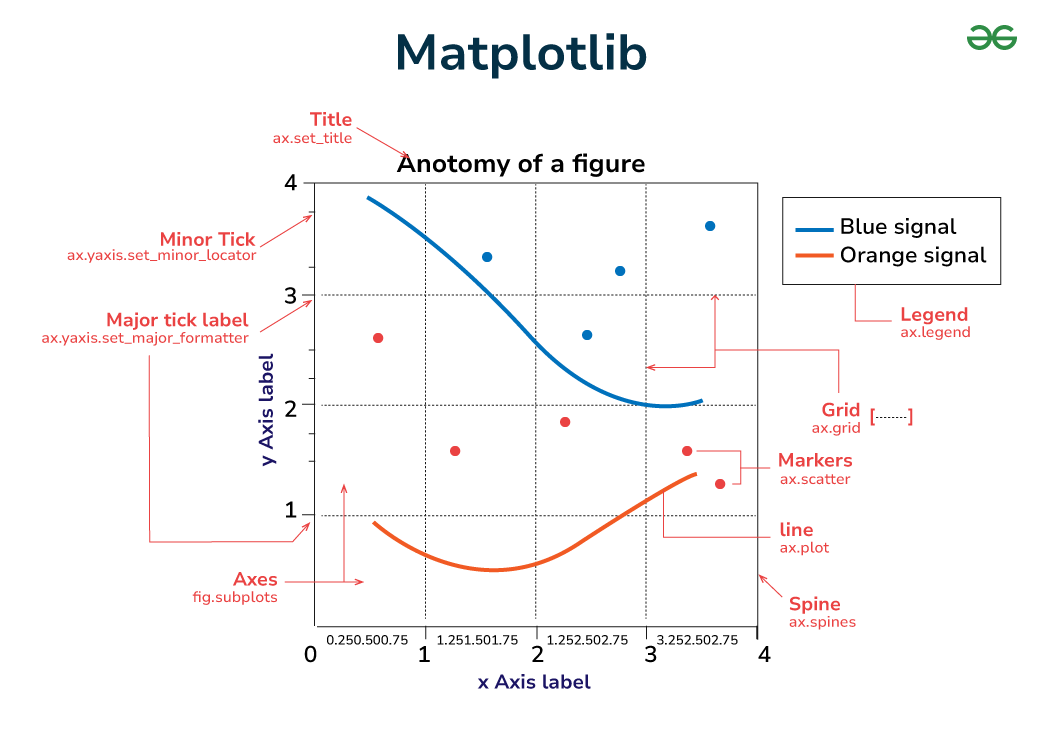
**Matplotlib:**

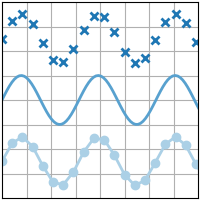
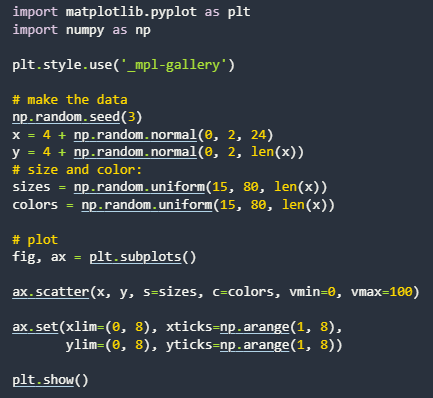
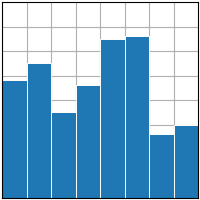
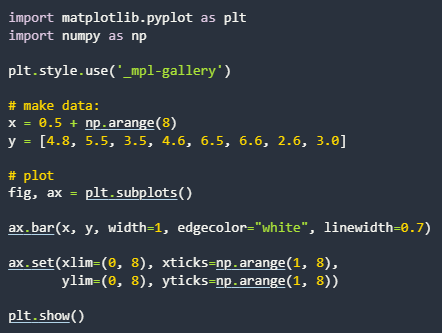
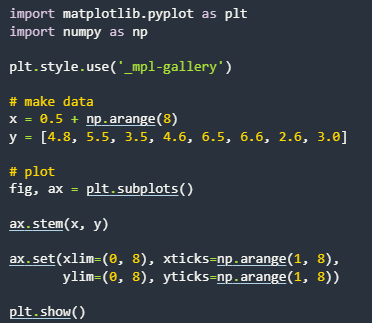
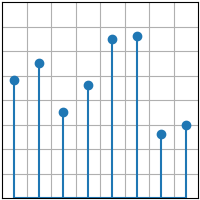
Matplotlib is a versatile and powerful visualization library in Python, widely appreciated for its simplicity and effectiveness. It is built upon NumPy arrays and designed to integrate seamlessly with the SciPy ecosystem. The library supports various plot types, such as line graphs, bar charts, scatter plots, histograms, and more.

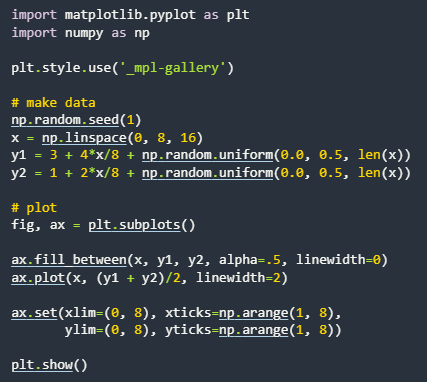
**Key Features of Matplotlib:**

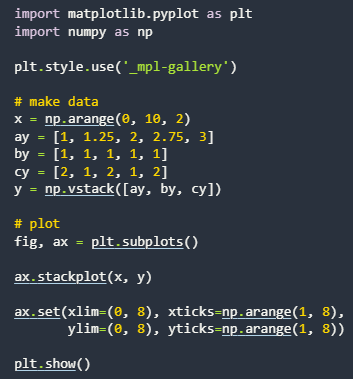
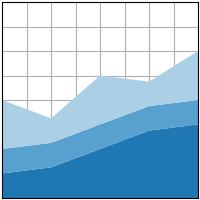
1. **Wide Variety of Plots:** Matplotlib supports numerous plot types, including line graphs, scatter plots, bar charts, histograms, pie charts, and more.
2. **Highly Customizable:** Users can control almost every aspect of the plot, such as line styles, colours, markers, labels, and annotations.
3. **Integration with NumPy:** The library works directly with NumPy arrays, making data plotting straightforward.
4. **Extensibility:** Matplotlib can be extended with add-ons like Seaborn, Pandas’ plotting utilities, and Basemap for geographical visualizations.
5. **Cross-Platform Compatibility:** The library is platform-independent and functions smoothly on Windows, macOS, and Linux.
6. **Interactive Capabilities:** It supports interactive plotting with widgets and event handling, allowing dynamic data exploration.

