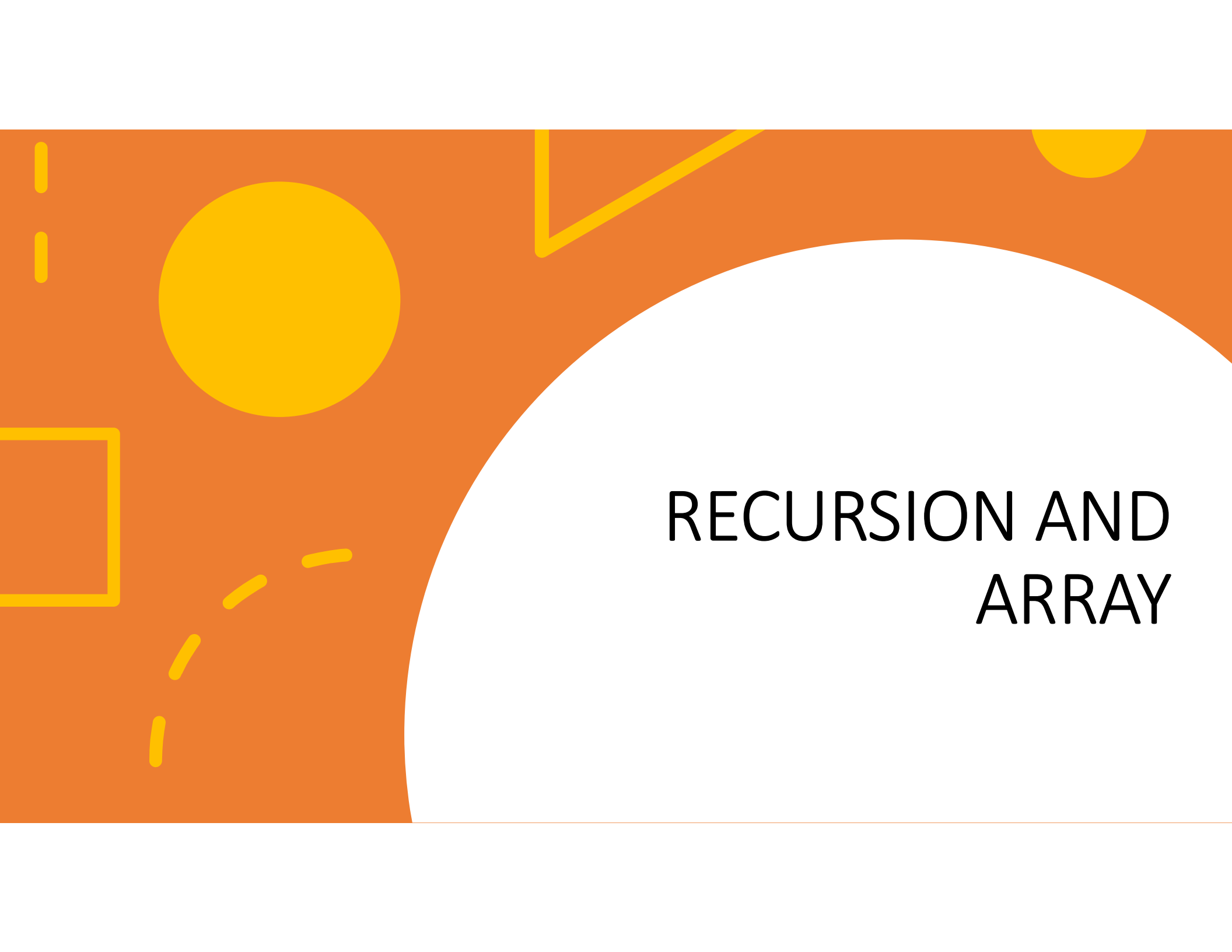
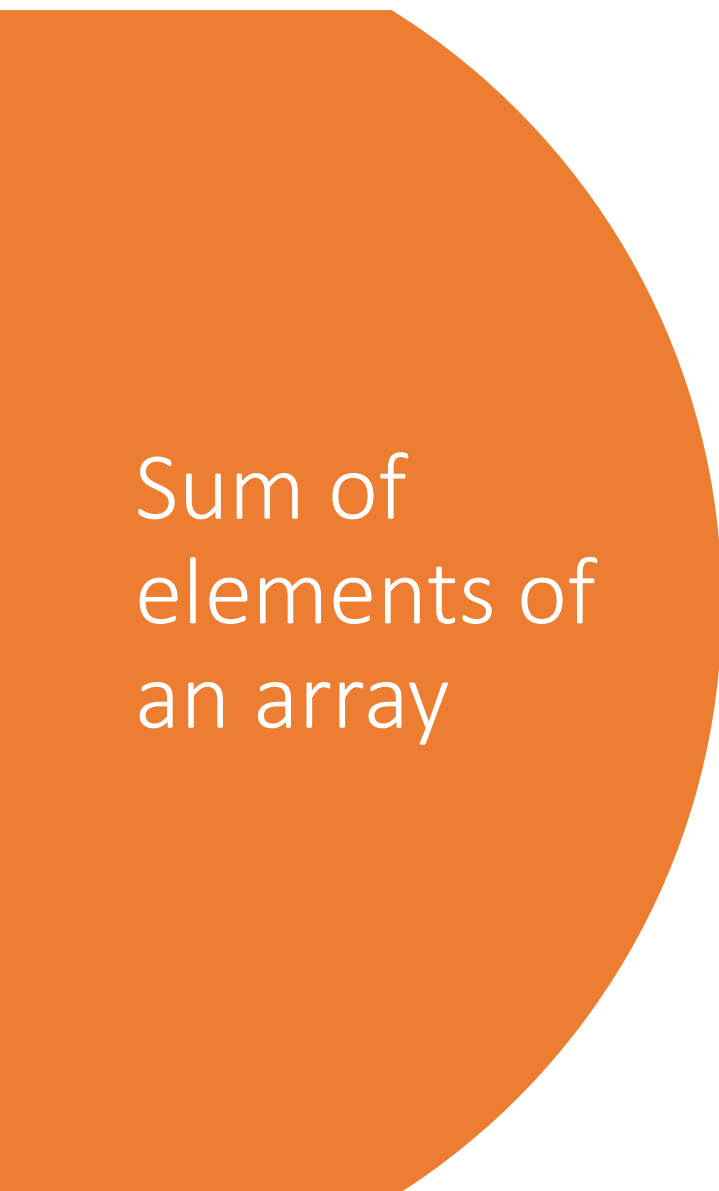




