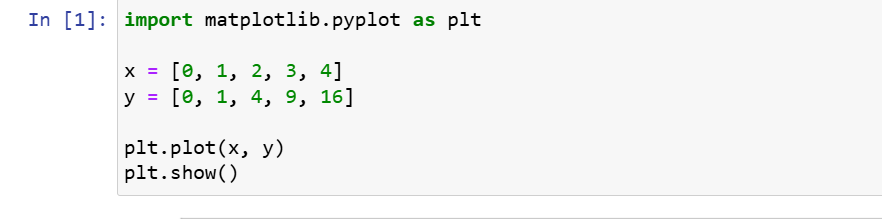
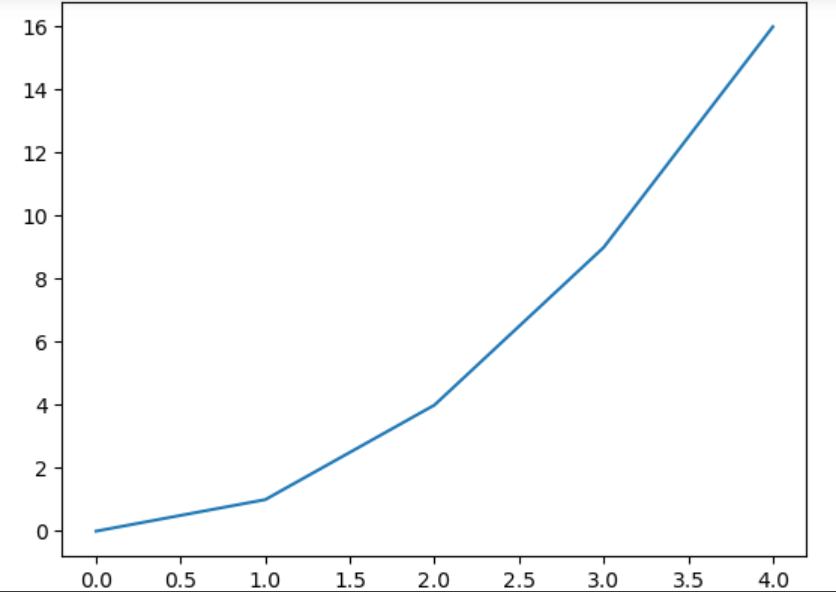
**VISUALIZATION**

Data visualization means turning complicated data into pictures like charts, graphs, or maps. These pictures help people see patterns, trends, and unusual points in the data. It makes big sets of information easier to understand. When data is shown visually, it's quicker to read and analyze. This helps people make smarter and faster decisions based on the data.

**Data Visualization with Matplotlib:**

Matplotlib is a powerful and versatile open-source plotting library for Python, designed to help users visualize data in a variety of formats.

**Example:**

****

**Components or Parts of Matplotlib:**

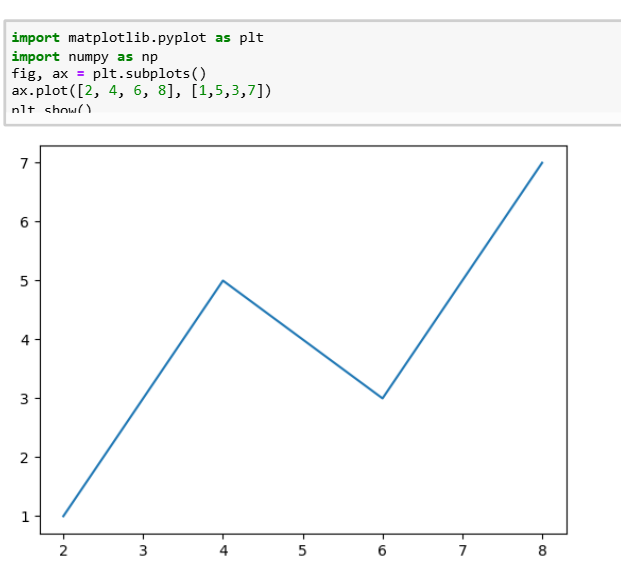
The parts of a Matplotlib include:

* **Figure**: The overarching container that holds all plot elements, acting as the canvas for visualizations.
* **Axes**: The areas within the figure where data is plotted; each figure can contain multiple axes.
* **Axis**: Represents the x-axis and y-axis, defining limits, tick locations, and labels for data interpretation.
* **Lines and Markers**: Lines connect data points to show trends, while markers denote individual data points in plots like scatter plots.
* **Title and Labels**: The title provides context for the plot, while axis labels describe what data is being represented on each axis.

**Matplotlib Pyplot:**

[Pyplot](https://www.geeksforgeeks.org/python/pyplot-in-matplotlib/) is a module within Matplotlib that provides a MATLAB-like interface for making plots. It simplifies the process of adding plot elements such as lines, images, and text to the axes of the current figure.

