# DATA VISUALIZATION

What is Data Visualization?

Representing huge text written data from files (CSV, JSON, etc.) into a graphical/visual format so that the trends, patterns and insights can be quickly and intuitively perceived and understood hence, cutting down the time and effort required to read/interpret the raw data.

A lot of data visualization is conducted in data science and machine learning processes, and also the processes can be benefitted from the data visualization.

The need of data visualization-

1.Quickest way to understand and represent data.

2.Universally understandable, whether an individual is from a technical background or not, people of all age groups can grasp what the data represents.

3. In technical wise, it is also used in k-means clustering to find the optimal number of clusters, understanding the data before building the model, finding better parameters for the model training.

4.Used in outlier detection, correlation analysis.

There are many libraries which help in data visualization.

This documentation would be mainly focusing on the libraries Matplotlib and Seaborn.

# MATPLOTLIB

[we have assumed that the library is already installed if not: [Installation — Matplotlib 3.10.1 documentation](https://matplotlib.org/stable/install/index.html)]

The csv file used: <https://www.csdojo.io/s/countries.csv>

Matplotlib has two different interfaces-

1. The “pyplot” interface which is for simpler tasks
2. The “axes” interface, more flexible and uses an OOPs approach

This will focus on the pyplot approach.

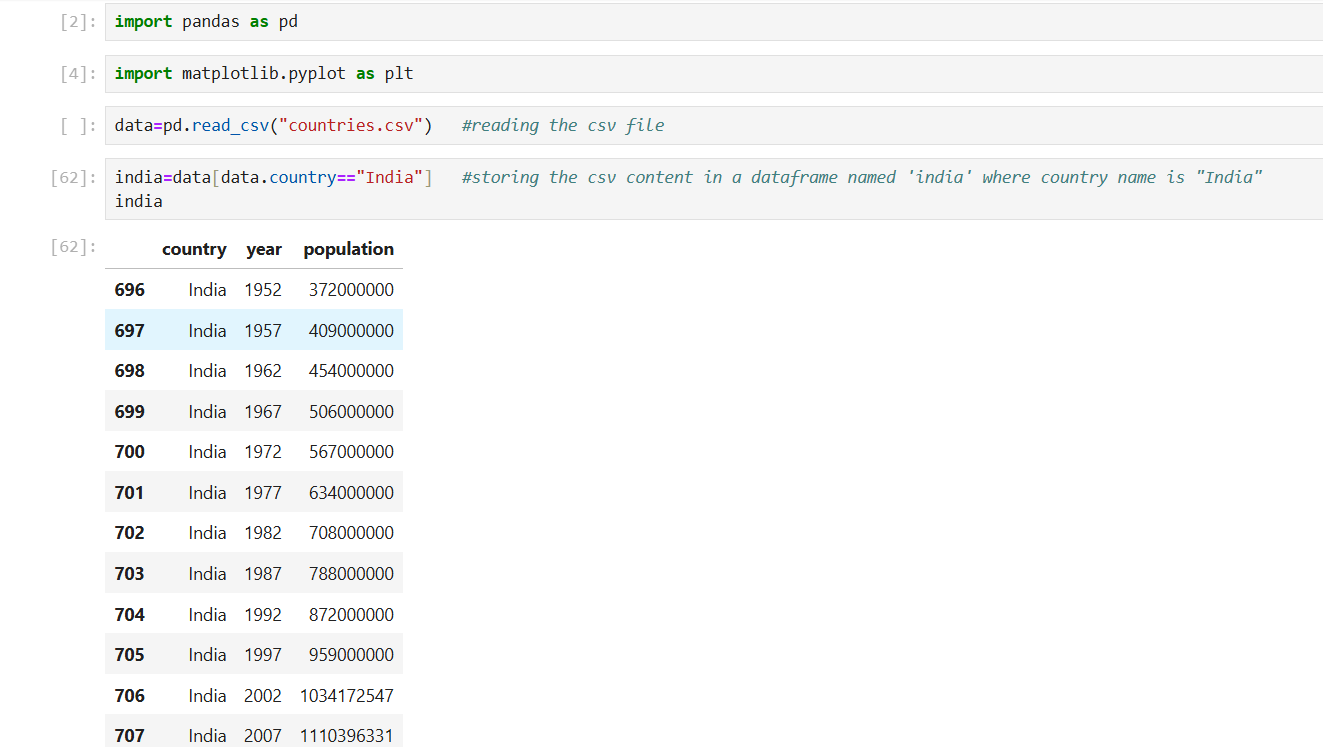
Matplotlib is one of the most popular choices for data visualization due to its customization properties.

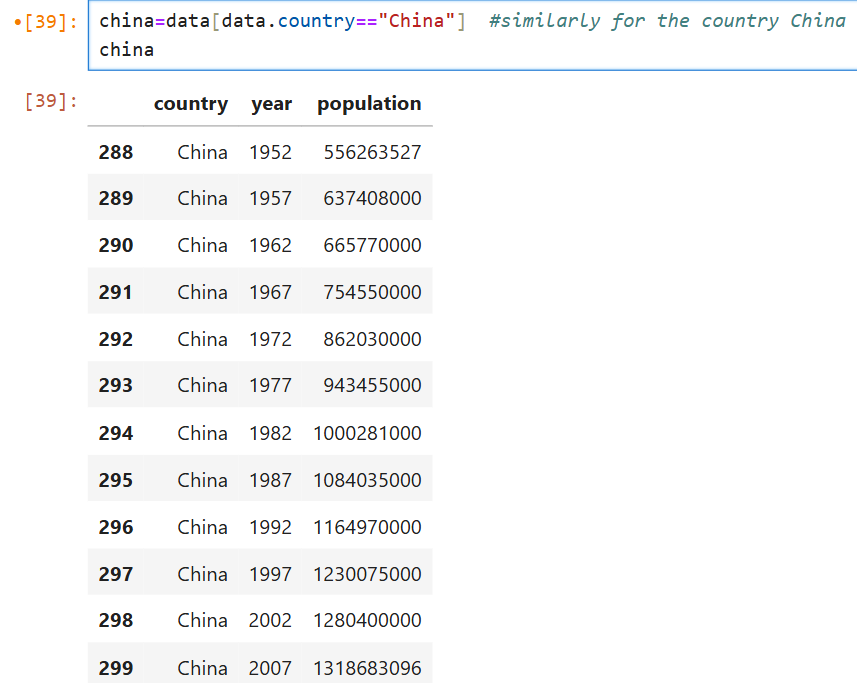
Works well with Pandas, NumPy and, ML frameworks like PyTorch.