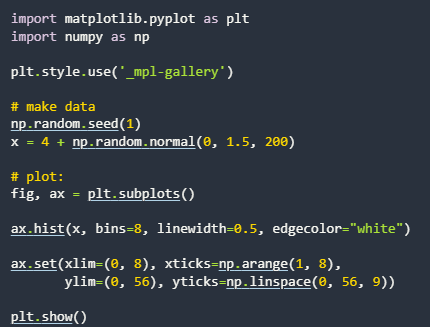
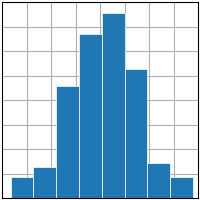
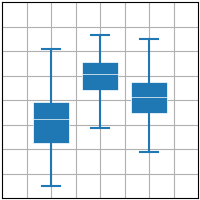
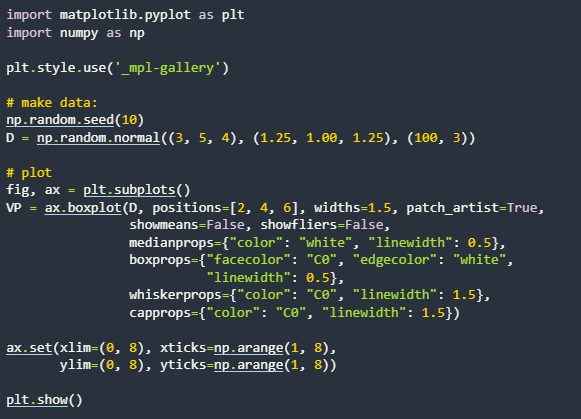
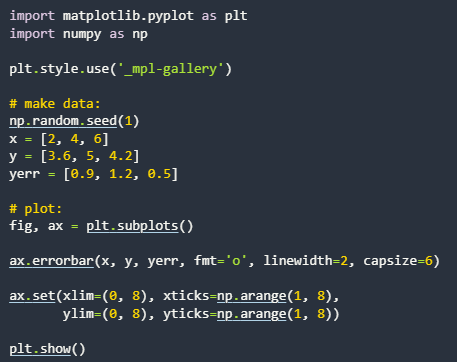
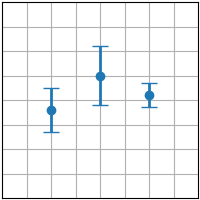
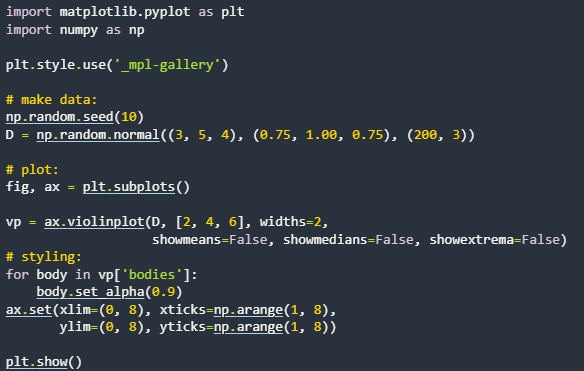
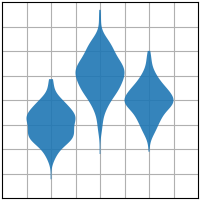
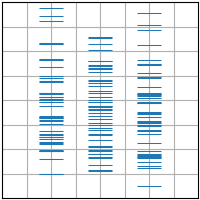
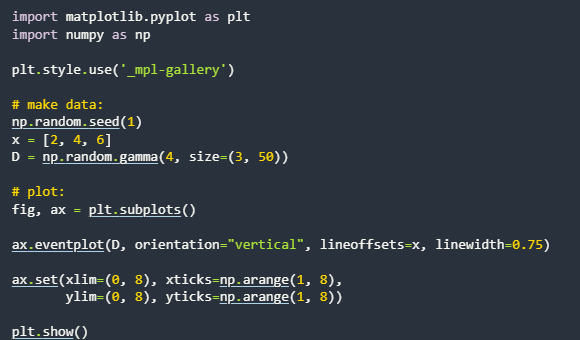
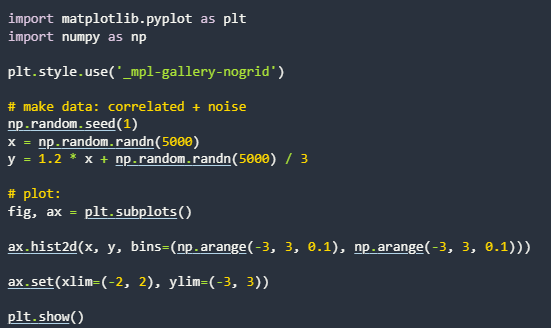
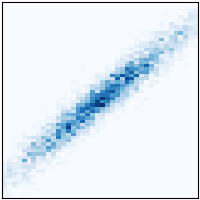
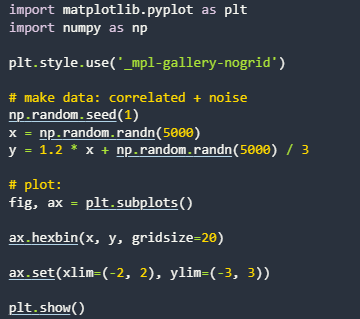
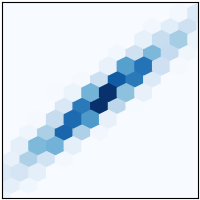
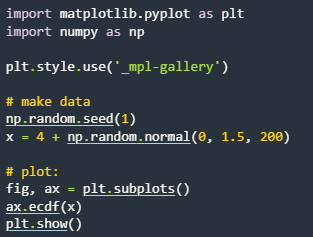
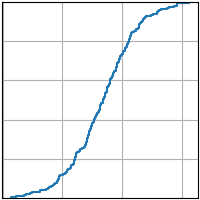
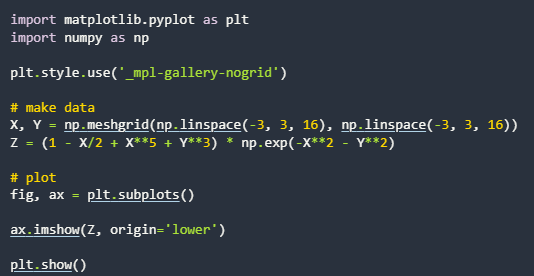
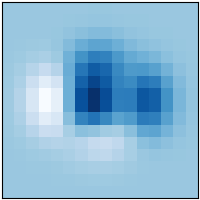
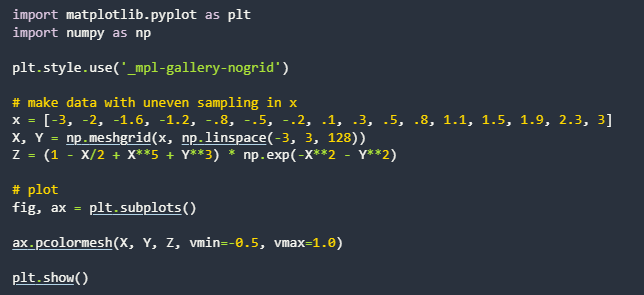
