**VISUALIZATION**

Visualization is the graphical representation of information and data. It uses charts, graphs, maps, and other visual tools to help people understand trends, patterns, and outliers in data.

Visualization in Data Science = Seeing the data story clearly.  
It turns raw numbers into meaningful visuals, helping both data scientists and non-technical users gain insights and take action.

Here's a comparison between Matplotlib and Plotly, two popular Python libraries used for data visualization:

**MATPLOTLIB**

Matplotlib is a comprehensive, open-source plotting library for Python that provides tools for creating static, animated, and interactive visualizations. It's widely used in data analysis and scientific research due to its flexibility and integration with NumPy, pandas, and other libraries.

**Key Features:**

* It Produces high-quality 2D graphics (and some 3D with mpl\_toolkits).
* Matlab-like syntax via pyplot, which is intuitive for those with MATLAB experience.
* Matplotlib provides full control over plot elements (axes, labels, grid, legends).
* Supports various backends for rendering.
* Integrates with Jupyter notebooks and GUI toolkits (Tkinter, PyQt, etc.).
* It also has customizable themes and styles (plt.style.use('ggplot'), etc.)

**Typical Use Cases:**

* Data exploration
* Report visualizations
* Scientific plotting
* Dashboard and GUI integrations

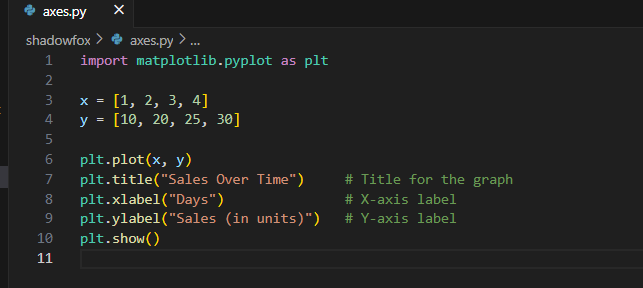
**Plot Elements :**

* **Axes**
* An Axes in Matplotlib is the actual plot area where data is drawn.
* It contains:
  + X and Y (and sometimes Z) axis
  + Title
  + Tick marks
  + Grid lines
  + Labels

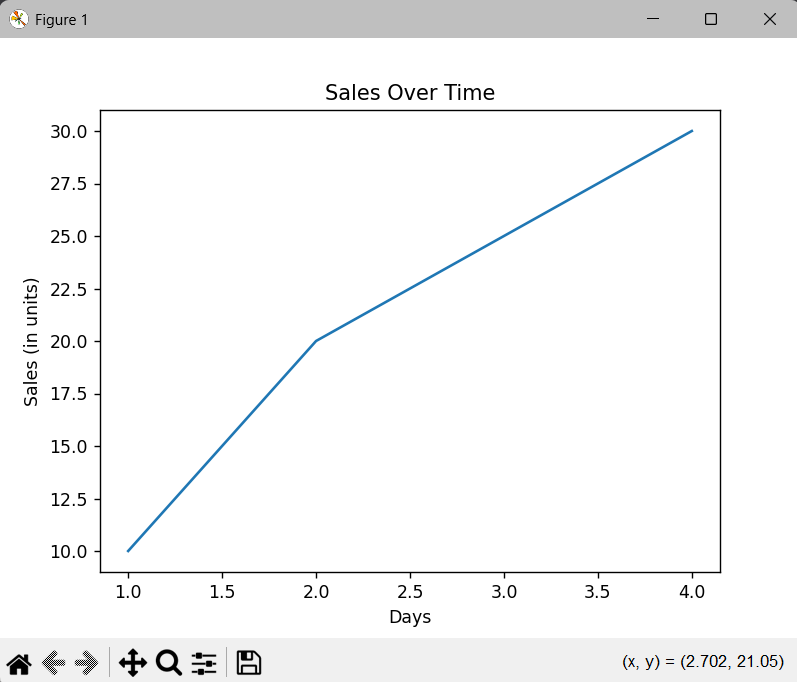
Basic Axes Methods:

* plt.xlabel(): Sets label for the x-axis.
* plt.ylabel(): Sets label for the y-axis.
* plt.title(): Adds a title to the graph.
* plt.axis(): Sets limits or scales for axes.

Example :

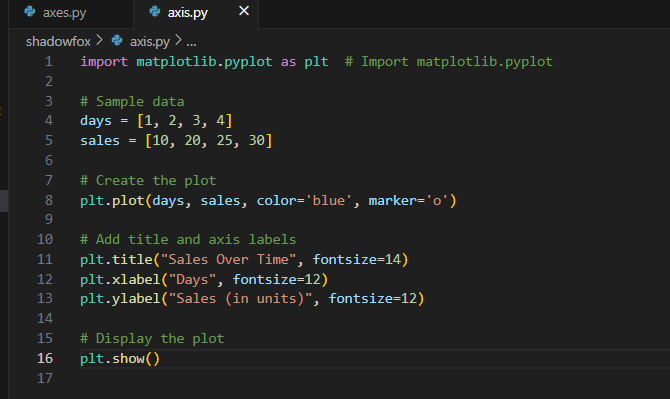


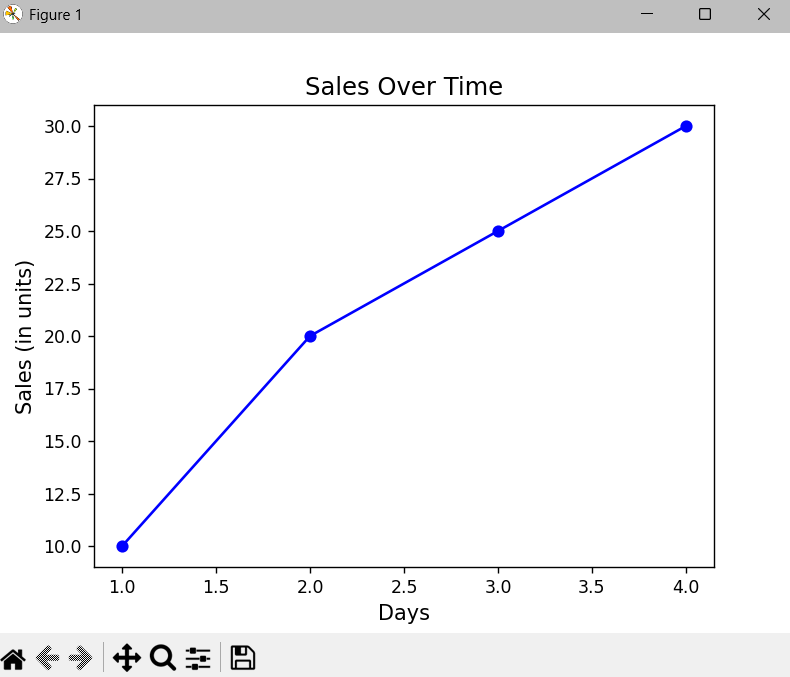
Output



* **Axis**
* It is used to **clarify** what the axes represent.
* It improves readability for users unfamiliar with the data.