Has a strong community with 15,000+ Stack Overflow answers and counting.

Other data visualization libraries are based on it, example Seaborn.

\*\*\*\*\*\*\*\*\*\*\*\*

Importing the necessary libraries, modules and reading csv files.



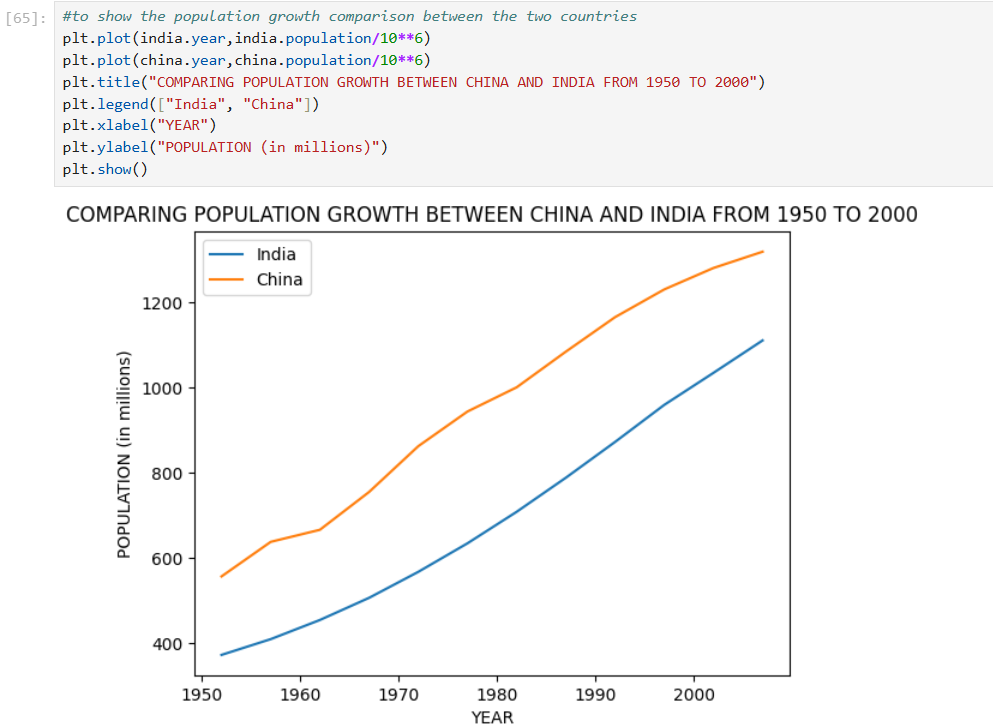
# Line plot

Shows trends over time by connecting datapoints with lines.

Mostly used to see how an independent variable influences the dependent variable(s). Used in linear regression (single and multiple).

Time series data, continuous data.

Example: Tracking the population growth over the time



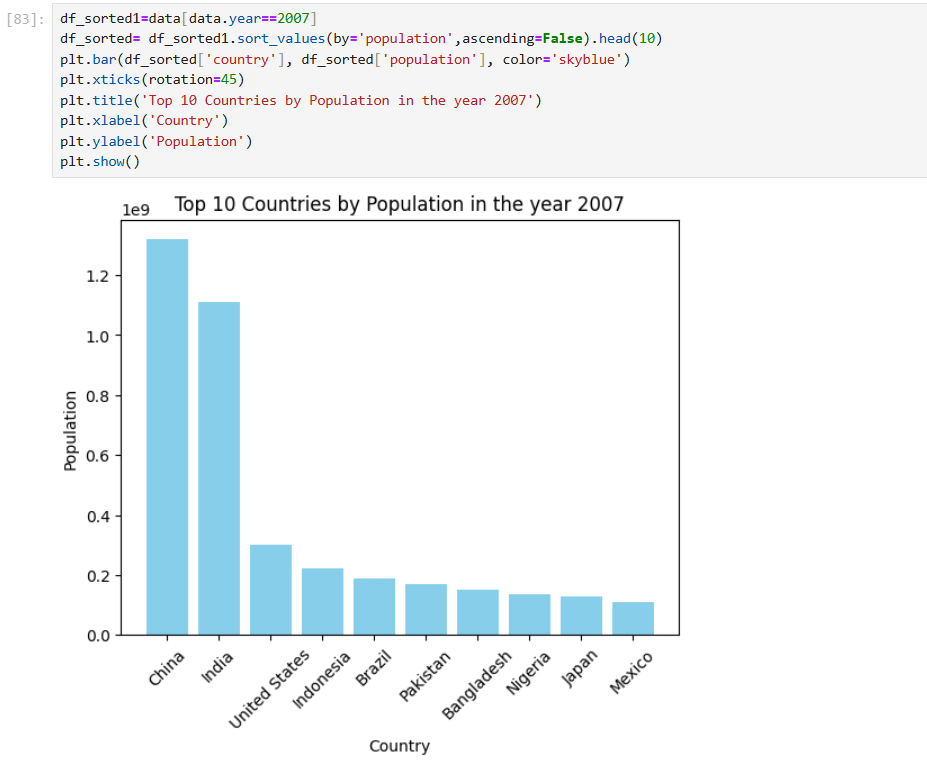
# Bar chart

Used to show differences between categories and understand relationships in the data.

When there are records with categories that need to be arranged from highest to lowest.

To explain the relationship between categories, which helps in discovering patterns and tendencies.