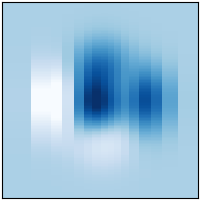
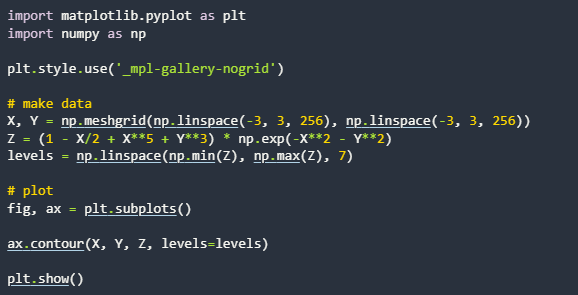
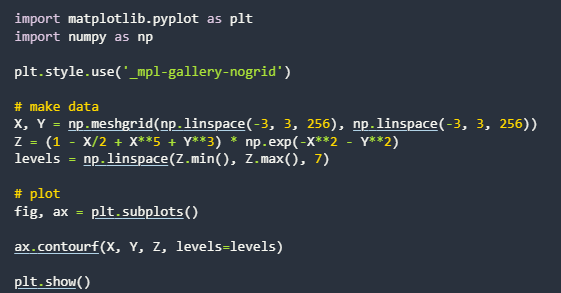
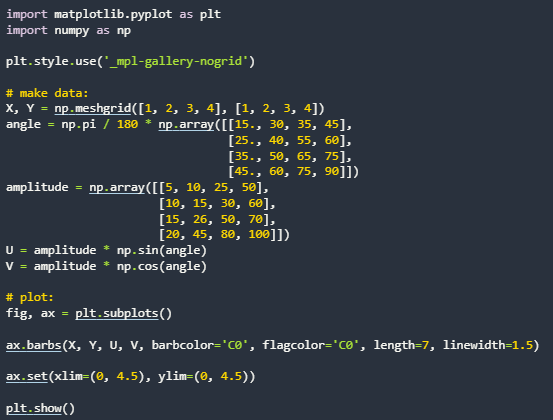
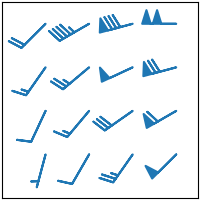
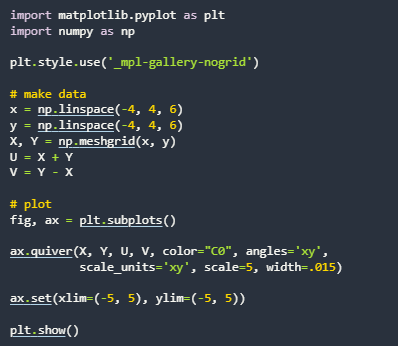
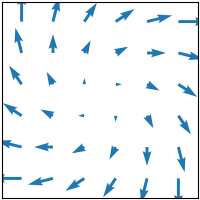
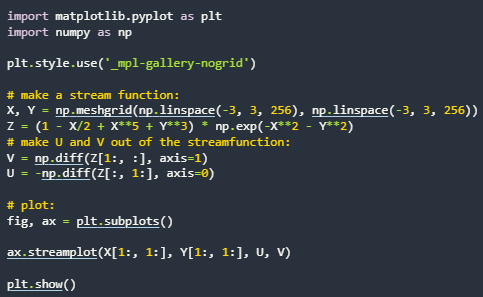
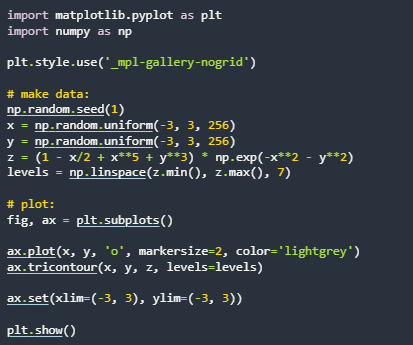
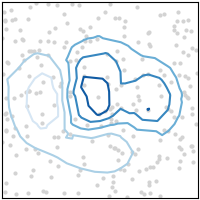
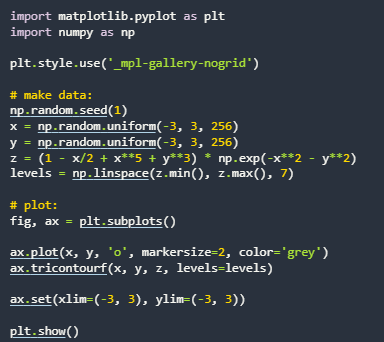
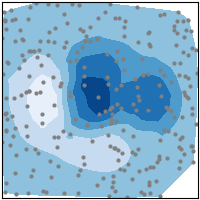
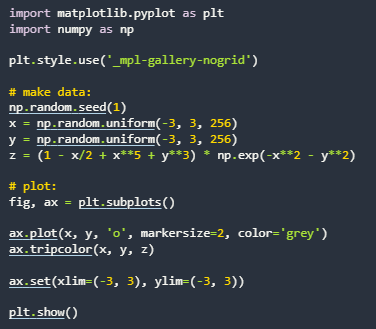
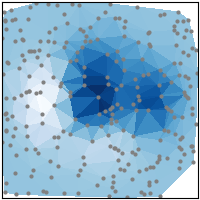
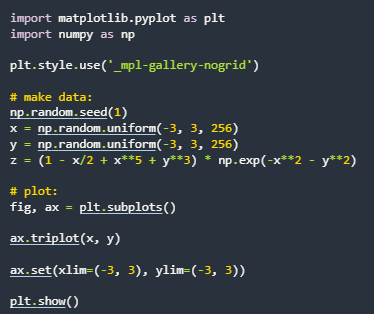
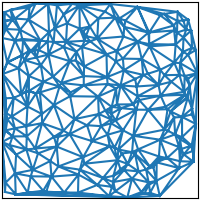
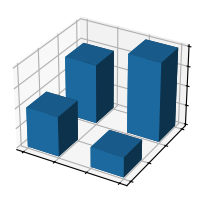
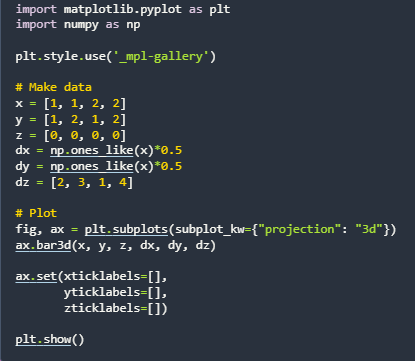
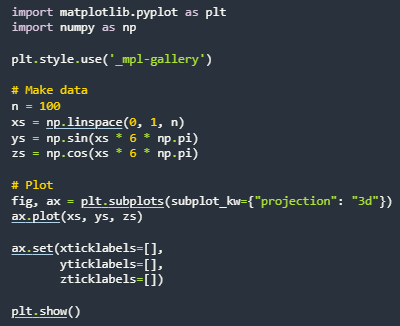
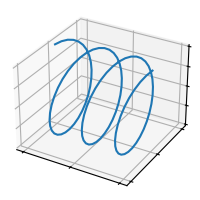