Example:



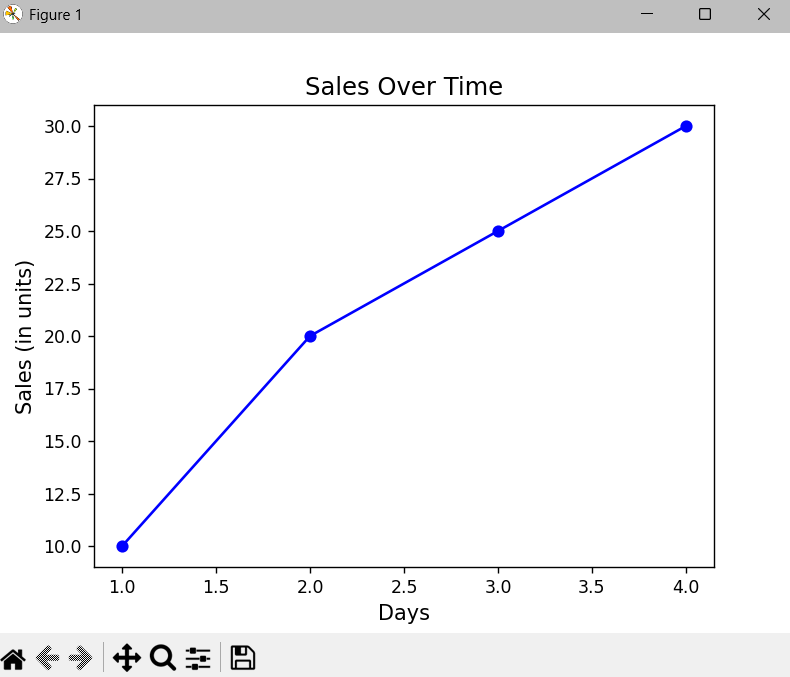
Output :  


* **Legend**
* A legend explains what different lines or markers represent.
* It is useful when plotting multiple datasets.
* Use the label argument in plot(), and then call plt.legend().

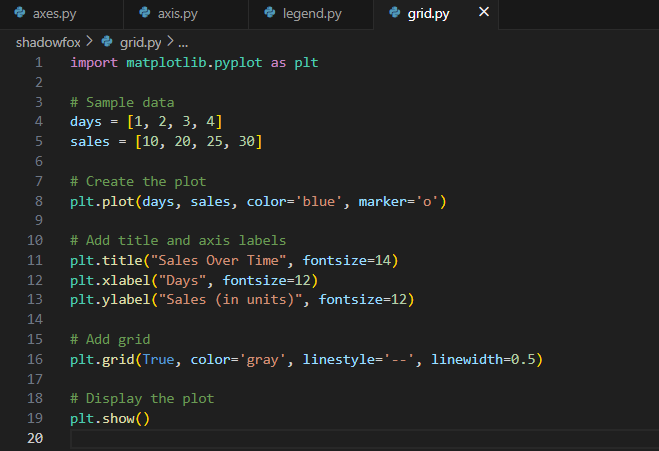
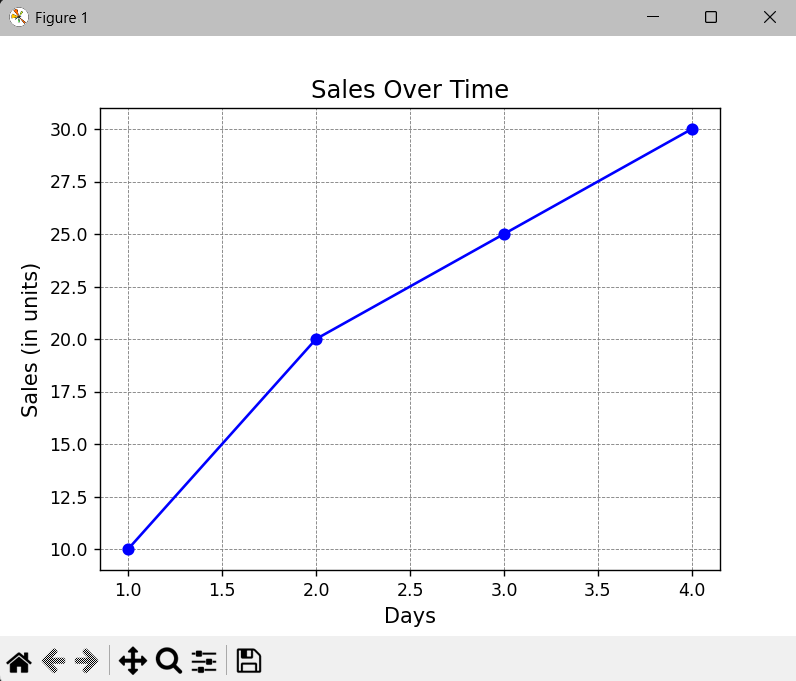
Example:



Output:



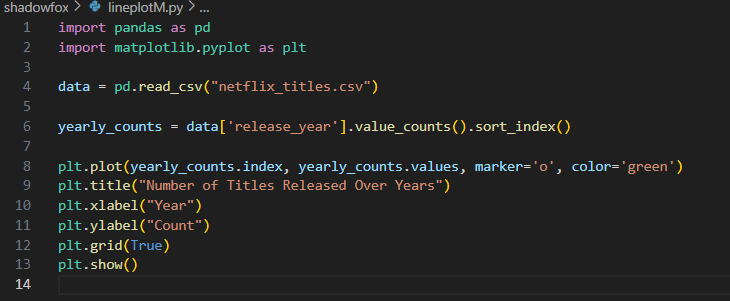
* **Grid**
* The grid adds reference lines across the plot.
* Helps in aligning and reading values more accurately.
* plt.grid(True): Turns the grid on.
* Options like color, linestyle, linewidth, and axis can be used for customization.

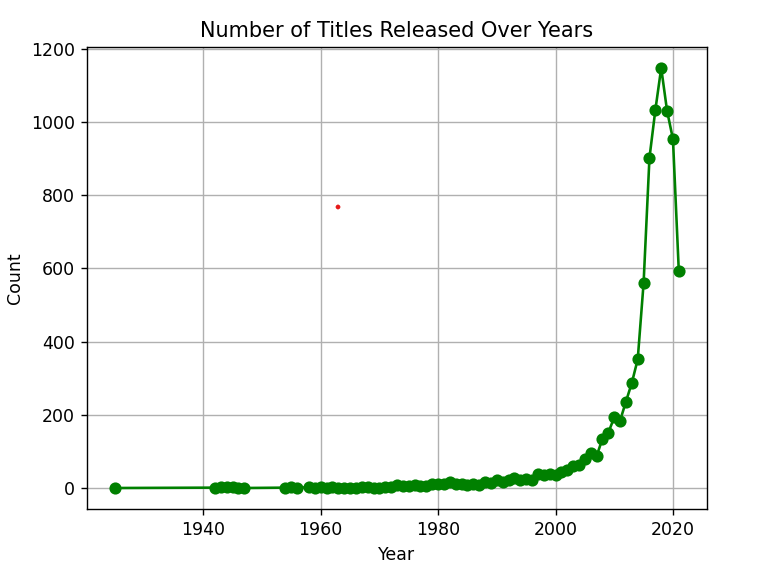
Example: Output: 

