Interactive Data Visualization with Plotly

Data Visualization Course – Lecture 5

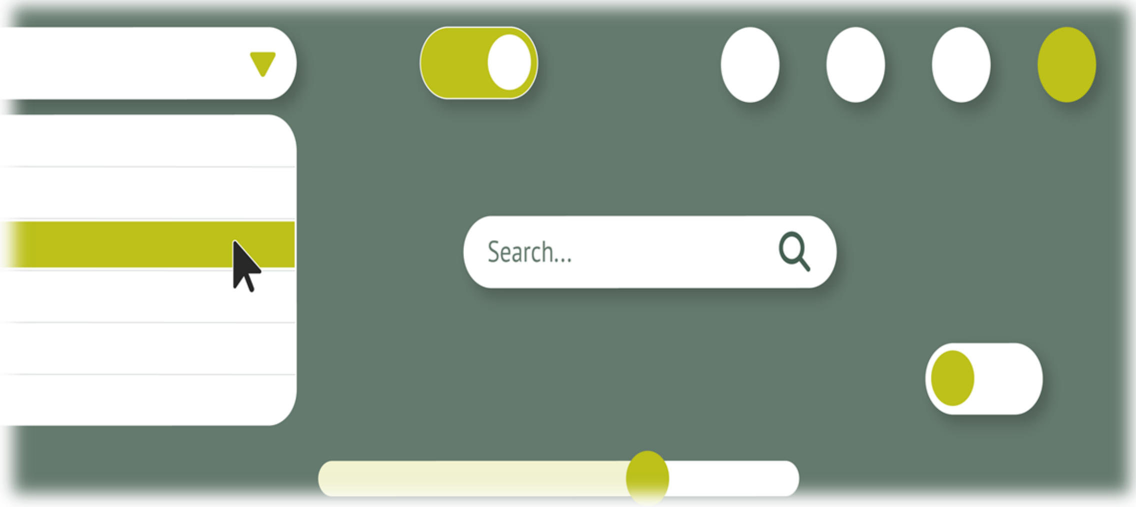
Dr. Muhammad Sajjad

R.A: Imran Nawar

December 2024

Overview

- Introduction to Interactive Data Visualization
  - What is Interactive Data Visualization?
  - Why use Interactive Data Visualization?
  - Benefits and Use Cases
- Getting Started with Plotly
  - Installation and Setup
  - Plotly's Architecture in Python
  - Basic Plot Types (Line, Scatter, Bar)
- Advanced Features of Plotly
  - Animations and Transitions
  - Interactive Widgets
  - Multi-axis and subplots
- Customizing Plotly Visualizations
  - Layout and Styling (titles, labels, legends, themes and templates)
  - Adding Interactivity (hover effects, zooming)
- Comparing Plotly with Other Visualization Libraries
- Future Trends in Interactive Visualization
- **Coding**: Creating Interactive Visualizations



Introduction to Interactive Data Visualization

What is Interactive Data Visualization?

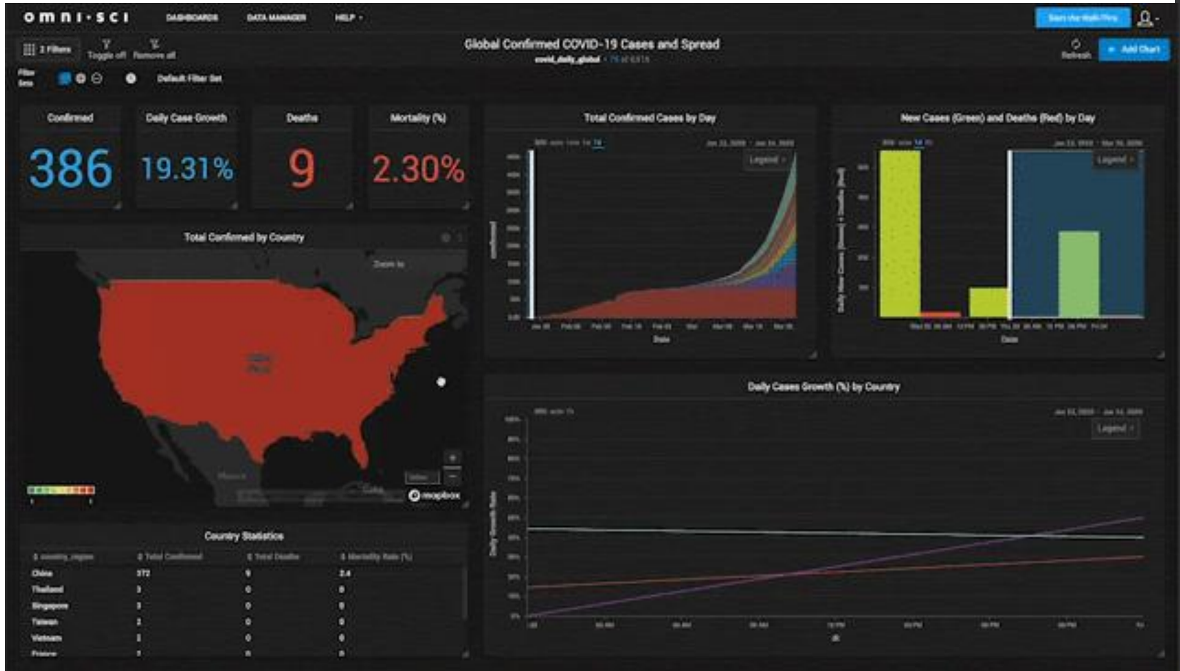
- A type of data visualization that allows users to interact with data in order to explore and understand it.
- It uses tools to create visual data representations that users can explore and analyze interactively.
- Helps users gain deeper insights by interacting with data directly.

- Interactive visualizations enhance data exploration by enabling users to dynamically engage with and manipulate data through actions like zooming, filtering, and hovering.

Why Use Interactive Data Visualization?

- Allows dynamic exploration and manipulation of data, rather than static views.
- Supports actions like zooming, filtering, and comparing to discover patterns, trends, and relationships.
- Enhances the ability to uncover insights and outliers.

HEAVY.AI's Interactive Data Visualization of Global Confirmed COVID-19 Cases and Spread.



Country Statistics

country_region	Total Confirmed	Total Deaths	Mortality Rate (%)
China	372	9	2.4
Thailand	3	0	0
Singapore	3	0	0
Taiwan	2	0	0
Vietnam	2	0	0
France	1	0	0

Introduction to Interactive Data Visualization

Benefits:

- **Enhanced Understanding:** Allows users to explore data patterns and trends interactively.
- **Improved Decision Making:** Provide quick insights through real-time interaction.
- **Engagement:** Keeps the audience actively involved and focused.
- **Customization:** Users can filter, sort, and adjust views to highlight specific insights.