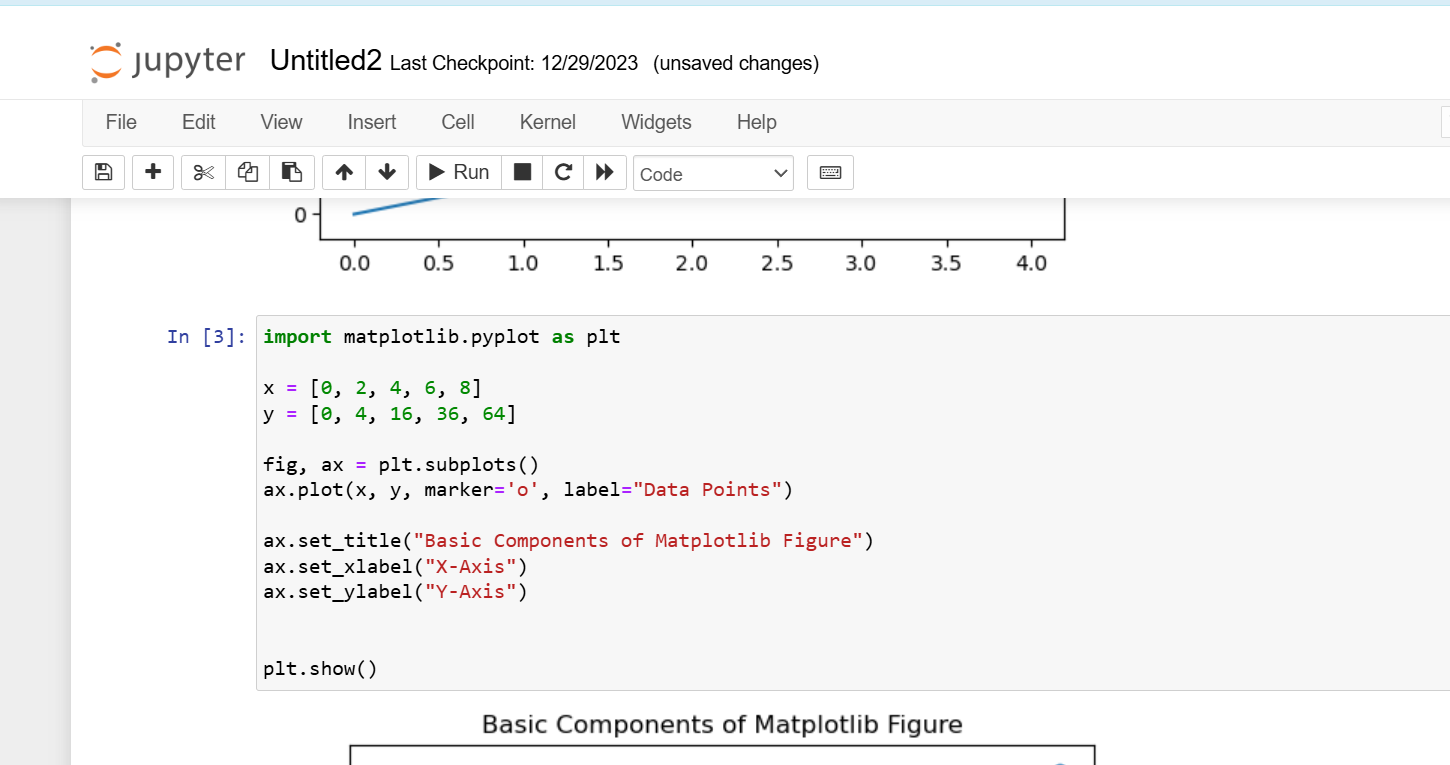
**Sample Example:**

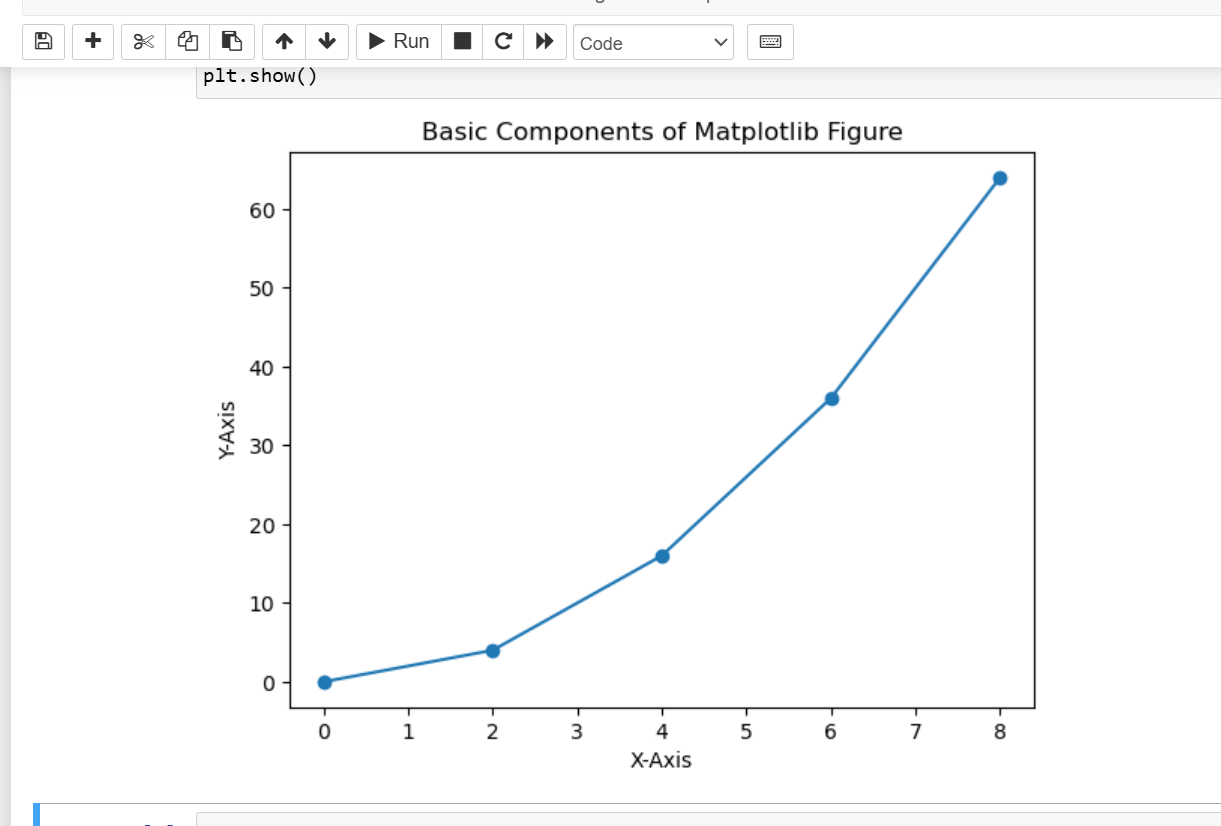
****

**Steps to Use Pyplot:**

* Import Matplotlib: Start by importing matplotlib.pyplot as plt.
* Create Data: Prepare your data in the form of lists or arrays.
* Plot Data: Use plt.plot() to create the plot.
* Customize Plot: Add titles, labels, and other elements using methods like plt.title(), plt.xlabel(), and plt.ylabel().
* Display Plot: Use plt.show() to display the plot.

**Example:**

****

****

**Different Types of Plots in Matplotlib:**

Matplotlib offers a wide range of plot types to suit various data visualization needs. Here are some of the most commonly used types of plots in Matplotlib:

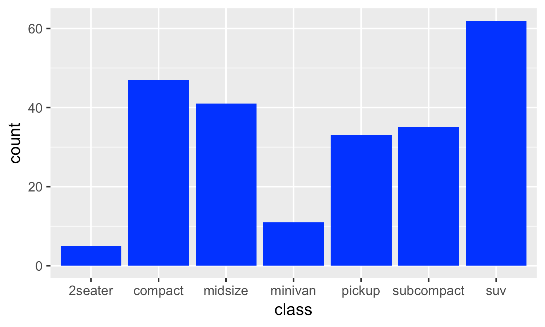
1. Line Graph

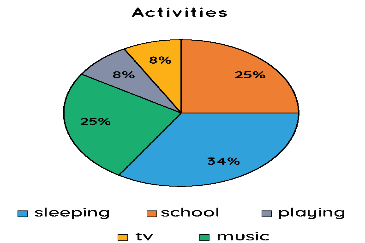
2. Bar Chart

3. Histogram

4. Scatter Plot

5. Pie Chart

6. 3D Plot

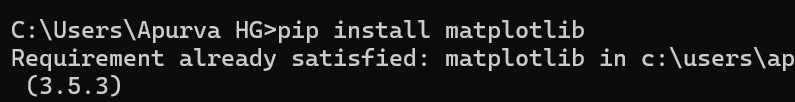


Bar chart Pie chart

**How to Install Matplotlib on python?**

**Installation Using PIP:**

pip install matplotlib command can be used to install it. Users who prefer to use pip can use the below command to install Matplotlib:



This will download and install the latest version of Matplotlib from the Python Package Index (PyPI).

