Data Visualization using Python

Importance of Data Visualization

- **Record Information:**
  Visualization serves as a means to record and represent data in a visual format, making it easier to interpret and analyze complex information.
- **Analyze Data to Support Reasoning:**
  Visualizing data helps in analyzing patterns, trends, and relationships within the data, enabling better decision-making and reasoning.
- **Confirm Hypotheses:**
  By visually representing data, individuals can validate or invalidate hypotheses, providing a clear and visual way to confirm or refute assumptions.
- **Communicate Ideas to Others:**
  Visualization aids in effectively communicating insights and findings to a broader audience, facilitating better understanding and engagement.

Useful Python Libraries

- **NumPy**
  NumPy is an important library in scientific computing and data manipulation. NumPy is highlighted for its efficiency in handling numerical computations and its integration with other Python libraries like Matplotlib and pandas. It is a fundamental tool for working with n-dimensional arrays and performing array manipulation operations. NumPy plays a crucial role in data analysis and visualization, making it a key resource for users seeking to analyze and visualize data effectively

- **Pandas**
  Pandas is highlighted in the document as a fundamental tool for efficiently handling and analyzing tabular data. It offers powerful data structures like DataFrame, which organizes data in rows and columns, and Series, which represents a single column. Pandas is particularly well-suited for working with structured data, implementing various data operations, and simplifying tasks such as data manipulation and analysis. Its integration with other Python libraries like NumPy and Matplotlib enhances its functionality for data processing and visualization, making it a versatile and essential tool for users working with tabular data sets.

- **Matplotlib**
  Matplotlib is a versatile Python library used for basic plotting tasks. It is highly customizable, allowing users to create a wide range of visualizations tailored to their specific needs. Matplotlib seamlessly integrates with other popular Python libraries like NumPy and pandas, enhancing its capabilities for data visualization and analysis.

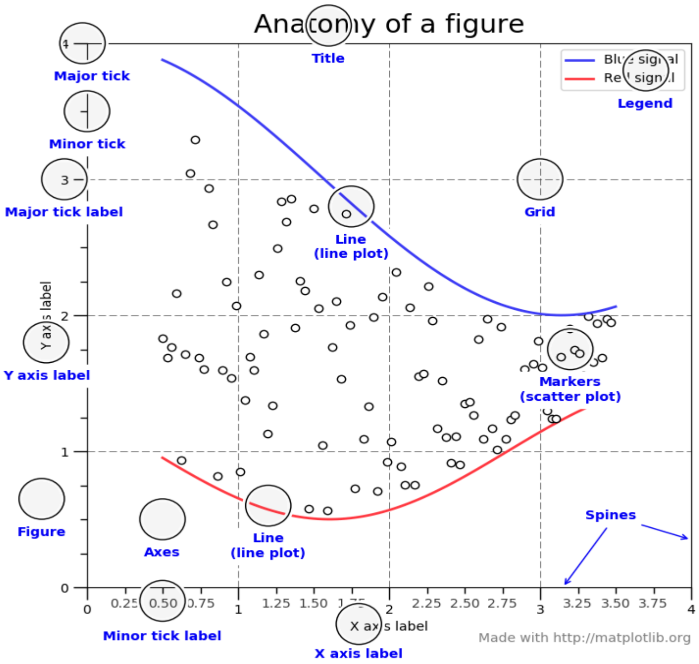
- **Seaborn**
  Seaborn is a Python library specializing in statistical data visualization. It offers a simplified syntax with default themes, making it easy to create visually appealing plots. Seaborn is seamlessly integrated with pandas data frames, enhancing its compatibility with tabular data structures. Under the hood, Seaborn utilizes Matplotlib for rendering plots, combining the ease of use with the power of Matplotlib for statistical visualization tasks.

- **Bokeh**
  Bokeh, as highlighted in the document, is a Python library known for interactive visualization capabilities. It requires modern web browsers for optimal performance and seamlessly integrates with JavaScript for creating dynamic and interactive plots. Bokeh enables users to build web-ready visualizations that enhance user engagement and exploration of data through interactive plots and dashboards.