Enhanced Understanding

Improved Decision Making

Engagement

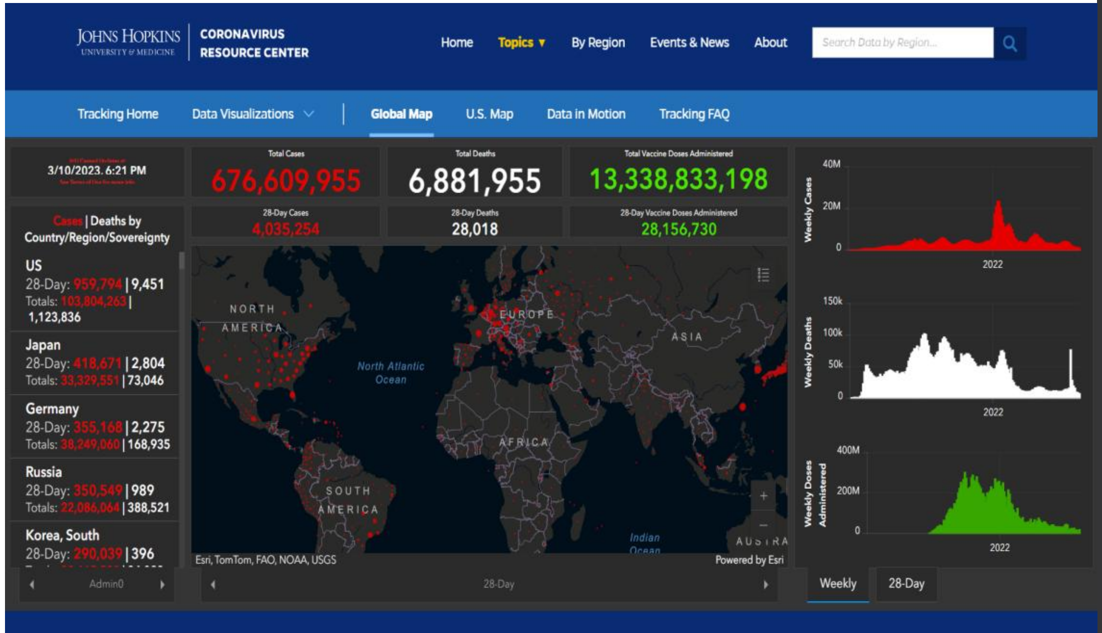
Customization

Use Cases (Real-World Applications):

- **Business Analytics**: Sales dashboards, performance tracking.
- **Finance**: Stock market trends, risk assessment.
- **Education**: Interactive learning tools, research data presentations.
- **Healthcare**: Patient data visualization, epidemiological tracking.

Real-World Example:

- **COVID-19 Dashboard by Johns Hopkins University:** Interactive maps and charts tracking the pandemic's progression.



3/10/2023. 6:21 PM

Total Cases 676,609,955

Total Deaths 6,881,955

Total Vaccine Doses Administered 13,338,833,198

28-Day Cases 4,035,254

28-Day Deaths 28,018

28-Day Vaccine Doses Administered 28,156,730

Cases | Deaths by Country/Region/Sovereignty

US
28-Day: 959,794 | 9,451
Totals: 103,804,263 | 1,123,836

Japan
28-Day: 418,671 | 2,804
Totals: 33,329,551 | 73,046

Germany
28-Day: 355,168 | 2,275
Totals: 38,249,060 | 168,935

Russia
28-Day: 350,549 | 989
Totals: 22,086,064 | 388,521

Korea, South
28-Day: 290,039 | 396

Getting Started with Plotly

What is Plotly?

- A powerful open-source graphing library for Python, R, and JavaScript.
- Creates interactive, publication-quality visualizations.
- Supports a wide range of chart types and interactive features.
- Enables Python users to create stunning web-based visualizations.
- Integrates seamlessly with web applications and Jupyter notebooks.
- Visualizations can be saved as standalone HTML files or displayed directly in Jupyter notebooks.

Installation and Setup

- Install Plotly using pip.

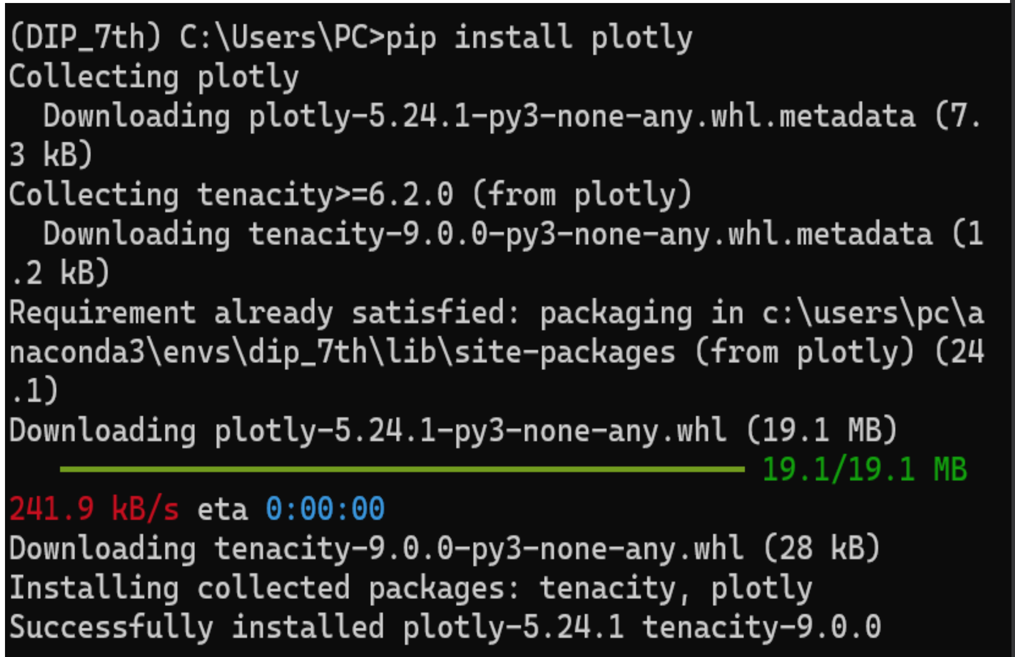
```
pip install plotly
```

- For notebook environments, install additional package *notebook* if required:

```
pip install “notebook>=5.3” “ipwidgets>=7.5”
```

- Basic usage in Python:

```
import plotly.graph_objects as go
import plotly.express as px
```



```
(DIP_7th) C:\Users\PC>pip install plotly
Collecting plotly
  Downloading plotly-5.24.1-py3-none-any.whl.metadata (7.
3 kB)
Collecting tenacity>=6.2.0 (from plotly)
  Downloading tenacity-9.0.0-py3-none-any.whl.metadata (1
.2 kB)
Requirement already satisfied: packaging in c:\users\pc\a
naconda3\envs\dip_7th\lib\site-packages (from plotly) (24
.1)
Downloading plotly-5.24.1-py3-none-any.whl (19.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 19.1/19.1 MB
241.9 kB/s eta 0:00:00
Downloading tenacity-9.0.0-py3-none-any.whl (28 kB)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.24.1 tenacity-9.0.0
```