RECURSION AND ARRAY

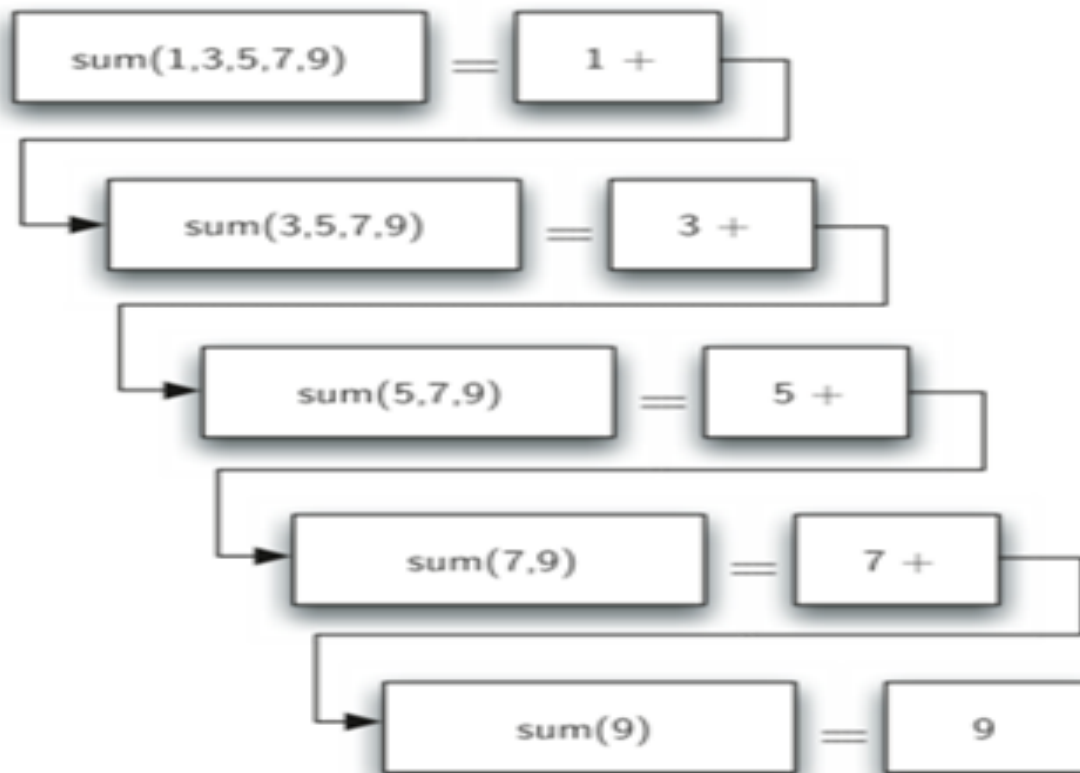
A large orange shape on the left side of the slide, consisting of a rectangle with a quarter-circle cutout on its right side.

