

# Data Analytics

## Lab # 8

### Visualization with Matplotlib

Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers. Visualizing data is an essential part of data analysis and machine learning. In this lab we'll use Python library Matplotlib to learn and apply some popular data visualization techniques.

We'll use the matplotlib.pyplot module for basic plots like line & bar charts. It is often imported with the alias plt.

### Line Chart

The line chart is one of the simplest and most widely used data visualization techniques. A line chart displays information as a series of data points or markers connected by straight lines. You can customize the shape, size, color, and other aesthetic elements of the lines and markers for better visual clarity.

Here's a Python list showing the yield of apples (tons per hectare) over six years.

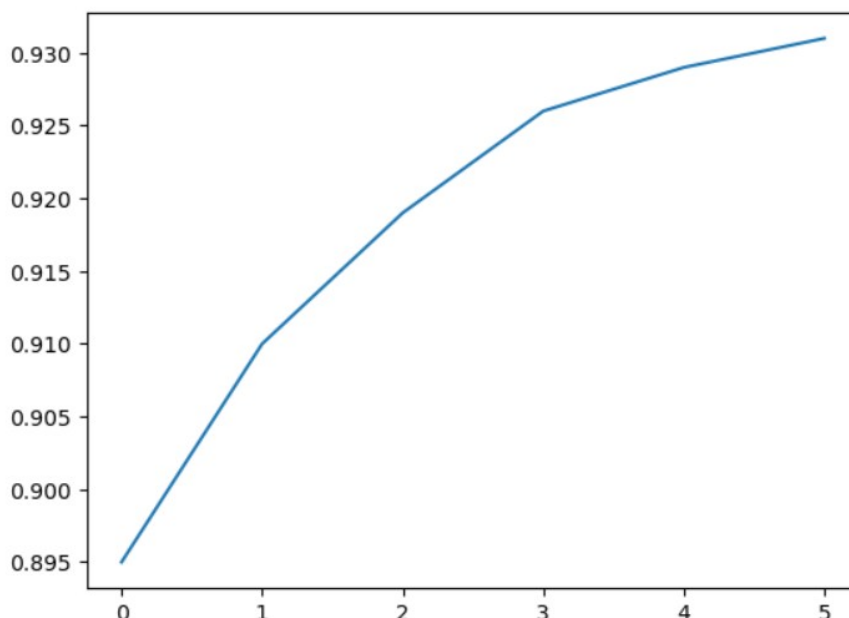
```
In [1]: import matplotlib.pyplot as plt
```

```
In [2]: yield_apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931]
```

We can visualize how the yield of apples changes over time using a line chart. To draw a line chart, we can use the plt.plot function.

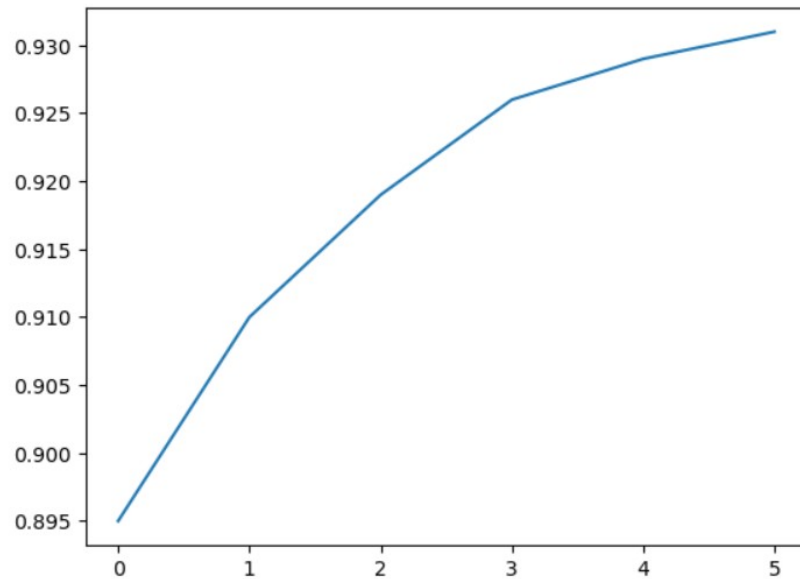
```
In [3]: plt.plot(yield_apples)
```

```
Out[3]: [<matplotlib.lines.Line2D at 0x7f51366d90c0>]
```



Calling the `plt.plot` function draws the line chart as expected. It also returns a list of plots drawn [`<matplotlib.lines.Line2D at 0x7ff70aa20760>`], shown within the output. We can include a semicolon (;) at the end of the last statement in the cell to avoiding showing the output and display just the graph.

```
In [4]: plt.plot(yield_apples);
```