Plotly's Architecture in Python

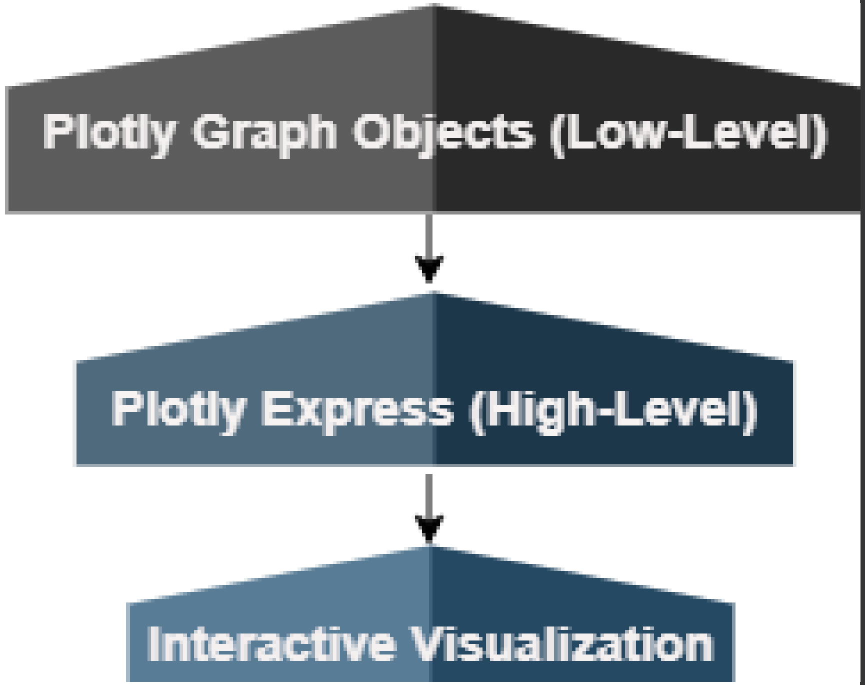
Plotly has a modular architecture, offering flexibility in creating interactive visualizations.

- Two primary interfaces:
  1. **Plotly Express**
  2. **Plotly Graph Objects**.

Plotly Express

- A high-level interface for quick and easy plotting.
- Built on top of Plotly Graph Objects.

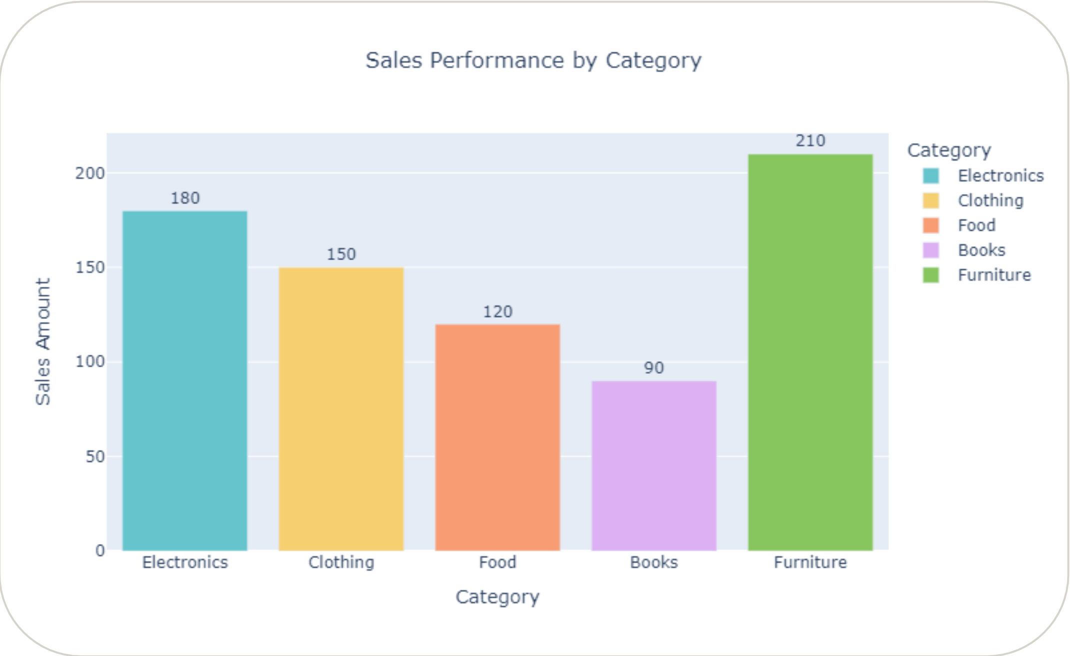
- **Features of Plotly Express**:
  - **Simplicity and Ease of Use**: Create plots with concise and straightforward syntax.
  - **Quick Visualization**: Ideal for exploratory data analysis , enabling easy creation of scatter plots, bar charts, and line plots with just a few lines of code.
  - **Automation**: Automatically manages plot elements such as axis labels, titles, and legends, allowing users to focus on analysis.


Plotly Graph Objects (Low-Level)
Plotly Express (High-Level)
Interactive Visualization

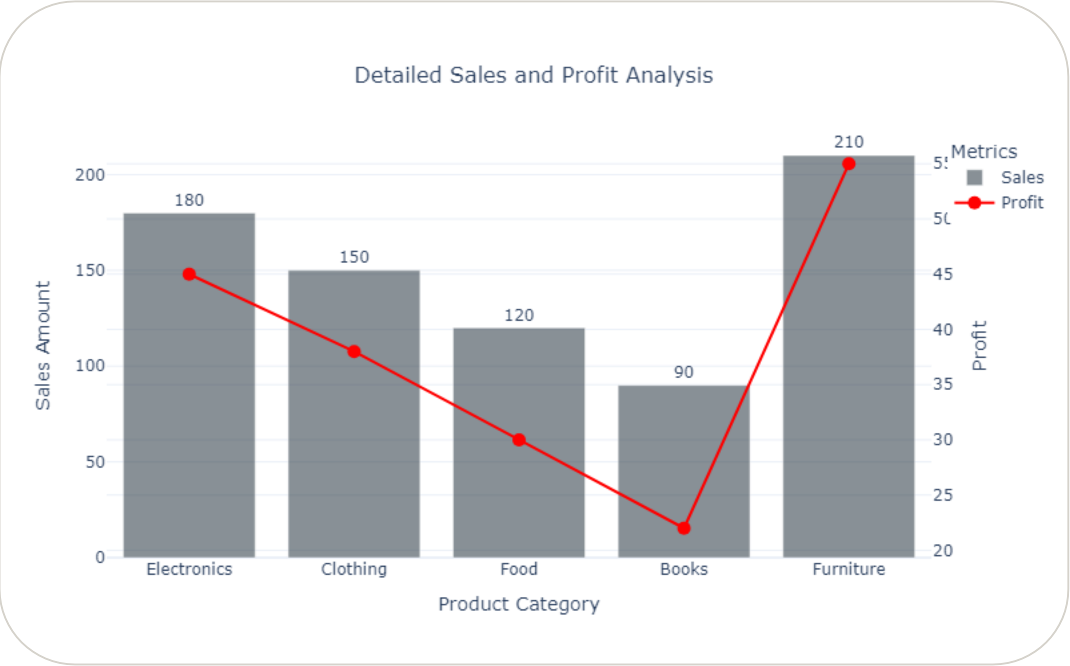
Plotly’s Architecture in Python

Plotly Graph Objects

- A lower-level interface for advanced and detailed customization.
- Provides full control over every aspect of the plot, making it suitable for complex, highly customized visualizations.