**GRAPH TYPES IN MATPLOTLIB**

* **Line Plot**

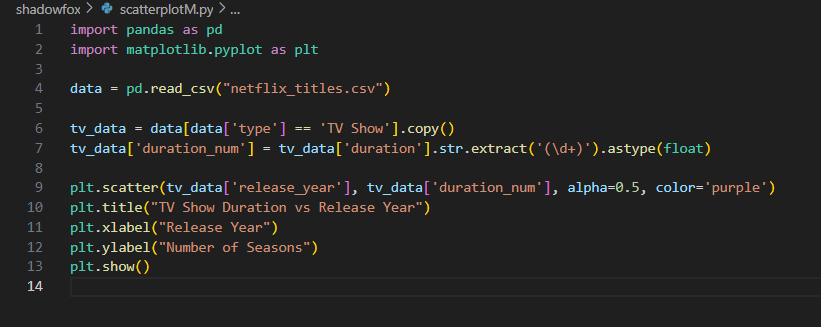
Shows relationships between two numerical values. Common for time series and trends.

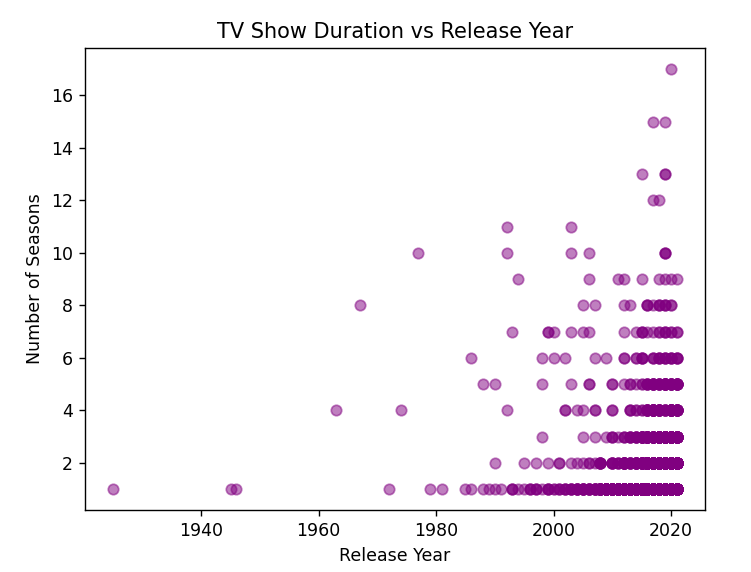
Example: 

Output: 

* **Scatter Plot**

Plots points based on two variables. Good for finding correlations and outliers.

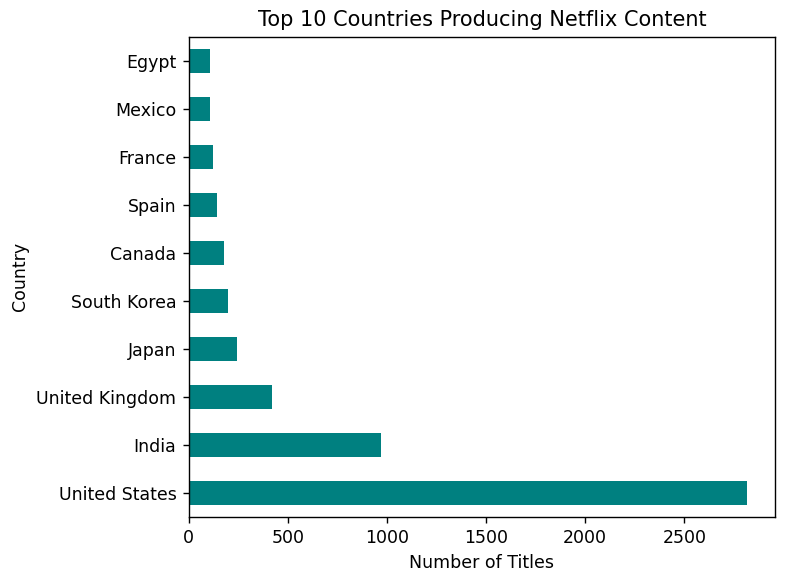
Example: 

Output: 

* **Barchart(Horizontal)**

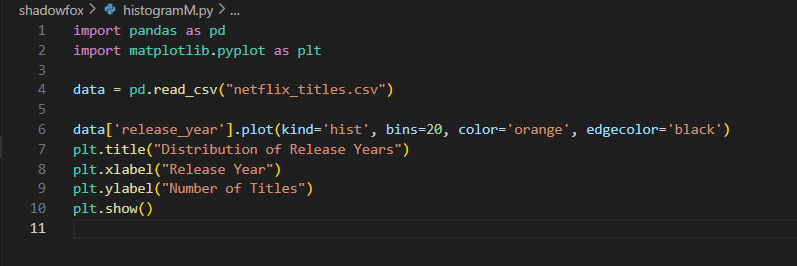
Displays categorical data with rectangular bars. Can be vertical or horizontal.

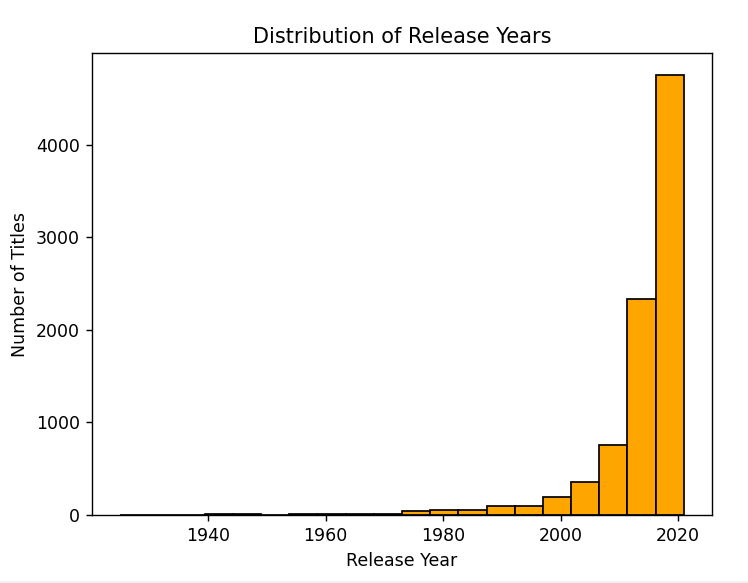
Example: 

Output: 

