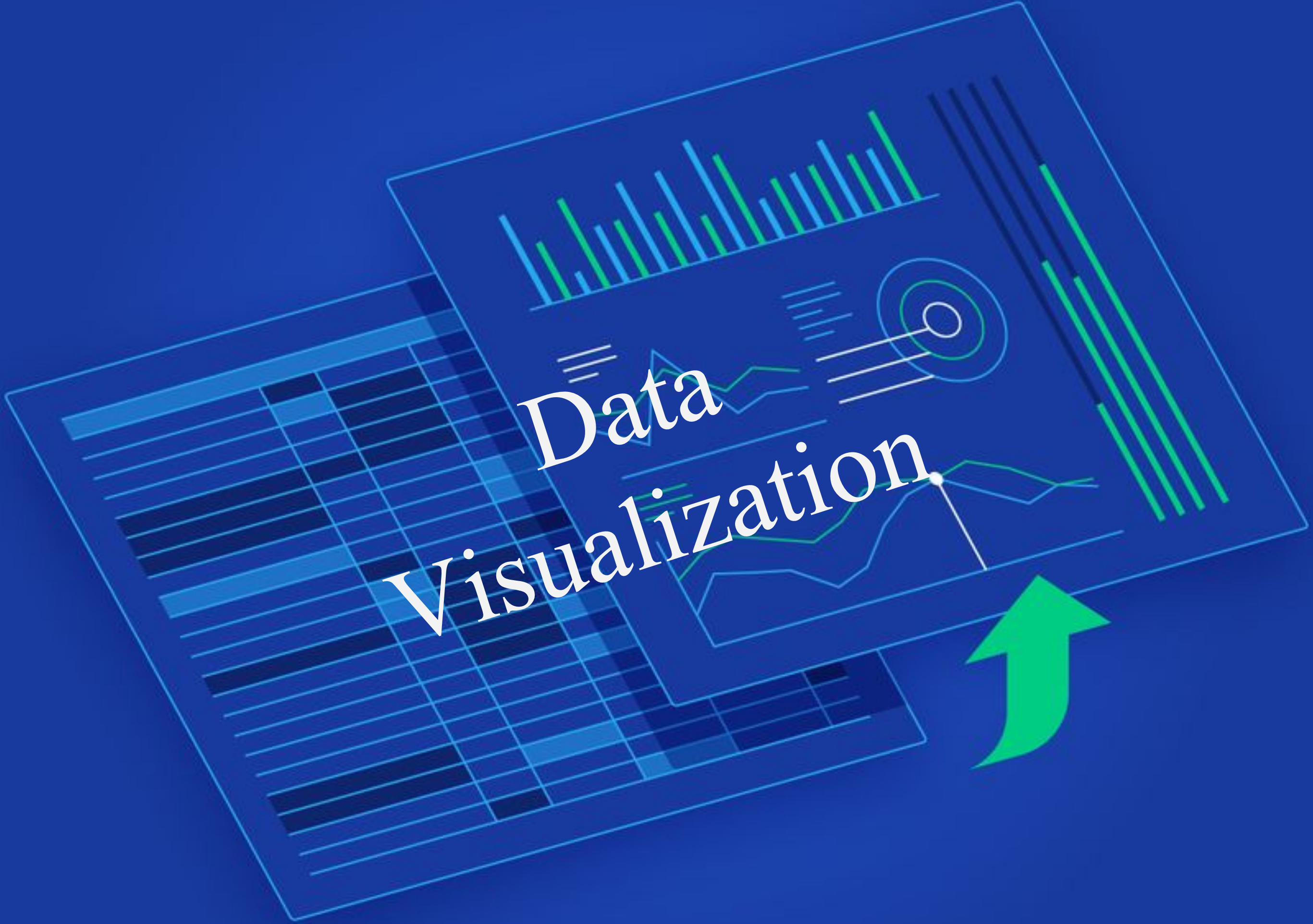
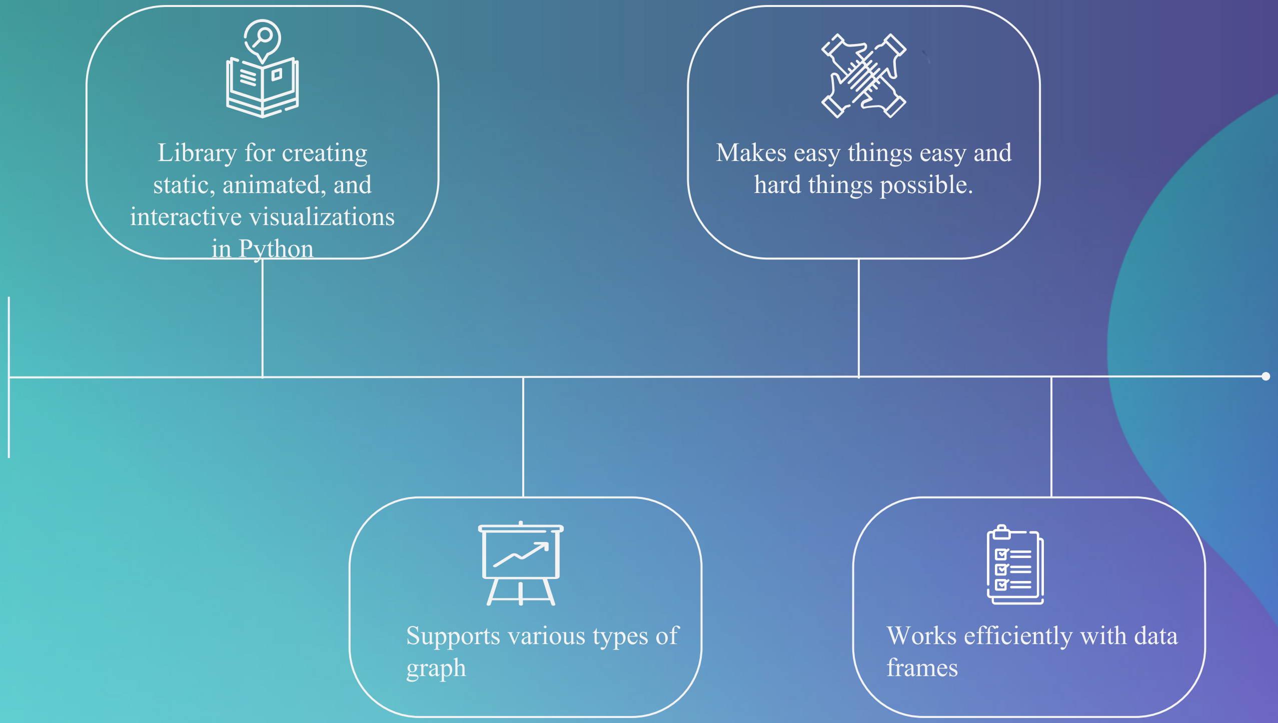
Data Visualization



