Assignment No. 08

Aim:
Data Visualization I
1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.
2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

Prerequisites:
1. Prior knowledge of Python programming.
2. Google Colab / Python IDE
3. Jupyter Notebook

Objectives: To check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram. Use the Seaborn library to see if we can find any patterns in the data.

Theory:
1. Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt # Importing the required libraries
import seaborn as sns
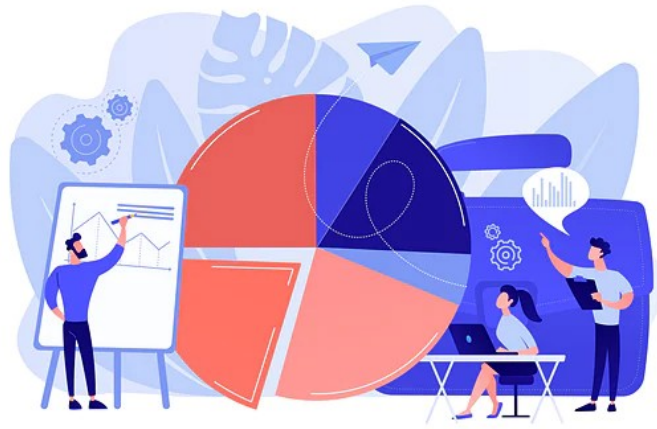%matplotlib inline

2. Data Visualization:
The process of finding trends and correlations in our data by representing it pictorially is called Data Visualization. To perform data visualization in python, we can use various python data visualization modules such as Matplotlib, Seaborn, Plotly, etc.

3. What is Data Visualization?
Data visualization is a field in data analysis that deals with visual representation of data. It graphically plots data and is an effective way to communicate inferences from data. Using data visualization, we can get a visual summary of our data. With pictures, maps and graphs, the human mind has an easier time processing and understanding any given data. Data visualization plays a significant role in the representation of both small and large data sets, but it is especially useful when we have large data sets, in which it is impossible to see all of our data, let alone process and understand it manually.

4. Data Visualization in Python

Python offers several plotting libraries, namely Matplotlib, Seaborn and many other such data visualization packages with different features for creating informative, customized, and appealing plots to present data in the most simple and effective way.

5. Matplotlib and Seaborn

Matplotlib and Seaborn are python libraries that are used for data visualization. They have inbuilt modules for plotting different graphs. While Matplotlib is used to embed graphs into applications, Seaborn is primarily used for statistical graphs.

But when should we use either of the two? Let's understand this with the help of a comparative analysis. The table below provides comparison between Python's two well-known visualization packages Matplotlib and Seaborn.

Matplotlib	Seaborn
It is used for basic graph plotting like line charts, bar graphs, etc.	It is mainly used for statistics visualization and can perform complex visualizations with fewer commands.
It mainly works with datasets and arrays.	It works with entire datasets.
Seaborn is considerably more organized and functional than Matplotlib and treats the entire dataset as a solitary unit.	Matplotlib acts productively with data arrays and frames. It regards the aces and figures as objects.
Seaborn has more inbuilt themes and is mainly used for statistical analysis.	Matplotlib is more customizable and pairs well with Pandas and Numpy for Exploratory Data Analysis.

Let's consider the apple yield (tons per hectare) in Kanto. Let's plot a line graph using this data and see how the yield of apples changes over time. We start by importing Matplotlib and Seaborn.

```python
import matplotlib.pyplot as plt
import seaborn as sns
```
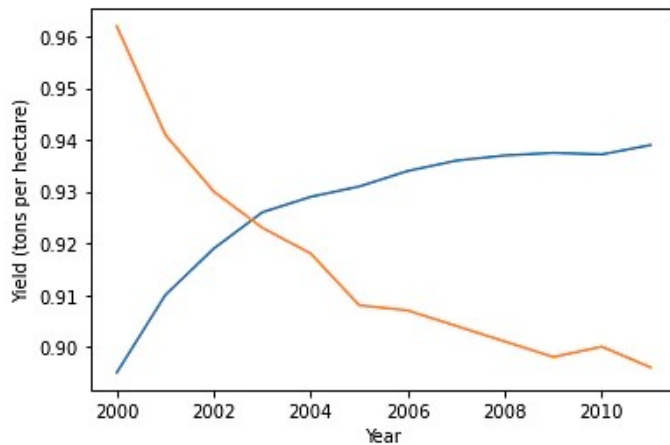
Using Matplotlib

To plot multiple datasets on the same graph, just use the plt.plot function once for each dataset. Let's use this to compare the yields of apples vs. oranges on the same graph.

```
years = range(2000, 2012)
apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931, 0.934, 0.936, 0.937, 0.9375, 0.9372, 0.939]
oranges = [0.962, 0.941, 0.930, 0.923, 0.918, 0.908, 0.907, 0.904, 0.901, 0.898, 0.9, 0.896, ]
```

```
plt.plot(years, apples)
plt.plot(years, oranges)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)');
```