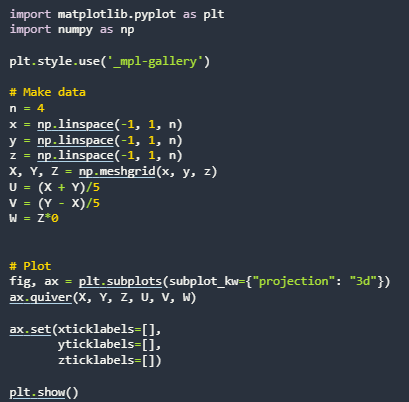
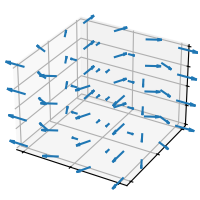
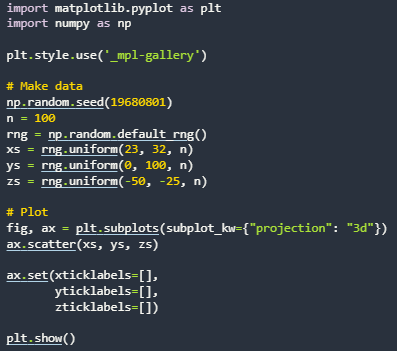
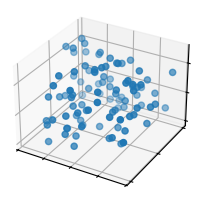
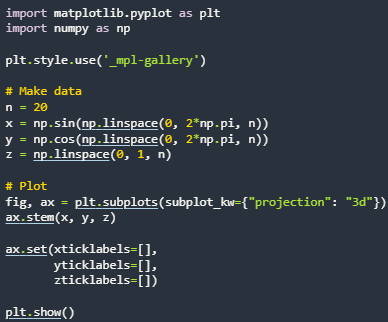
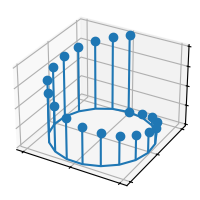
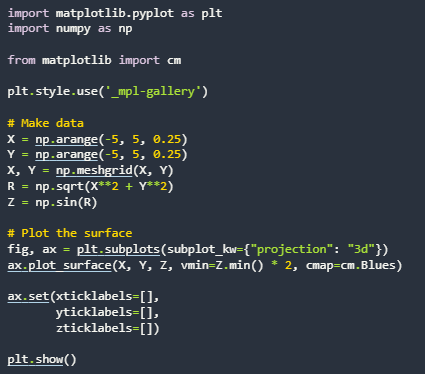
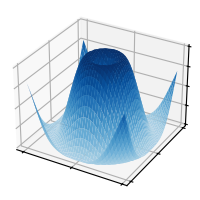
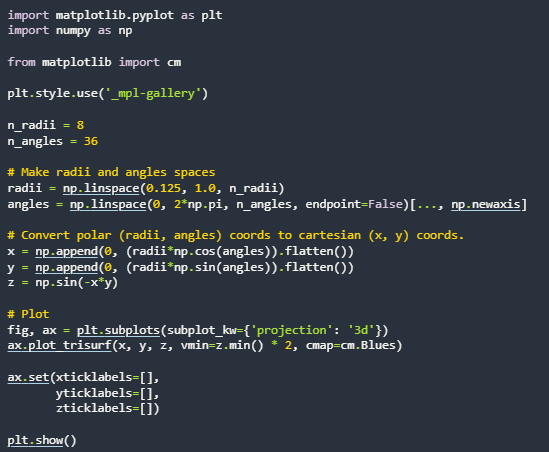
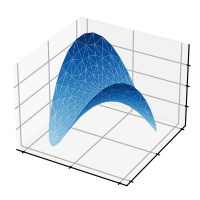
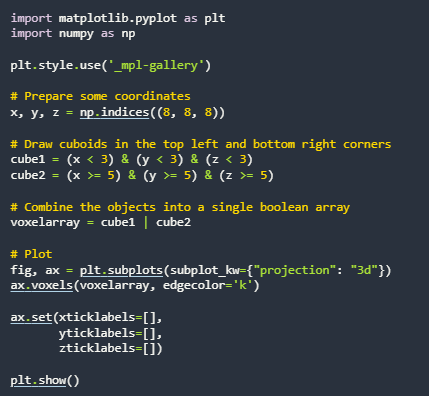
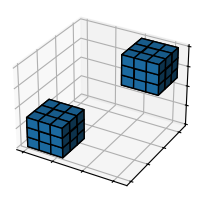
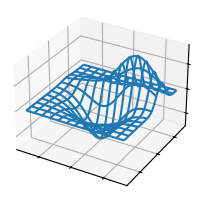
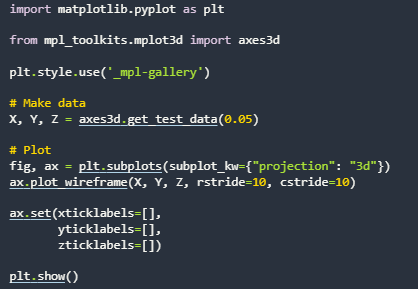
**Understanding Matplotlib Figures:** In Matplotlib, a figure acts as the main container that encompasses all the elements of a plot. It represents the canvas or window where the visualization is rendered.