Introduction

- Representation of data through common graphics, such as charts, plots.
- Helps to perform complex data analysis and recognize patterns.
- We use different libraries for plotting
- By combining user-friendly and aesthetically pleasing features, these visualizations make research and data analysis much quicker and are also a powerful communication tool.

matplotlib

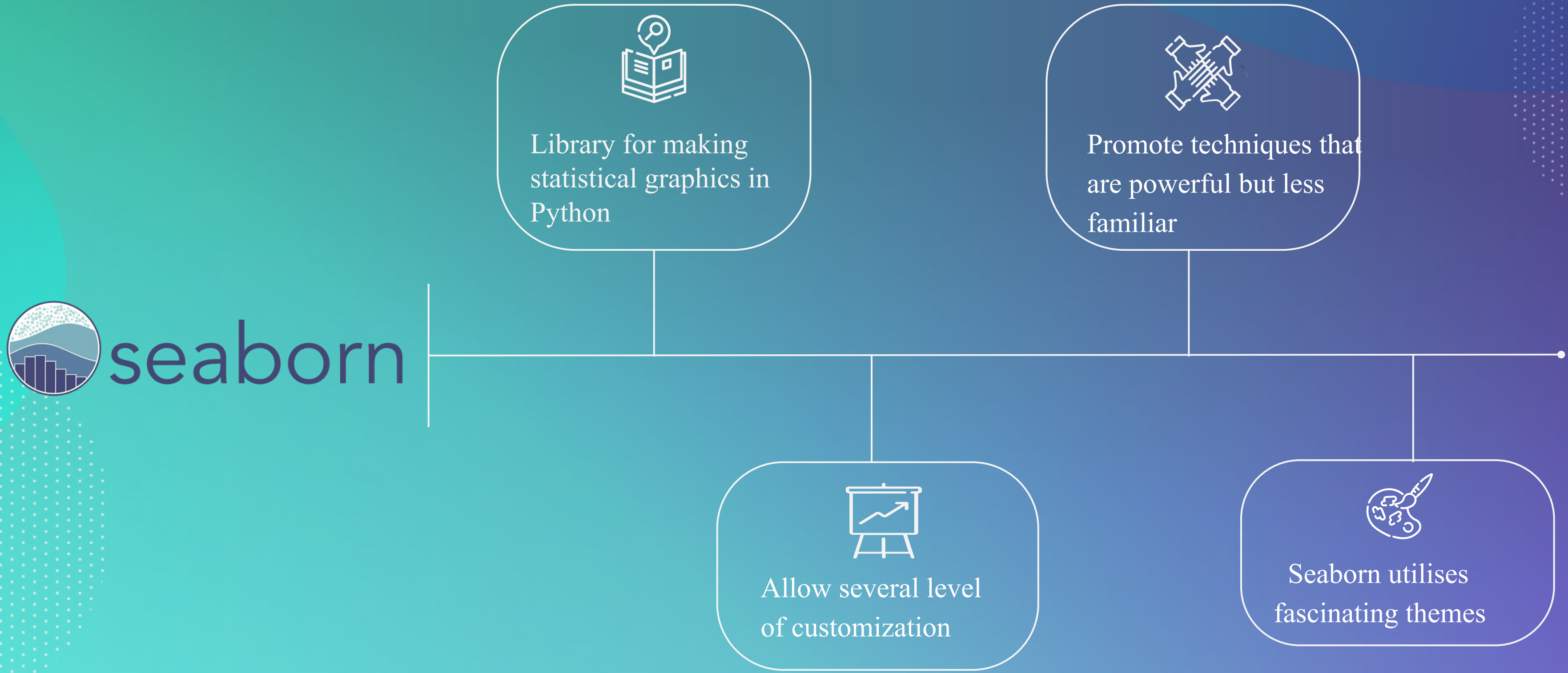


Library for creating static, animated, and interactive visualizations in Python

Makes easy things easy and hard things possible.

Supports various types of graph

Works efficiently with data frames



seaborn

Library for making statistical graphics in Python

Promote techniques that are powerful but less familiar