* **Histogram**

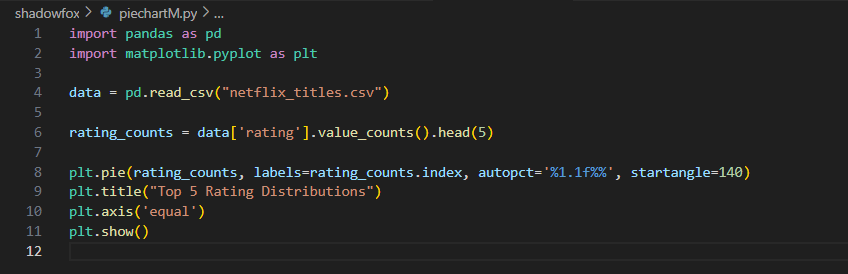
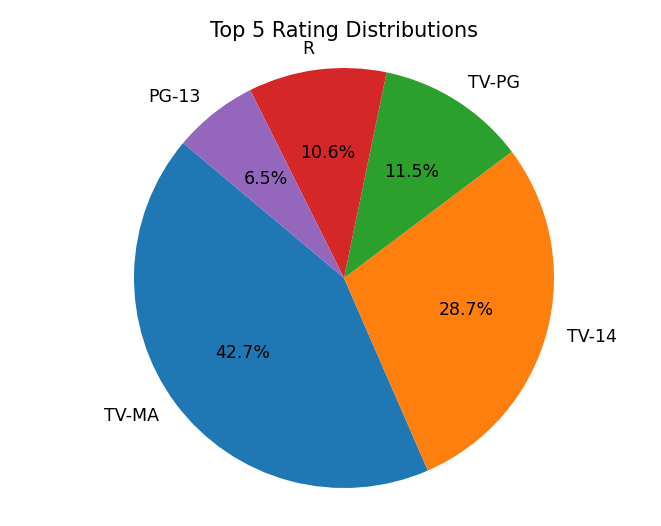
Groups continuous data into bins. Shows data distribution.

Example: 

Output: 

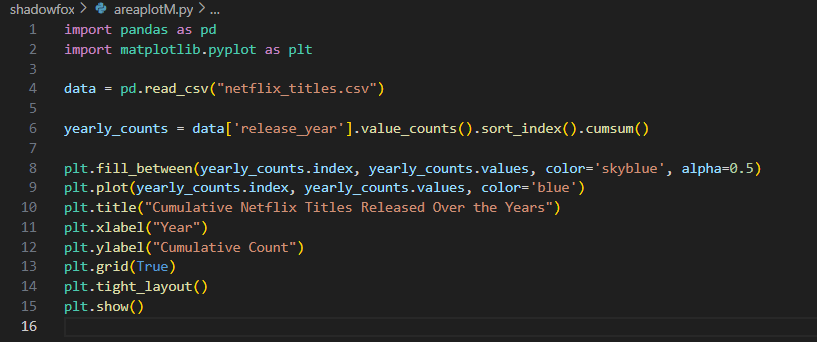
* **Pie Chart**

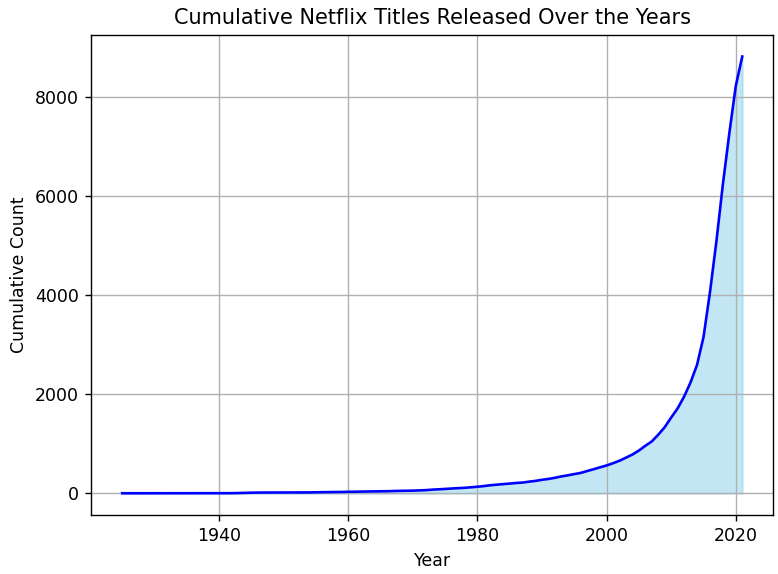
Displays proportions of a whole. Uses slices of a circle.

Example: Output: 

* **AreaPlot**

Fills the area under the line plot. Good for cumulative data.

Example: 

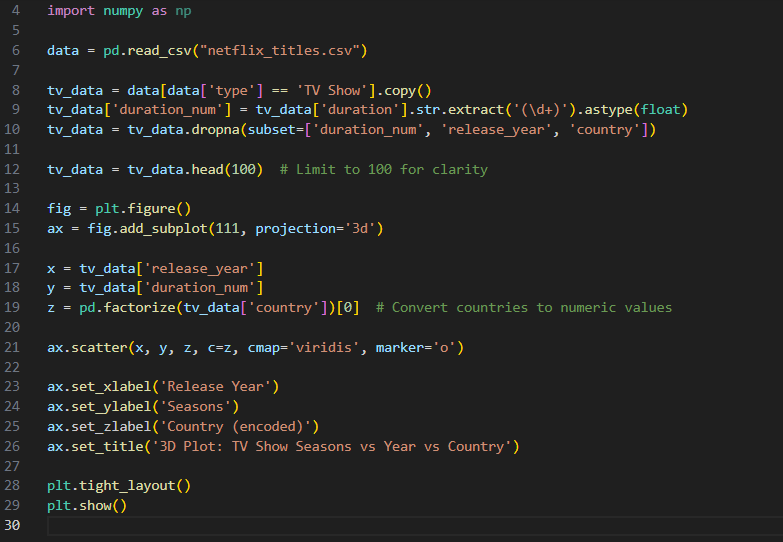
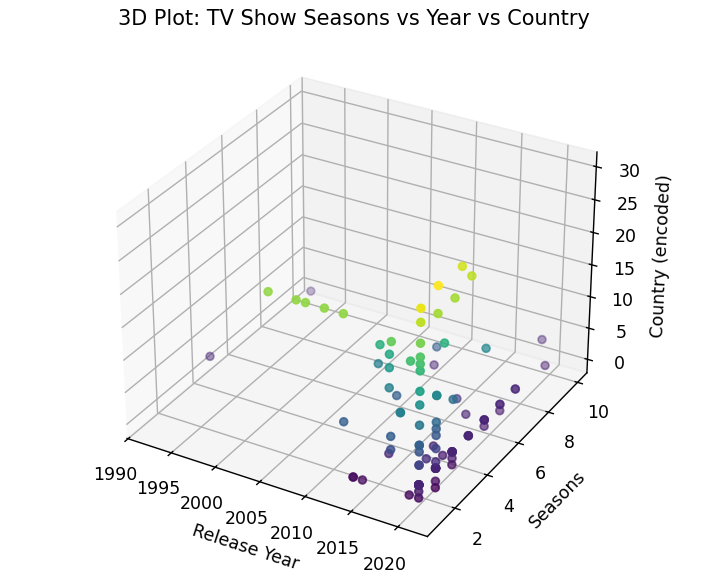
Output: 

* **3D Plot (Using mpl\_toolkits.mplot3d)**

Three-dimensional visualizations.