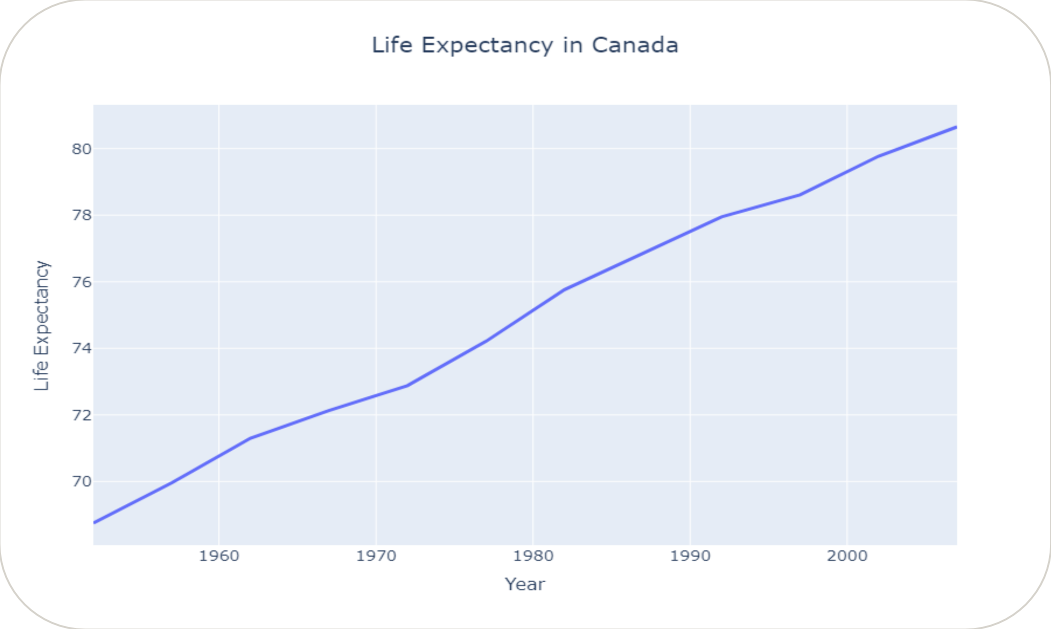
- **Features of Plotly Express**:
  - **Detailed Customization:** Offers fine-grained control over trace properties, layouts, and annotations.
  - **Complex Plotting:** Combines multiple traces and layers to create sophisticated visualizations.
  - **Manual Configuration:** Requires manual setup of plot structure, offering greater flexibility for complex visualizations.



Basic Plot Types

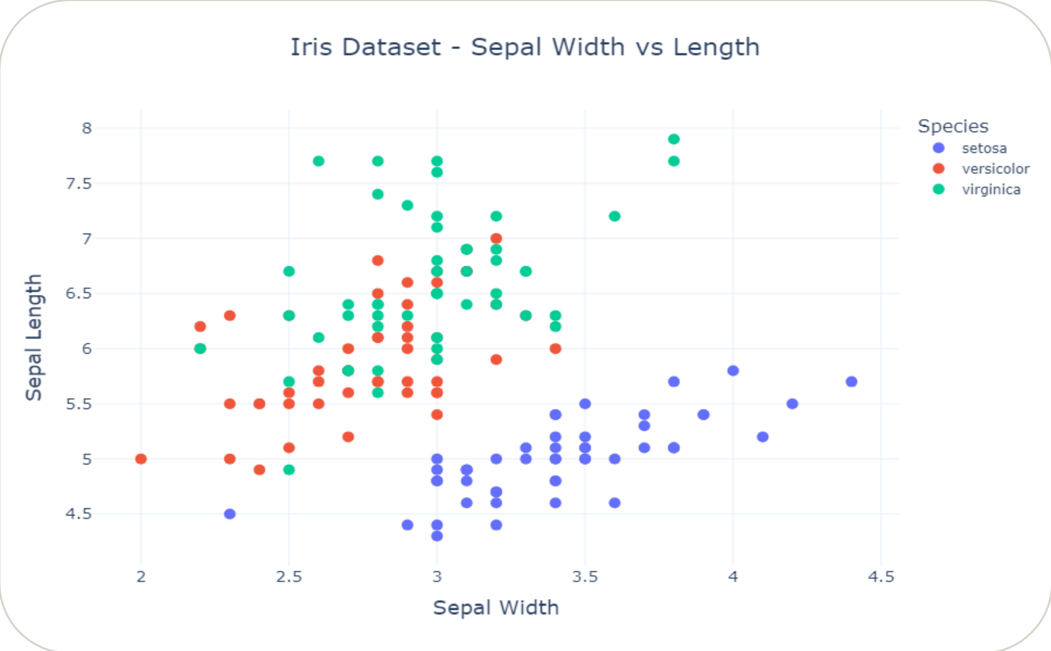
1. **Line Plots:**
   - Show trends over time
   - **Example**:
     - Showing life expectancy in a country
     - Stock prices over time.

```
import plotly.express as px

df = px.data.gapminder().query(“country==‘Canada‘”)
fig = px.line(df, x=“year”, y=“lifeExp”, title='Life
expectancy in Canada')
fig.show()
```



2. **Scatter Plots:**
   - Show the relationship between two variables.
   - **Example:**
     - Height vs Weight of individual

```
import plotly.express as px

df = px.data.iris()
fig = px.scatter(df, x=“sepal_width”, y=“sepal_length”,
color=“species”,
          title=“Iris Dataset – Sepal Width vs
Length”)
fig.show()
```

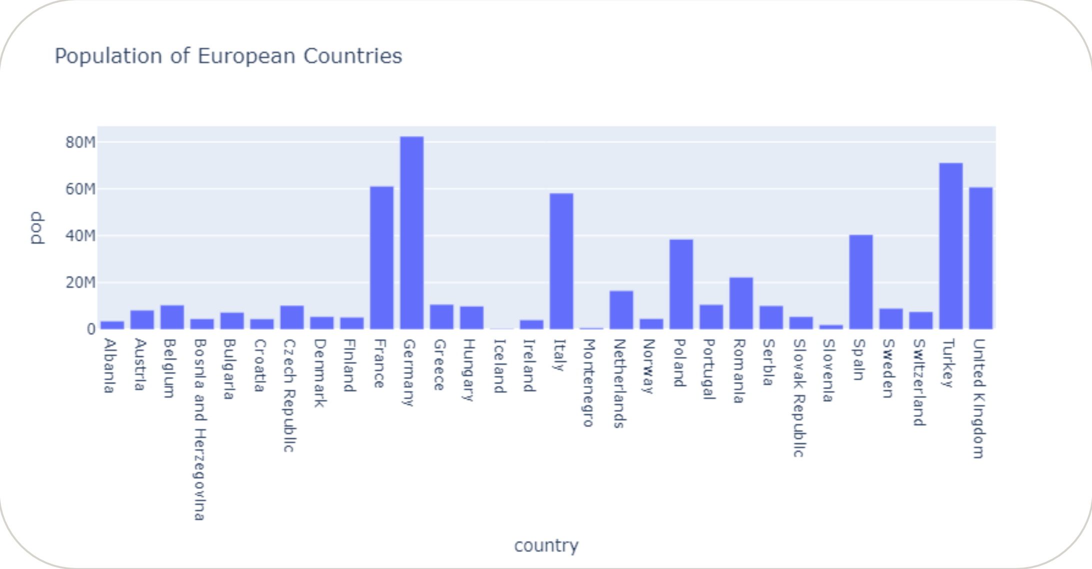


Basic Plot Types

3. Bar Charts:

- Comparing quantities across different categories.
- **Example**:
  - Sales per region.

```
import plotly.express as px

df = px.data.gapminder().query(“year == 2007”).query(“continent
== ‘Europe’”)
fig = px.bar(df, x=“country”, y=“pop”, title="Population of
European Countries")
fig.show()
```



