**MATPLOTLIB:**

**Matplotlib** is easy to use and an amazing visualizing library in Python. It is built on NumPy arrays and designed to work with the broader SciPy stack and consists of several plots like line, bar, scatter, histogram, etc.

## Key Features of Matplotlib:

1. **Versatility**: Matplotlib can generate a wide range of plots, including line plots, scatter plots, bar plots, histograms, pie charts, and more.
2. **Customization**: It offers extensive customization options to control every aspect of the plot, such as line styles, colours, markers, labels, and annotations.
3. **Integration with NumPy**: Matplotlib integrates seamlessly with NumPy, making it easy to plot data arrays directly.

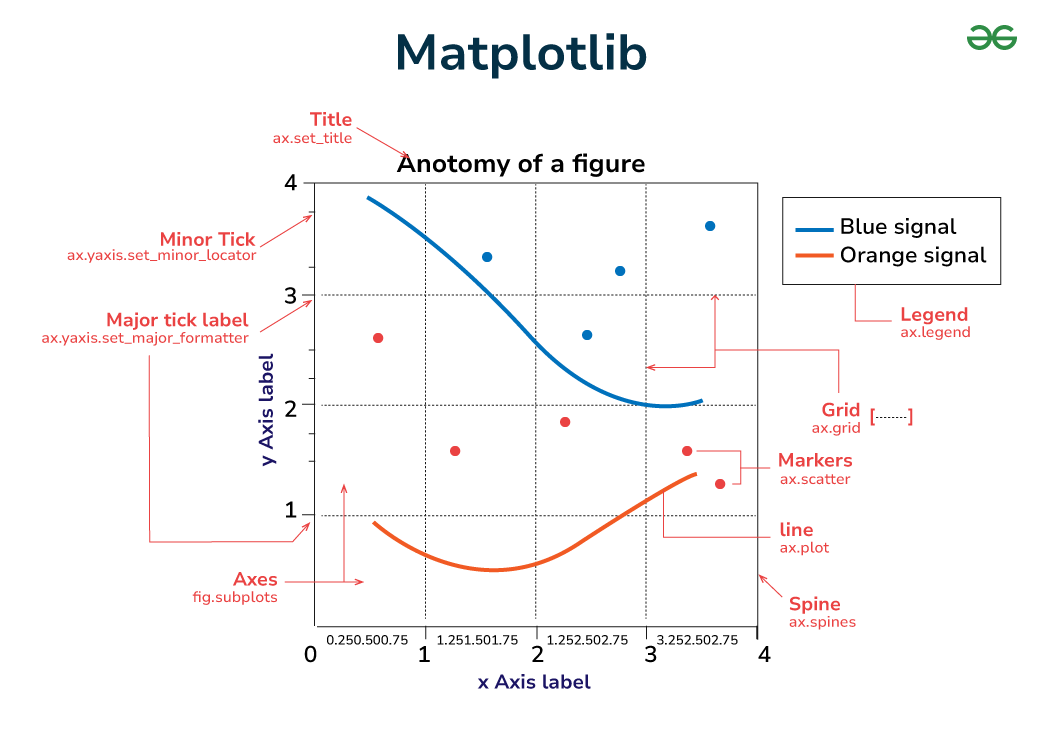
**4.Extensible**: Matplotlib is highly extensible, with a large ecosystem of add-on toolkits and extensions like Seaborn, Pandas plotting functions, and Basemap for geographical plotting.

5.**Cross-Platform**: It is platform-independent and can run on various operating systems, including Windows, macOS, and Linux.

**6.Interactive Plots**: Matplotlib supports interactive plotting through the use of widgets and event handling, enabling users to explore data dynamically.

**What is a Matplotlib Figure?**

In Matplotlib, a figure is the top-level container that holds all the elements of a plot. It represents the entire window or page where the plot is drawn.



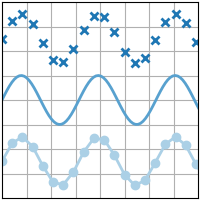
The graphs in Matplotlib are classified based on the data we are trying to visualise This results in the 5 main types of data that matplotlib can handle:

**1)Pairwise Data:**

Allows us to see both the distribution of a single variable and the relationship between two variables. Here are the types of graphs:

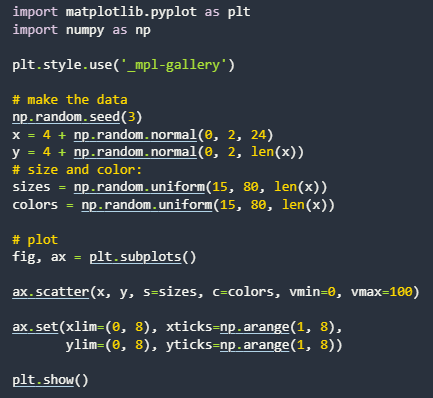
**plot(x,y):**

This helps us project a basic plot using given x and y coordinate values.



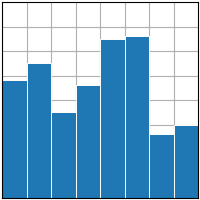
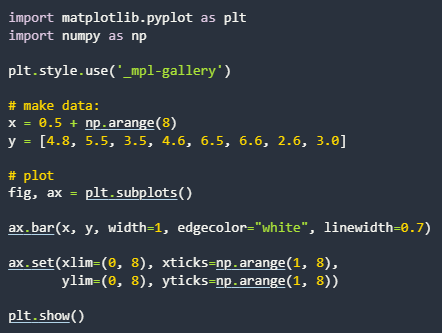
**scatter(x,y):**

This helps us create a scatter plot for pairwise data.



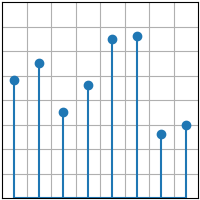
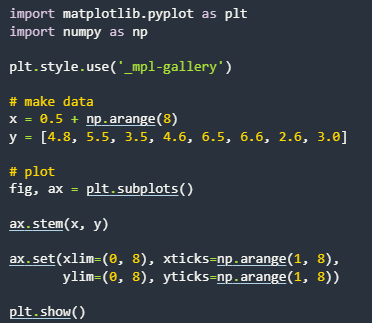
**bar(x,height):**

Helps you create a bar plot for a given batch of variables.