Sum of
elements of
an array

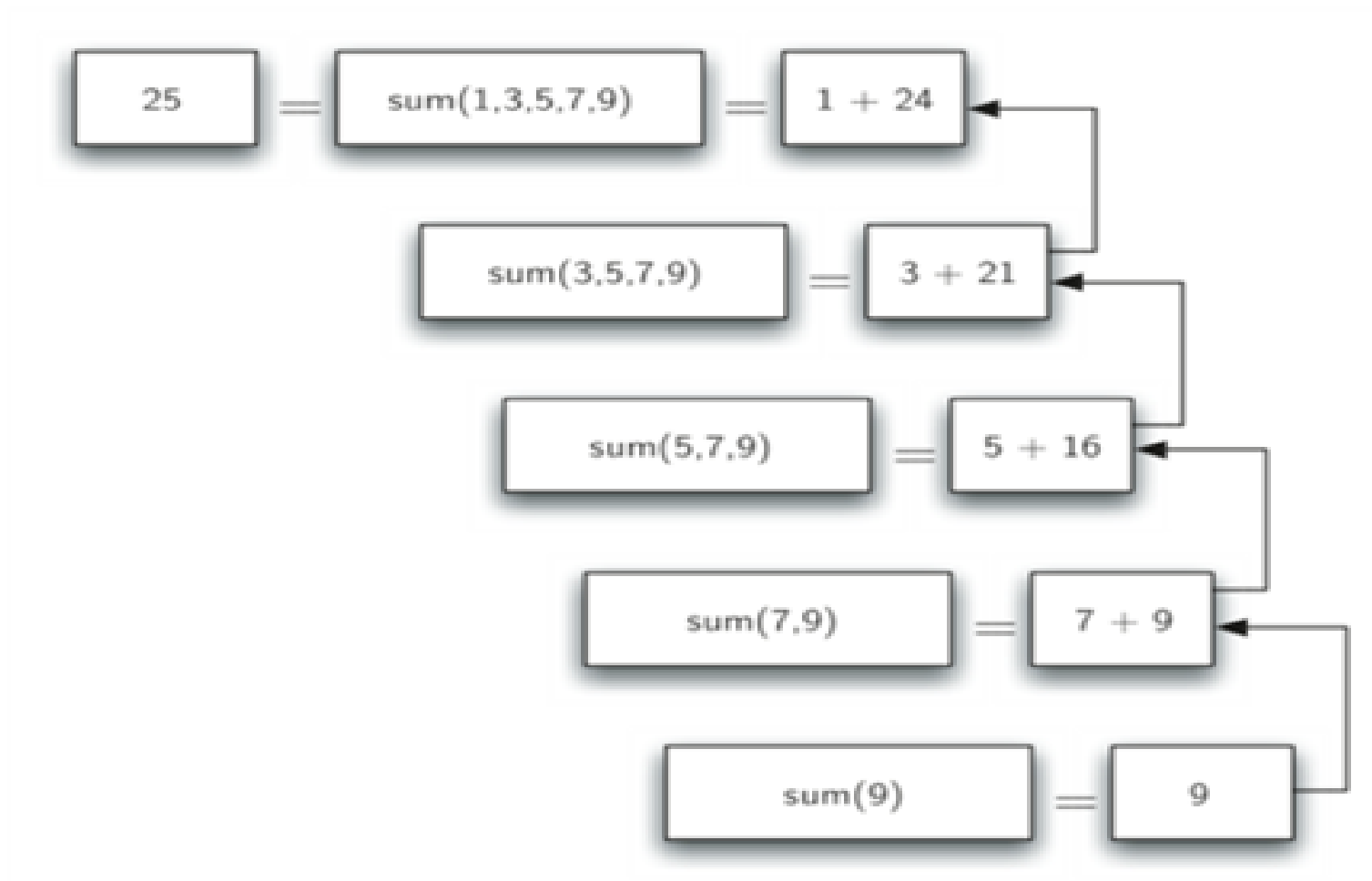
```
def getSum(piece):  
    if len(piece)==0:  
        return 0  
    else:  
        return piece[0] + getSum(piece[1:])  
print (getSum([1,3,5,7,9]))
```