**Plot Types Based on Data Visualization:**

1. **Pairwise Data:**
   * plot(x, y): Creates a simple line plot using x and y coordinates.
   * scatter(x, y): Generates a scatter plot to depict pairwise data.
   * bar(x, height): Produces a bar chart for categorical data.
   * stem(x, y): Classifies variables as discrete or continuous.
   * fill\_between(x, y1, y2): Shades the area between specified ranges.



* + stackplot(x, y): Displays stacked data as a filled plot.

1. **Statistical Distributions:**
   * hist(x): Displays numeric value distributions as bars.
   * boxplot(X): Visualizes data distribution using quartiles.
   * errorbar(x, y, yerr, xerr): Represents variability or uncertainty in data measurements.
   * violinplot(D): Combines box plot and density plot characteristics.
   * eventplot(D): Illustrates sequences of events over time.
   * hist2d(x, y): Creates a 2D histogram.
   * hexbin(x, y, c): Generates a hexagonal binning plot.
   * pie(x): Displays data as slices of a circular pie chart.
   * ecdf(x): Computes and plots the empirical cumulative distribution function.
2. **Gridded Data:**
   * imshow(Z): Renders a 2D array as an image.
   * pcolormesh(X, Y, Z): Creates a pseudo-color plot with an irregular grid.
   * contour(X, Y, Z): Plots contour lines.
   * contourf(X, Y, Z): Generates filled contour plots.
   * barbs(X, Y, U, V): Displays wind barb symbols.
   * quiver(X, Y, U, V): Plots a vector field.
   * streamplot(X, Y, U, V): Draws streamlines of vector flows.
3. **Irregularly Gridded Data:**
   * tricontour(x, y, z): Draws contour lines on unstructured triangular grids.
   * tricontourf(x, y, z): Creates filled contours on triangular grids.
   * tripcolor(x, y, z): Displays pseudocolor plots of triangular grids.
   * triplot(x, y): Renders triangular grids with markers and lines.
4. **3D and Volumetric Data:**
   * barplot3d(x, y, z, dx, dy, dz): Creates 3D bar plots.
   * plot(xs, ys, zs): Produces 3D line plots.
   * quiver(X, Y, Z, U, V, W): Plots 3D vector fields.
   * scatter(xs, ys, zs): Creates 3D scatter plots.
   * stem(x, y, z): Plots 3D stem graphs.
   * plot\_surface(X, Y, Z): Displays 3D surface plots.
   * plot\_trisurf(x, y, z): Draws triangulated surfaces.
   * voxels([x, y, z], filled): Represents 3D pixel data (voxels).
   * plot\_wireframe(X, Y, Z): Creates a wireframe of a surface using gridlines.

**Pandas:**

Pandas is a robust, open-source Python library designed for data manipulation and analysis. It offers specialized data structures and operations for handling structured data, such as rows and columns, making it an essential tool for data scientists.

**Key Features for Data Visualization with Pandas:**

1. **Diverse Plotting Options:** Includes line plots, bar plots, histograms, box plots, and scatter plots.
2. **Customizable Visuals:** Titles, labels, and styles can be added to enhance clarity.
3. **Missing Data Handling:** Handles missing values gracefully, ensuring accurate visualizations.
4. **Integration with Matplotlib:** Leverages Matplotlib’s functionality for a wide range of plots.