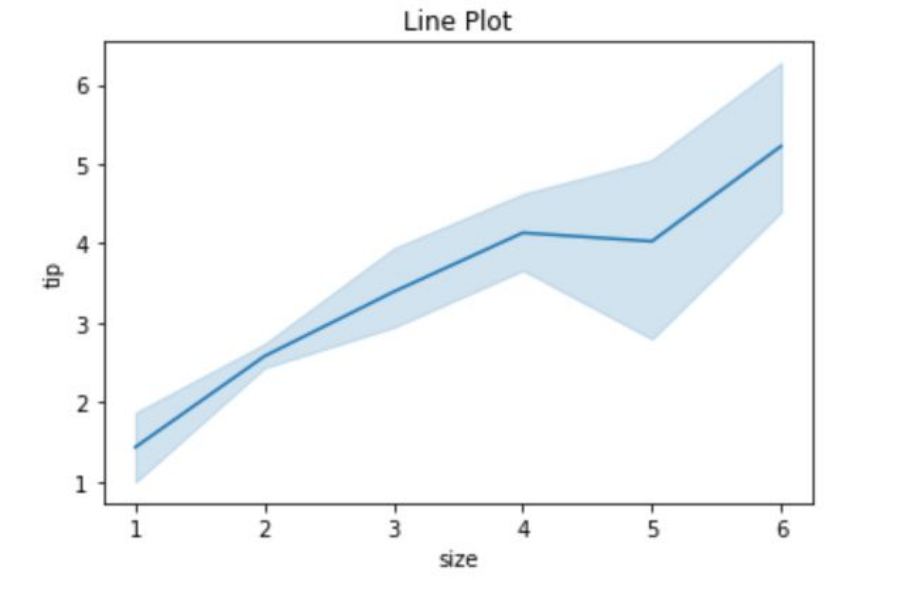
Allow several level of customization

Seaborn utilises fascinating themes

Line Plot with seaborn

- Display numerical values on one axis, and categorical values on other,These parameters control what visual semantics are used to identify the different subsets, these plot aggregates over multiple tip values at each value of size

- sns.lineplot() to draw a line plot using long-form data, assign the x and y variables

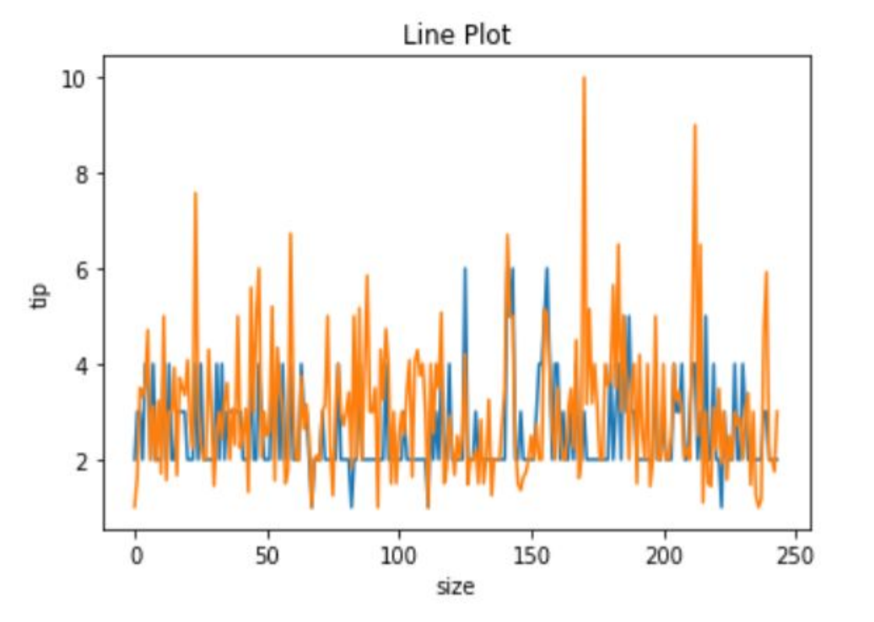


Line Plot with matplotlib

- Generic one automatically plots using lines .By default, each line is assigned a different style, this results in same line plot as before, as the values of Day are inferred.