**Verifying Installation:**

To check if Matplotlib has been installed correctly, run the following in your Python environment:

*import matplotlib  
print(matplotlib.\_\_version\_\_)*



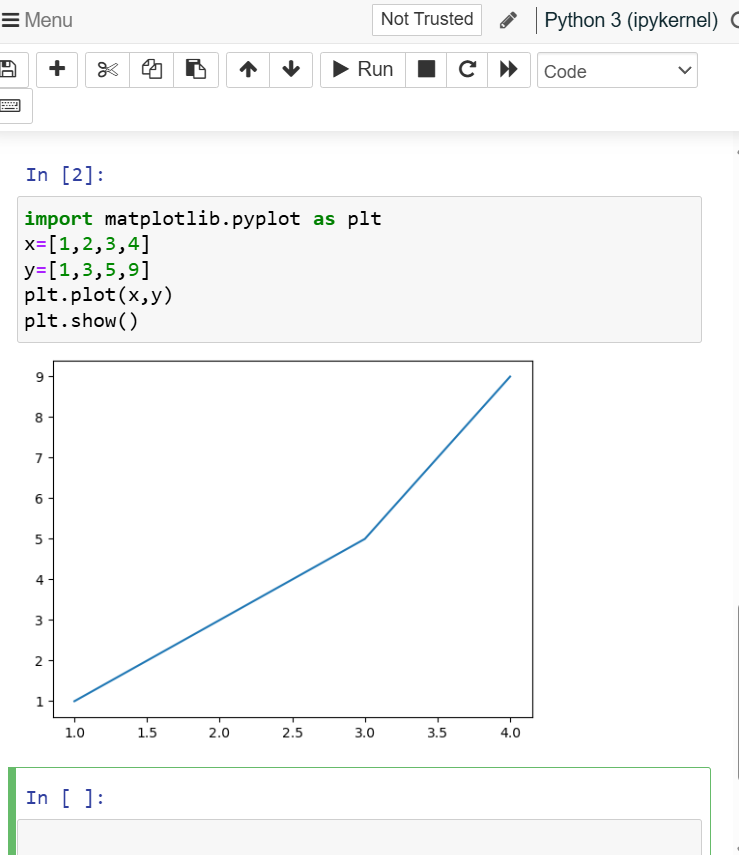
If it prints the version number, the installation was successful.

**Basic Plotting with Pyplot:**

1. **Line Plot:**

Line plots are simplest types of charts. The plot() function takes two lists: one for x-axis and one for y-axis and connects the points in a linear path.

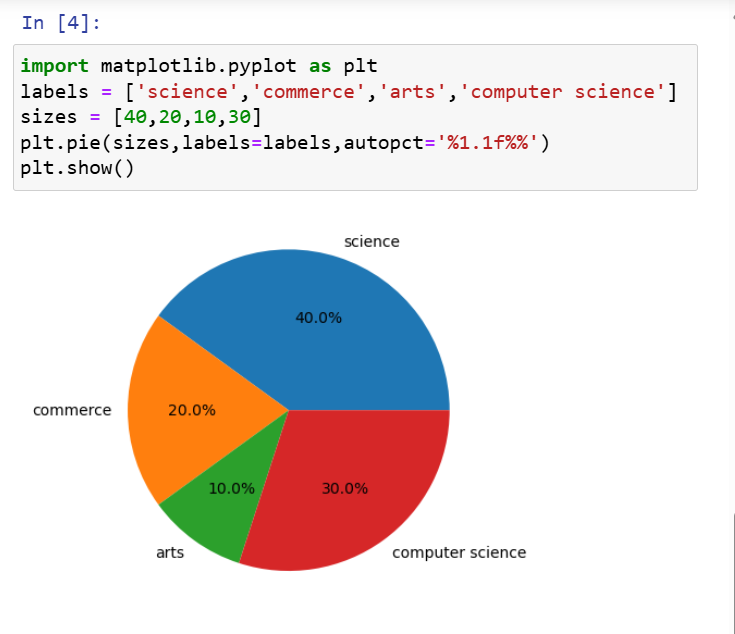
**Example:**

****

* **plt.plot(x, y):**creates a line plot by connecting **x**and **y**data points.
* **plt.show():** displays final plot window.

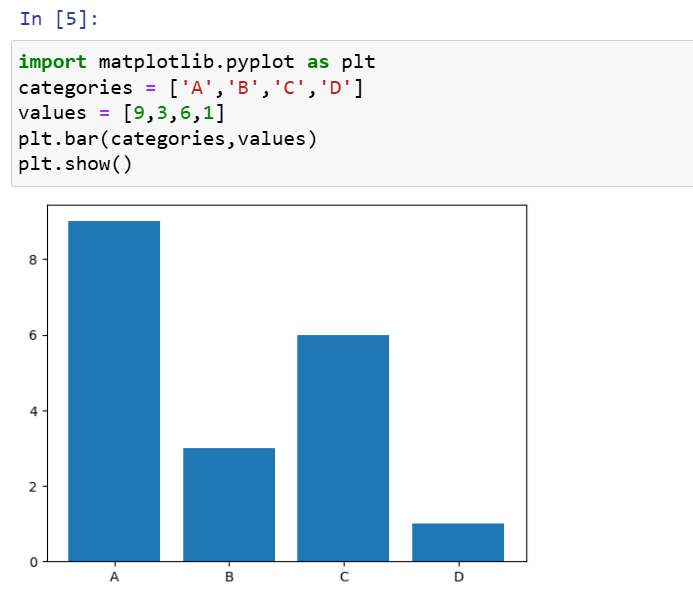
1. **Pie Charts:**

Pie charts show the share of each category in a whole. The pie() function takes a list of values and optional labels to represent each slice of the pie.



**3. Bar Charts:**

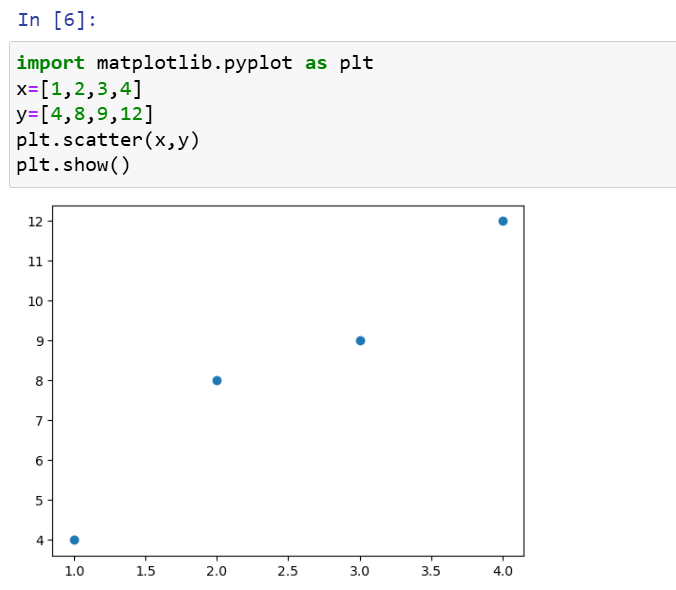
Bar charts are ideal for comparing quantities across categories. The**bar()** function takes two lists one for **category**labels (x-axis) and one for their corresponding **values**(y-axis).