Real-World Examples

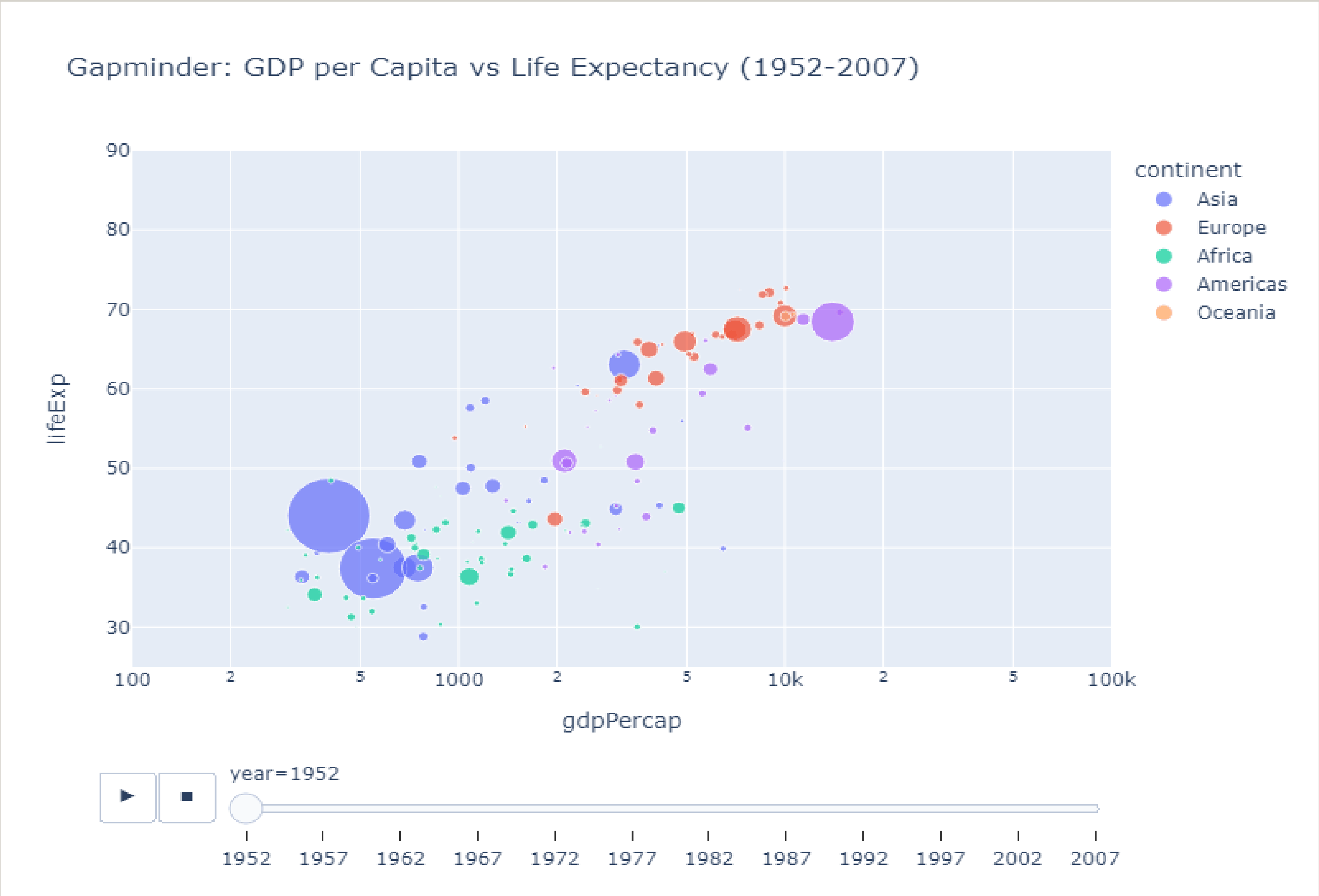
- **Line Chart:** Tracking website traffic over a month.
- **Scatter Plot:** Analyzing the correlation between advertising spend and sales.
- **Bar Chart:** Comparing product performance across different markets.

Advanced Features of Plotly

➢ Animations and Transitions

- **Dynamic Visualizations**: Bring data to life by showcasing changes over time or across different states.
- **Applications**: Highlight trends, growth patterns, or evolving relationships (e.g., population growth, financial performance, or climate change).
- **Example**: Animated bubble charts (Gapminder-style) showing global development indicators like GDP and life expectancy.

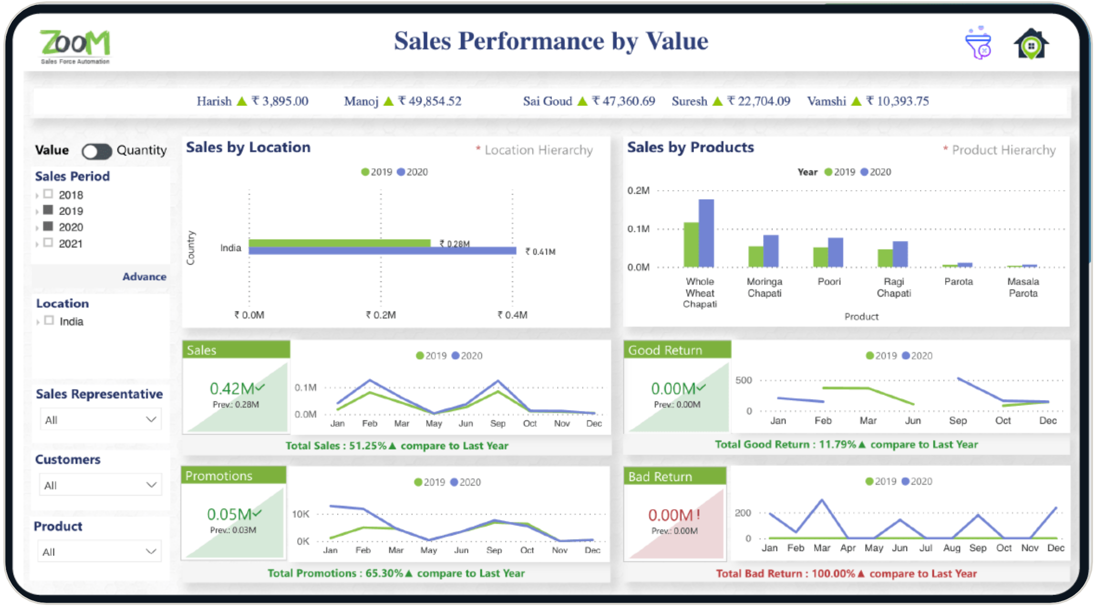
This animated scatter plot shows the relationship between GDP per capita and life expectancy (1952-2007). Bubble size represents population, with colors indicating continents for global comparisons.