- This can help in the modification of better visualization. %matplotlib inline makes your plot outputs appear and be stored within the notebook.

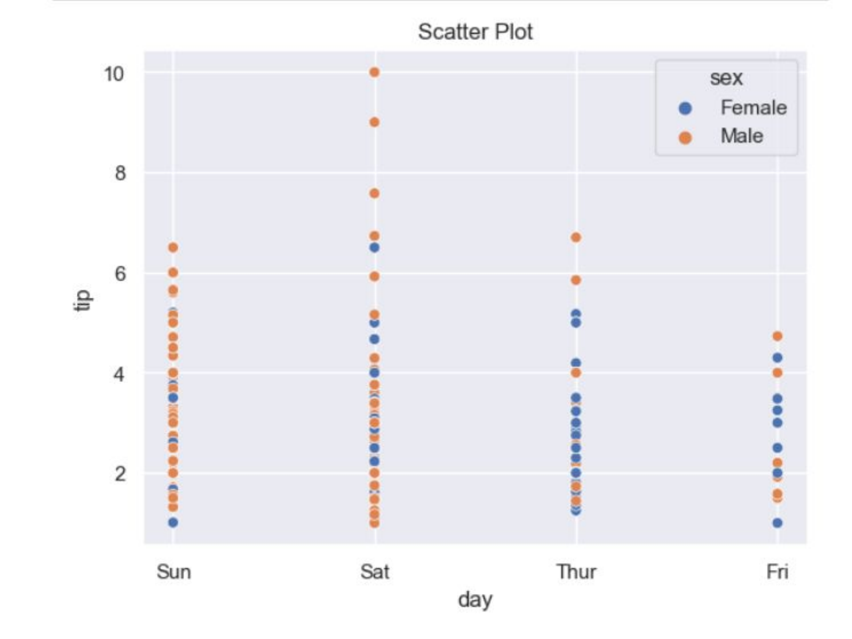
- The plt.plot() function takes additional arguments that can be used



Scatter Plot with seaborn

- Displays the relationship between 2 variables.The circle is used to represent the data point, this style of plot can be hard to interpret and is often ineffective. Using redundant semantics can be helpful for making graphics more accessible.

- sns.scatterplot() function is used to plot the data and depict the relationship between the values using the scatter visualization in seaborn

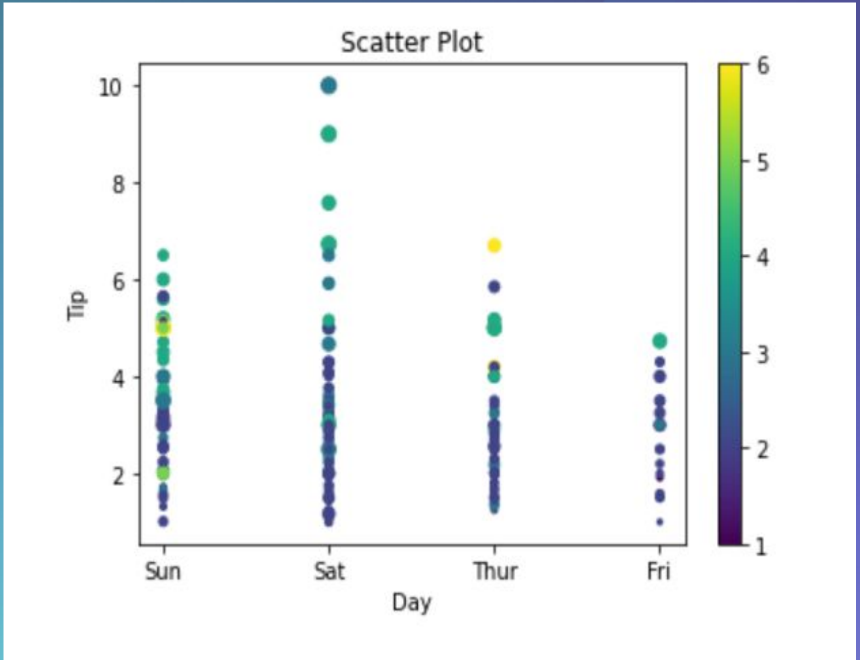


Scatter Plot with matplotlib

- Observe relationship between variables Day and tip, Represent by a marker the position depends on its values

- There are a few outliers, but majority follows this hypothesis by looking into the size of dot.

