-source, high-level visualization library. - Viewable in Jupyter notebooks, HTML files, or online.	- Offers intuitive graphs for versatile solutions. - Renowned for crafting custom visualizations. - Interactive, open-source, high-level visualization library.	- User-friendly visualization creation with enhanced representations. - Integrated with Pandas for data storage. - Based on ggplot2 from R for robust plotting capabilities.	- User-friendly API based on Vega-lite JSON spec. - Source available on GitHub. - Dependencies include Python 3.6, entry points, jsonschema, NumPy, Pandas, Toolz.	- Three map packages maintain compact module size. - Provides interactive data exploration and filtration. - Optimized with rich support for visionary solutions in complex problems.

Matplotlib	Seaborn	Plotly	Bokeh	GGplot	Altair	Pygal
+ Widely used. + Versatile library. + Integrates well with many flatforms.	+ Best for statistical plots. + Simplifies complex plotting tasks	+ Allows 40 unique chart and plot types. + Create interactive plots easily.	+ Highest level of control. + Interactive plots. + High performance	+ Wide range of default themes and aesthetics for visually appealing plots.	+ Minimal code. + Declarative grammar on both visuals and interaction.	+ Provide variety of output types: like SVGs. + Attractive chart in a few code lines.
- Must manually configure most parts. - Require more coding.	- Limited customizable options (less flexibility).	- Limited features in the free version.	- No pre-set defaults. - Requires understanding of JavaScript for advanced customization (BokehJS).	- Not suitable for creating highly customized graphics.	- Limited customization options. - Smaller community and available resources.	- Limited interactivity. - Not suitable for large data.

EVALUATION

- Community support and documentation.
- Compatibility with widely used Data Tools.
- Open-source nature.
- Industry adoption.
- Easy to understand and use.

- Performance optimization for Big Data.
- Data cleaning and preprocessing.
- Limited documentation for Big Data use cases.

QUESTION AND ANSWER

Thank You