**Data Visualization with Python**

**Week 2**

**Data Visualization :**

Data visualization is the graphical representation of data and information. It involves the process of creating visual representations of data. Data visualization can take many different forms, from basic charts and graphs to more complex interactive dashboards, maps, and infographics.

**Area Plot:**

Area plots are particularly effective and depict data with a cumulative nature, such as tracking stock market performance, visualizing population demographics, or displaying the distribution of resources across various sectors.

**Histogram Plot:**

A histogram is a way of representing the frequency distribution of a numeric data set.

The way it works is that it partitions the spread of the numeric data into bins, assigns each data point in the data set to a bin, and then counts the number of data points assigned to each bin. So, the vertical axis is essentially the frequency or the number of data points in each bin.

# Bar Charts (Dataframe) [¶](https://jupyterlab-0-labs-prod-jupyterlab-us-east-0.labs.cognitiveclass.ai/user/gautamishu5/lab/tree/labs/authoride/IBMSkillsNetwork%2BDV0101EN/labs/v4/DV0101EN-Exercise-Area-Plots-Histograms-and-Bar-Charts_.ipynb#Bar-Charts-(Dataframe)-)

A bar plot is a way of representing data where the length of the bars represents the magnitude/size of the feature/variable. Bar graphs usually represent numerical and categorical variables grouped in intervals.

To create a bar plot, we can pass one of two arguments via kind parameter in plot():

* kind=bar creates a vertical bar plot
* kind=barh creates a horizontal bar plot

**Vertical bar plot**

In vertical bar graphs, the x-axis is used for labelling, and the length of bars on the y-axis corresponds to the magnitude of the variable being measured. Vertical bar graphs are particularly useful in analyzing time series data. One disadvantage is that they lack space for text labelling at the foot of each bar.

# Pie Charts

A pie chart is a circular graphic that displays numeric proportions by dividing a circle (or pie) into proportional slices. You are most likely already familiar with pie charts as it is widely used in business and media. We can create pie charts in Matplotlib by passing in the kind=pie keyword.

Plot the data. We will pass in kind = 'pie' keyword, along with the following additional parameters:

* autopct - is a string or function used to label the wedges with their numeric value. The label will be placed inside the wedge. If it is a format string, the label will be fmt%pct.
* startangle - rotates the start of the pie chart by angle degrees counterclockwise from the x-axis.
* shadow - Draws a shadow beneath the pie (to give a 3D feel).

When the numbers and text overlap in some instances. Let's make a few modifications to improve the visuals:

* Remove the text labels on the pie chart by passing in legend and add it as a seperate legend using plt.legend().
* Push out the percentages to sit just outside the pie chart by passing in pctdistance parameter.
* Pass in a custom set of colors for continents by passing in colors parameter.
* **Explode** the pie chart to emphasize the lowest three continents (Africa, North America, and Latin America and Caribbean) by passing in explode parameter.

# Box Plots [¶](https://jupyterlab-0-labs-prod-jupyterlab-us-east-0.labs.cognitiveclass.ai/user/gautamishu5/lab/tree/labs/authoride/IBMSkillsNetwork%2BDV0101EN/labs/v4/DV0101EN-Exercise-Pie-Charts-Box-Plots-Scatter-Plots-and-Bubble-Plots.ipynb#Box-Plots-)

A box plot is a way of statistically representing the distribution of the data through five main dimensions:

* **Minimum:** The smallest number in the dataset excluding the outliers.
* **First quartile:** Middle number between the minimum and the median.
* **Second quartile (Median):** Middle number of the (sorted) dataset.
* **Third quartile:** Middle number between median and maximum.
* **Maximum:** The largest number in the dataset excluding the outliers.

One of the key benefits of box plots is comparing the distribution of multiple datasets. In one of the previous labs, we observed that China and India had very similar immigration trends. Let's analyze these two countries further using box plots.

**Subplots**

Often times we might want to plot multiple plots within the same figure. For example, we might want to perform a side by side comparison of the box plot with the line plot of China and India's immigration.

To visualize multiple plots together, we can create a **figure** (overall canvas) and divide it into **subplots**, each containing a plot. With **subplots**, we usually work with the **artist layer** instead of the **scripting layer**.