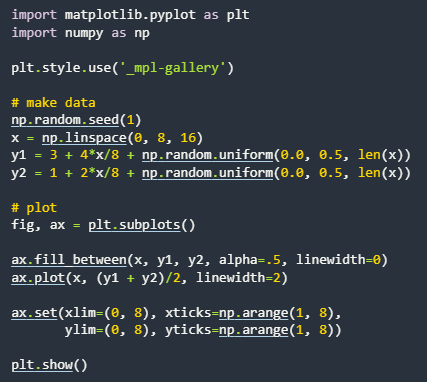
**stem(x,y):**

Helps classify either discrete or continuous variables.

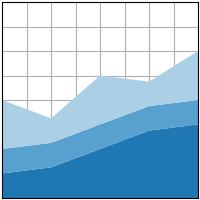
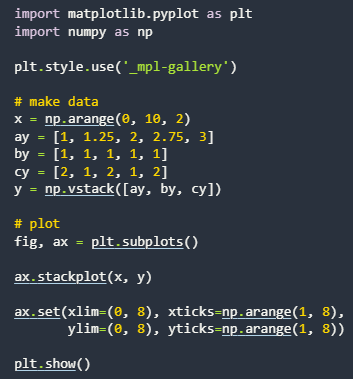


**fill\_between(x,y1,y2):**

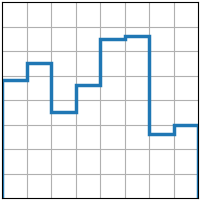
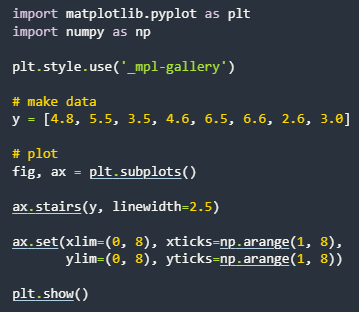
Fills the graph between a given range of values.



**stackplot(x,y):**



**stairs(values):**

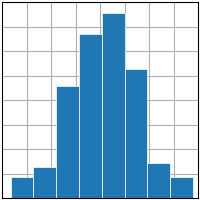
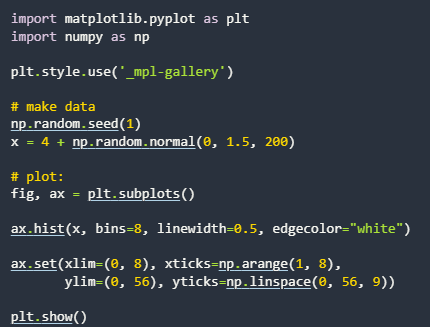


**2)Statistical Distributions:**

It helps map data in a probabilistic format.Which help predict certain outcomes.

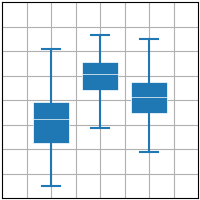
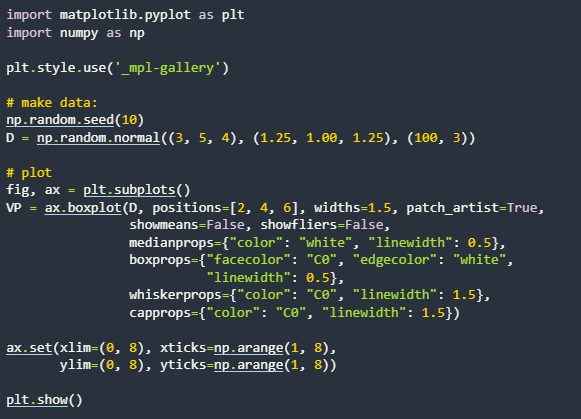
**hist(x):**

Helps plot distribution of numeric values as series of bars.



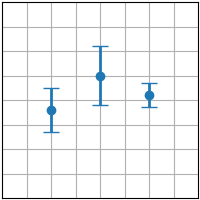
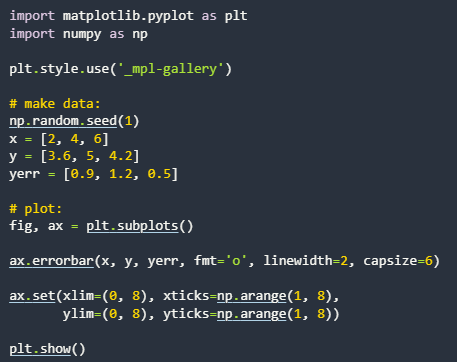
**boxplot(X):**

A graphical method to visualize data distribution for gaining insights and making informed decisions.



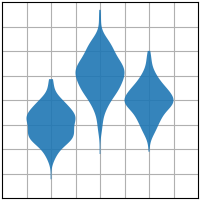
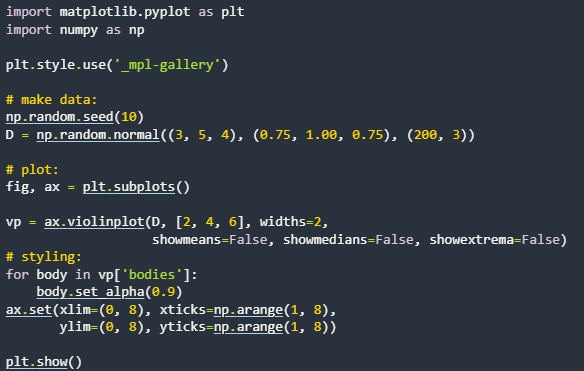
**errorbar(x,y,yerr,xerr):**

A graphical representation of the variability of data in a chart, and is used to indicate the uncertainty or error in a reported measurement.



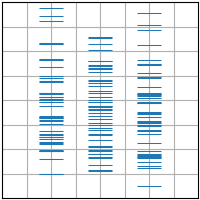
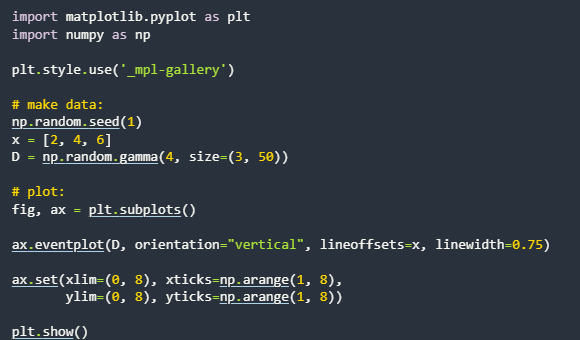
**violinplot(D):**

Depicts distributions of numeric data for one or more groups using density curves.