Ex. Below shows the top 10 countries with the highest population in 2007



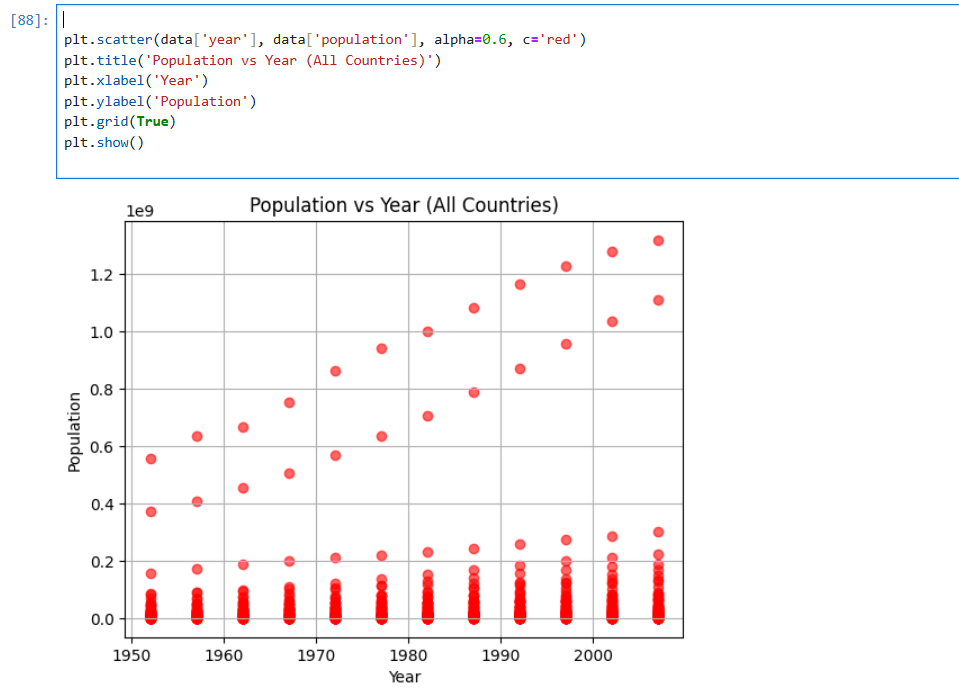
# Scatter Chart

It uses dots to represent data points showing the relationship between two numerical variables.

To spot the natural groupings in the data (segmentation, classification).

They are used for identifying outliers or unusual remark for the facts.

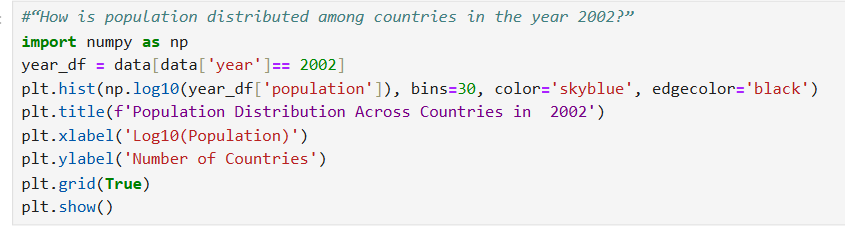
Ex. In the scatter plot given below it shows how for certain countries-India and China the growth has been different as compared to others.



# Histogram

Plots the distribution of a numeric variable’s values as a series of bars.

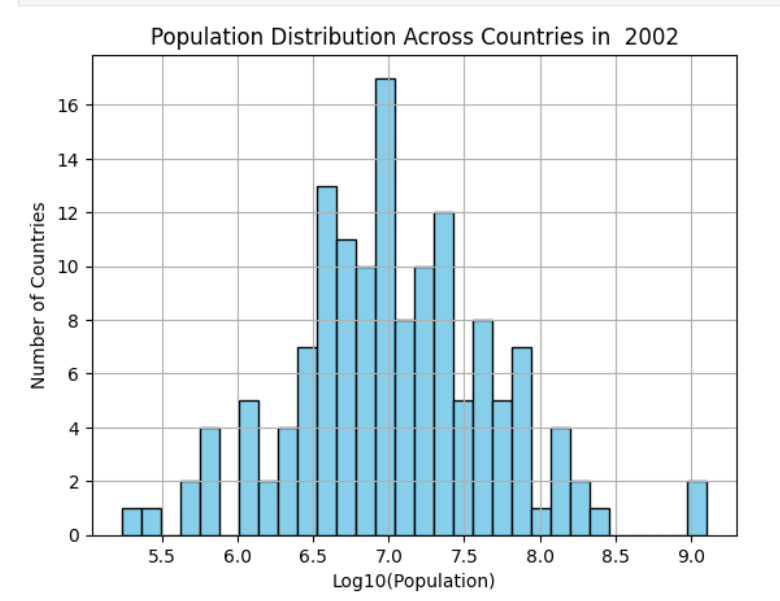
Each bar comprises of a range of numeric values, the bar’s height indicates the frequency of data points with a value within the corresponding bin.



The population is scaled on a log10 scale, in order to standardize and make scaling understandable.

Bins also known as intervals means the population has been divided into 30 intervals/groups, each group ranging within a certain range of values.

The graph below shows how the population of the countries of the world had been distributed in the year 2002.



Takeaways:

1. More than 16 countries had population ranging around 10 million in the year 2002.

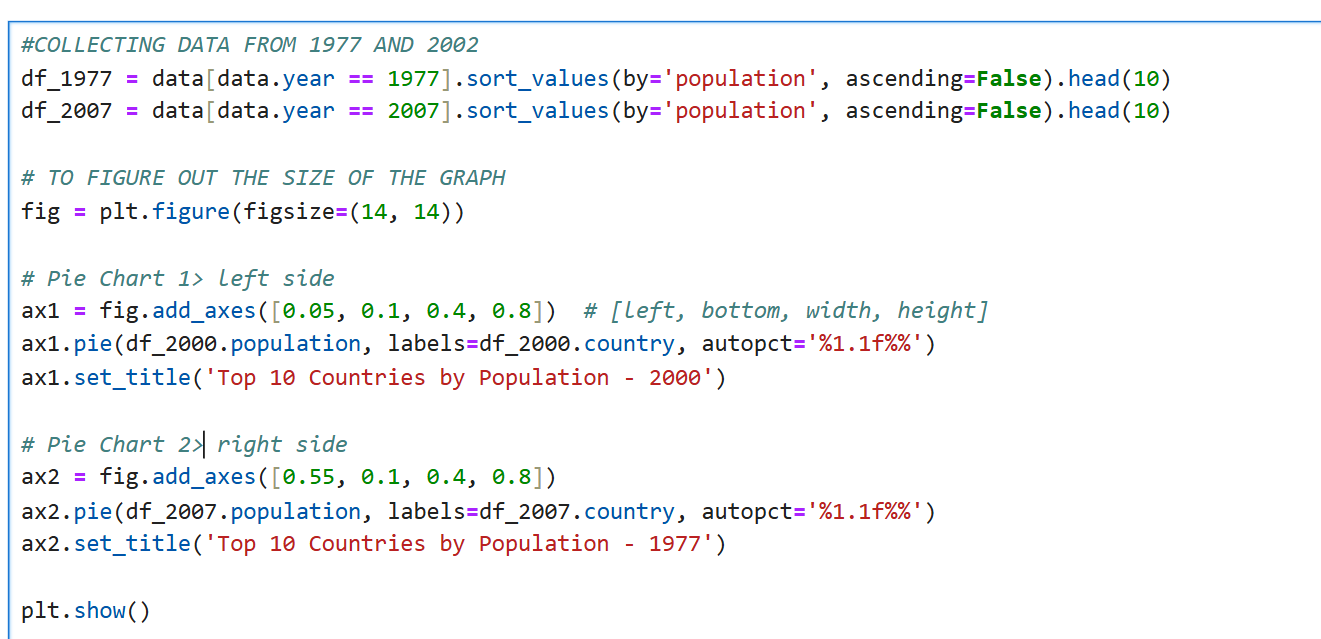
# PIE CHART

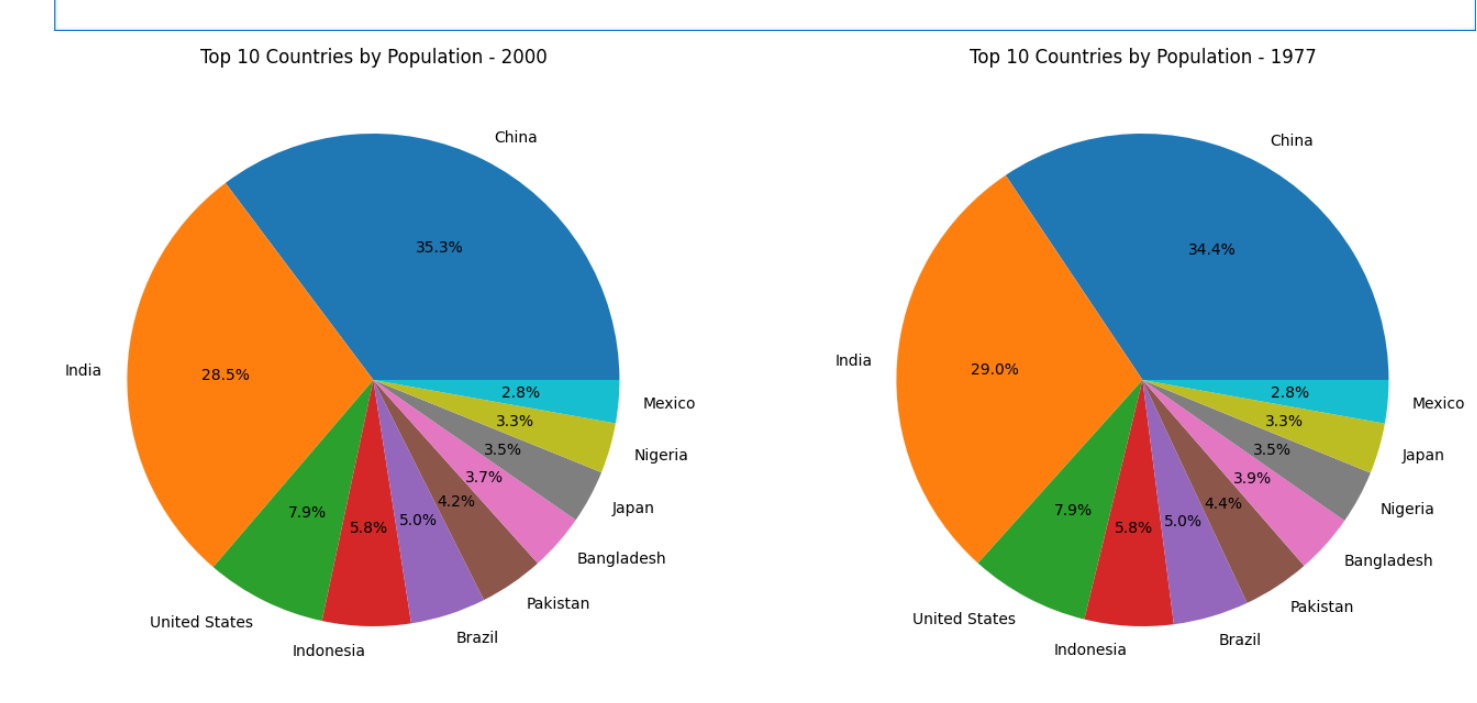
The size of each slice directly corresponds to the percentage or proportion of the data it represents.

Useful for comparing the contribution of each category to the overall total.

Mostly effective when there are a limited number of categories.

Becomes less effective when representing a large number of categories, as it may overlap or become difficult to compare slices accurately.