**Use Case:**

Surface plots, scientific visualization.

Example: Output: 

PLOTLY

Plotly is a powerful, open-source library for creating interactive visualizations in Python. Unlike Matplotlib, Plotly produces highly interactive and web-friendly charts (e.g., zoom, hover, pan) with ease. It's often used in data science dashboards and web apps.

**Key Features:**

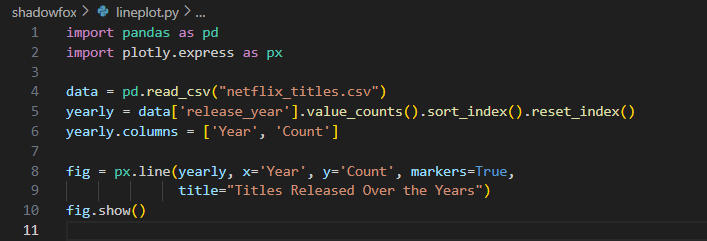
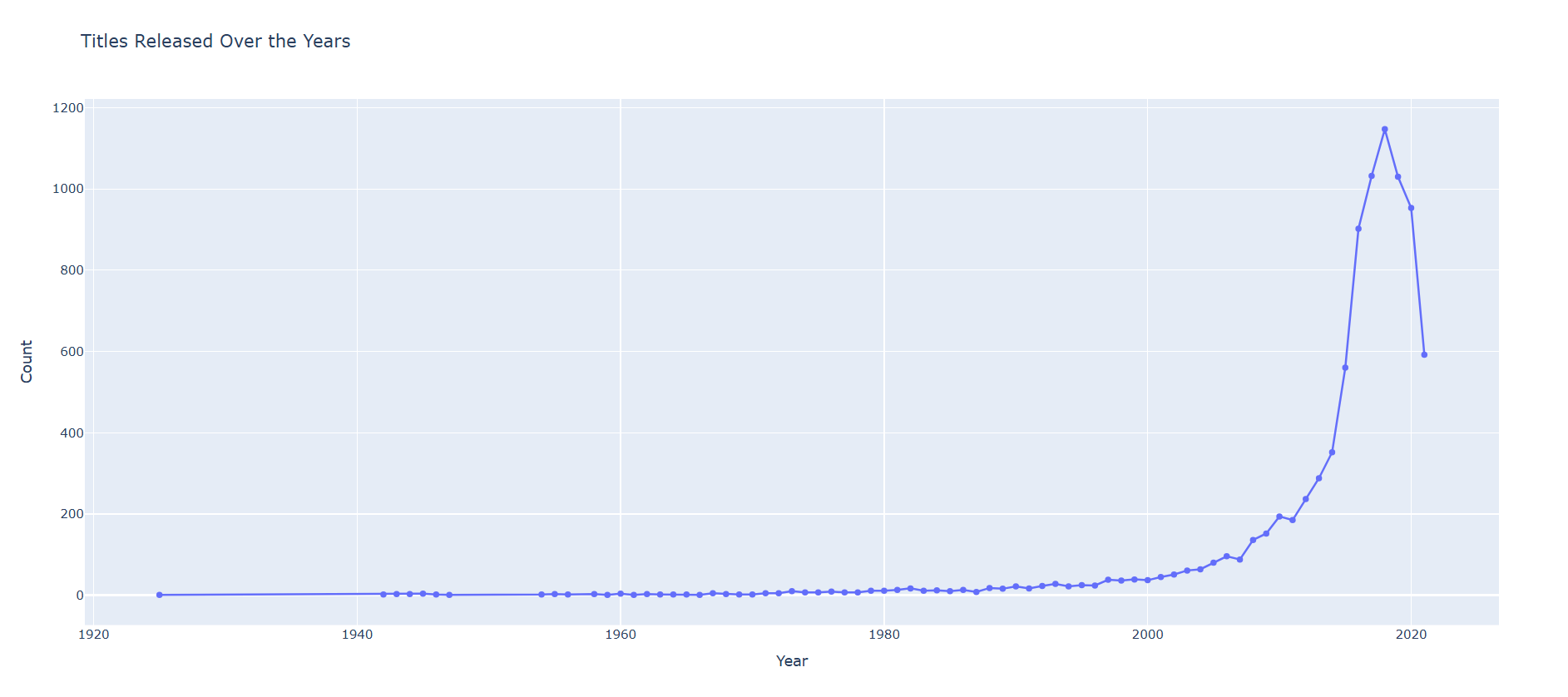
* Interactive visualizations (hover tooltips, zoom, etc.)
* Browser-friendly (works in Jupyter Notebooks and web apps)
* Easy-to-use API
* Wide variety of chart types
* Built-in themes and styling options

**GRAPH TYPES IN PLOTLY**

* **Line Plot**

Used to visualize data trends over a continuous interval or time. **Difference from Matplotlib:**

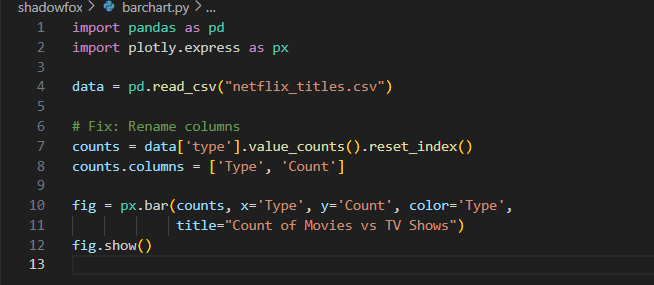
* Matplotlib: plt.plot() is static.
* Plotly: Interactive with zoom, hover, export as PNG.
* Easier to label, responsive in web apps.