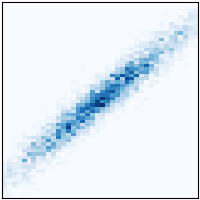
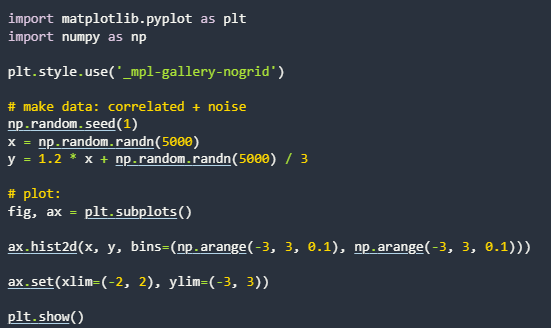
**eventplot(D):**

The sequence of events in a story, where each event is connected to the next by cause and effect.

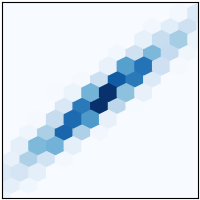
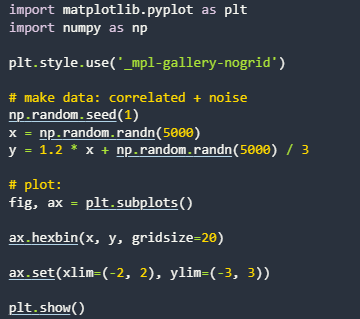


**hist2d(x,y):**

A 2D Histogram plot.



**hexbin(x,y,c):**Makes a hexagonal binning plot of points of x,y.



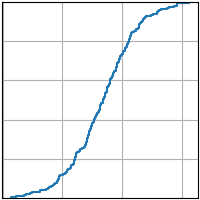
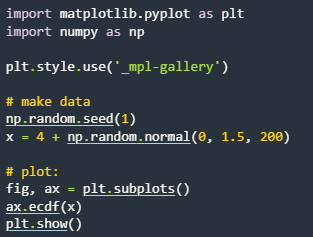
**pie(x):**

A way of summarizing a set of nominal data or displaying the different values of a given variable.



**ecdf(x):**

Compute and plot the empirical cumulative distribution function of x.

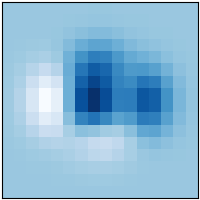
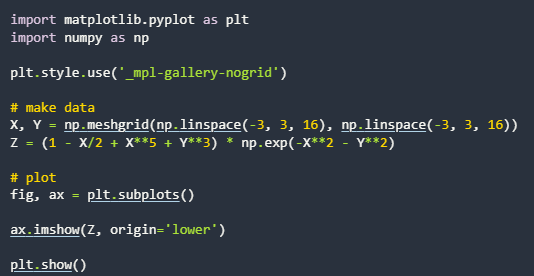


**3)Gridded Data:**

A collection of values or measurements that are organized in a grid at regular intervals.

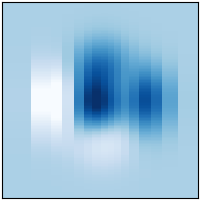
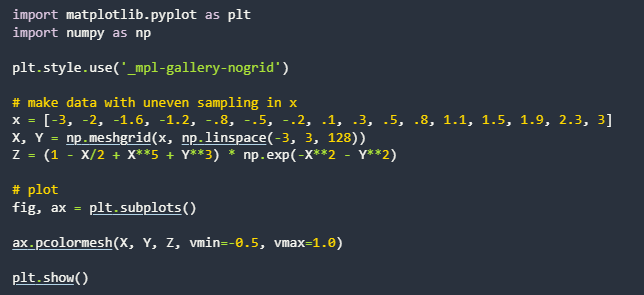
**imshow(Z):**

Display data as an image, i.e., on a 2D regular raster.



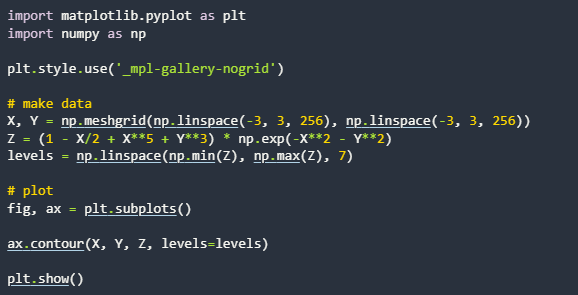
**pcolormesh(X,Y,Z):**

Create a pseudocolor plot with a non-regular rectangular grid.



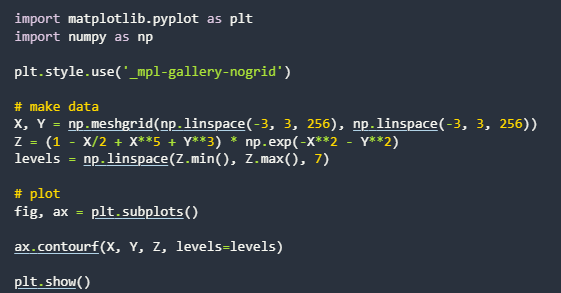
**contour(X,Y,Z):**

Plots Contour Lines.



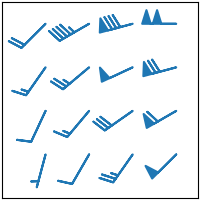
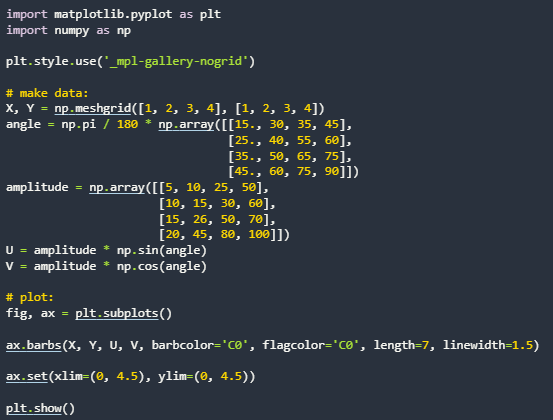
**contourf(X,Y,Z):**

Plot filled contours.