**Example:**

**4. Scatter Plots**

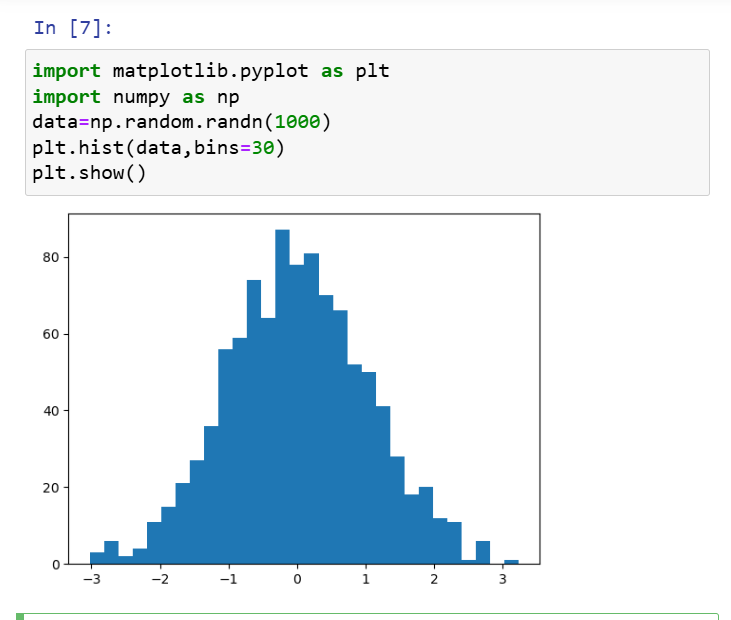
Scatter plots are used for displaying individual data points and showing relationship between two variables. The**scatter()** function takes two lists like plot(), but it only plots the individual points without connecting them.

**Example:**



**5. Histograms:**

Histograms are used to visualize distribution of a dataset. The hist() function takes a list or array of numerical data and divides it into bins.

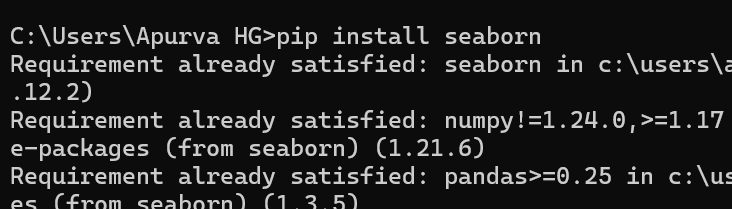
* **np.random.randn(1000):** generates **1000**random numbers from a standard normal distribution.
* **plt.hist(data, bins=30):** creates a histogram of the data with 30 bins.

**Seaborn**

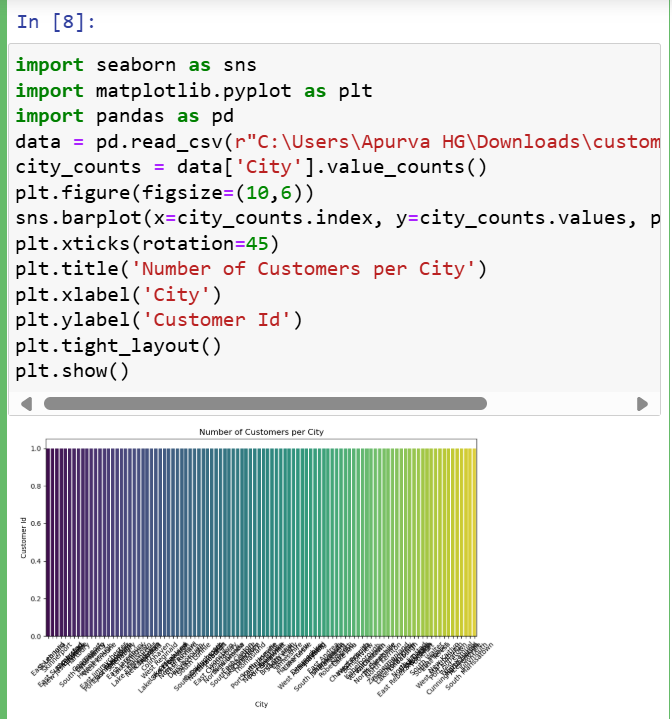
**Seaborn**is a high-level interface built on top of the Matplotlib. It provides beautiful design styles and color palettes to make more attractive graphs.

To install seaborn type the below command in the terminal.

**pip install seaborn**

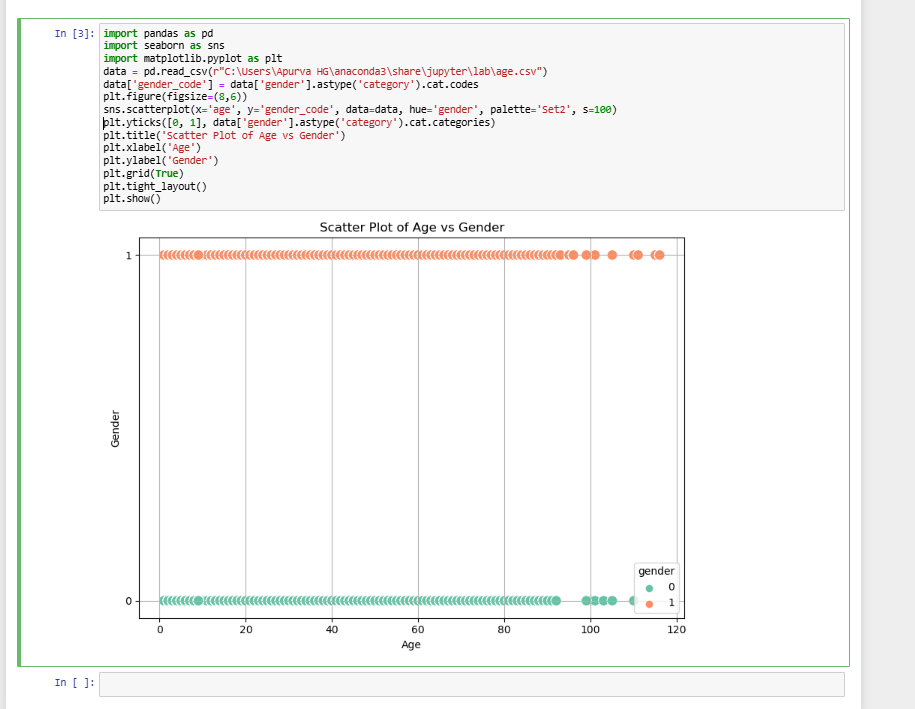


Seaborn is built on the top of Matplotlib, therefore it can be used with the Matplotlib as well. Using both Matplotlib and Seaborn together is a very simple process.

**Example:**

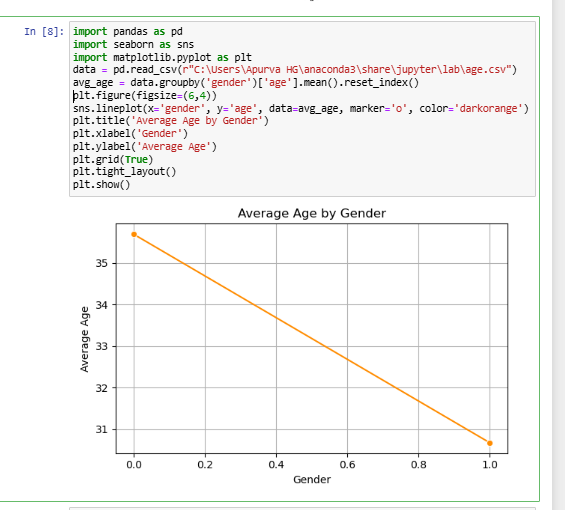
**Scatter Plot**

[Scatter plot](https://www.geeksforgeeks.org/python/scatterplot-using-seaborn-in-python/) is plotted using the scatterplot() method. This is similar to Matplotlib, but additional argument data is required.