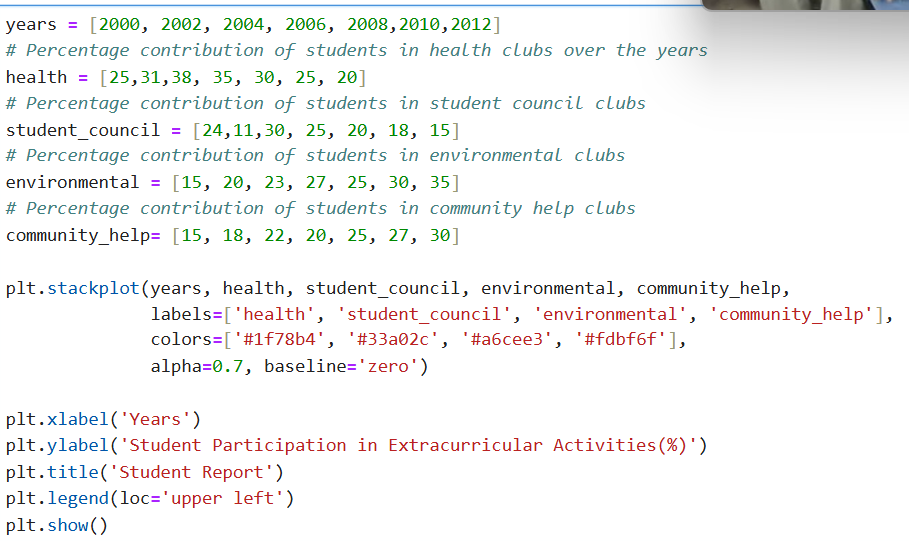


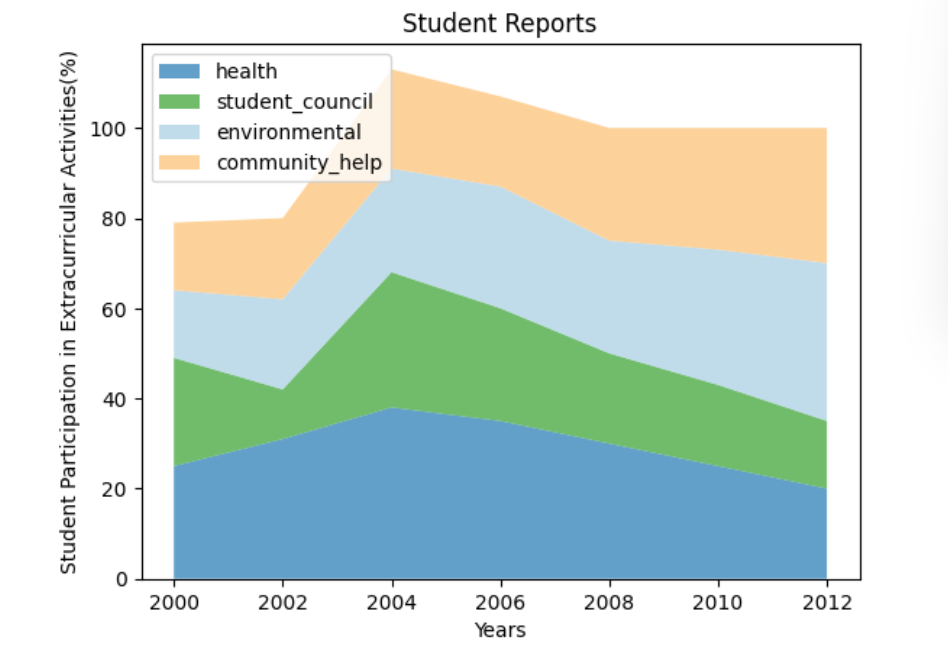
# AREA PLOT

An extension of the basic line chart.

Useful to show the composition of a whole, along with the individual components, as well as how they change over time or across categories.

In a stacked area chart, each line is stacked on top of the previous one. The stacking illustrates how the total changes over time or across categories, as well as the contribution of each category to the whole.





# SEABORN

Seaborn is a data visualization library built on top of matplotlib and closely integrated with Pandas data structures in Python.

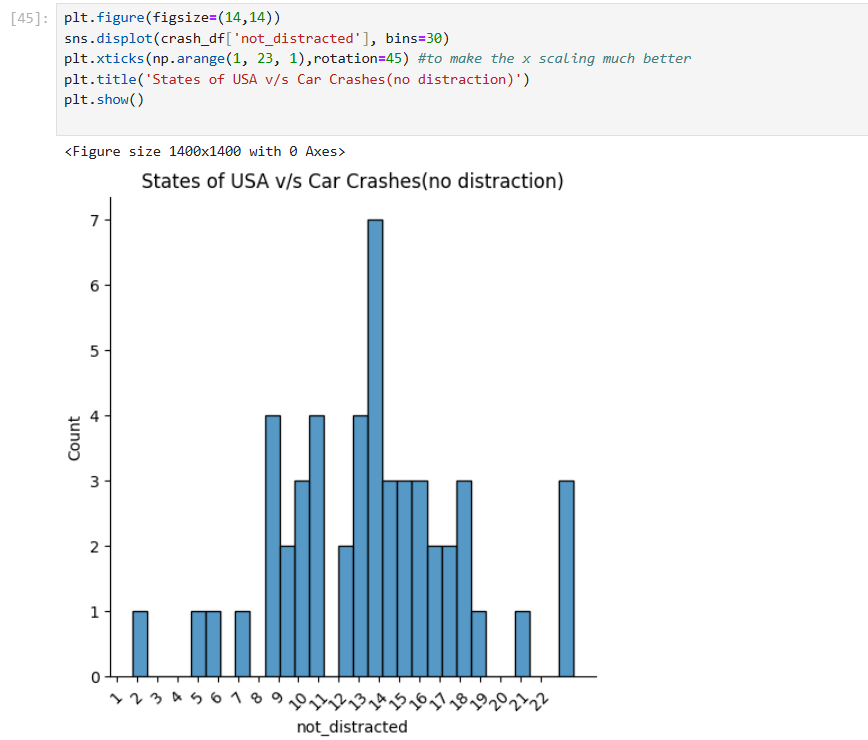
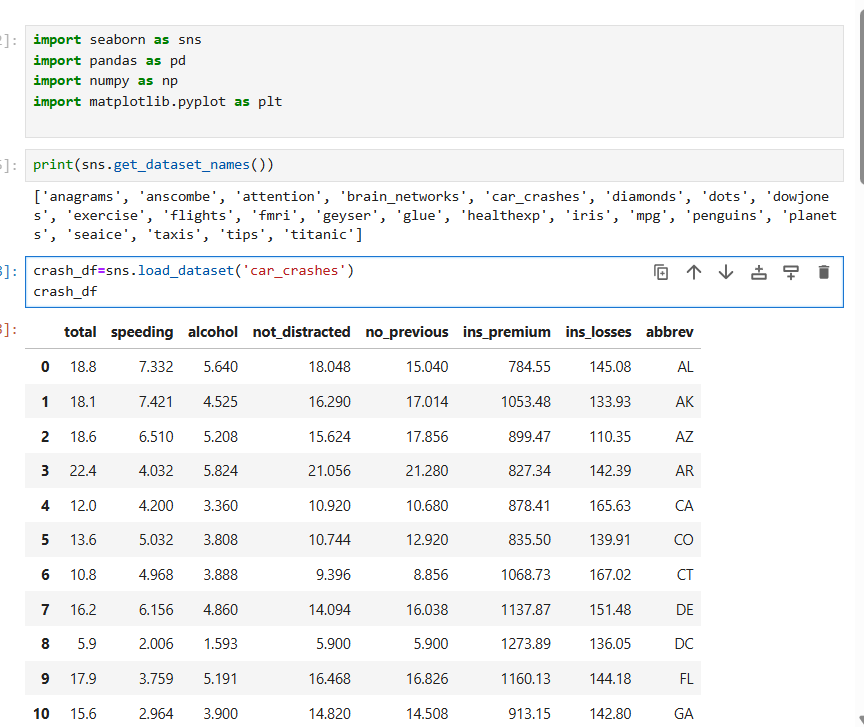
Focused on statistical visualization.