Types of Plots for Effective Representation

- **Line plots**
  Used for displaying trends and patterns in numeric data over time, allowing for comparisons and predictions based on historical data.
- **Bar plots**
  Ideal for comparing data across different categories, especially nominal or ordinal, and can effectively show changes over time for multiple categories.
- **Scatter plots**
  Useful for visualizing relationships and correlations between two variables, helping in identifying patterns and trends in the data.
- **Box plots**
  Provide a visual summary of the distribution of numerical data, including the range, spread, and central tendency, making them suitable for comparing data across different categories.
- **Histograms**
  Display the frequency distribution of numerical data, showing the distribution shape and patterns within the data set.

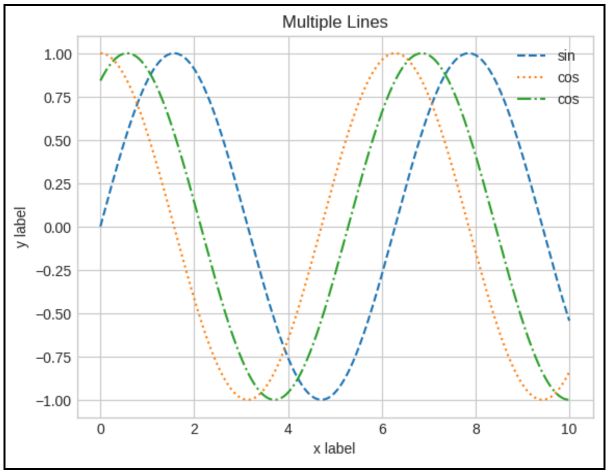
Matplotlib

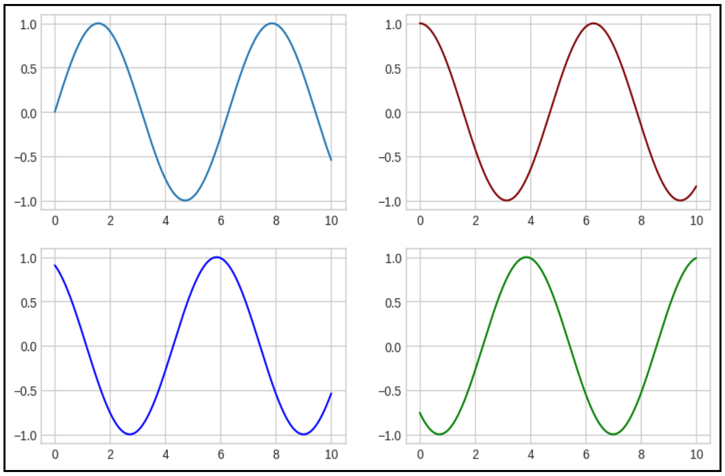


- In Matplotlib, the figure (an instance of the class plt.Figure) can be thought of as a single container that contains all the objects representing axes, graphics, text, and labels. The axes (an instance of the class plt.Axes) is what we see above: a bounding box with ticks and labels, which will eventually contain the plot elements that make up our visualization.
- In Matplotlib, there are various kind of plots like, Line plots, Bar plots, Scatter plots, Box plots and Histograms that are highly customizable, which allows users to create a wide range of visualizations tailored to their specific needs

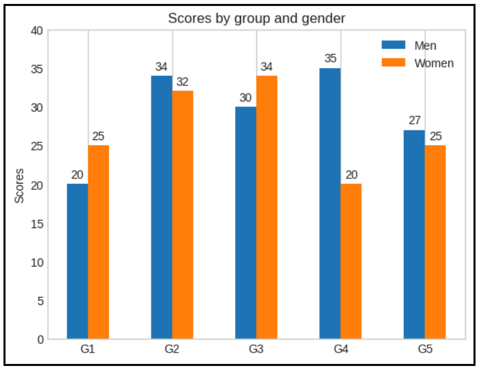
Following are some example visualizations that can be done using Matplotlib.