USING STACK TO EVALUATE



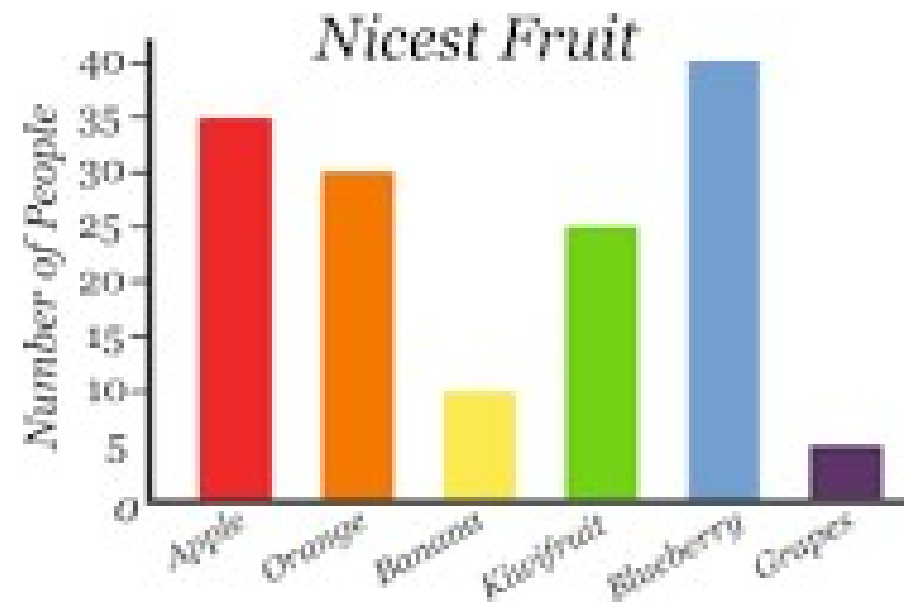
USING STACK TO EVALUATE



Binary Search

```
def binary_search(alist, start, end, key):  
  
    if start > end:  
        return -1  
    mid = (start + end)//2  
    if key > alist[mid] :  
        return binary_search(alist, mid + 1, end, key)  
    elif key < alist[mid] :  
        return binary_search(alist, start, mid-1, key)  
    else:  
        return mid  
  
alist=[33,44,55,66,77]  
key = int(input('The number to search for: '))  
  
index = binary_search(alist, 0, len(alist), key)  
if index < 0:  
    print(key,' was not found.')  
else:  
    print(key,' was found at index ',index)
```

DATA VISUALIZATION USING PYPLOT





DATA VISUALIZATION

Data Visualization is an important part of business activities as organizations nowadays collect a huge amount of data. Sensors all over the world are collecting climate data, user data through clicks, car data for prediction of steering wheels etc. All of these data collected hold key insights for businesses and visualizations make these insights easy to interpret.

Data is only as good as it's presented.



Why are visualizations important?

Visualizations are the easiest way

- To analyze and absorb information.
- Help to easily understand the complex problem.
- They help in identifying patterns, relationships, and outliers in data.
- It helps in understanding business problems better and quickly.
- It helps to build a compelling story based on visuals.
- Analysis