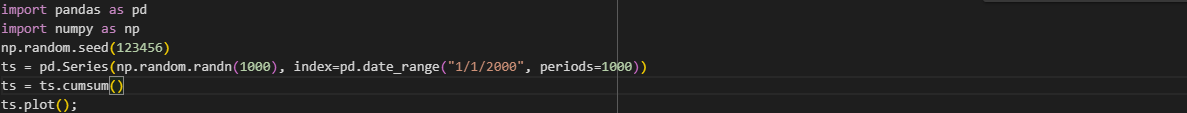
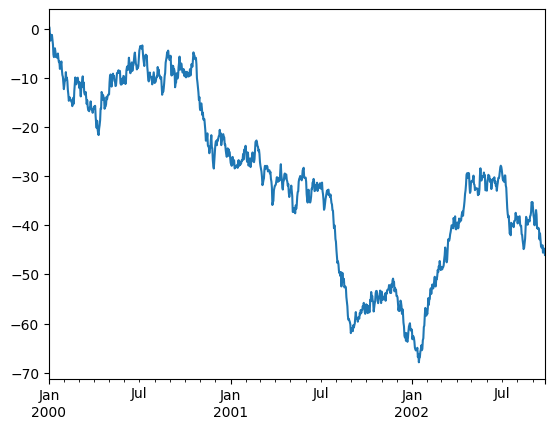
Basic plots in Pandas are created using the built-in plot() method. For instance, you can use df.plot(kind='hist') to generate a histogram or replace 'hist' with other keywords such as 'box', 'barh', etc.

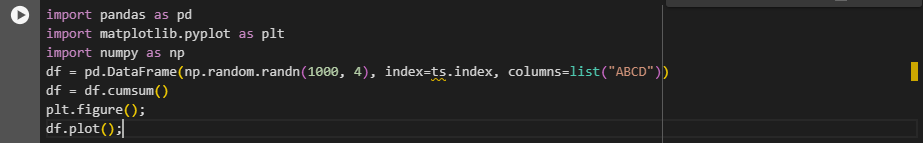
**Common Plot Types in Pandas:**

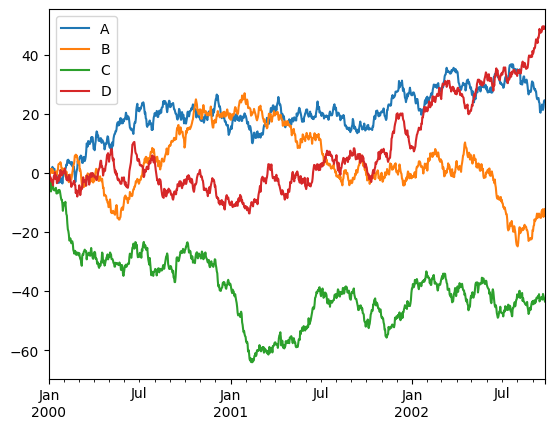
**PLOT:**

The “plot” method on Series and DataFrame is just a simple wrapper around “plt.

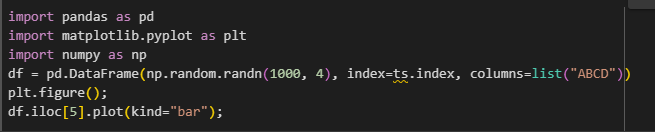
Let us look at a simple line plot in pandas:

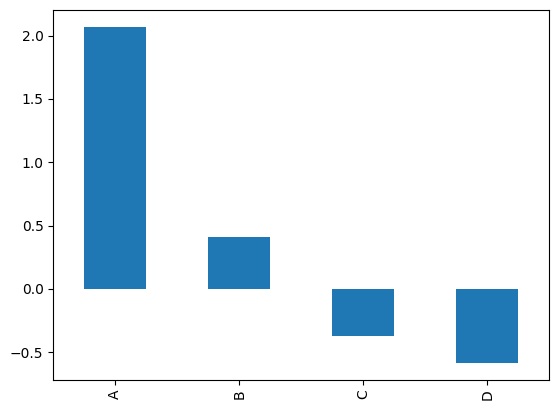
**SINGLE LINE:**

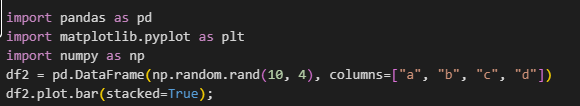
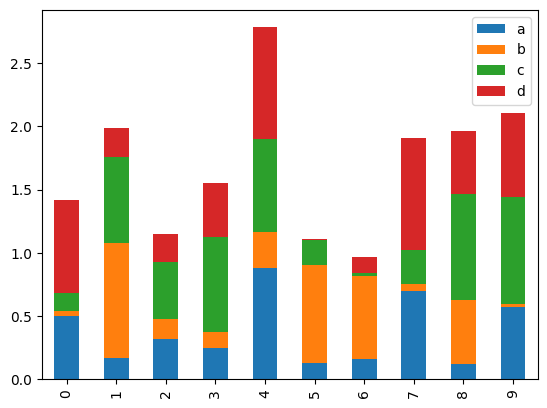
**MULTI-LINE:**

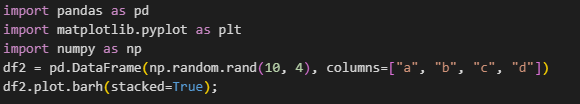
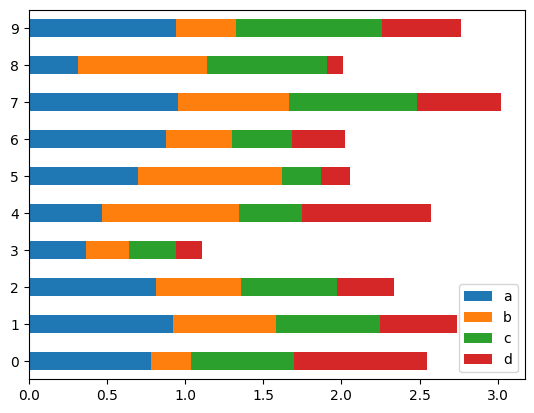