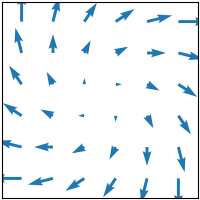
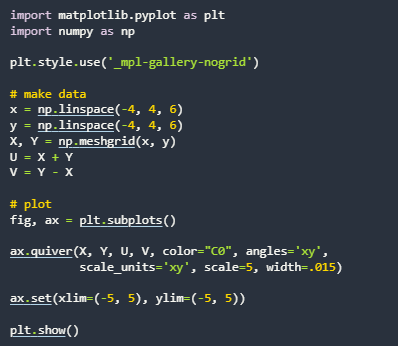
**barbs(X,Y,U,V):**

Plot a 2D field of wind barbs.



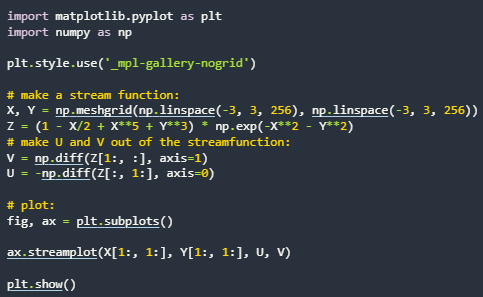
**quiver(X,Y,U,V):**

A field of arrows.



**steamplot(X,Y,U.V):**

Draw Streamlines of a vector flow.

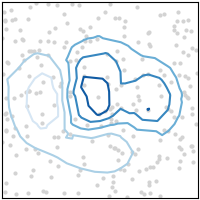
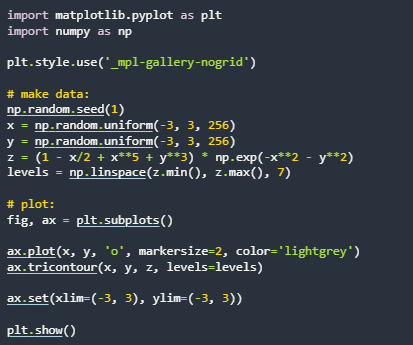


**4)Irregularly Gridded Data:**

An unstructured grid or irregular grid is a tessellation of a part of the Euclidean plane or Euclidean space by simple shapes, such as triangles or tetrahedra, in an irregular pattern.

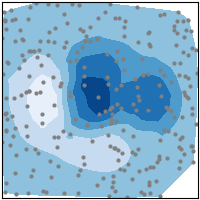
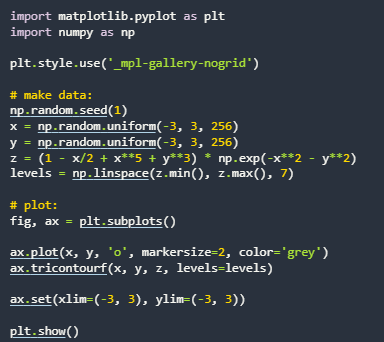
**tricontour(x,y,z):**

Drawing contour lines on an unstructured triangular grid.



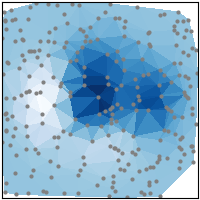
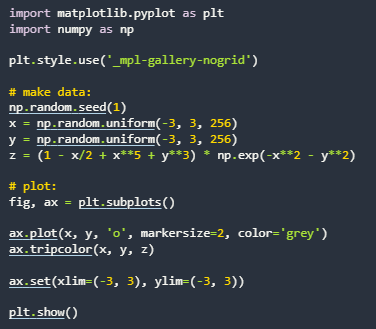
**tricontourf(x,y,z):**

Drawing filled contours on an unstructured triangular grid.



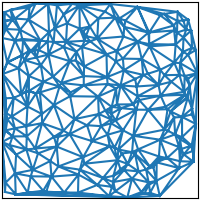
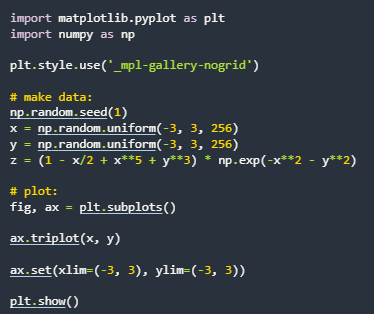
**tripcolor(x,y,z):**

Create a pseudocolor plot of an unstructured triangular grid.



**triplot(x.y):**

Draw an unstructured triangular grid as lines and/or markers.

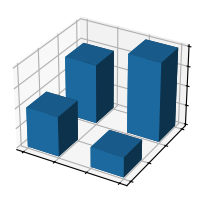
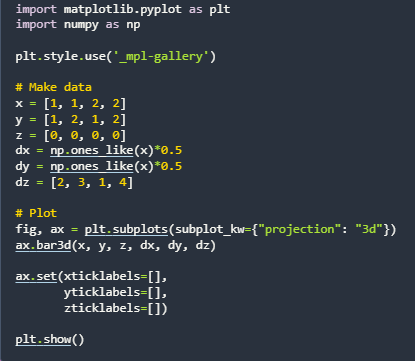


**5)3D and Volumetric Data:**

a collection of samples in 3D space, where each sample has coordinates and a value that represents a property like color, density, or pressure.

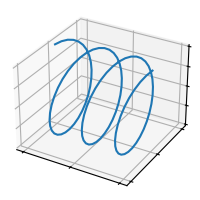
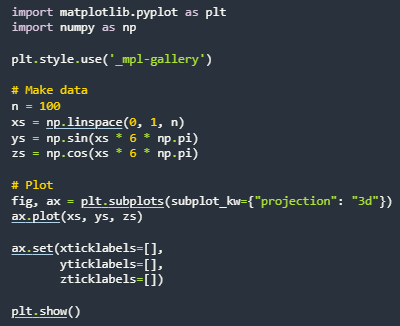
**barplot3d(x,y,z,dx,dy,dz):**

This method creates three-dimensional barplot where the width, depth, height, and color of the bars can all be uniquely set.