****

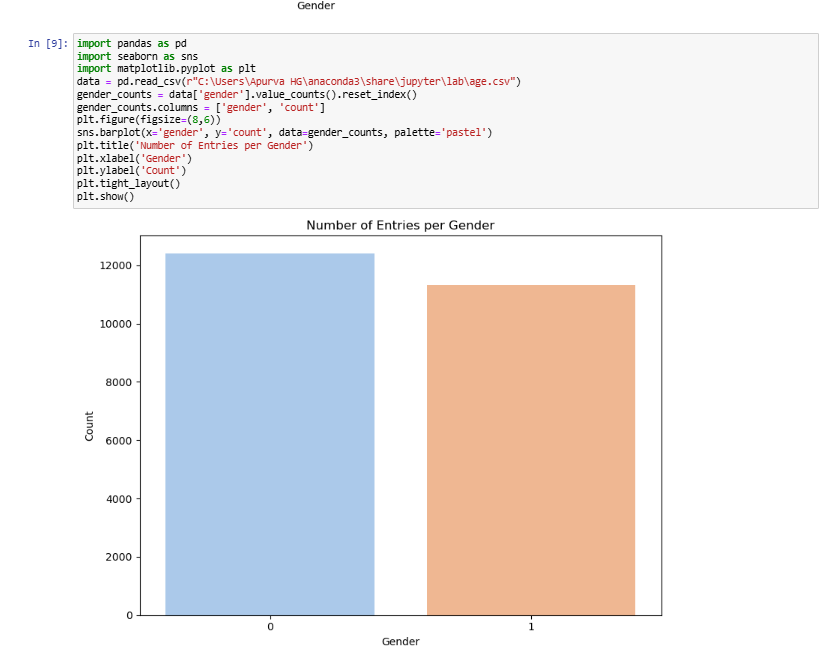
**Line Plot:**

[Line Plot](https://www.geeksforgeeks.org/data-visualization/data-visualization-with-seaborn-line-plot/) in Seaborn plotted using the [lineplot()](https://www.geeksforgeeks.org/python/seaborn-lineplot-method-in-python/) method.  In this, we can pass only the data argument also.

****

**Bar Plot**

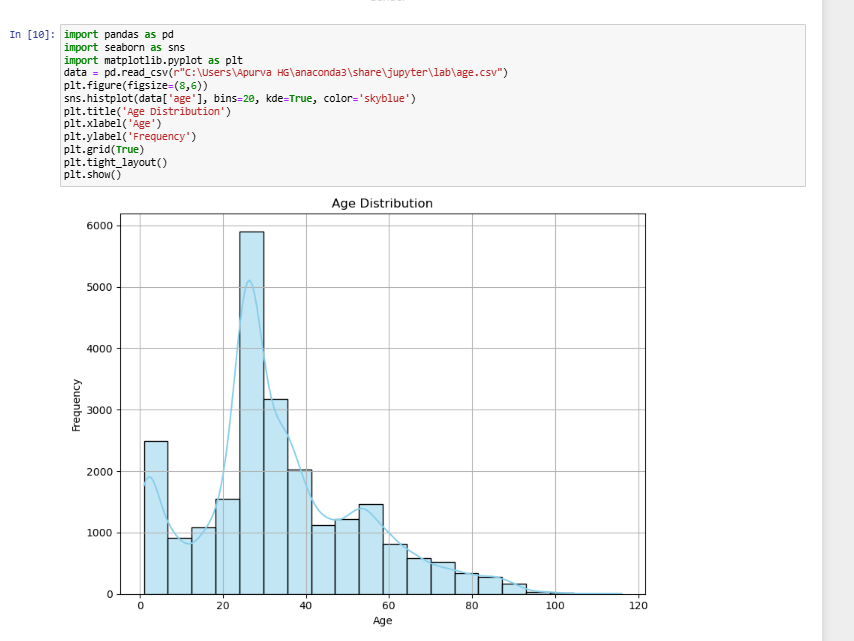
[Bar Plot](https://www.geeksforgeeks.org/python/barplot-using-seaborn-in-python/) in Seaborn can be created using the [barplot()](https://www.geeksforgeeks.org/data-visualization/seaborn-barplot-method-in-python/) method.

**Example:**

**Histogram:**

The histogram in Seaborn can be plotted using the histplot() function.

**Example:**

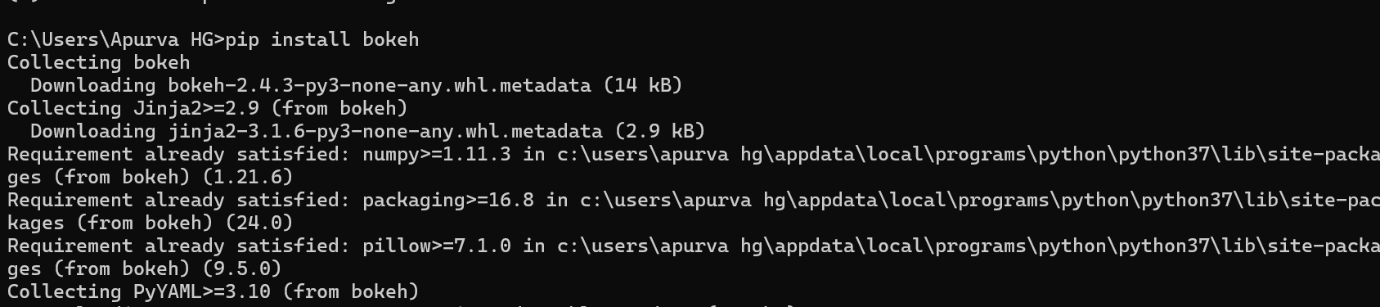
****

**Bokeh**

Bokeh renders its plots using HTML and JavaScript that uses modern web browsers for presenting elegant, concise construction of novel graphics with high-level interactivity.

To install this type the below command in the terminal.

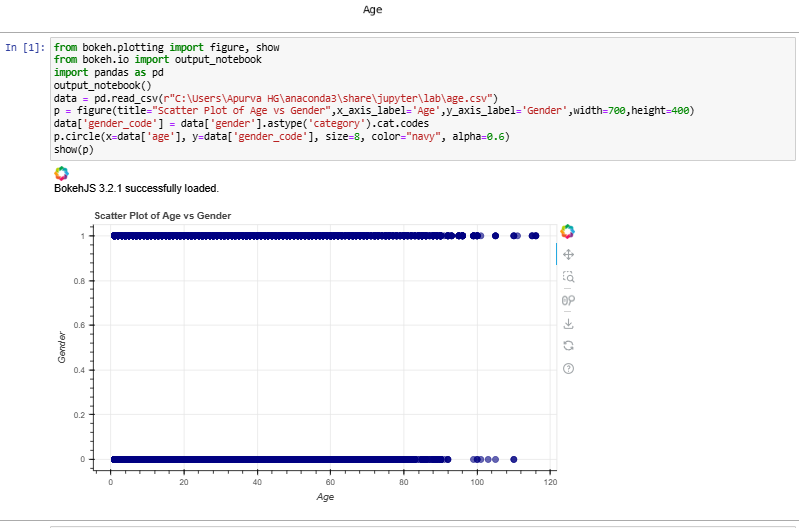
**pip install bokeh**

****

**Scatter Plot**

[Scatter Plot](https://www.geeksforgeeks.org/data-visualization/python-bokeh-plotting-a-scatter-plot-on-a-graph/) in Bokeh can be plotted using the scatter() method of the plotting module. Here pass the x and y coordinates respectively.