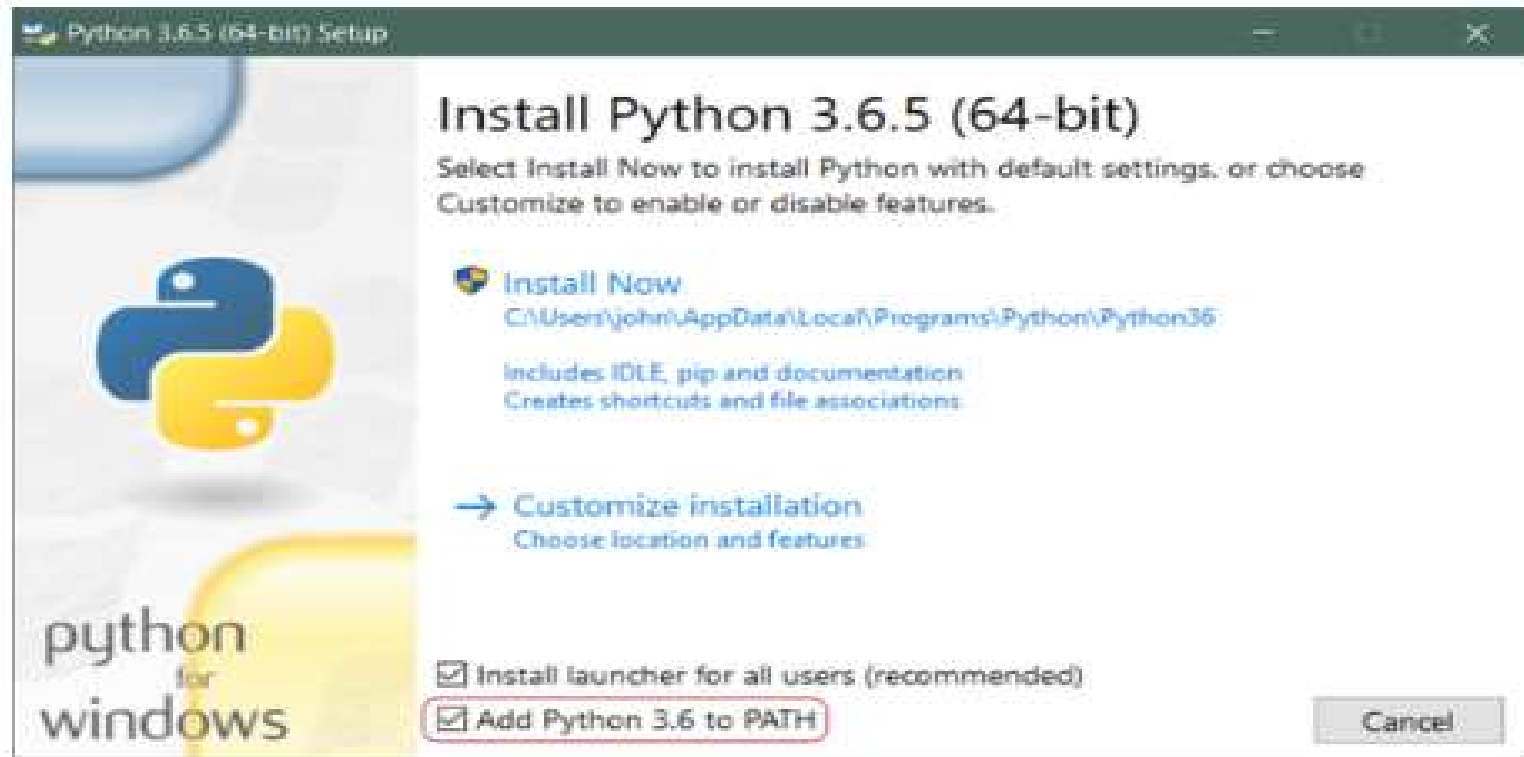
Data visualization contd.

- Data visualization basically refers to the graphical or visual representation of information and data using charts, graphs, maps
- Data visualization helps for better understanding of the meaning of data.

Matplotlib

- Matplotlib is a 2-D plotting library that helps in visualizing figures. Matplotlib emulates Matlab like graphs and visualizations. Matlab is not free, is difficult to scale and as a programming language is tedious. So, matplotlib in Python is used as it is a robust, free and easy library for data visualization.
- pyplot is a collection of methods which exists in matplotlib library.
- Pyplot allow user to construct 2D plots easily and interactively.

Setting path



All Apps Documents Web More

Best match

-  **Edit the system environment variables**
Control panel

Settings

-  **Edit environment variables for your account** >

Search the web

-  **environment** - See web results >

Edit the system environment variables
Control panel

 Open

Python 3.6.5 (64-bit)
Python with default settings, or choose alternative features.

Local Programs\Python\Python36

Documentation

Associations

Python (recommended)

Cancel

AutoSave Off data visualization.pptx Search madhu singh

File Home Insert Design Transitions Animations Slide Show Review View Help

Paste New Slide Layout Reset Section B I U S Aa AV A Paragraph Drawing Editing Voice Designer

Clipboard Find Replace Select Dictate Design Ideas

System Properties

Computer Name Hardware Advanced System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...

User Profiles
Desktop settings related to your sign-in
Settings...

Startup and Recovery
System startup, system failure, and debugging information
Settings...

Environment Variables...

OK Cancel Apply

Design Ideas
Generating Design Ideas...