 Seaborn makes it easy to switch between different visual representations by using a consistent dataset-oriented API.

# DISTRIBUTUION PLOT

Like a histogram.

Depending on the data has been segregated into bins (intervals), how the data is distributed.



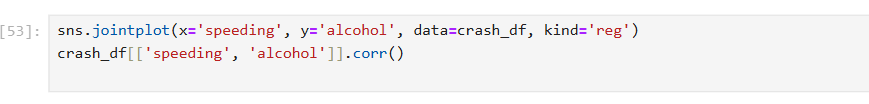
Takeaways:

1. There are 7 states of USA which fall under the criteria that 14% of car crashes did not involve any distraction from the driver itself.

# JOINT PLOT

It is used to visualize the relationship between two variables. It combines-

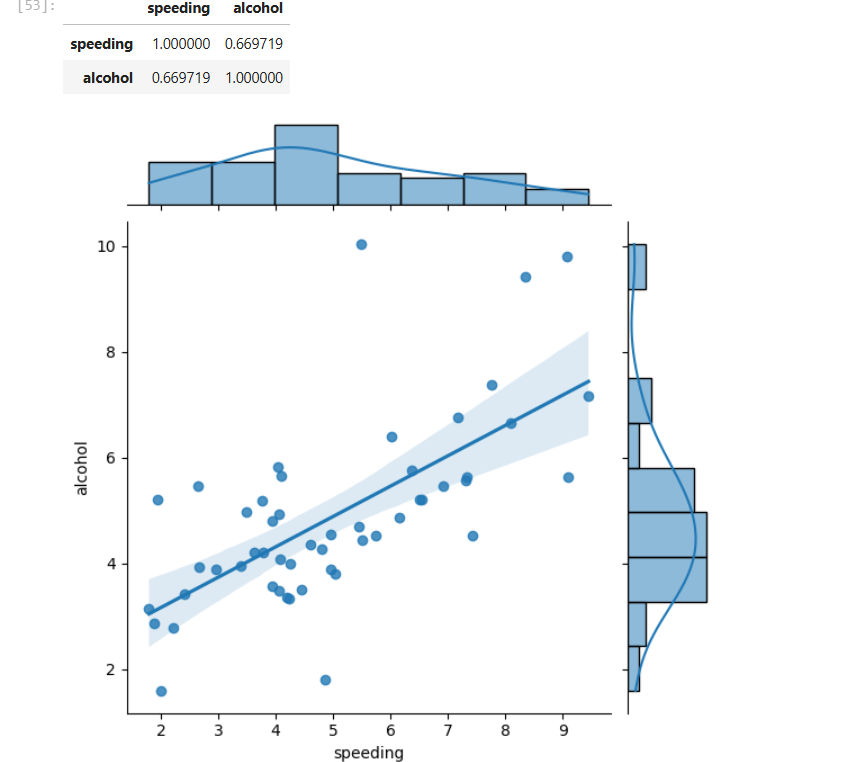
1. A scatter plot (or regression, hexagon) in the center
2. Two histograms or KDE plot (distribution plots) on the margins- one for each variable



x-axis: percentage of crashes involving speeding in each state

y-axis: % of crashes involving alcohol in each state

kind=’reg’ ONE of the methods to show the correlation between speeding and inebriation.



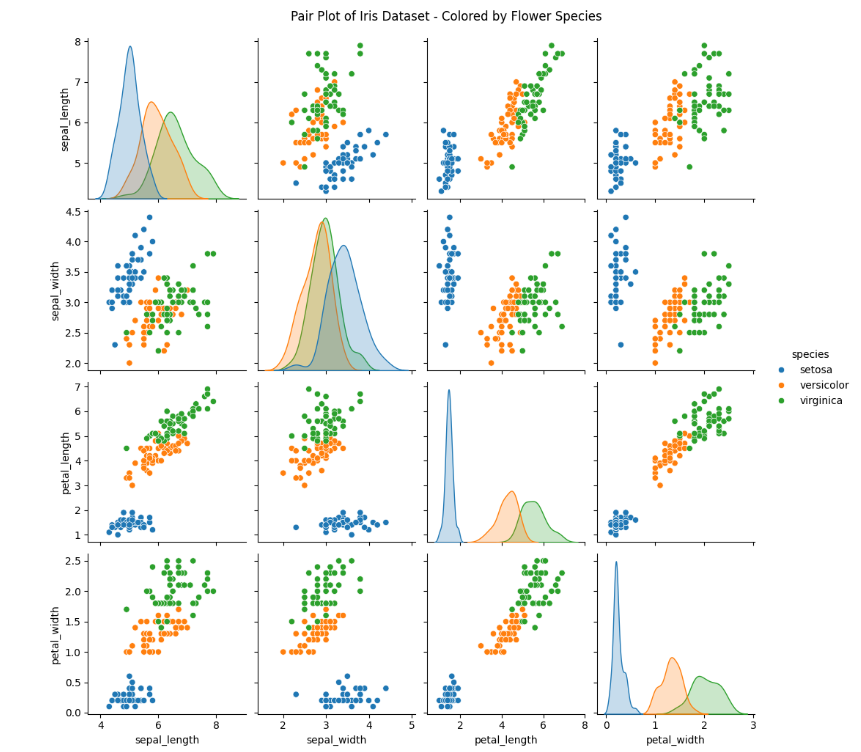
Takeaways:

1.The correlation value is positive and approx. 0.67 showing a strong correlation. Thus, we can conclude as speeding involvement increases, alcohol involvement tends to increase too.