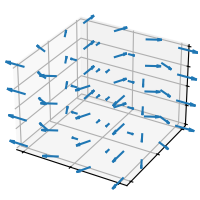
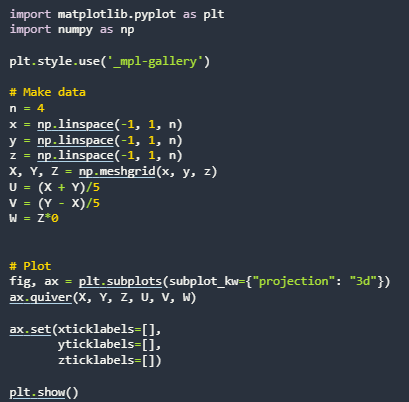
**plot(xs,ys,zs):**

Draws the basic plot in a 3d axis.



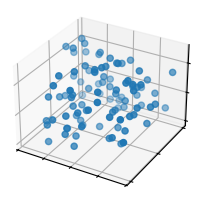
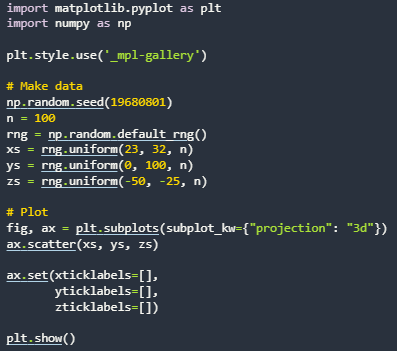
**quiver(X,Y,Z,U,V,W):**

Plots quivers in a 3d field.



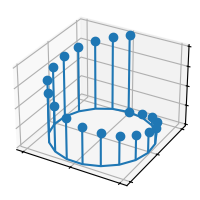
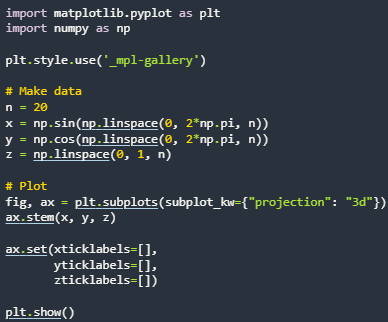
**scatter(xs,ys.zs):**

Plots a scatter plot in a 3d field.



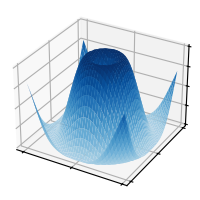
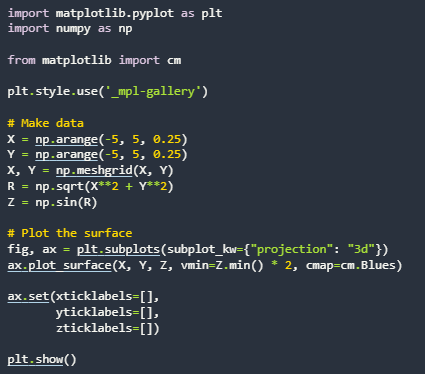
**stem(x,y,z):**

Plots a stem graph in a 3d field.



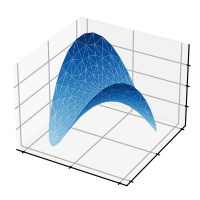
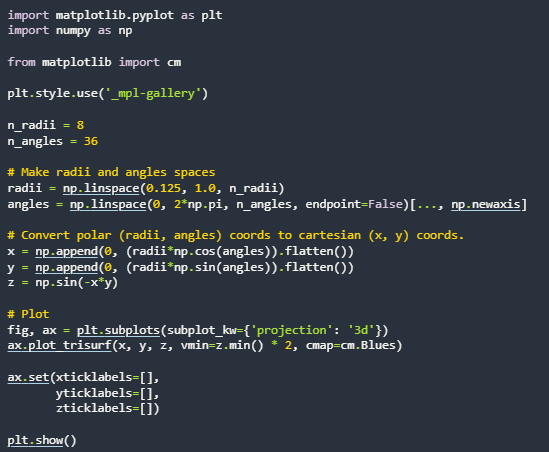
**plot\_surface(X.Y,Z):**

Plots a 3D plane in a 3D field.



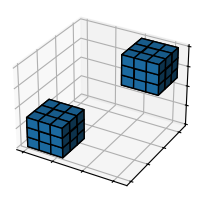
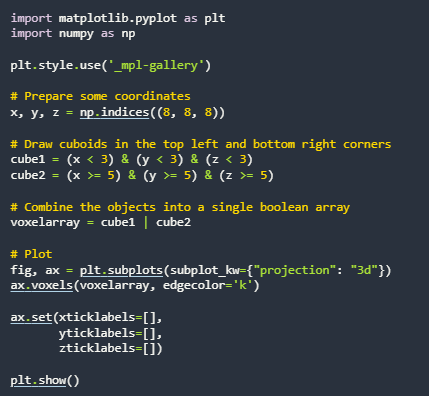
**plot\_trisurf(x,y,z):**

plots a triangulated surface in a 3D plane:



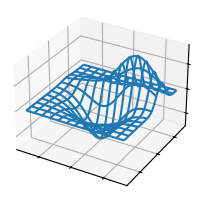
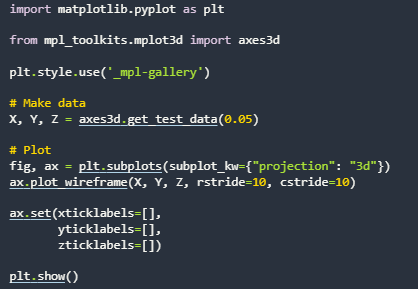
**voxels([x,y,z],filled):**

A voxel is a three-dimensional counterpart to a pixel . It prints said voxels in a 3D field.



**plot\_wireframe(X,Y,Z):**

Plots the wireframe of a specified surface using grid lines.



**PANDAS:**

Pandas is a powerful open-source data analysis and manipulation library for Python. It provides data structures and functions needed to work with structured data seamlessly. The library is particularly well-suited for handling labeled data, such as tables with rows and columns, making it a staple in the data science community.