Advanced Features of Plotly

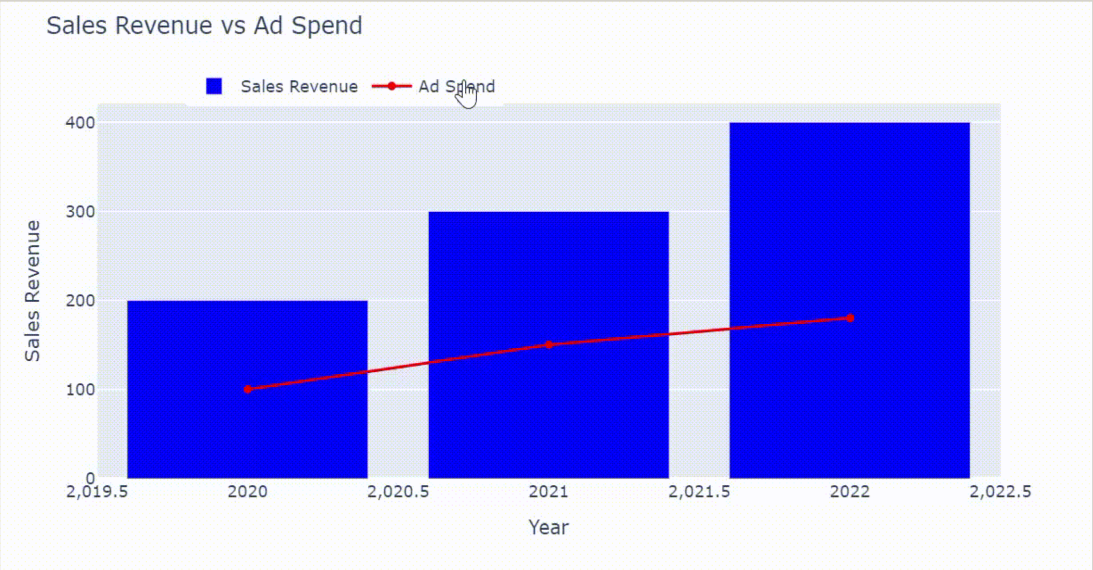
Interactive Widgets

- **Custom Interactivity**: Enhance user experience with sliders, dropdowns, and buttons for tailored data exploration.
- **Applications**: Enable filtering, zooming, and dynamic customization of plots without page reloads.
- **Example**: A dashboard where users can explore sales performance by region, time, or product category.



Multi-Axis and Subplots

- **Complex Layouts**: Create advanced visualizations by combining multiple plots or adding secondary axes for deeper analysis.
- **Applications**: Compare diverse metrics or overlay datasets with different scales in a single view.
- **Example:** A dual-axis chart showing sales revenue (bar plot) and ad spend (line plot) to analyze their relationship over time.



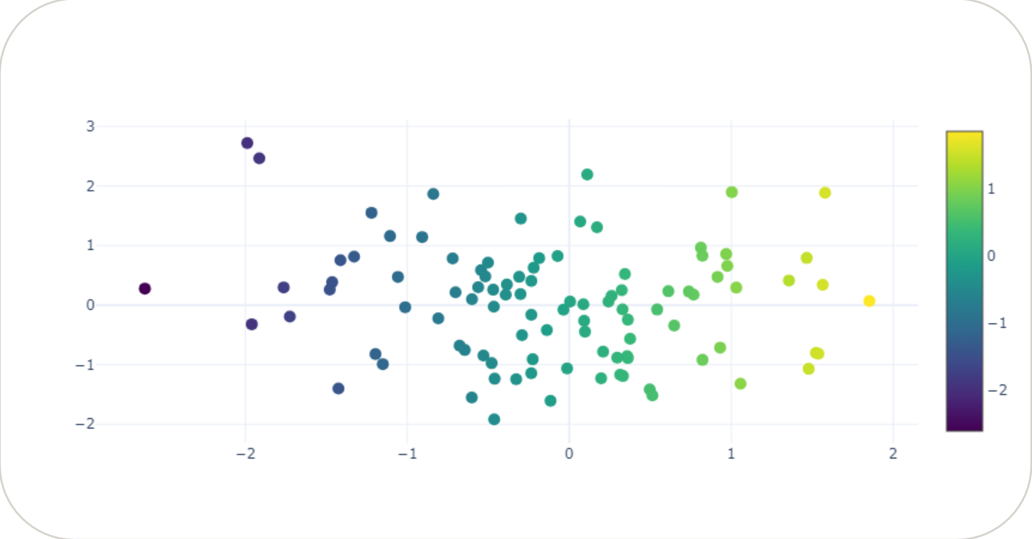
Customizing Plotly Visualizations

1. **Layout and Styling**
   - Plotly offers extensive customization options for layouts and styles.
   - Add margins for spacing, include descriptive titles, and customize axis properties.
   - **Example**: Customize a scatter plot by adding margins, titles, and axis labels to improve readability.

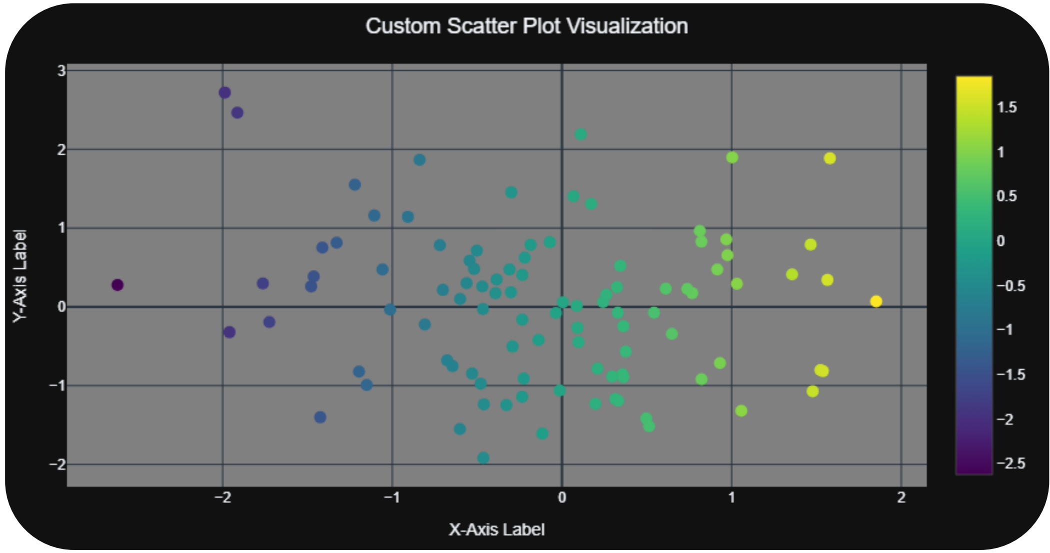
2. **Titles, Labels, and Legends**
   - **Improve Readability**:
     - Use clear, descriptive titles and axis labels.
     - Position legends strategically to avoid visual clutter.