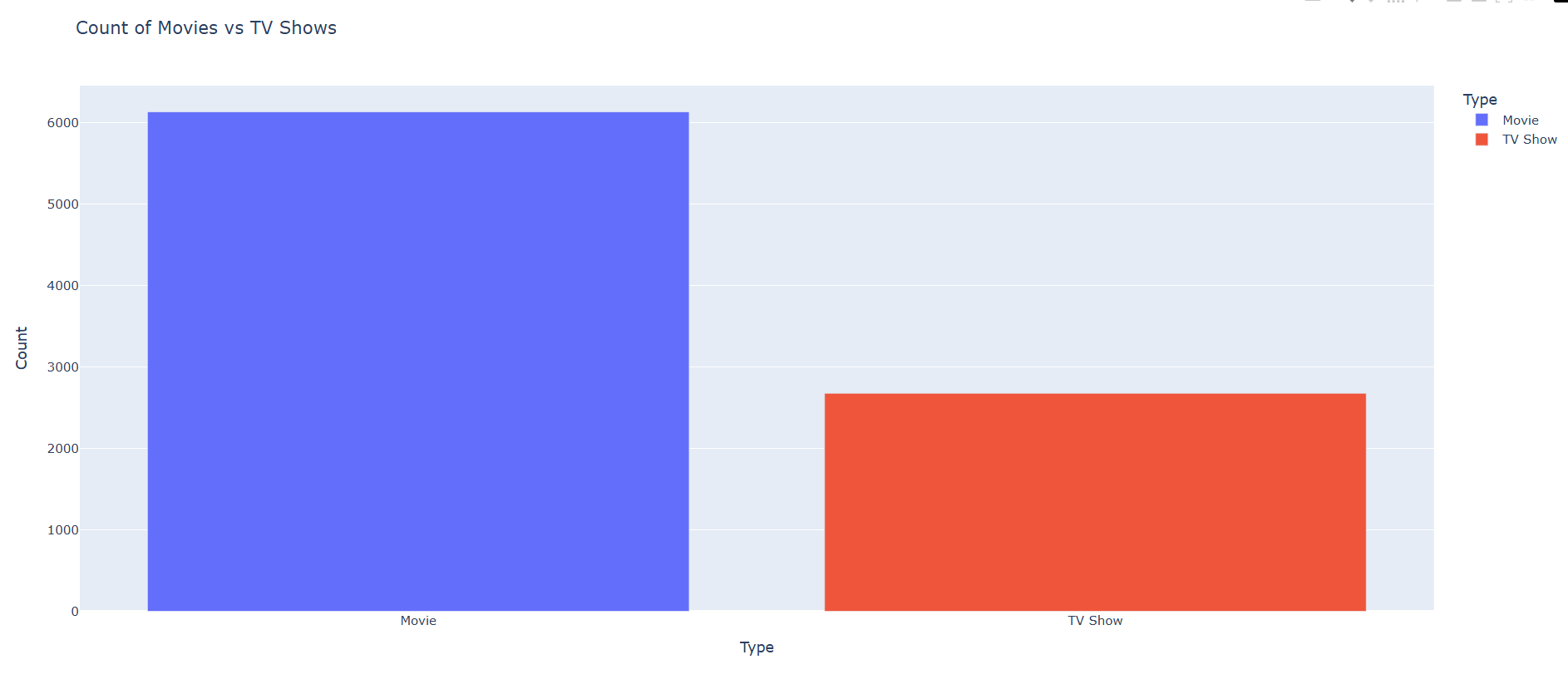
ExampleOutput: 

* **Bar Chart**  
  Used to compare quantities of different categories.

**Difference from Matplotlib:**

* Plotly provides hover tooltips and styling out-of-the-box.
* Matplotlib requires more code for aesthetics.
* Plotly is better for dashboards.

Example: 

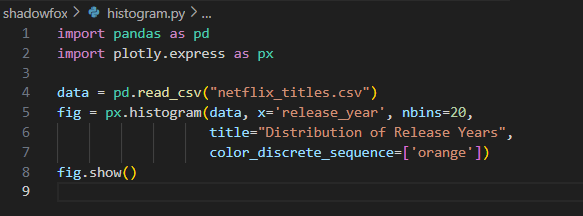
Output: 

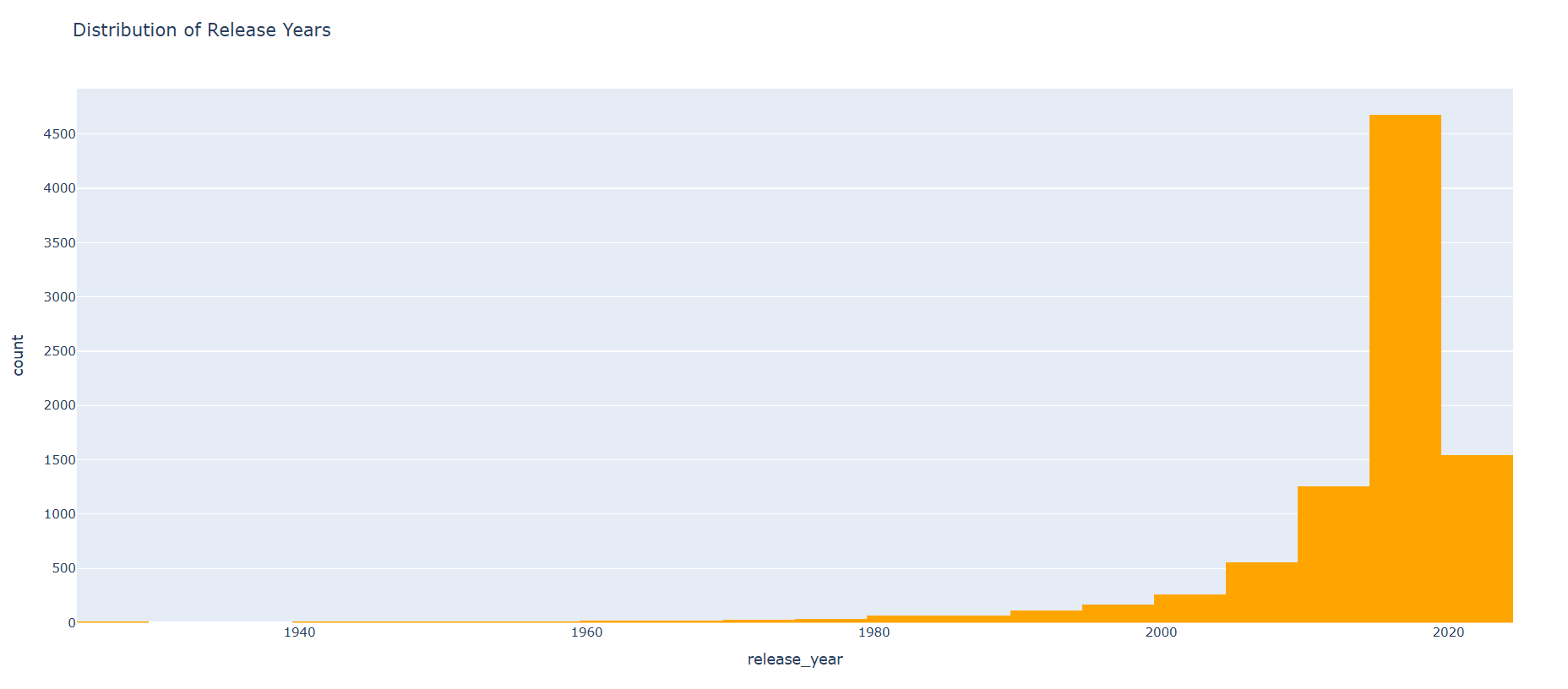
* **Histogram**

Displays the frequency distribution of a dataset.

**Difference from Matplotlib:**

* Plotly histograms allow bin resizing and hover.
* Interactive exploration is easy in Plotly.
* Matplotlib is static and suited for print.