To show each data point on our graph, we can highlight them with markers using the marker argument. Many different marker shapes like a circle, cross, square, diamond, etc. are provided by Matplotlib.
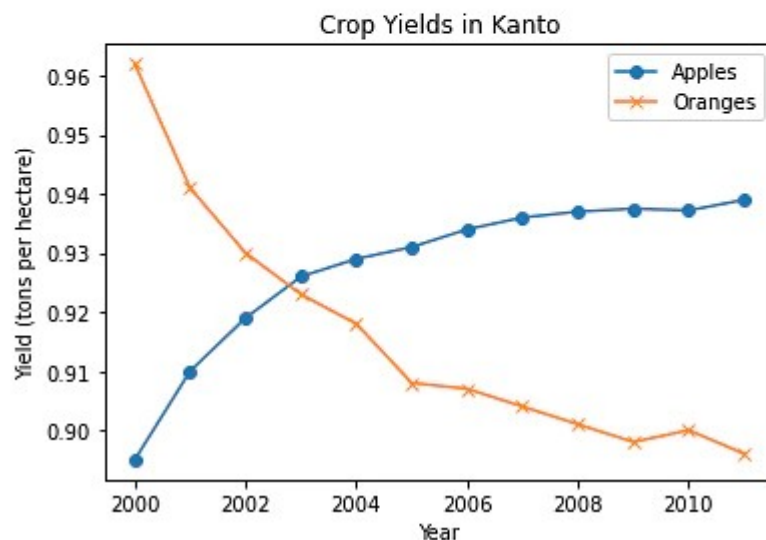
```
plt.plot(years, apples, marker='o')
plt.plot(years, oranges, marker='x')

plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')

plt.title("Crop Yields in Kanto")
plt.legend(['Apples', 'Oranges'])
```

```
<matplotlib.legend.Legend at 0x2054a8940d0>
```

Using Seaborn

An easy way to make your charts look beautiful is to use some default styles from the Seaborn library. These can be applied globally using the sns.set_style function.
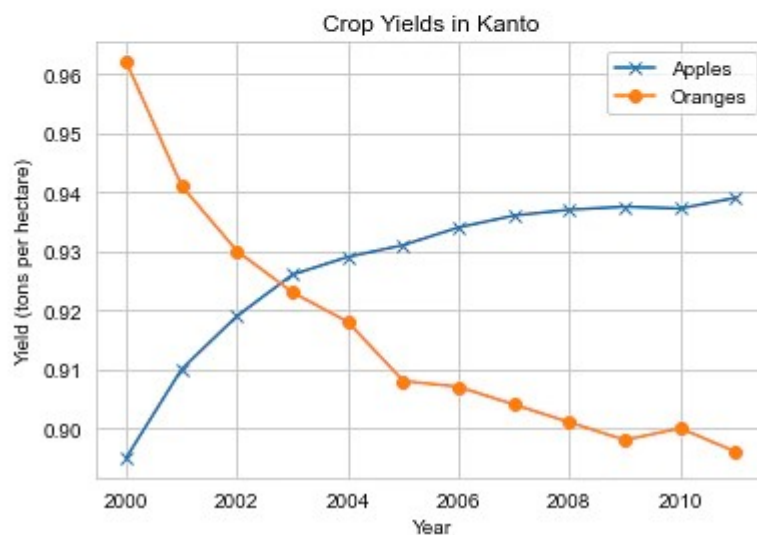
```
sns.set_style("whitegrid")
```

```
plt.plot(years, apples, marker = 'x')
plt.plot(years, oranges, marker = 'o')

plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')

plt.title("Crop Yields in Kanto")
plt.legend(['Apples', 'Oranges'])
```

```
<matplotlib.legend.Legend at 0x2054aa1d1f0>
```
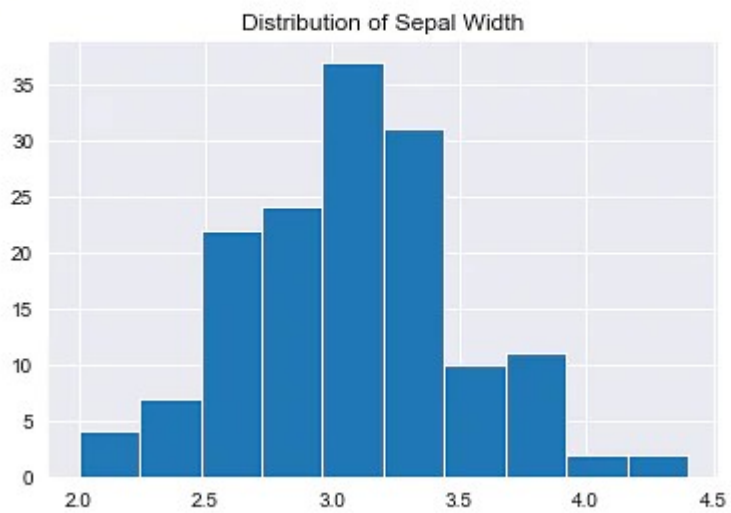

6. Histograms

A Histogram is a bar representation of data that varies over a range. It plots the height of the data belonging to a range along the y-axis and the range along the x-axis. Histograms are used to plot data over a range of values. They use a bar representation to show the data belonging to each range. Let's again use the 'Iris' data which contains information about flowers to plot histograms.

```
flowers_df = sns.load_dataset("iris")
flowers_df.sepal_width
```

```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      ...
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
Name: sepal_width, Length: 150, dtype: float64
```

Now, let's plot a histogram using the hist() function.

```
plt.title("Distribution of Sepal Width")
plt.hist(flowers_df.sepal_width)
```


Distribution of Sepal Width

Conclusion:

Thus we have studied how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram also the use of Seaborn library to see if we can find any patterns in the data.