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

Pandas offers several features that make it a great choice for data visualization:

1. **Variety of Plot Types**: Pandas supports various plot types, including line plots, bar plots, histograms, box plots, and scatter plots, catering to different visualization needs.
2. **Customization**: Users can customize plots by adding titles, labels, and styling, enhancing the readability and aesthetics of the visualizations.
3. **Handling of Missing Data**: Pandas efficiently handles missing data, ensuring that visualizations accurately represent the dataset without errors.
4. **Integration with Matplotlib**: Pandas seamlessly integrates with Matplotlib, allowing users to create a wide range of static, animated, and interactive plots.

To perform basic plotting with Pandas, **we can leverage the built-in plot () method**, which is a wrapper around Matplotlib’s plotting functions. You can also just call df.plot(kind=’hist’) or replace that kind argument with any of the key terms shown in the list above (e.g. ‘box’, ‘barh’, etc). Let us take a look at.

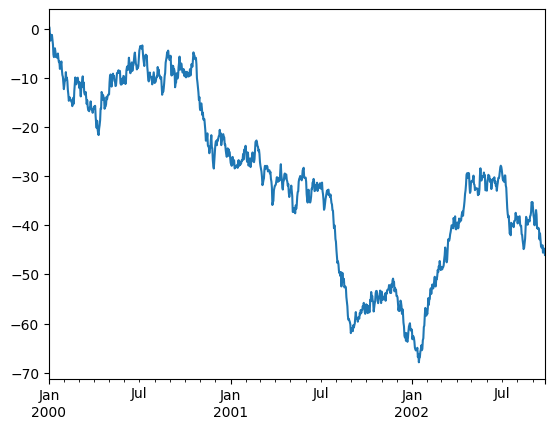
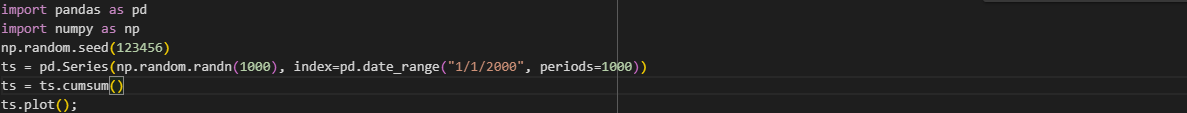
There are a total of 11 plots in pandas let us take a look at these plots.

**PLOT:**

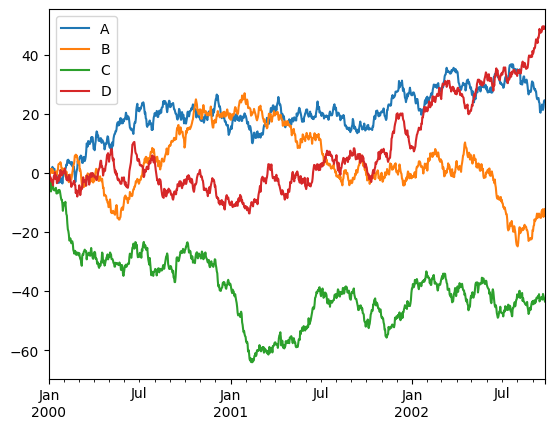
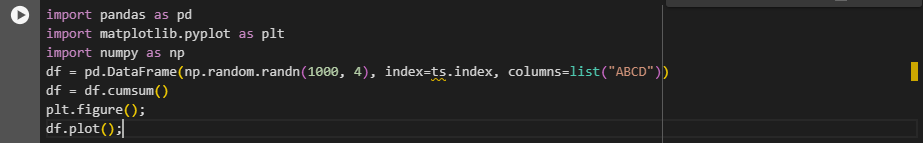
The “plot” method on Series and DataFrame is just a simple wrapper around “plt.

Let us look at a simple line plot in pandas:

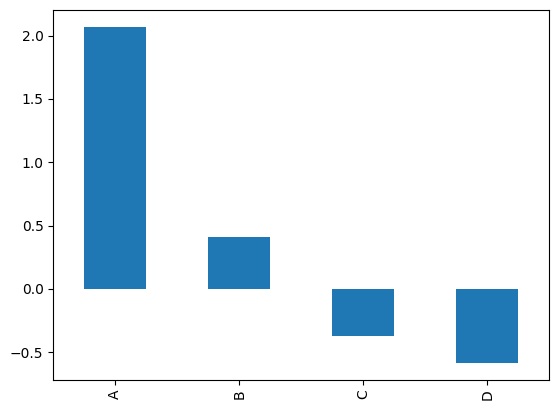
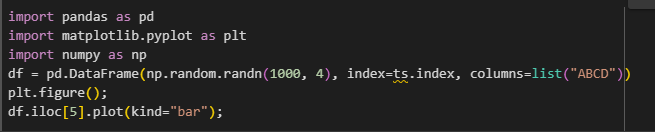
**SINGLE LINE:**



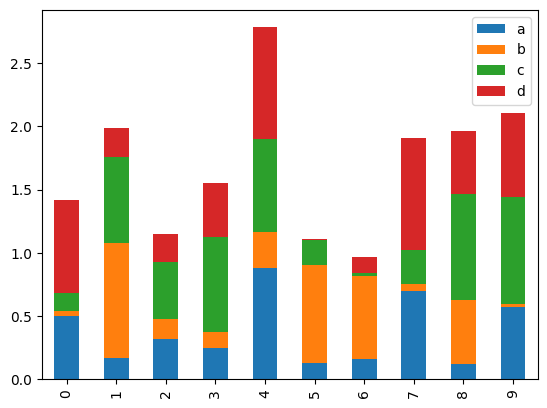
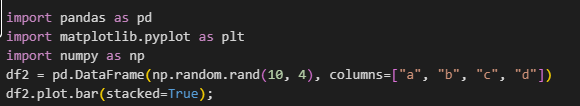
**MULTI-LINE:**