- plt.scatter() plots scatter dots, depends on whether the variable is inferred to represent “numeric” or “categorical” data. In particular, numeric variables are represented with a sequential colormap by default in matplotlib.

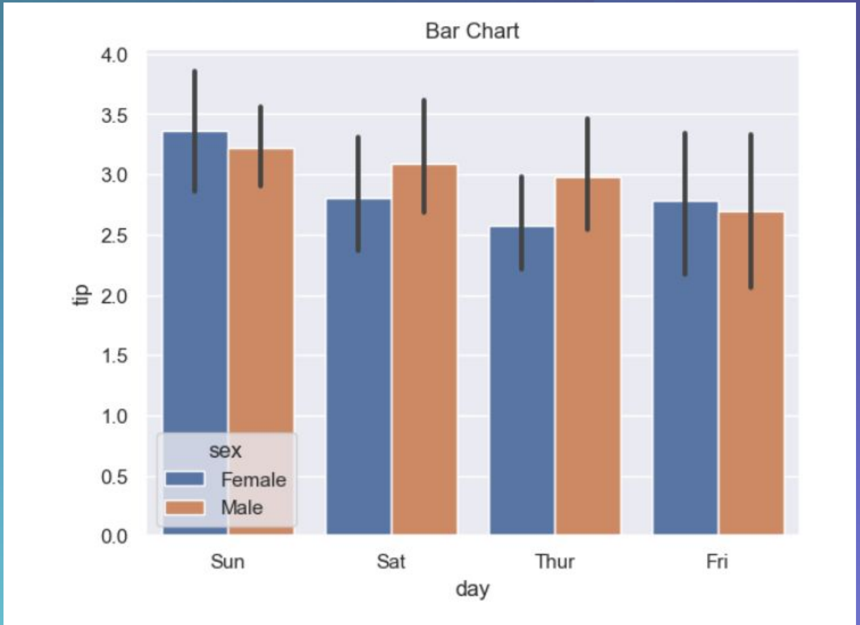


Bar Plot with seaborn

- ❑ Show point estimates and confidence intervals as rectangular bars, this function always treats one of the variables as categorical and draws data at ordinal positions (0, 1, … n)

- ❑ Bar plot represents an estimate of central tendency for a numeric variable with the height of each rectangle and provides some indication of the uncertainty around that estimate using error bars

- ❑ sns.barplot() to use this plot we choose a categorical column for the x-axis and a numerical column for the y-axis

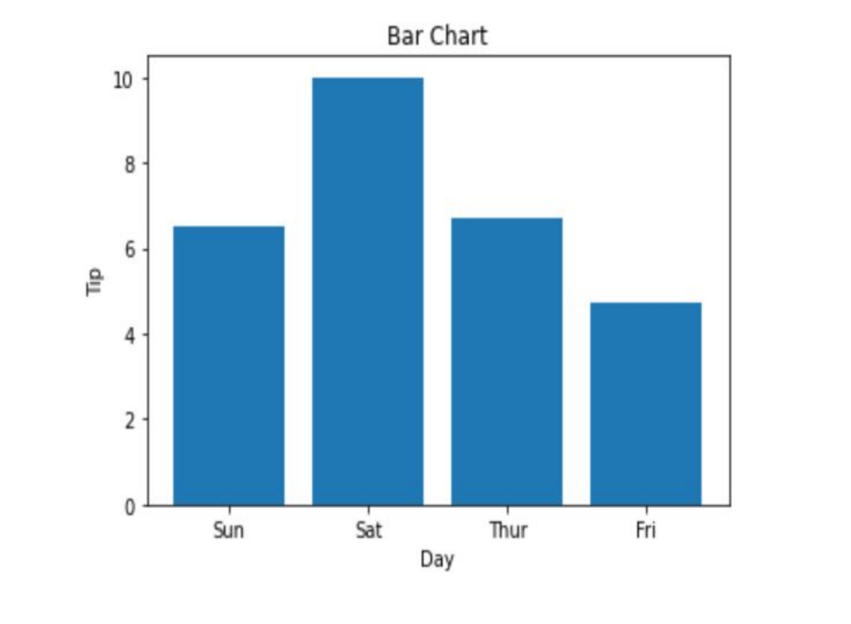


Bar Plot with matplotlib

❑ (h) or (l) proportional to values, the bars are positioned at Day with the given alignment. The vertical baseline is bottom (default 0).

❑ Shows comparisons among discrete categories, many parameters can take either a single value applying to all bars or a sequence of values, one for each bar.

❑ plt.bar() function that can be used in the MATLAB style use as well as object oriented API.