Slide 8 of 27

Type here to search

9:25 PM 3/31/2020

AutoSave Off data visualization.pptx Search

File Home Insert Design Transitions Animations Slide Show Review View

Paste New Slide Layout Reset Section Paragraph

Clipboard

4 5 6 7 8 9 10 11

Slide 9 of 28

System Properties

Computer Name Hardware Advanced System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...

User Profiles
Desktop settings related to your sign-in
Settings...

Startup and Recovery
System startup, system failure, and debugging information
Settings...

Environment Variables...

OK Cancel Apply

Type here to search

AutoSave Off data visualization.pptx Search madhu singh

File Home Insert Design Transitions Animations Slide Show Review View Help Picture Format

Remove Background Corrections Color Artistic Effects Compress Pictures Change Picture Transparency Reset Picture

System Properties

Computer Name Hardware Advanced System

You must be logged on as an Administrator to perform this action.

Performance

Visual effects, processor scheduling, memory management, and virtual memory.

User Profiles

Desktop settings related to your sign-in.

Startup and Recovery

System startup, system failure, and debugging.

OK

Environment Variables

User variables for madhusingh

Variable	Value
OneDrive	C:\Users\madhusingh\OneDrive - Delhi Public School Ghaziabad
OneDriveCommercial	C:\Users\madhusingh\OneDrive - Delhi Public School Ghaziabad
Path	C:\Users\madhusingh\AppData\Local\Programs\Python\Python37...
script	C:\Users\madhusingh\AppData\Local\Programs\Python\Python37...
TEMP	C:\Users\madhusingh\AppData\Local\Temp
TMP	C:\Users\madhusingh\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
configsetroot	C:\WINDOWS\ConfigSetRoot
DriverData	C:\Windows\System32\Drivers\DriverData
easyplusdk	"C:\Program Files (x86)\Common Files\lenovo\easyplusdk\bin"
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32;Wb...

New... Edit... Delete

OK Cancel

Bring Forward Send Backward Selection Pane Arrange Size

Design Ideas

Click to add title

Slide 9 of 28

Type here to search

Notes

9:26 PM 3/31/2020

AutoSave Off data visualization.pptx Search madhu singh

File Home Insert Design Transitions Animations Slide Show Review View Help

Clipboard Paste New Slide Layout Reset Section B I U S AV

System Properties Computer Name Hardware Advanced Sys You must be logged on as an Administrator to Performance Visual effects, processor scheduling, memor User Profiles Desktop settings related to your sign-in Startup and Recovery System startup, system failure, and debu

Environment Variables User variables for madhusingh

Variable	Value
OneDrive	C:\Users\madhusingh\OneDrive - Delhi Public School Ghaziabad
OneDriveCommercial	C:\Users\madhusingh\OneDrive - Delhi Public School Ghaziabad
Path	C:\Users\madhusingh\AppData\Local\Programs\Python\Python37...
script	C:\Users\madhusingh\AppData\Local\Programs\Python\Python37...
TEMP	C:\Users\madhusingh\AppData\Local\Temp
TMP	C:\Users\madhusingh\AppData\Local\Temp

New User Variable

Variable name: path

Variable value: C:\Users\madhusingh\AppData\Local\Programs\Python\Python37-32\scripts