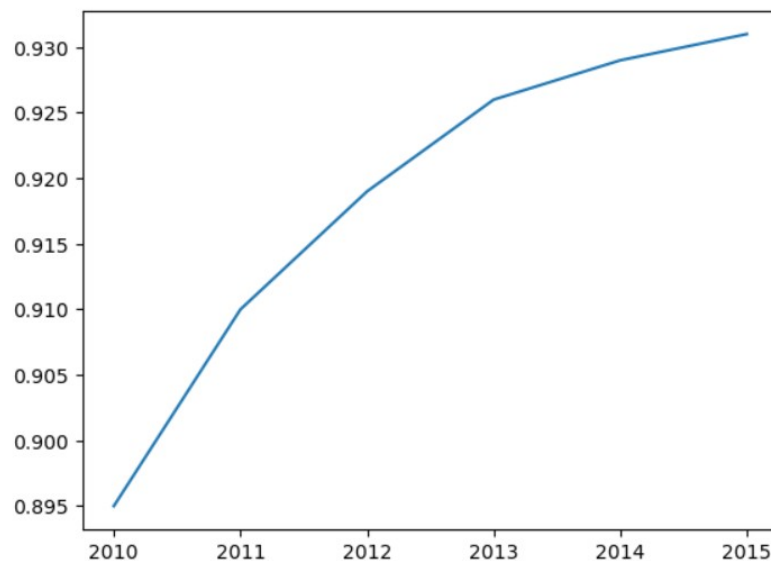


## Customizing the X-axis

The X-axis of the plot currently shows list element indexes 0 to 5. The plot would be more informative if we could display the year for which we're plotting the data. We can do this by two arguments `plt.plot`

```
In [5]: years = [2010, 2011, 2012, 2013, 2014, 2015]
yield_apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931]
```

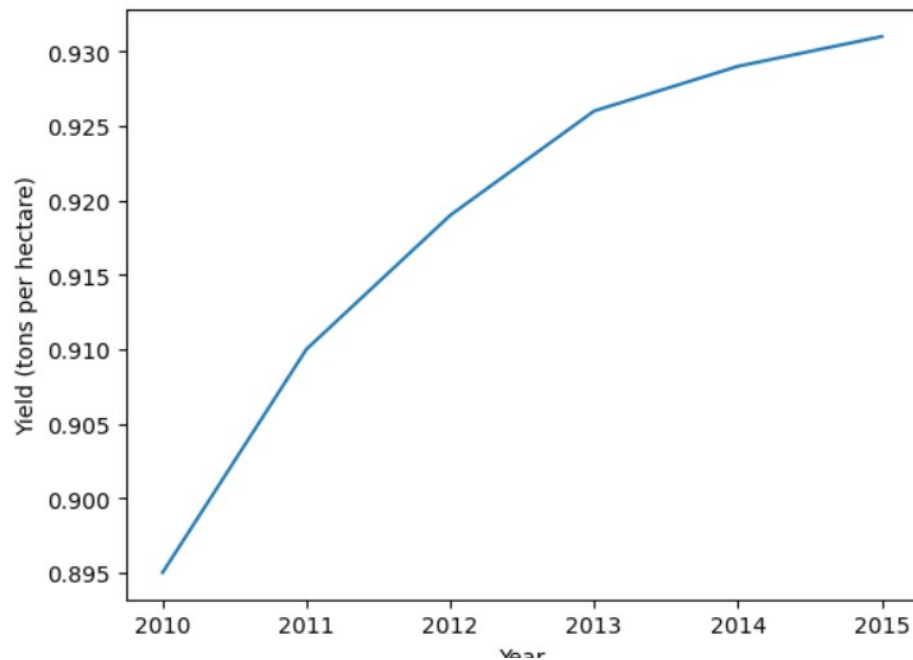
```
In [6]: plt.plot(years, yield_apples);
```



## Axis Labels

We can add labels to the axes to show what each axis represents using the `plt.xlabel` and `plt.ylabel` methods.

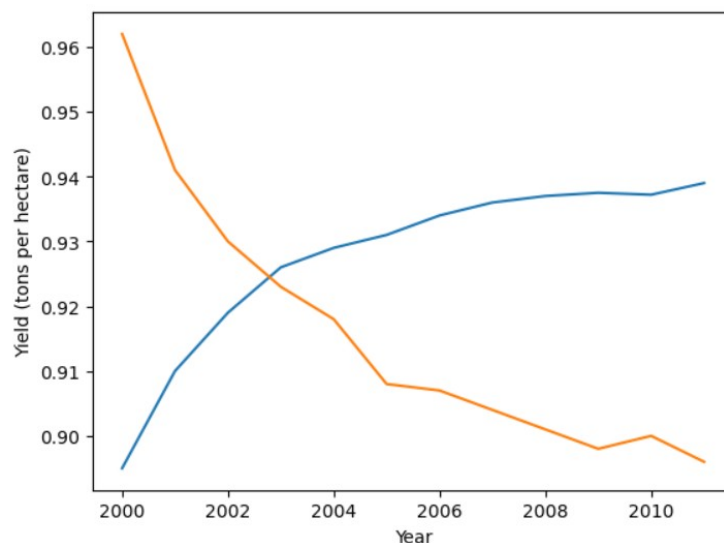
```
In [7]: plt.plot(years, yield_apples)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)');
```



## Plotting Multiple Lines

You can invoke the `plt.plot` function once for each line to plot multiple lines in the same graph. Let's compare the yields of apples vs. oranges

```
In [9]: plt.plot(years, apples)
plt.plot(years, oranges)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)');
```