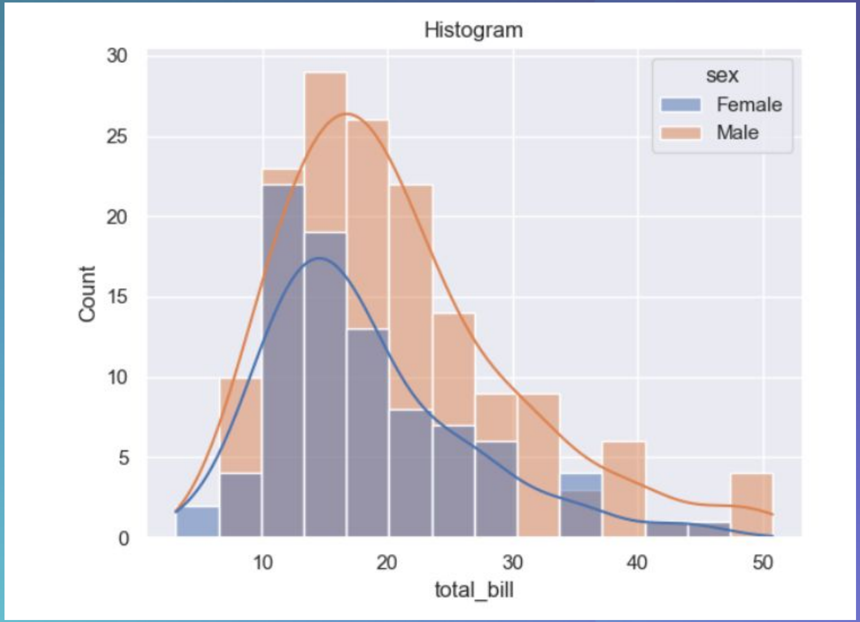


Histogram with seaborn

- It can add a smooth curve obtained using a (kde), Here these represents right skewed histogram.

- KDE can produce a plot that is less cluttered and more interpretable, especially when drawing multiple distributions along with bar plor

- sns.histplot() this function combines the matplotlib hist function (with automatic calculation of a good default bin size)

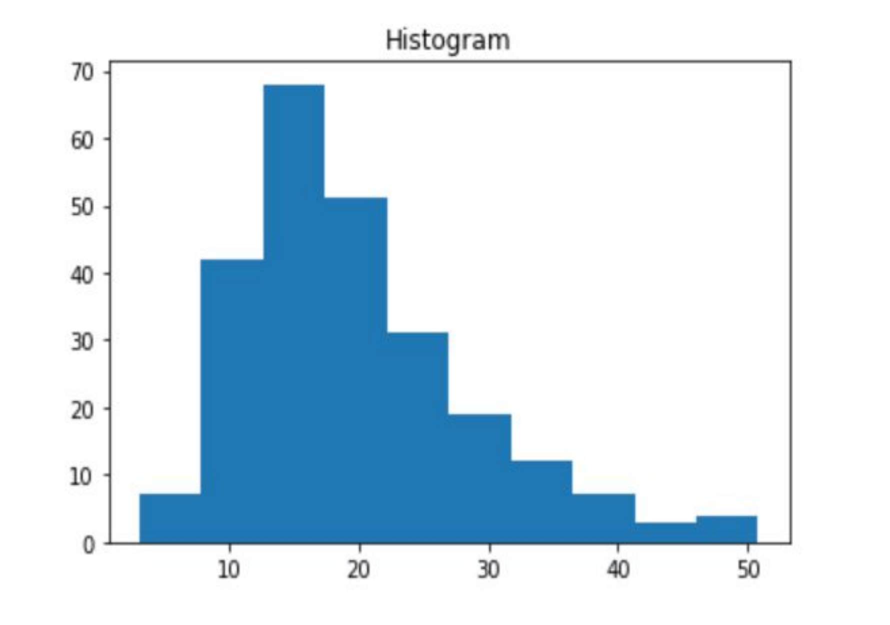


Histogram with matplotlib

- Graph showing frequency distributions and ‘bar’ is a traditional bar-type histogram. If multiple data are given the bars are arranged side by side.

- Divide the entire range of values into a series of intervals, Give idea about shape and spread of the data

- plt.hist() function in pyplot module of matplotlib library is used to plot a histogram,