3. **Themes and Templates**
   - Utilize Plotly's pre-defined themes for cohesive visuals.
   - **Example:**
     - `plotly_dark` is ideal for dark-mode visualizations.

Without Customization

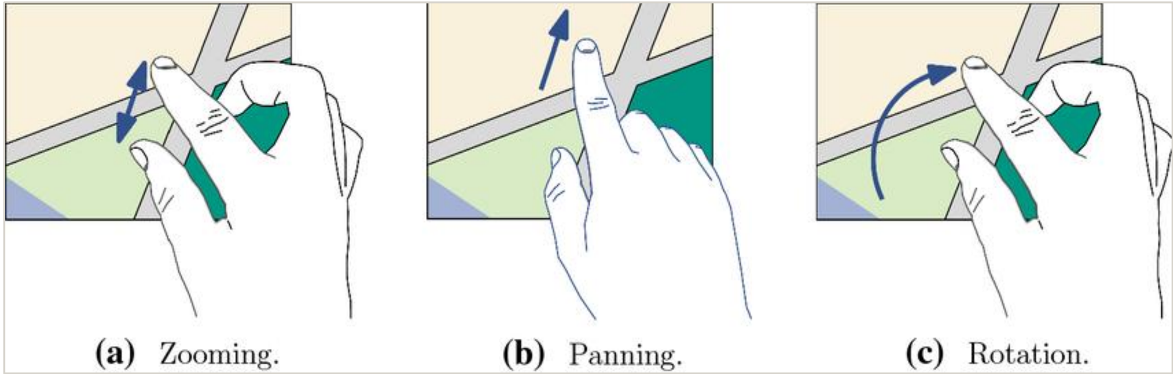


With Customization



Customizing Plotly Visualizations – Adding Interactivity

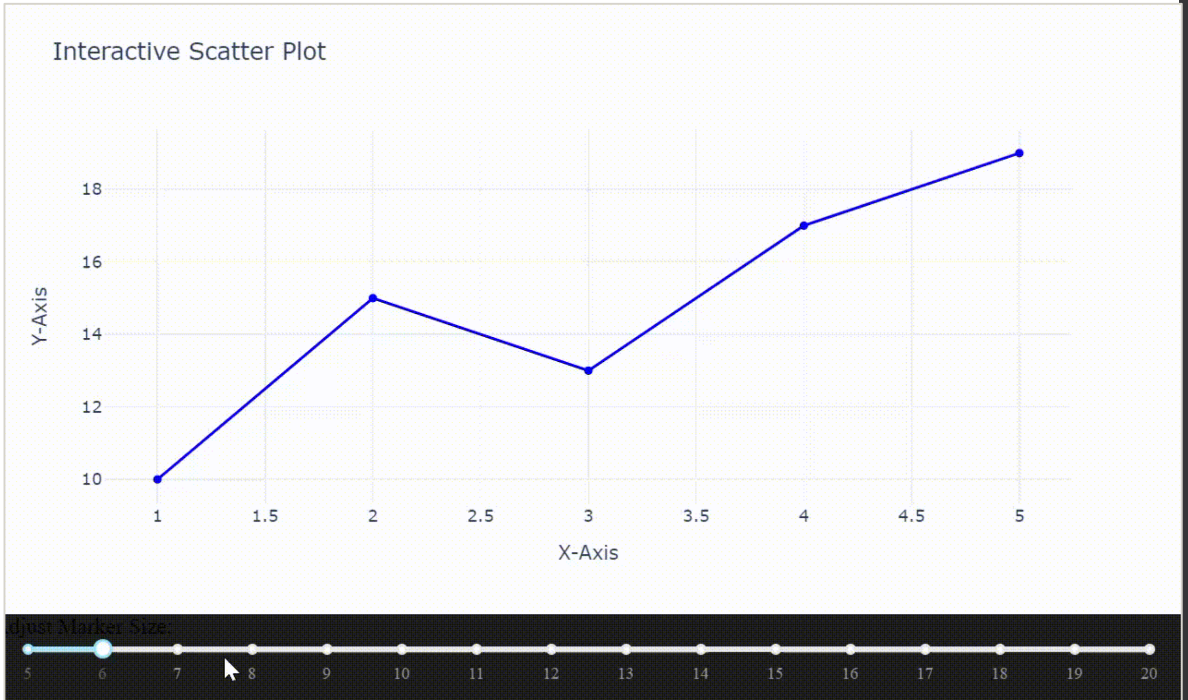
1. **Hover Effects**
   - **Purpose:** Provide contextual information dynamically when hovering over data points.
   - **Example:** Show precise values or annotations on a scatter plot.



(a) Zooming. (b) Panning. (c) Rotation.

2. **Zooming**
   - Enable users to zoom into specific data ranges for detailed exploration.

3. **Interactive Widgets and Callbacks**
   - Use Plotly Dash to incorporate interactive elements such as dropdown menus, sliders, and buttons.
   - Configure callbacks to filter, update, or transform data interactively.



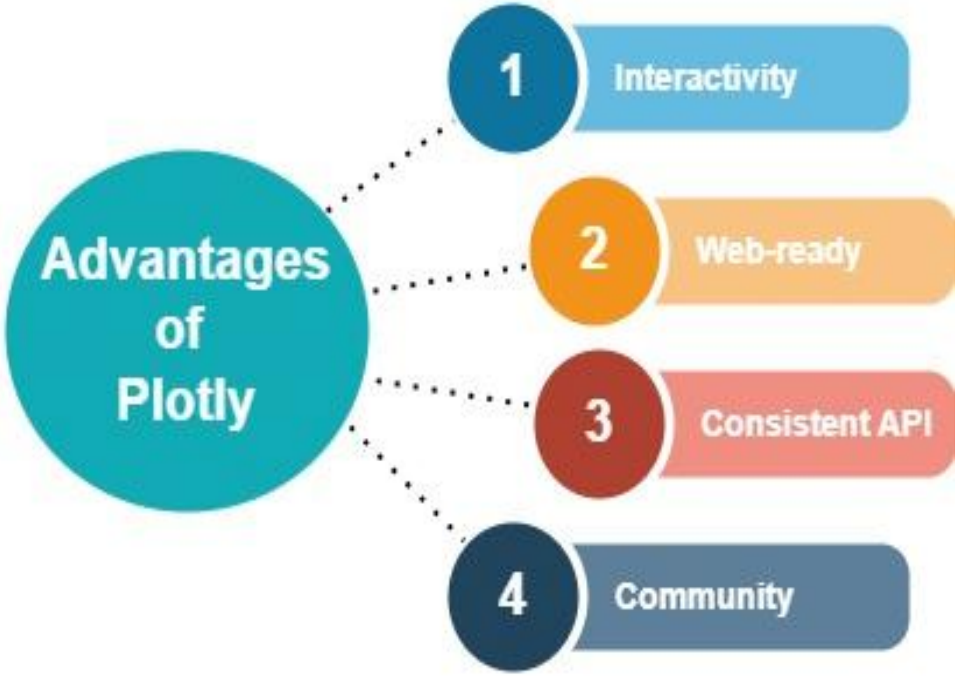
Interactive Scatter Plot

djust Marker Size:

Comparing Plotly with Other Visualization Libraries

Advantages of Plotly

- **Interactivity**: Provides interactive visualizations out of the box.
- **Web-ready**: Easily embed visualizations into web applications.
- **Diverse chart types**: Supports a wide range of chart types, including 3D and statistical plots.
- **Consistent API**: Offers a uniform API across Python, R, and JavaScript
- **Community and documentation**: Strong community support and extensive documentation.