# Pair Plots:

1. Quick EDA (exploratory data analysis tool).
2. Easily detect variables which might be related





Takeway/Insights-

1. Setosa forms a distinct cluster in most plots, especially those involving petal measurements – setosa has noticeably small petal lengths and widths.
2. Versicolor and virginia overlap in many feature combinations. Somewhat separable using petal\_length and petal\_width.
3. Petal dimensions are better at classification.

# Categorical Plots

1. BAR PLOT and STRIPPLOT

Bar plot:

(a)shows the average y value for each x value

(b)the height of the bar is the mean of all the y values for corresponding x-values

Strip plot:

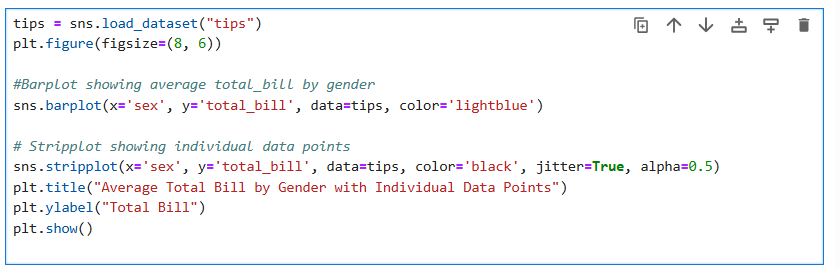
(a)shows each individual point

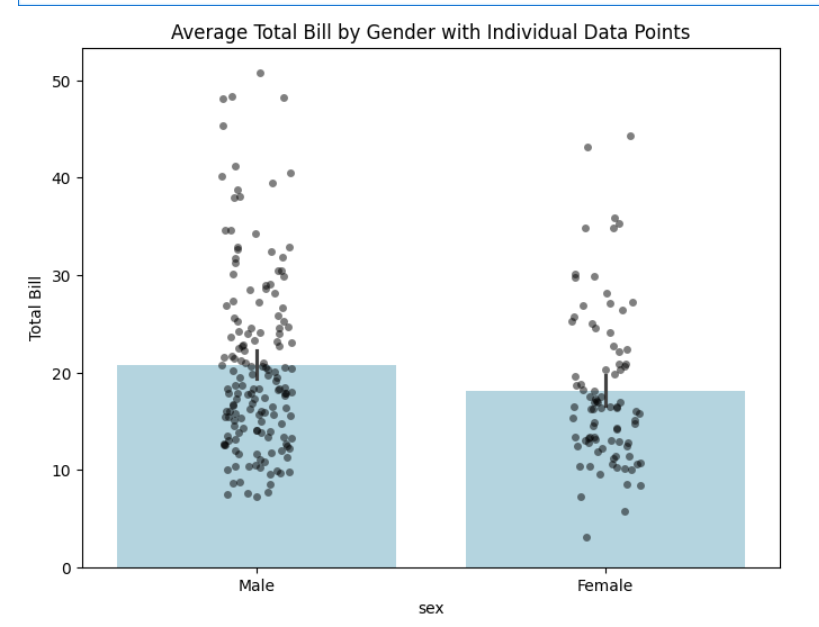
(b) clustered region denotes a higher density of data points

Takeaways

1.The average (bar) lies within the range where the y values for an x value mostly fall and the cluster will also be visible.

2.If the data is skewed, the mean (bar) can shift from the densest cluster. If the estimator parameter is equal to median, then the bar will overlap with the densest cluster even if skewed.





The estimator by default is mean and since the dataset is normally distributed, the densest cluster and the mean (bar) overlap.

# BOX PLOT (BOX AND WHISKER PLOT)

Used for comparing distributions and spotting outliers.

The center line of the box plot denotes the median.

The box spans from the **1st quartile (Q1)** to the **3rd quartile (Q3)**.

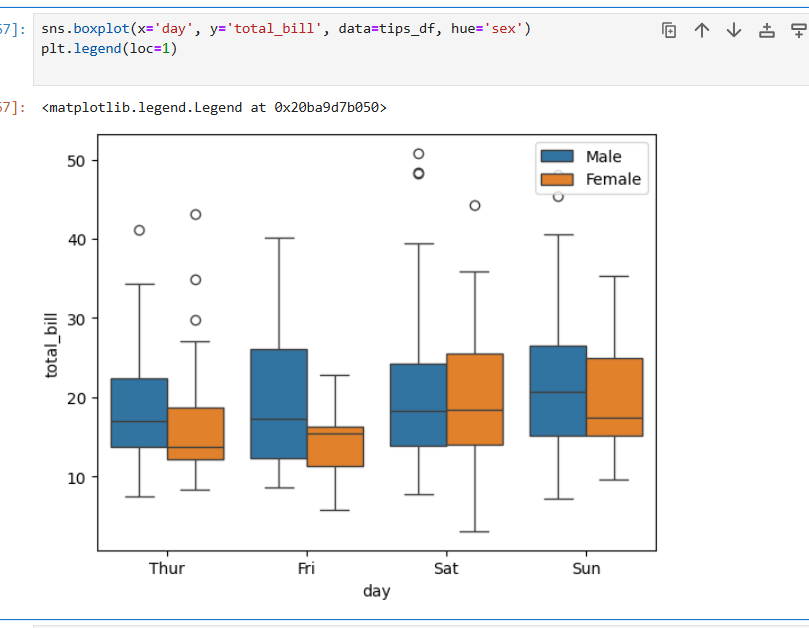
Q1 is the 25th percentile (25% of data below it).

Q3 is the 75th percentile (75% of data below it).

The box shows the **middle 50%** of the data.

The whiskers ( t shaped structures) show the maximum and minimum values.

The diagrams outside the box plot are outliers.



Depending on the gender of the person, we can see for each day of the week which gender is most likely to have larger bills.

For example, on the day Saturday females had greater bills than males, thus indicating more tips from the females.

# VIOLIN PLOT

While a box plot corresponds to the data points, violin plot utilizes a KDE (Kernel Density Estimate) estimation to create the violin plot.