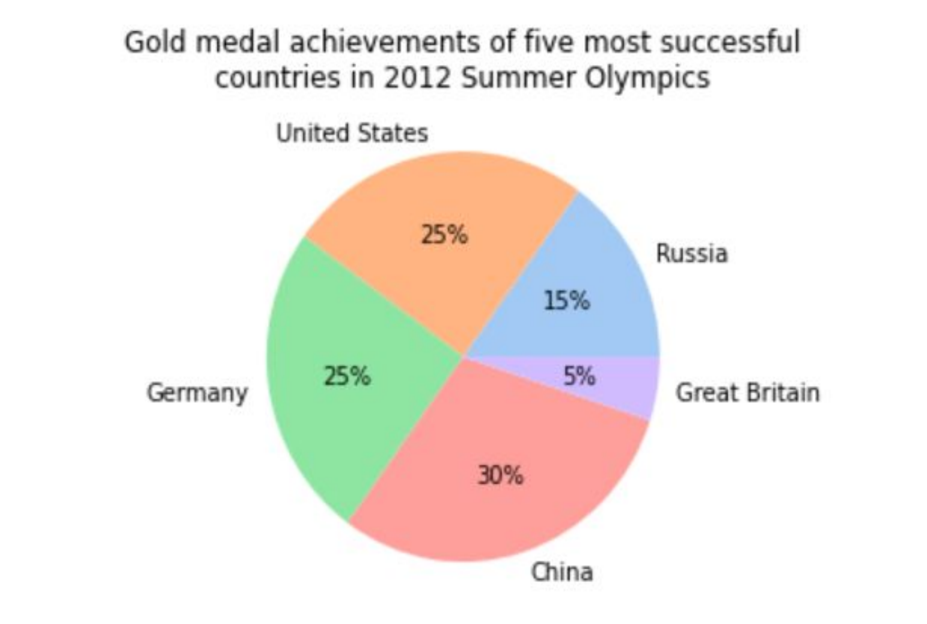


Pie Chart with seaborn

- Divided into slices to illustrate numerical proportion, Hard to work out exact values

- Provides a high-level interface for drawing attractive and informative statistical graphic

- colors = sns.color_palette('pastel') we have to pass the color pallet as a string inside the seaborn attribute plt.pie() we have to use the pie attribute of Matplotlib and the color pallets of Seaborn.

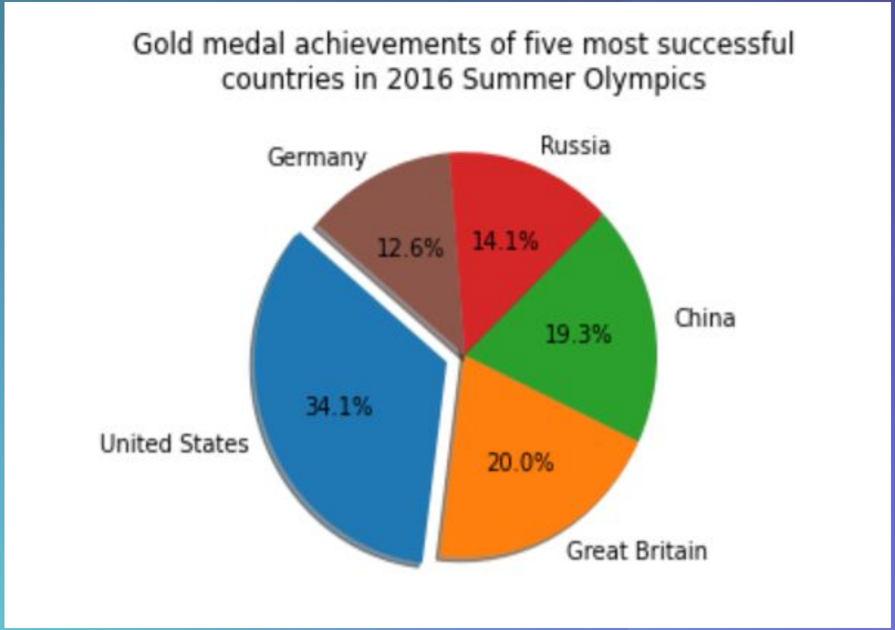


Pie Chart with matplotlib

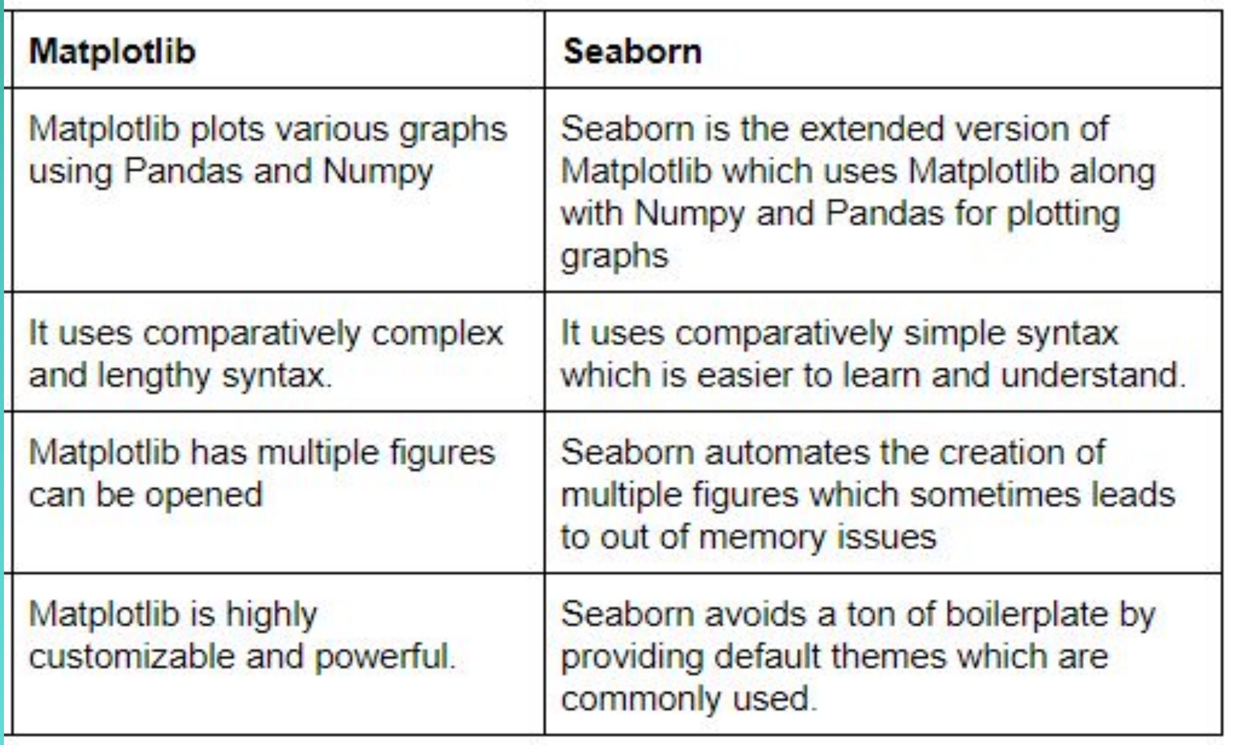
- Generates a pie diagram representing data in an array with labels seperated by it’s value.