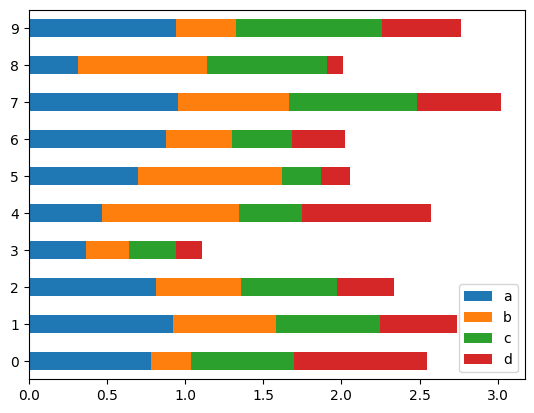
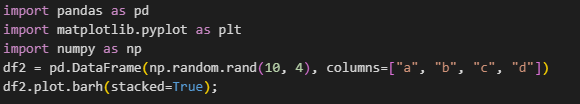
**Barplot:**



**Stacked barplot:**

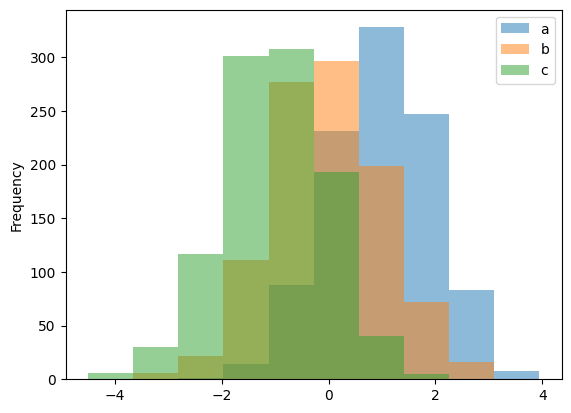
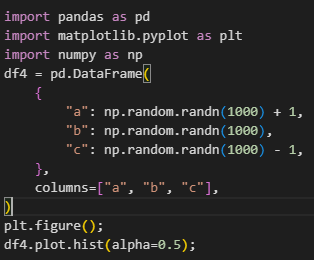


**Stacked Horizontal barplot:**



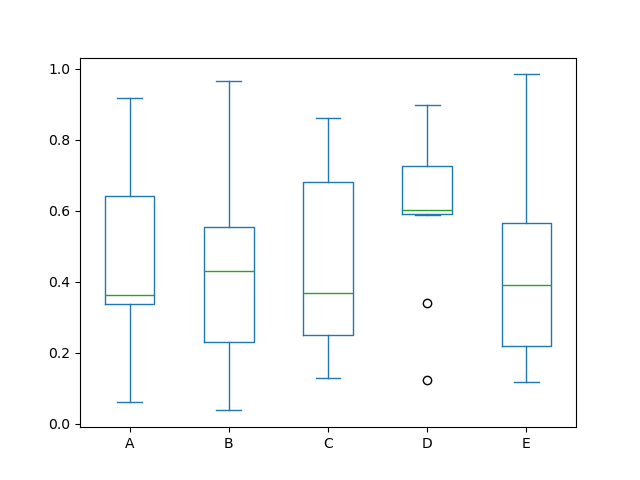
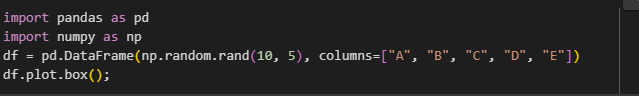
**Histogram:**

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.



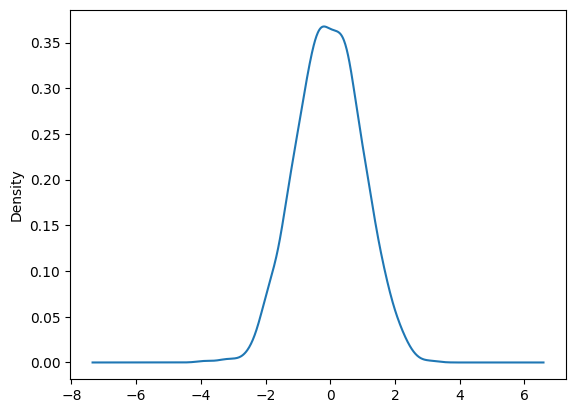
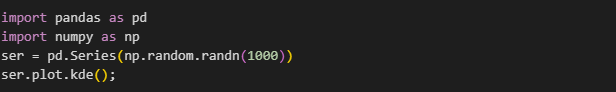
**Box plot:**

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.



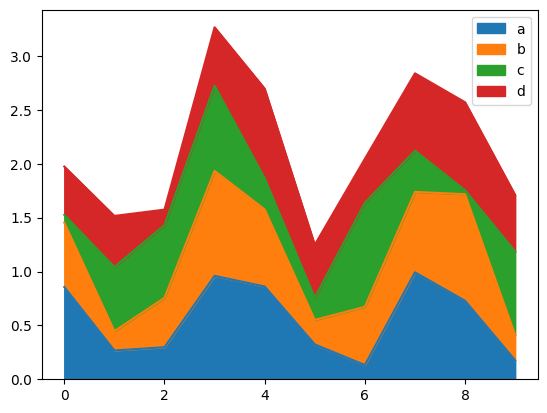
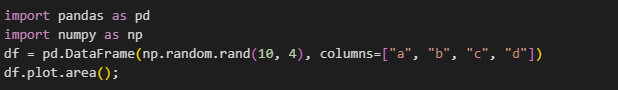
**kde or density plot:**

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.



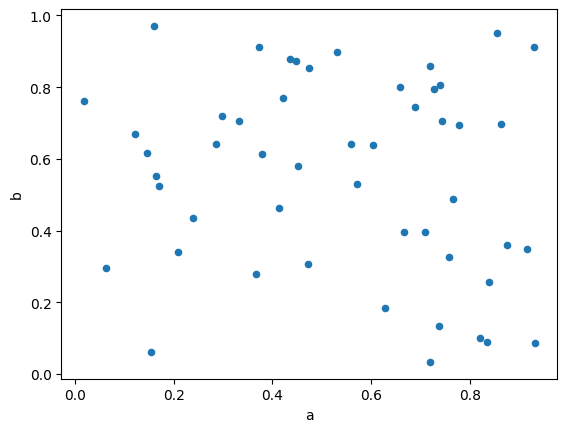
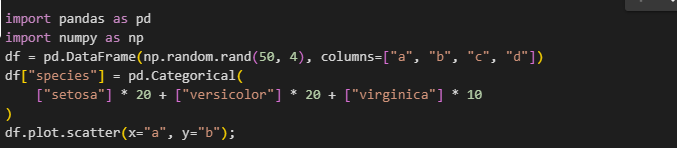
**Area Chart:**

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.