Browse Directory... Browse File... OK Cancel

Path

Design Ideas

Click to add title

Click to add title

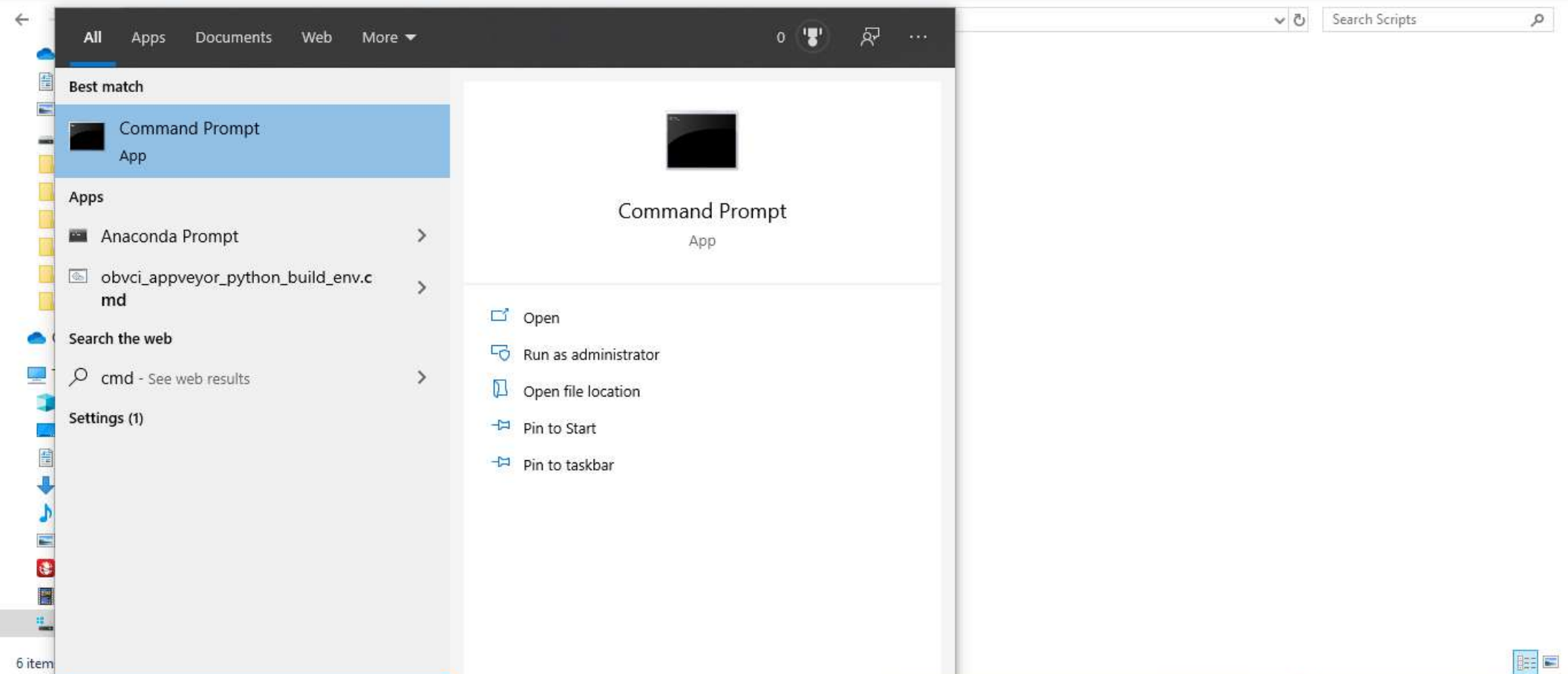
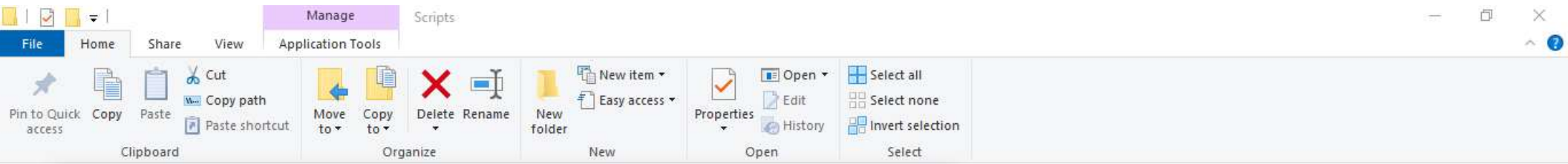
Click to add title