**Barplot:**



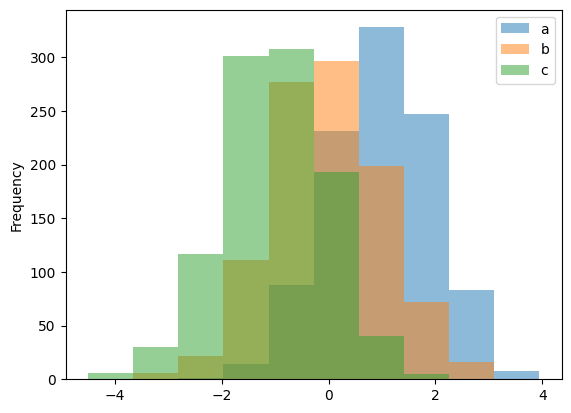
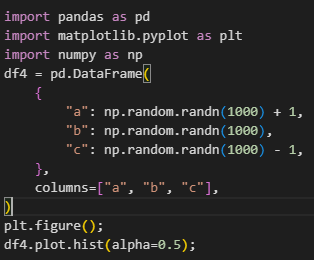


**Stacked barplot:**

**Stacked Horizontal barplot:**

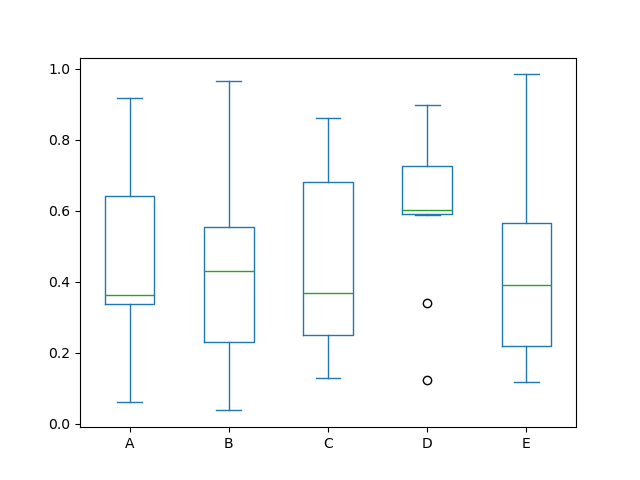
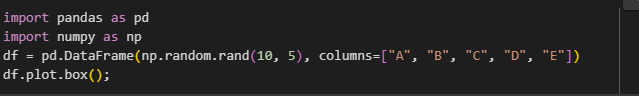
**Histogram:**

A histogram is a type of chart that shows the frequency distribution of [data points](https://www.techtarget.com/whatis/definition/data-point) across a continuous range of numerical values. The values are grouped into bin or buckets that are arranged in consecutive order along the horizontal [x-axis](https://www.techtarget.com/whatis/definition/x-and-y-coordinates) at the bottom of the chart. Each bin is represented by a vertical bar that sits on the x-axis and extends upward to indicate the number of data points within that bin.



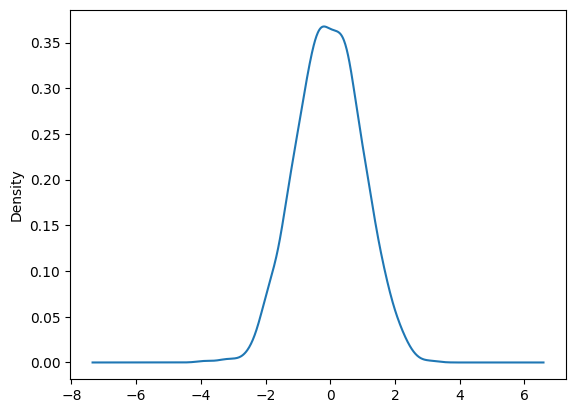
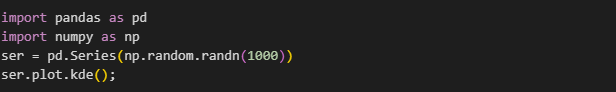
**Box plot:**

Box Plot is a graphical method to visualize data distribution for gaining insights and making informed decisions. Box plot is a type of chart that depicts a group of numerical data through their quartiles.

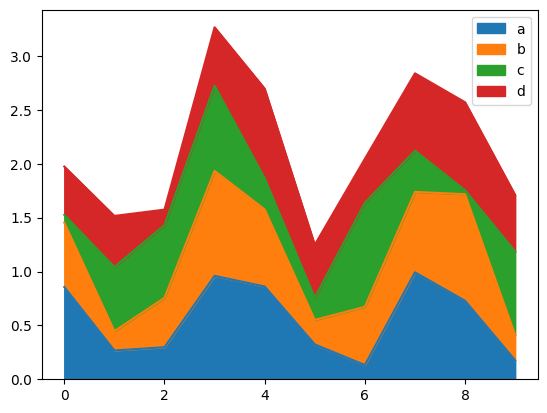
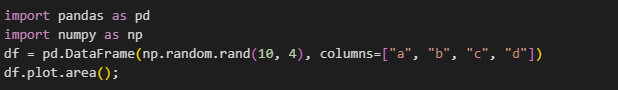


**kde or density plot:**

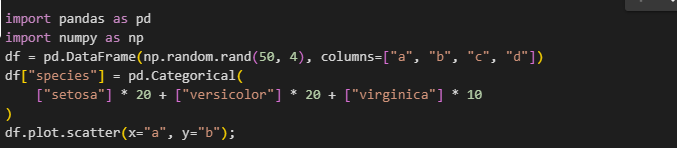
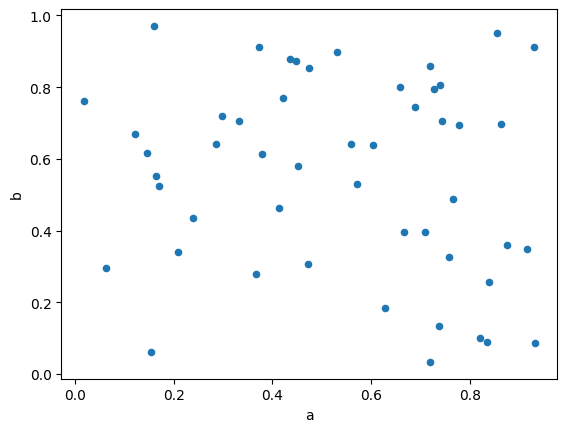
A density plot is a representation of the distribution of a numeric variable. It uses a kernel density estimate to show the probability density function of the variable (see more). It is a smoothed version of the histogram and is used in the same concept.