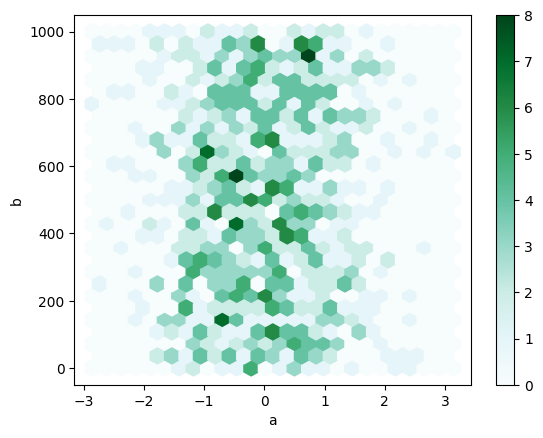
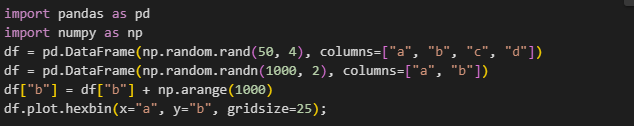
**Scatter Plot:**

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.



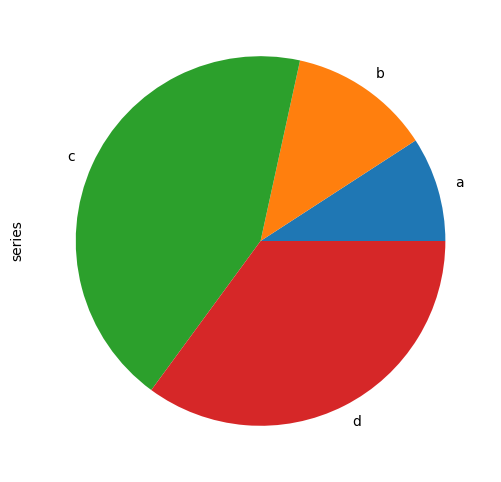
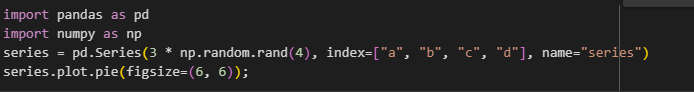
**Hexagonal Bin Plot:**

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.

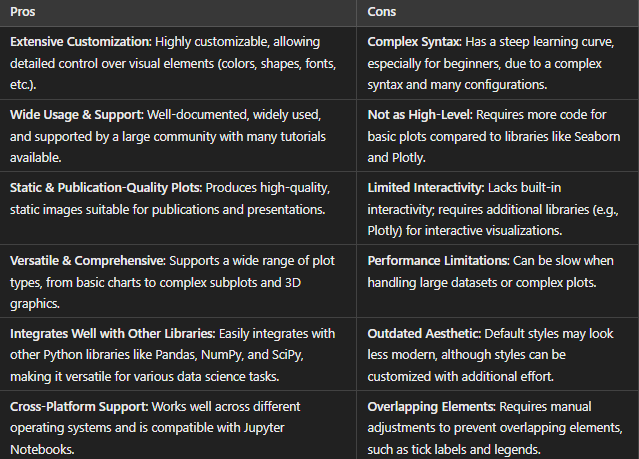


**Pie Chart:**

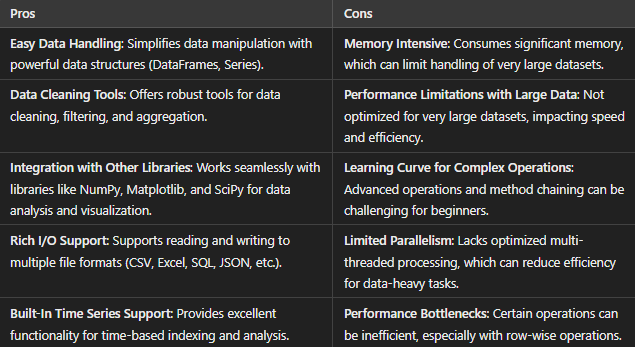
A pie chart is a type of graph representing data in a circular form, with each slice of the circle representing a fraction or proportionate part of the whole. All slices of the pie add up to make the whole equaling 100 percent and 360 degrees.



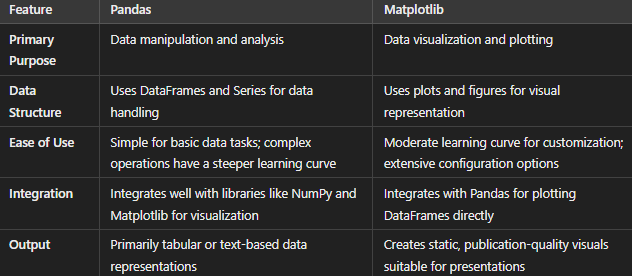
**MATPLOTLIB: Advantages and Disadvantages**



**PANDAS: Advantages and Disadvantages**



**Comparison Between Pandas and Matplotlib:**



### Pandas Applications:

1. **Data Cleaning and Preprocessing**: Handling missing values, duplications, and transforming data for analysis.
2. **Data Aggregation and Grouping**: Summarizing data through grouping, filtering, and aggregation functions.
3. **Time Series Analysis**: Processing and analyzing time-stamped data for trend analysis and forecasting.
4. **Data Merging and Joining**: Combining datasets from multiple sources using merging and joining techniques.
5. **Exploratory Data Analysis (EDA)**: Generating summary statistics and quick insights from data before detailed analysis.

### Matplotlib Applications:

1. **Data Visualization**: Creating a wide range of charts (line, bar, histogram, scatter plots) to represent data visually.
2. **Trend Analysis**: Plotting data over time to identify patterns, trends, and seasonal variations.
3. **Statistical Analysis**: Visualizing statistical distributions and relationships in data (box plots, histograms).
4. **Presentation Graphics**: Generating high-quality, publication-ready visuals for research papers, presentations, and reports.
5. **Geospatial Visualization**: Creating basic geographic maps by plotting data points with Matplotlib's basemap or using coordinates.