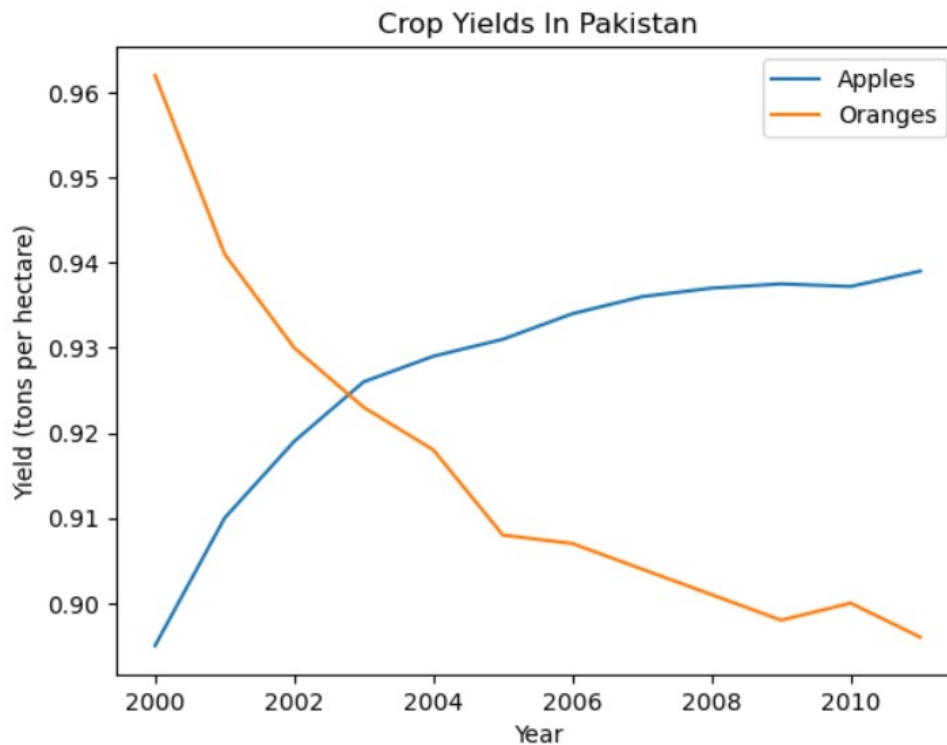


## Chart Title and Legend

To differentiate between multiple lines, we can include a legend within the graph using the `plt.legend` function. We can also set a title for the chart using the `plt.title` function.

```
In [11]: plt.plot(years, apples)
plt.plot(years, oranges)

plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields In Pakistan")
```



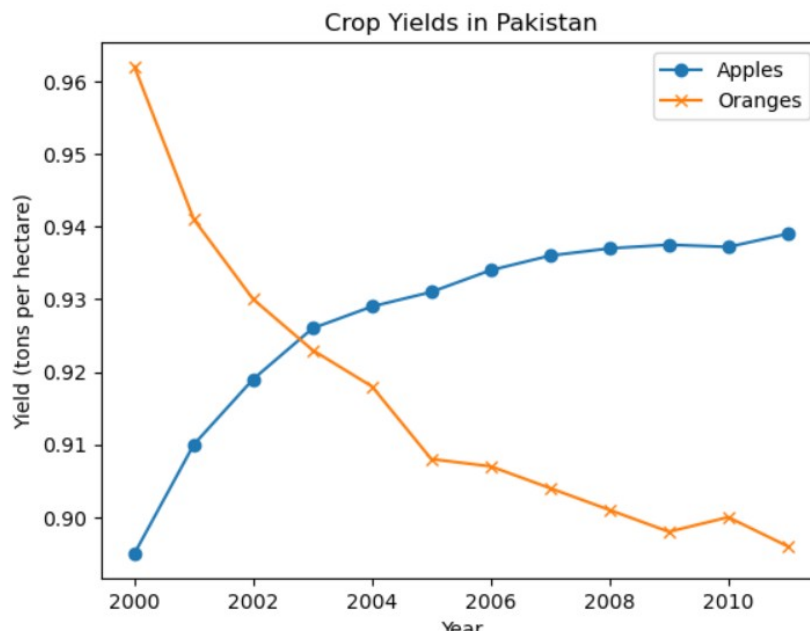
## Line Markers

We can also show markers for the data points on each line using the `marker` argument of `plt.plot`. Matplotlib provides many different markers, like a circle, cross, square, diamond, etc. You can find the full list of marker types here: [https://matplotlib.org/3.1.1/api/markers\\_api.html](https://matplotlib.org/3.1.1/api/markers_api.html)

```
In [15]: # plt.figure(figsize=(12, 6))
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 Pakistan")
plt.legend(['Apples', 'Oranges']);
```



## Tasks

consider the CVS file of Italy COVID cases

1. Display the graph of death cases verses months.
2. Display the graph of new cases verses months.
3. Compare the new cases and death cases day-wise on multi-line graph, mark the legends and properly label and title the graph.
4. Display how the new cases and new tests are related day-wise on multi-line graph, mark the legends and properly label and title the graph.