Typical syntax is :

fig **=** plt.figure() *# create figure*

ax **=** fig.add\_subplot(nrows, ncols, plot\_number) *# create subplots*

Where

* nrows and ncols are used to notionally split the figure into (nrows \* ncols) sub-axes,
* plot\_number is used to identify the particular subplot that this function is to create within the notional grid. plot\_number starts at 1, increments across rows first and has a maximum of nrows \* ncols as shown below.

**Tip regarding subplot convention**

In the case when nrows, ncols, and plot\_number are all less than 10, a convenience exists such that a 3-digit number can be given instead, where the hundreds represent nrows, the tens represent ncols and the units represent plot\_number. For instance,

subplot(211) **==** subplot(2, 1, 1)

produces a subaxes in a figure which represents the top plot (i.e. the first) in a 2 rows by 1 column notional grid (no grid actually exists, but conceptually this is how the returned subplot has been positioned).

Note how the box plot differs from the summary table created. The box plot scans the data and identifies the outliers. In order to be an outlier, the data value must be:

* larger than Q3 by at least 1.5 times the interquartile range (IQR), or,
* smaller than Q1 by at least 1.5 times the IQR.

Let's look at decade 2000s as an example:

* Q1 (25%) = 36,101.5
* Q3 (75%) = 105,505.5
* IQR = Q3 - Q1 = 69,404

Using the definition of outlier, any value that is greater than Q3 by 1.5 times IQR will be flagged as outlier.

Outlier > 105,505.5 + (1.5 \* 69,404)  
Outlier > 209,611.5

# Scatter Plots

A scatter plot (2D) is a useful method of comparing variables against each other. Scatter plots look similar to line plots in that they both map independent and dependent variables on a 2D graph. While the data points are connected together by a line in a line plot, they are not connected in a scatter plot. The data in a scatter plot is considered to express a trend. With further analysis using tools like regression, we can mathematically calculate this relationship and use it to predict trends outside the dataset.

# Bubble Plots [¶](https://jupyterlab-5-labs-prod-jupyterlab-us-east-0.labs.cognitiveclass.ai/user/gautamishu5/lab/tree/labs/authoride/IBMSkillsNetwork%2BDV0101EN/labs/v4/DV0101EN-Exercise-Pie-Charts-Box-Plots-Scatter-Plots-and-Bubble-Plots.ipynb#Bubble-Plots-)

A bubble plot is a variation of the scatter plot that displays three dimensions of data (x, y, z). The data points are replaced with bubbles, and the size of the bubble is determined by the third variable z, also known as the weight. In maplotlib, we can pass in an array or scalar to the parameter s to plot(), that contains the weight of each point.

**Note:** Numpy arrays are usually used as the data source for plotting and also support mathematical functions.

# Summary: Basic and Specialized Visualization Tools

Congratulations! You have completed this module. At this point in the course, you know:

* A pie chart is a circular statistical graphic, divided into segments, to illustrate numerical proportion.
* The process of creating a pie chart involves importing Matplotlib to represent a large set of data over a period of time.
* A box plot is a way of statistically representing given data distribution through five main dimensions.
* The five main dimensions are minimum, first quartile, median, third quartile, and maximum.
* You can create a box plot using Matplotlib.
* A scatter plot displays values pertaining to typically two variables against each other.
* The process of creating a scatter plot involves importing Matplotlib to visualize a large set of data.
* Matplotlib is a versatile plotting library that offers a flexible interface for creating various types of plots.
* Matplotlib’s Pyplot module offers a convenient way to create and customize plots quickly.
* Data Storytelling is the ‘art of storytelling’ that involves creating a narrative around the data.
* Data visualization is an important aspect of data storytelling and involves creating engaging visuals.

**Waffle Charts:**

Waffle charts are a visualization technique that represents categorical data in the form of square tiles or cells.

These resemble a grid of equal-sized squares, with each square representing a specific value or category.

The size or color of the squares indicate the magnitude or proportion of each category.

Waffle charts effectively show the proportion or percentage of different categories within an overall composition.

The grid-like structure of waffle charts makes it easy to understand and interpret data even for nontechnical audiences.

**Usage of Waffle Charts :**

waffle charts are used for such use cases given below-

* market share analysis,
* demographic representation,
* project progress tracking,
* budget allocation,
* survey responses,
* election results, and
* product sales analysis.