Example: 

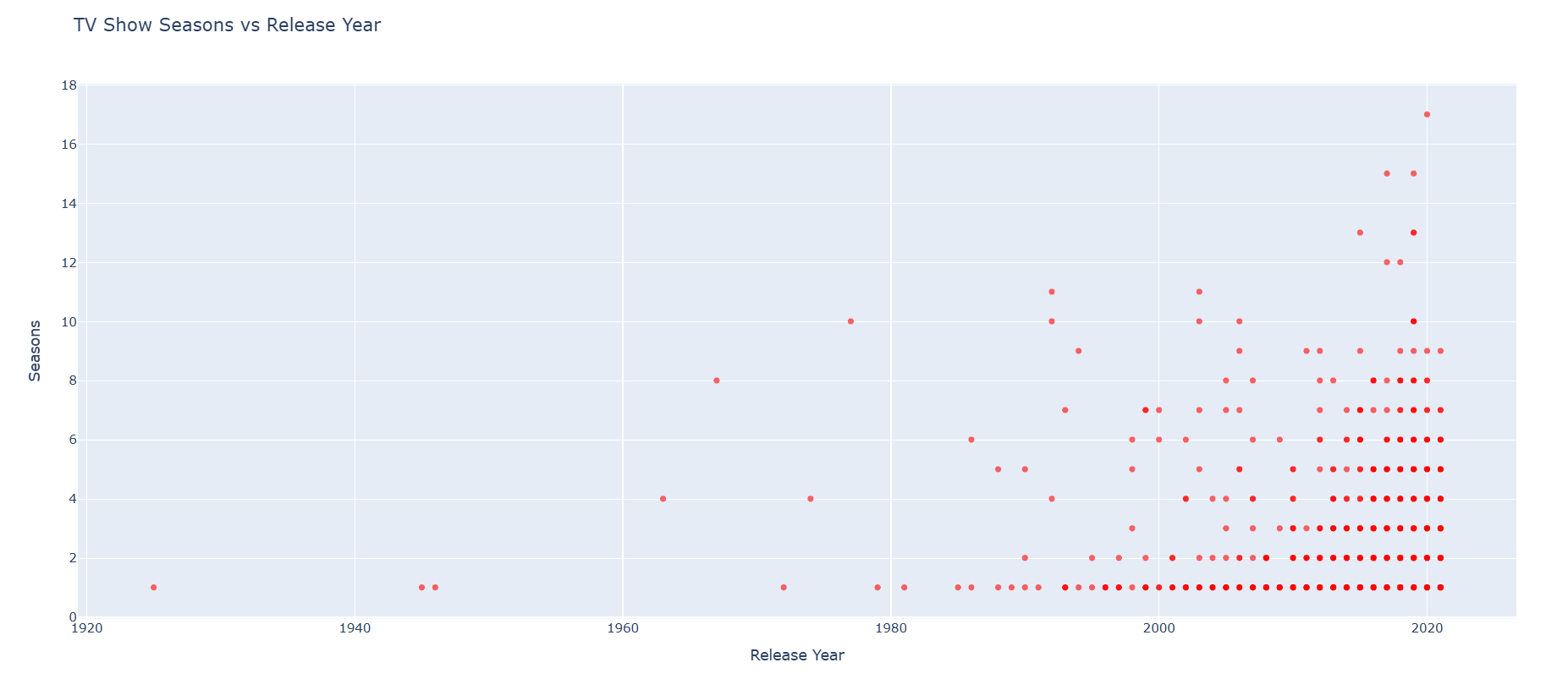
Output: 

* **Scatter Plot**  
  Used to identify relationships or correlations between two variables.

**Difference from Matplotlib:**

* Plotly allows interactive zoom, hover tooltips.
* Matplotlib scatter plots are static.
* Plotly can group by color/shape easily.

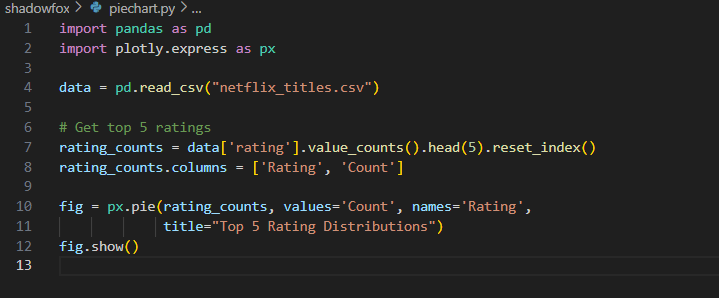
Example: 

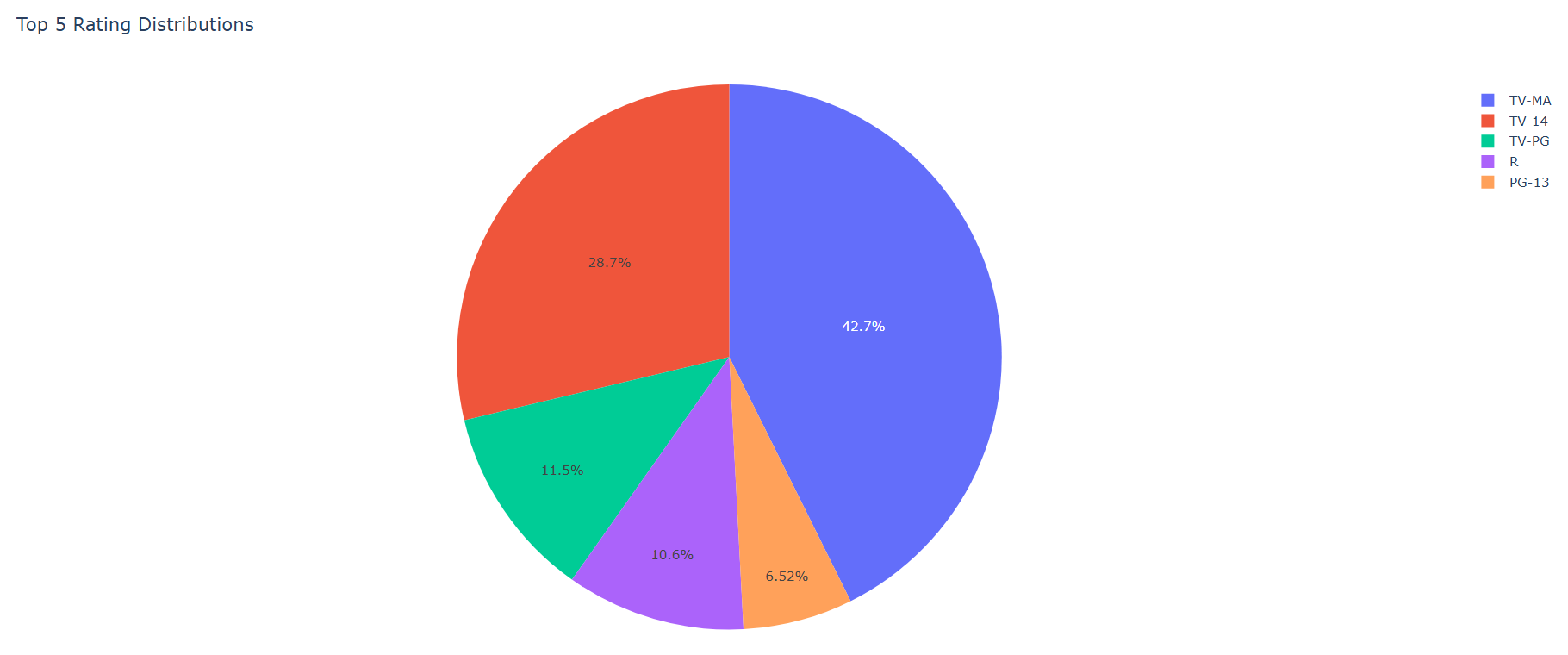
Output: 