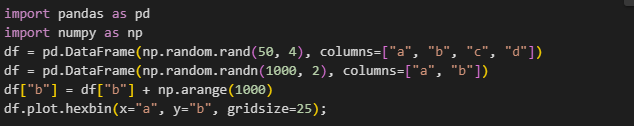
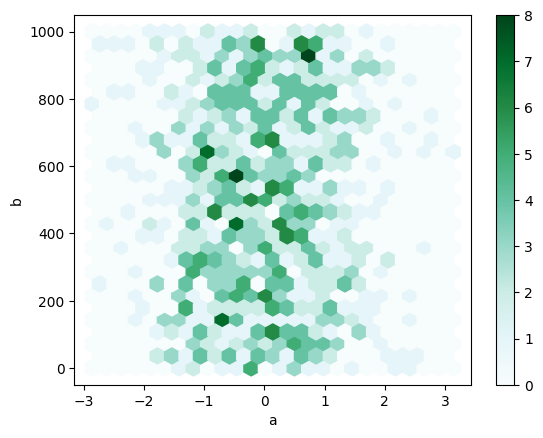
**Area Chart:**

An area chart or area graph displays graphically quantitative data. It is based on the line chart. The area between axis and line are commonly emphasized with colors, textures and hatchings. Commonly one compares two or more quantities with an area chart.

**Scatter Plot:**

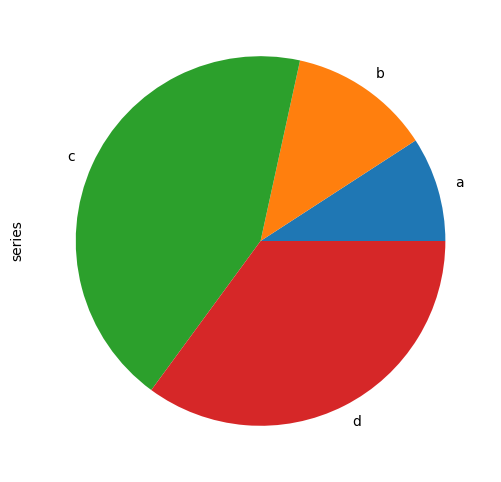
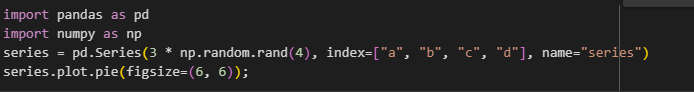
A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

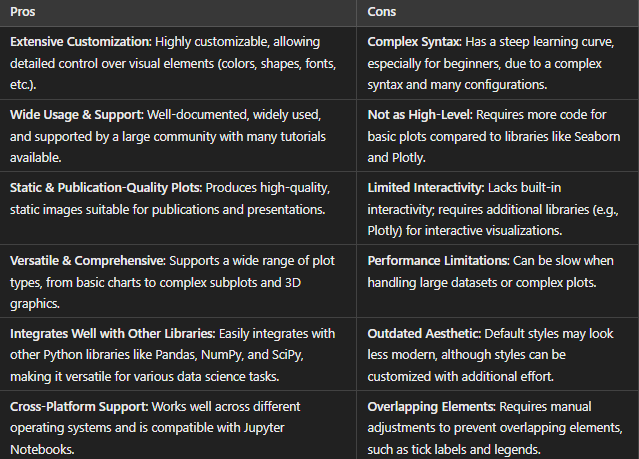
**Hexagonal Bin Plot:**

A hexagonal bin plot is a way to visualize data by grouping points into hexagonal bins and coloring the bins based on the number of points in each bin.

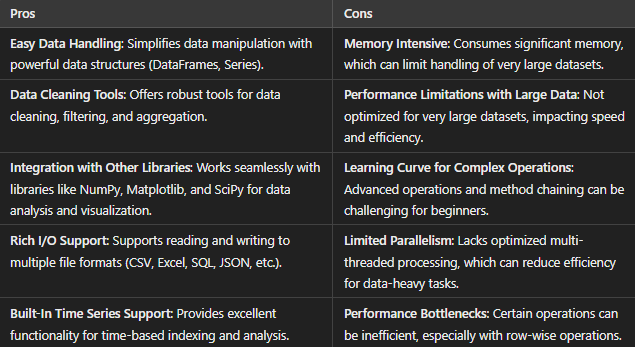
**Pie Chart:**

A pie chart is a type of graph representing data in a circular form, with each slice of the circle representing a fraction or proportionate part of the whole. All slices of the pie add up to make the whole equaling 100 percent and 360 degrees.

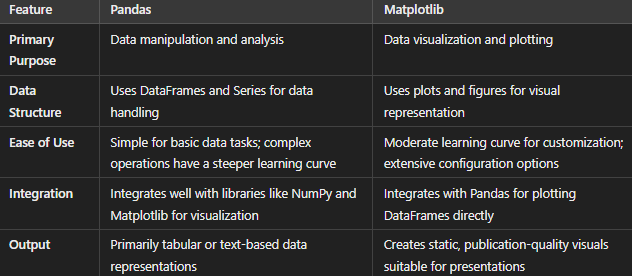


**MATPLOTLIB: Advantages and Disadvantages**

**PANDAS: Advantages and Disadvantages**



**Comparison Between Pandas and Matplotlib:**



**Applications of Pandas:**

1. **Data Cleaning:** Identifying and addressing missing or duplicate values.
2. **Data Aggregation:** Grouping and summarizing data.
3. **Time Series Analysis:** Analysing trends and patterns over time.
4. **Data Merging:** Combining datasets from various sources.
5. **Exploratory Data Analysis:** Generating insights from raw data.

**Applications of Matplotlib:**

1. **Data Visualization:** Creating detailed, high-quality charts.
2. **Trend Analysis:** Visualizing patterns and seasonality.
3. **Statistical Analysis:** Representing data distributions.
4. **Presentation Graphics:** Producing publication-ready visuals.
5. **Geospatial Mapping:** Plotting data points on geographical maps.