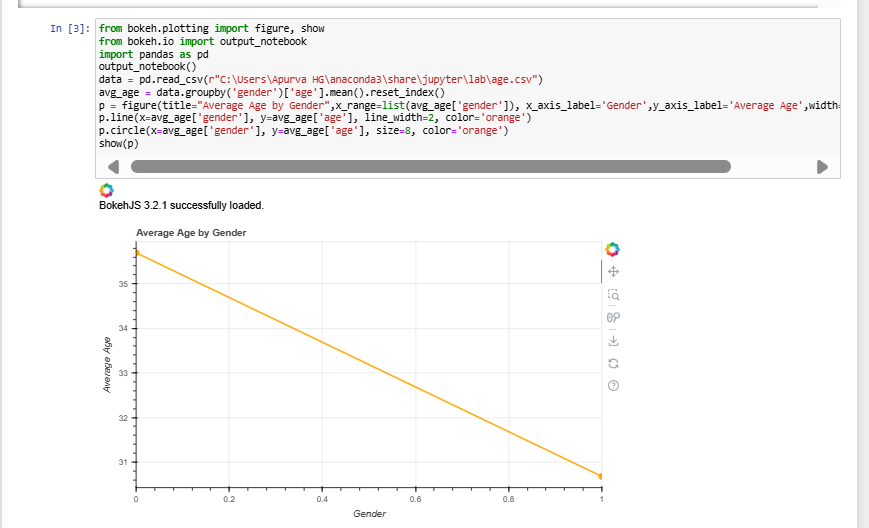
**Example:**

****

**Line Chart:**

 A [line plot](https://www.geeksforgeeks.org/data-visualization/python-bokeh-plotting-a-line-graph/) can be created using the line() method of the plotting module.

**Example:**

****

**Bar Chart:**

Bar Chart can be of two types horizontal bars and vertical bars. Each can be created using the hbar() and vbar() functions of the plotting interface respectively.

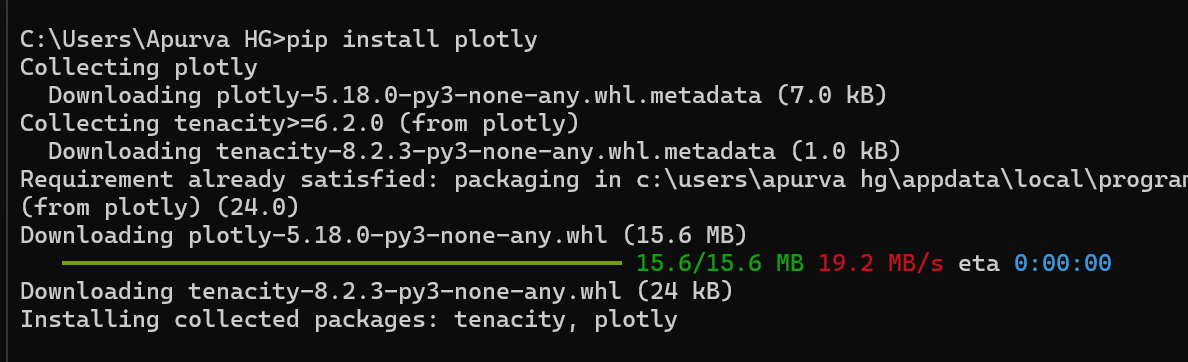
**Example:**

****

**Plotly**

* Plotly has hover tool capabilities that allow us to detect any outliers or anomalies in numerous data points.
* It allows more customization.
* It makes the graph visually more attractive.

To install it type the below command in the terminal.

**pip install plotly**

**Scatter Plot:**

[Scatter plot](https://www.geeksforgeeks.org/python/scatter-plot-using-plotly-in-python/) in Plotly can be created using the [scatter()](https://www.geeksforgeeks.org/python/plotly-express-scatter-function-in-python/) method of plotly.express. Like Seaborn, an extra data argument is also required here.

**Example:**

****

**Line Chart:**

[Line plot](https://www.geeksforgeeks.org/python/line-chart-using-plotly-in-python/) in Plotly is much accessible and illustrious annexation to plotly which manage a variety of types of data and assemble easy-to-style statistic. With [px.line](https://www.geeksforgeeks.org/python/plotly-express-line-function-in-python/) each data position is represented as a vertex.

**Example:**

****

**Bar Chart:**

Bar Chart in Plotly can be created using the bar() method of plotly.express class.

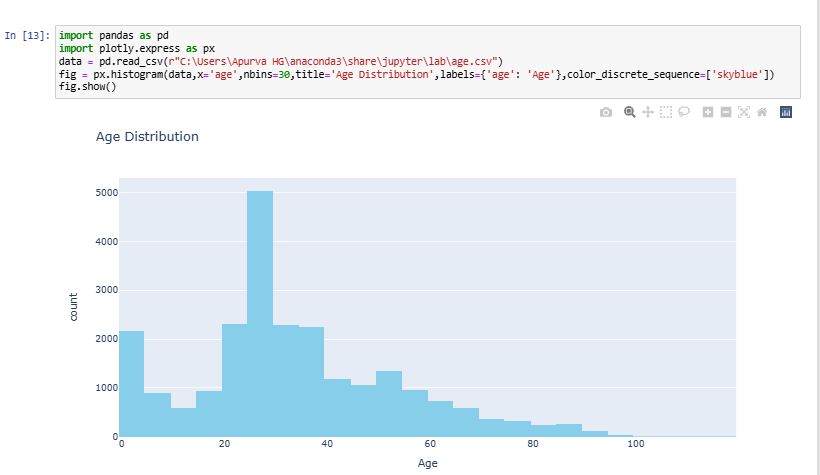
**Example:**

****

**Histogram:**

In plotly, [histograms](https://www.geeksforgeeks.org/python/histogram-using-plotly-in-python/) can be created using the histogram() function of the plotly.express class.