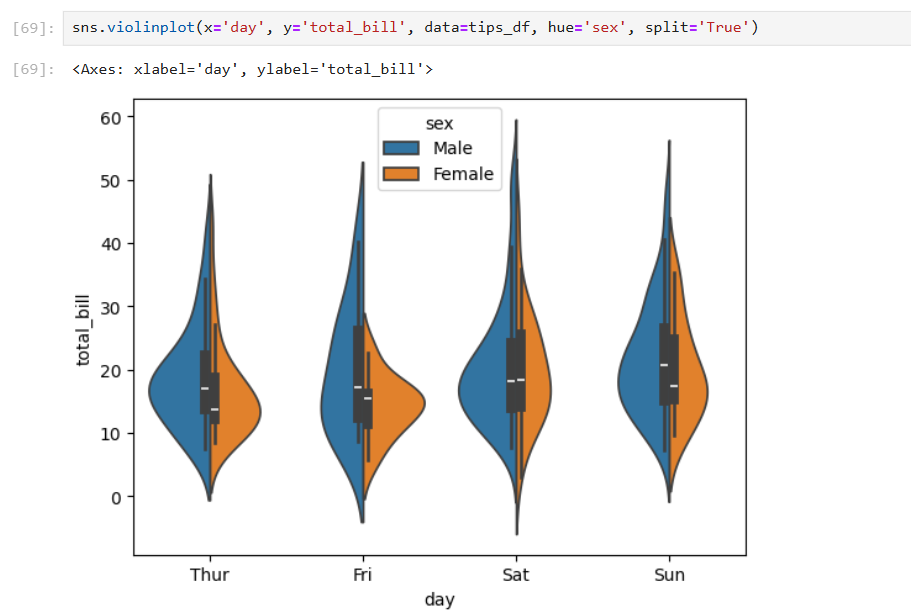
The width of each curve corresponds with the approximate frequency of data points in each region.

In the middle of each density curve is a small box plot.

Note: In a KDE, each data point contributes a small area around its true value. The shape of this area is called the kernel function.

Kernels can have different width, or bandwidth, affecting the influence of each data point.

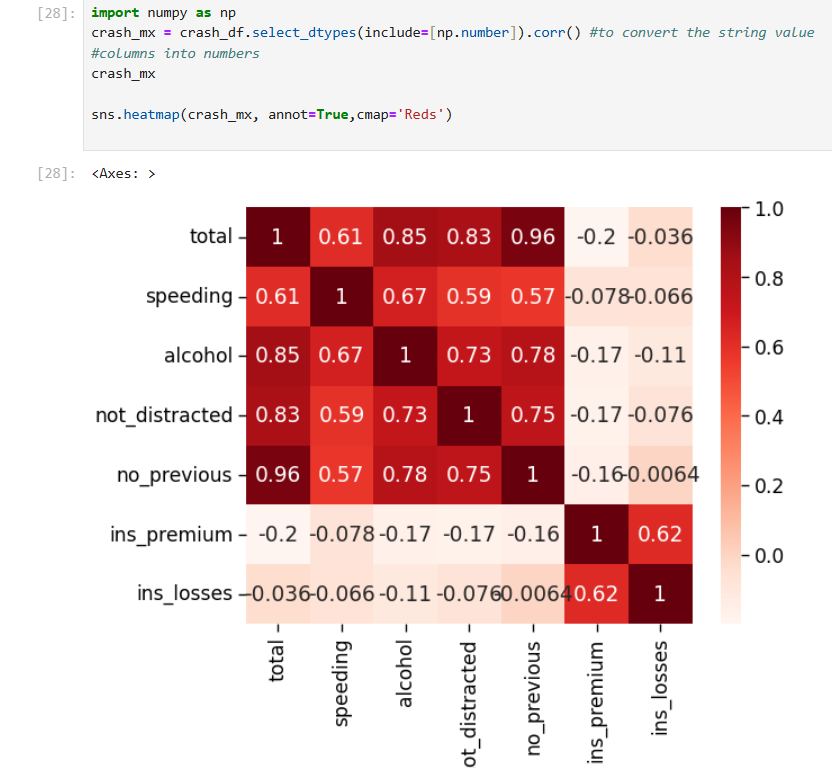
Bandwidth can be determined by the user itself depending on the shape and skew of the data to be plotted.



From the above violin plot, we can see that on Friday, a greater number of females have bills around 10-20 dollars (sharper curve) whereas the number of men paying bills ranging from 10-30 dollars is more or less equal (smoother curve).

# HEATMAPS

Used to show relationships between two variables, one plotted on each axis. By observing how each cell changes colors change across each axis, one can observe if there are any patterns in value for one or both variables.



From the heatmap we can see how speeding and alcohol are closely related (also people who have been involved in previous accidents also have high chances of being inebriated leading to the car crash) showing a darker color

# Key differences:

Matplotlib is a low-level library: Gives one control over plots but requires more code for complex visuals.

Seaborn is a high-level wrapper; Simplifies many tasks like plotting distributions, box plots, heatmaps, and regression lines.

|  |  |  |
| --- | --- | --- |
| ASPECTS | MATPLOTLIB | SEABORN |
| STRENGTHS | Good for customization.  Has a solid community.  Well connected with Numpy and Pandas.  Pyplot provides similar features and syntax as in MATLAB. Therefore. MAATLAB users can easily study it. | Extended version of Matplotlib which uses Matplotlib along with Numpy and Pandas for plotting graphs.  Statistical plots and fast EDA.  Comparatively simple syntax which is easier to learn and understand. |
| WEAKNESSES | Static, cannot make an interactive plot.  Uses comparatively complex and lengthy syntax. | Not interactive.  Not as customizable as Matplotlib. |
| CUSTOMIZATION OPTIONS | Extremely high. Can customize almost everything. | Less customizable than Matplotlib. |
| INTERACTIVITY | Not ideal for interactivity. | Also, not ideal for interactivity, inherits Matplotlib’s limitations. |
| PERFORMANCE WITH LARGE DATASETS | Decent for moderate data sizes, but may lag with very large datasets. | Much more functional and organized.  Treats the whole dataset as a single unit.  Better performance than Matplotlib. |