- Line Plots

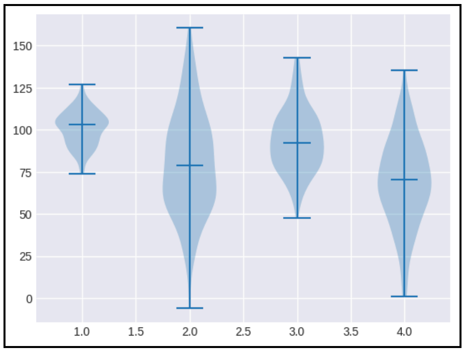




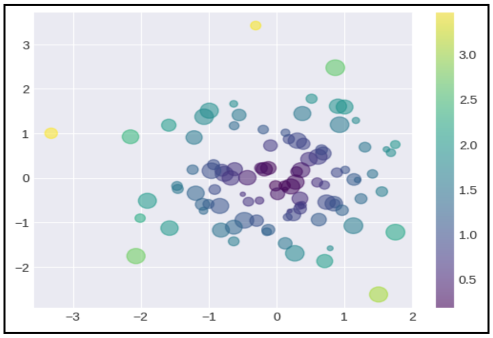
- Bar Plots



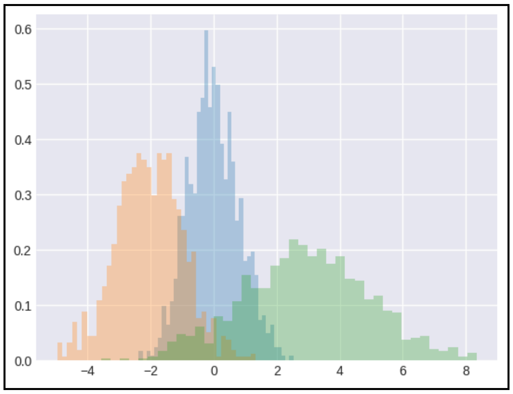
- Box Plots



- Scatter Plots

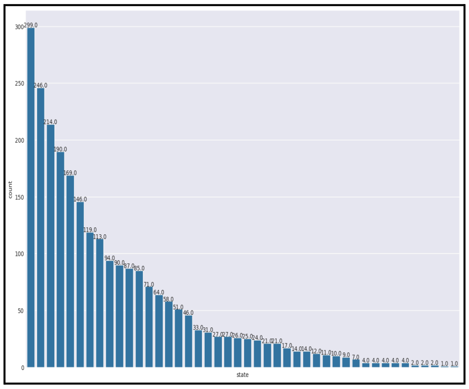


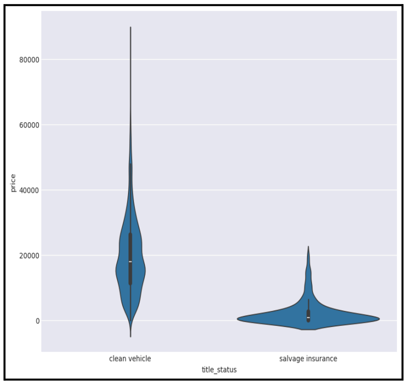
- Histogram

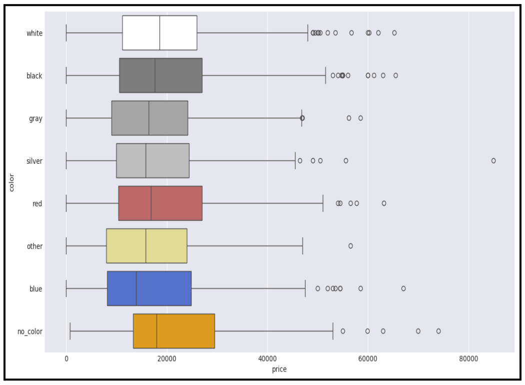


Seaborn

Seaborn is also similar to Matplotlib which offers a simplified syntax with default themes, making it easy to create visually appealing plots. Following are some examples.







You can refer to the following links for more examples and representation methods.

- Colab Notebook: https://colab.research.google.com/drive/1KEi49-UnPvzYsFqP0ivKn3hGw2_X_DGP?usp=sharing
- Google Drive: https://drive.google.com/file/d/1IALayh_wL6AChp2be30gHMvnUtrcW6se/view?usp=sharing
- GitHub Repo: https://github.com/UsithaDJay/python-data-viz.git