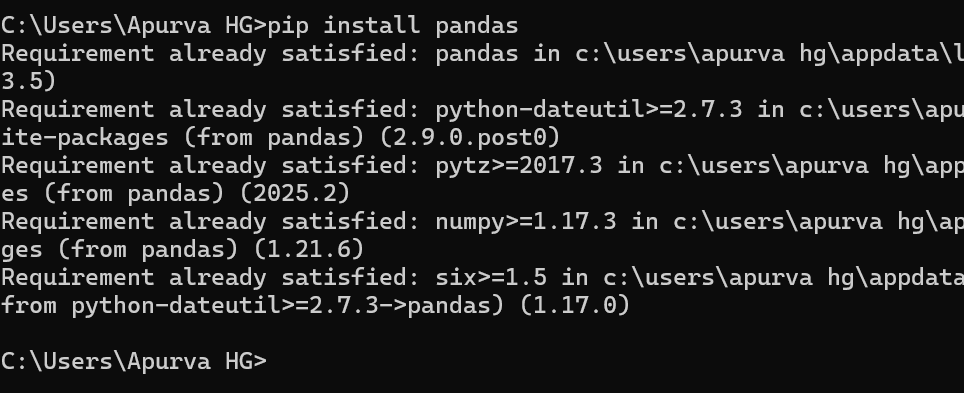
**Example:**

** Pandas**

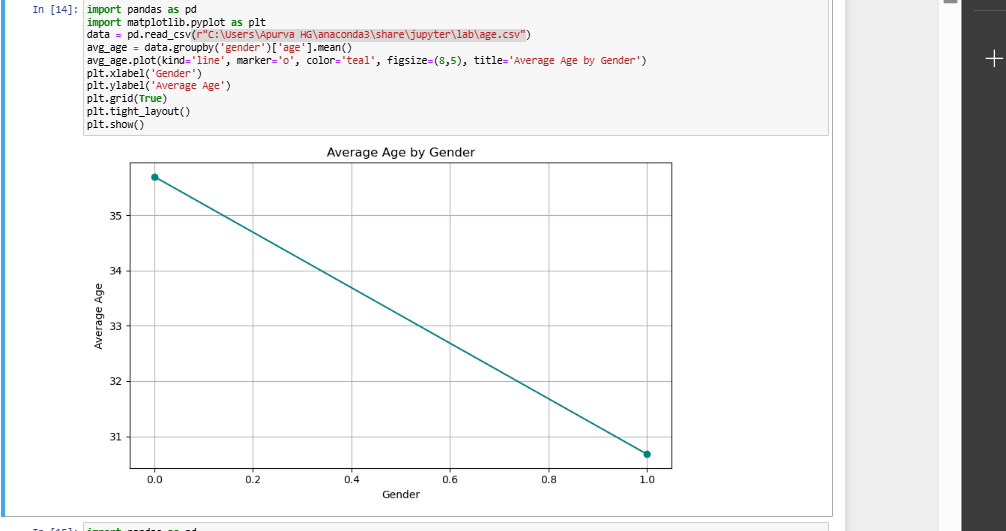
Pandas is open-source Python library which is used for data manipulation and analysis. It consists of data structures and functions to perform efficient operations on data.

**Installing Pandas**

**pip install pandas**

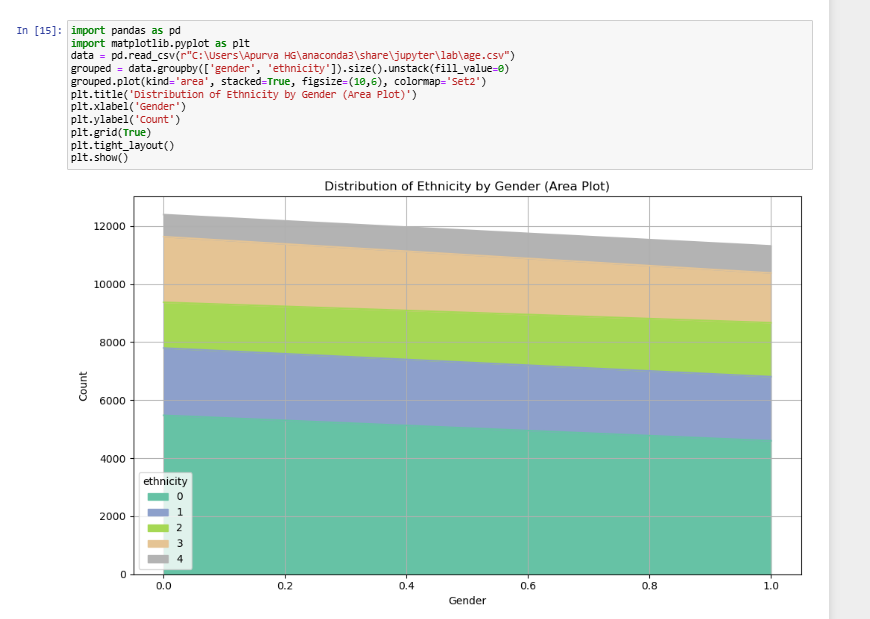


**1.Line Plot:**

A [Line plot](https://www.geeksforgeeks.org/data-visualization/what-is-line-plot/) is a graph that shows the frequency of data along a number line. It is best to use a line plot when the data is time series. It can be created using Dataframe.plot() function.

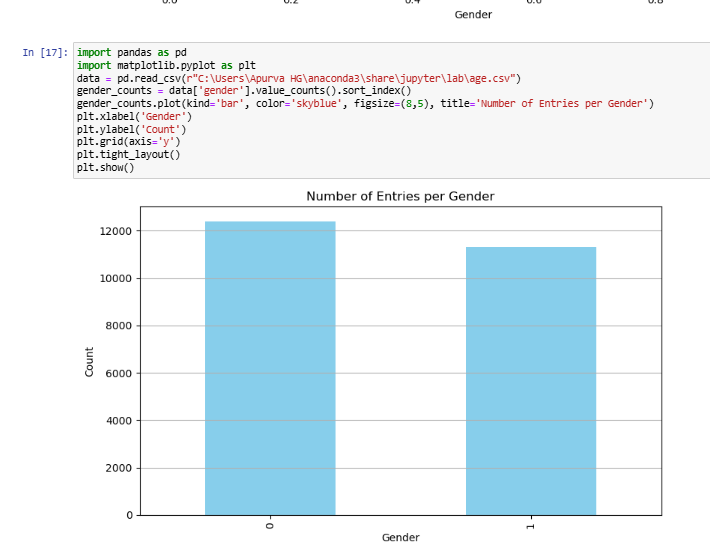
2.**Area Plots:**

[Area plot](https://www.geeksforgeeks.org/data-visualization/area-line-plot/) shows data with a line and fills the space below the line with color. It helps see how things change over time. we can plot it using DataFrame.plot.area() function.



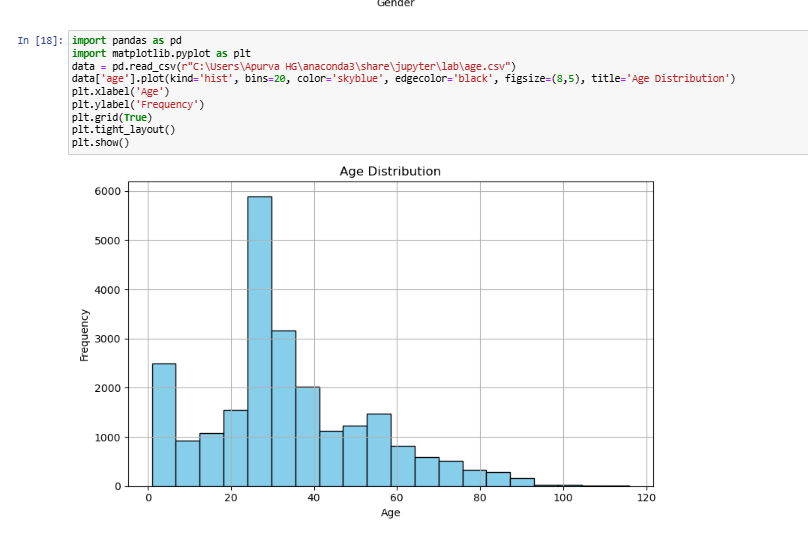
1. **Bar Plots:**

A[bar chart](https://www.geeksforgeeks.org/maths/bar-graphs/)presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally with DataFrame.plot.bar() function.