Limitations of Plotly

- **Performance**: Can be slower and resource-intensive with very large datasets.
- **Learning curve**: Syntax is more complex compared to simpler libraries like Seaborn.
- **Dependency**: Relies heavily on JavaScript, which may restrict usage in certain environments or projects.

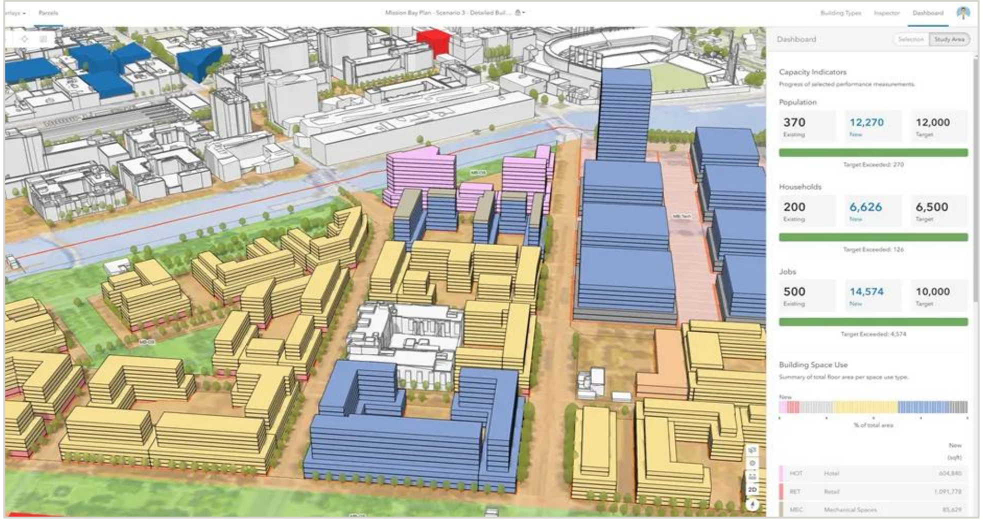
Comparing Plotly with Other Visualization Libraries

Library	Strengths	Limitations
Matplotlib	• Highly customizable, ideal for static plots.	• Lacks interactivity, steep learning curve.
Seaborn	• Simplifies statistical visualizations.	• Limited interactivity, fewer chart types.
Plotly	• Best for interactive, web-ready visualizations.	• Slower for large datasets, reliant on JavaScript
Bokeh	• Good for interactive web-based plots.	• Smaller community, less documentation than Plotly

Future Trends in Interactive Visualization

Integration with AI and Machine Learning

- Interactive visualizations will increasingly incorporate **predictive analytics**, **real-time data processing**, and **AI-driven insights.**
    - **Example:** Dashboards that update in real-time to predict sales trends, such as forecasting seasonal demand dynamically.



Dashboard

Capacity Indicators

Progress of selected performance measurements.

Population

370	12,270	12,000
Existing	New	Target

Target Exceeded: 270

Households

200	6,626	6,500
Existing	New	Target

Target Exceeded: 126

Jobs

500	14,574	10,000
Existing	New	Target

Target Exceeded: 4,574

Building Space Use

Summary of total floor area per space use type.

New

% of total area

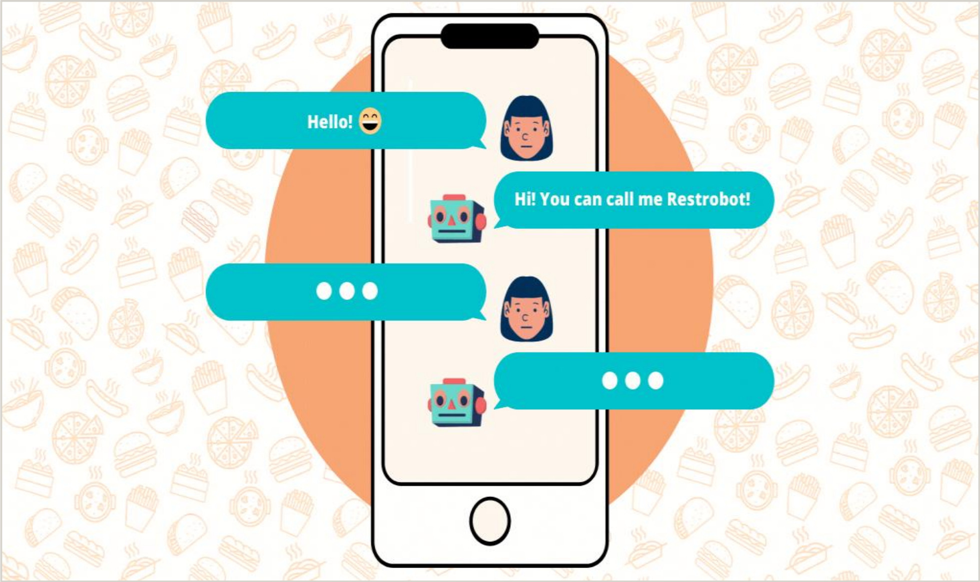
		New (sqft)
HOT	Hotel	604,840
RET	Retail	1,091,778
MSC	Mechanical Spaces	85,629

Virtual and Augmented Reality

- Engaging 3D environments will allow users to **explore data spatially**, creating deeper and more intuitive insights.
    - **Example:** A city planner using AR to overlay traffic and pollution data on a real-world map for optimized urban development.

Natural Language Interfaces

- The next generation of tools will enable users to interact with visualizations using **voice or text-based commands**, making data exploration simpler.
    - **Example:** Asking, “Show me last year’s profit growth by region” to generate an interactive heatmap instantly.



Summary and Next Steps

- **Conclusion**
  - Interactive data visualization empowers users to explore, analyze, and present data in a more dynamic and engaging way. Plotly, with its diverse features and interactivity, is a powerful tool for creating visually appealing, web-ready visualizations.
- **Key Takeaways:**
  - **Interactive Data Visualization** enables users to gain deeper insights and engage with data through real-time interactions, enhancing decision-making.
  - **Plotly** is a versatile library that provides interactive, web-ready visualizations, with a strong emphasis on ease of use and customization**.**
  - **Key Plotly features** include animations, interactive widgets, multi-axis charts, and extensive customization options for layouts, styling, and interactivity.
  - **Future trends** include integration with AI, immersive VR/AR environments, and natural language interfaces to further enhance user interaction with data
- **Next Lecture:** Image Data Visualization.

Thank You