* **Pie Chart**  
  Represents data as slices of a circle, showing proportions.

**Difference from Matplotlib:**

* Plotly shows percentages, values on hover.
* Interactive legend toggle available.
* Matplotlib pie charts are less visually appealing by default.

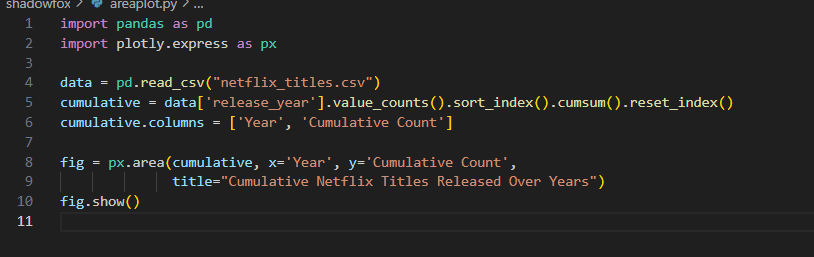
Example: 

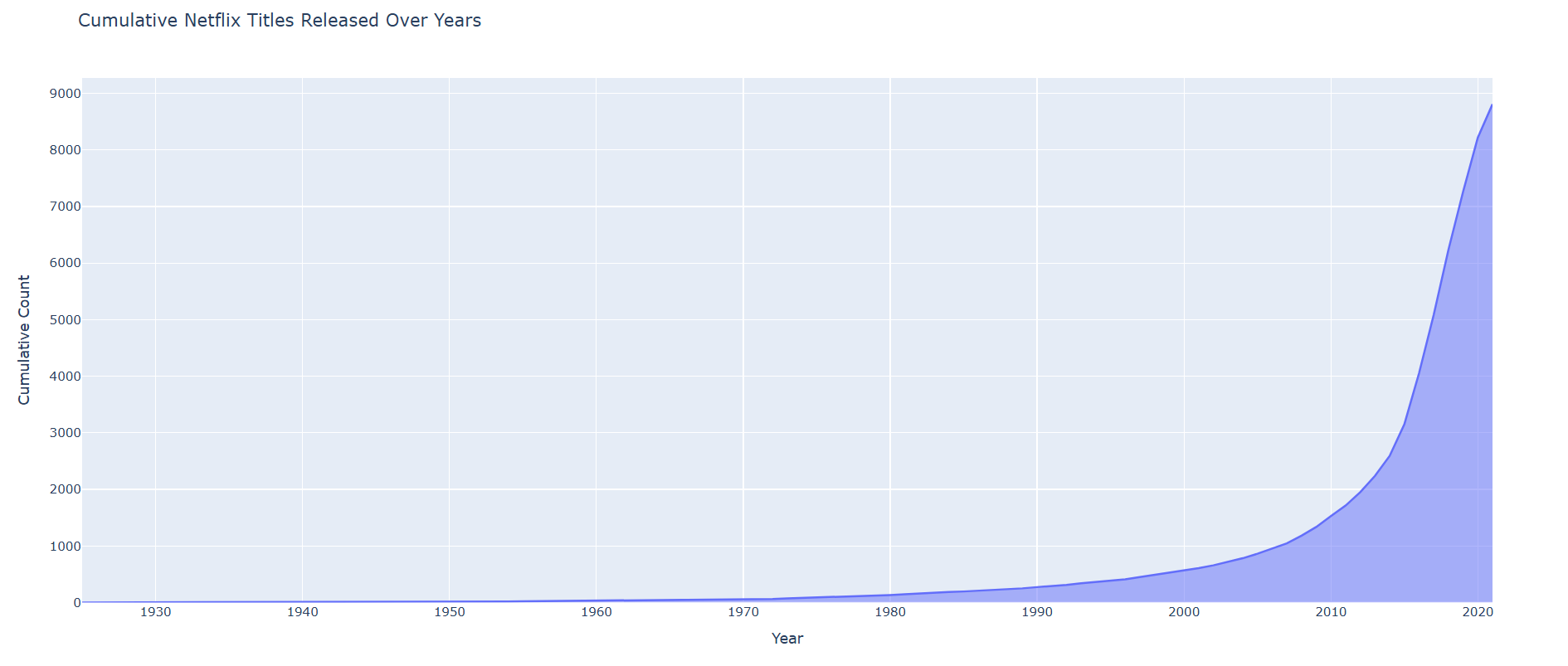
Output: 

* **Area Plot**  
  Used to display quantitative data with filled areas under the line.

**Difference from Matplotlib:**

* Matplotlib uses fill\_between(), but needs more customization.
* Plotly is visually smoother and responsive, great for web.

Example: 

Output: 

| **Feature** | **Matplotlib** | **Plotly** |
| --- | --- | --- |
| **Type** | 2D plotting library (static images) | Interactive graphing library |
| **Interactivity** | Limited (via toolkits like mpl\_interactive) | Highly interactive by default (zoom, hover, pan, etc.) |
| **Ease of Use** | Simpler for basic plots | Requires more structure but offers more advanced features |
| **Customization** | Extensive customization via code | Customizable with built-in tools and GUI-based editors |
| **Output Format** | Static (PNG, PDF, SVG) | Interactive (HTML, JSON, static images) |
| **Web Integration** | Not native, requires extra setup (e.g., Flask + Matplotlib) | Natively supports web integration via Dash or HTML exports |
| **3D Plotting** | Basic 3D support via mpl\_toolkits.mplot3d | Advanced and smoother 3D rendering |
| **Animation Support** | Yes (basic via FuncAnimation) | Yes (more dynamic and browser-based) |
| **Learning Curve** | Easier for beginners | Slightly steeper due to structured syntax |
| **Best Use Case** | Academic, print publications, static dashboards | Web dashboards, interactive reports, data exploration |

**DIFFERENCE BETWEEN MATPLOTLIB AND PLOTLY**

**Summary**

* **Matplotlib** is ideal for static plots and academic work.
* **Plotly** is better suited for interactive visualizations and web applications.