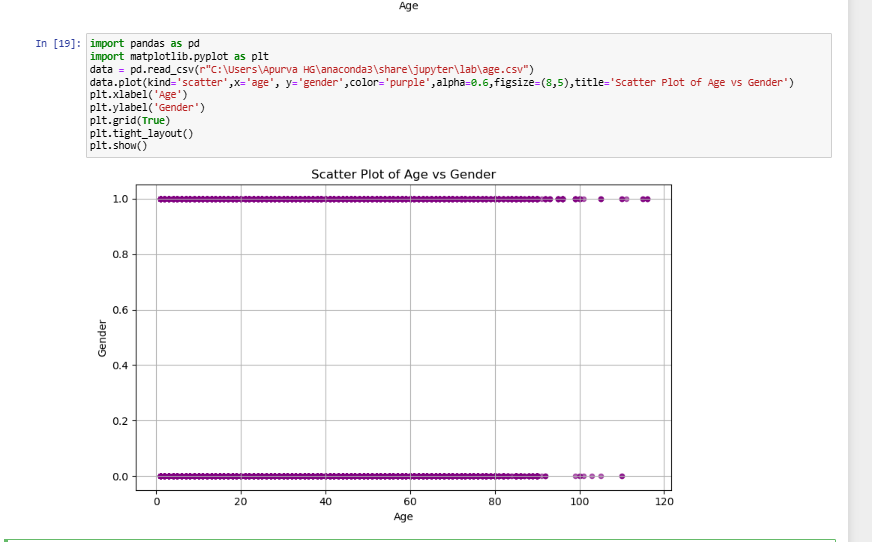


1. **Histogram:**

[Histograms](https://www.geeksforgeeks.org/maths/histogram/)help visualize the distribution of data by grouping values into bins. Pandas use DataFrame.plot.hist() function to plot histogram.

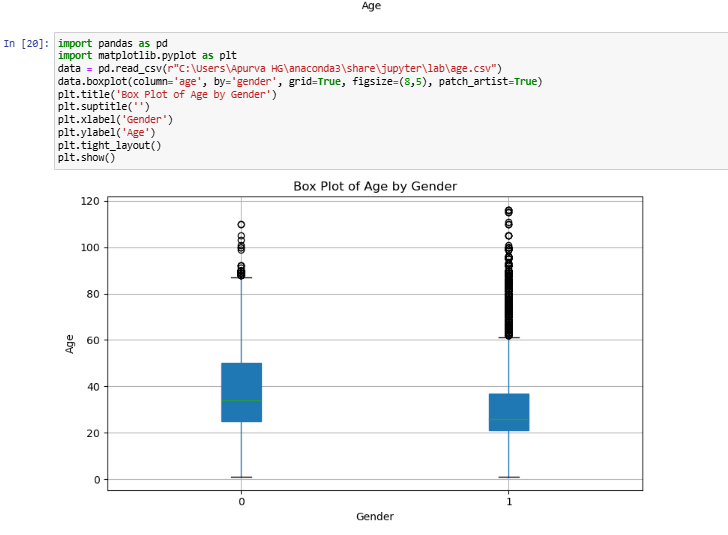


1. **Scatter Plot:**

[Scatter plots](https://www.geeksforgeeks.org/maths/scatter-plot/) are used when you want to show the relationship between two variables. They are also called correlation and can be created using DataFrame.plot.scatter() function.

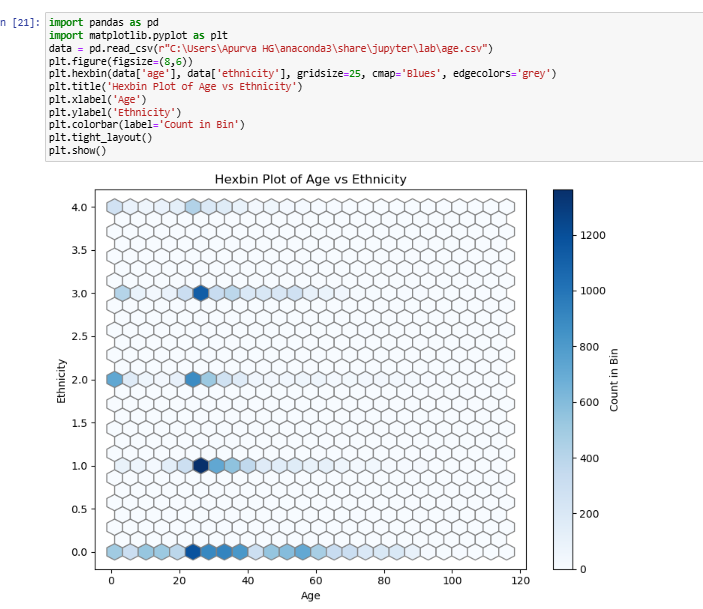
1. **Box Plots:**

A [box plot](https://www.geeksforgeeks.org/machine-learning/box-plot/) displays the distribution of data, showing the median, quartiles, and outliers. we can use DataFrame.plot.box() function or DataFrame.boxplot() to create it.



1. **Hexagonal Bin:**

[Hexagonal binning](https://www.geeksforgeeks.org/data-visualization/exploration-with-hexagonal-binning-and-contour-plots/)helps manage dense datasets by using hexagons instead of individual points. It’s useful for visualizing large datasets where points may overlap. Let's create the hexagonal bin plot.



**Matplotlib**

**Advantages:**

->Highly customizable with full control over plot elements.  
->Supports a wide range of static plot types for scientific use.

**Disadvantages:**  
->Syntax can be verbose and complex for beginners.  
->Lacks built-in interactivity and modern styling by default.

**Seaborn**

**Advantages:**  
->Beautiful default styles and color palettes.  
->Simplifies statistical plots like box, violin, and heatmaps.

**Disadvantages:**  
->Limited flexibility compared to Matplotlib.  
->No support for interactive or web-based visualizations.

**Pandas**

**Advantages:**  
->Quick one-liner plots directly from DataFrames.  
->Ideal for basic visualizations during data exploration.

**Disadvantages:**  
->Minimal customization and styling options.  
->Not suitable for complex or interactive plots.

**Plotly**

**Advantages:**  
->Fully interactive with hover, zoom, and pan features.  
->Supports 3D plots, animations, and web integration.

**Disadvantages:**  
->Slower performance with large datasets.  
->Less control over fine-tuned plot elements than Matplotlib.

**Bokeh**

**Advantages:**  
->Great for interactive and streaming visualizations.  
->Integrates well with web frameworks like Flask and Django.

**Disadvantages:**  
->Steeper learning curve for advanced features.  
->Less intuitive for statistical or quick exploratory plots.