- Particularly good at showing percentage or proportional data we can use lot of attributes to customize the pie chart according to data

- plt.pie() function to draw pie charts
  explode used to draw attention to a specific portion of the pie.
  autopct string, used to label the wedges with their % numeric value.



Matplotlib vs Seaborn



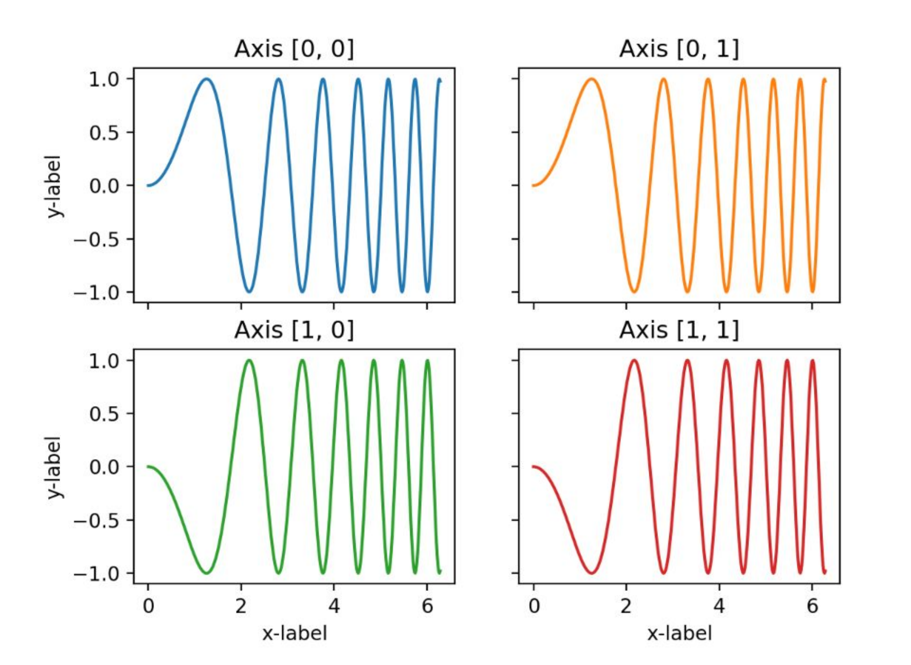
Matplotlib	Seaborn
Matplotlib plots various graphs using Pandas and Numpy	Seaborn is the extended version of Matplotlib which uses Matplotlib along with Numpy and Pandas for plotting graphs
It uses comparatively complex and lengthy syntax.	It uses comparatively simple syntax which is easier to learn and understand.
Matplotlib has multiple figures can be opened	Seaborn automates the creation of multiple figures which sometimes leads to out of memory issues
Matplotlib is highly customizable and powerful.	Seaborn avoids a ton of boilerplate by providing default themes which are commonly used.

Multiple Figures in Matplotlib opened

- Figures are identified via a figure number that is passed to x,y-labels
  we switch back to axes and make changes

- When stacking in two directions, the returned axs is a 2D NumPy array.

- Each Axes is scaled individually. Thus, if the ranges are different the tick values of the subplots do not align



Conclusion

Matplotlib is the most prominent one. Seaborn is also a great package which offers a lot more appealing plot and even it uses matplotlib as its base layer, where this task of learning from data is done in a huge extent on these recent days.

Both are in consideration. Visualization enables us to recognize emerging trends and respond rapidly on the grounds of what we see.