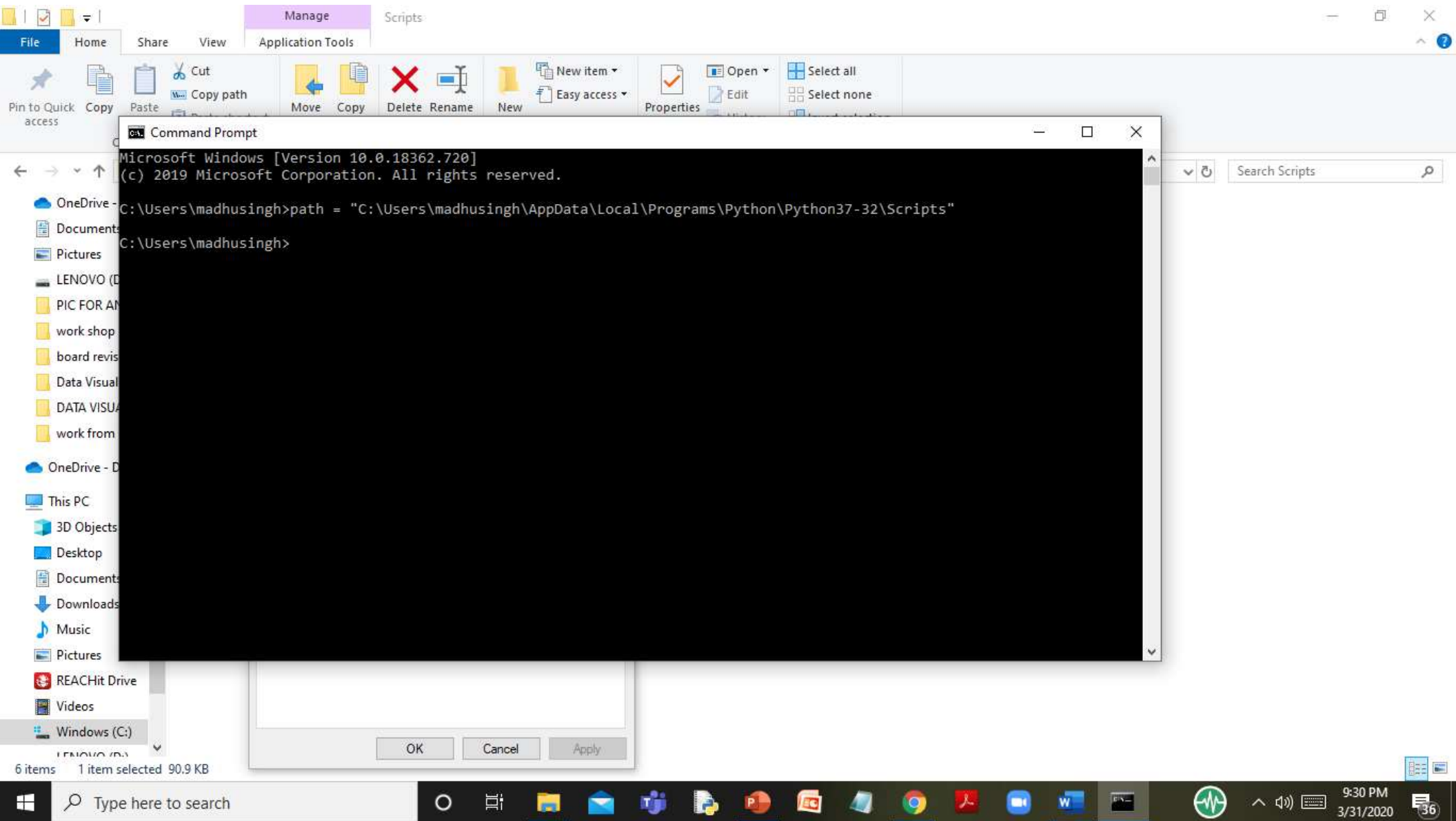
Click to add title

Slide 11 of 30

Type here to search

9:27 PM 3/31/2020





To install the matplotlib :

- `pip install matplotlib`

```
(base) C:\>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\niti\anaconda3\lib\site-packages (3.0.2)
Requirement already satisfied: numpy>=1.10.0 in c:\users\niti\anaconda3\lib\site-packages (from matplotlib) (1.15.4)
Requirement already satisfied: cycler>=0.10 in c:\users\niti\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\niti\anaconda3\lib\site-packages (from matplotlib) (1.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\niti\anaconda3\lib\site-packages (from matplotlib) (2.3.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\niti\anaconda3\lib\site-packages (from matplotlib) (2.5)
Requirement already satisfied: six in c:\users\niti\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib) (1.12.0)
Requirement already satisfied: setuptools in c:\users\niti\anaconda3\lib\site-packages (from kiwisolver>=1.0.1->matplotlib) (40.6.3)

(base) C:\>pip install pandas
Requirement already satisfied: pandas in c:\users\niti\anaconda3\lib\site-packages (0.23.4)
Requirement already satisfied: python-dateutil>=2.5.0 in c:\users\niti\anaconda3\lib\site-packages (from pandas) (2.7.0)
Requirement already satisfied: pytz>=2011k in c:\users\niti\anaconda3\lib\site-packages (from pandas) (2018.7)
Requirement already satisfied: numpy>=1.9.0 in c:\users\niti\anaconda3\lib\site-packages (from pandas) (1.15.4)
Requirement already satisfied: six>=1.5 in c:\users\niti\anaconda3\lib\site-packages (from python-dateutil>=2.5.0->pandas) (1.12.0)
```

Installing Matplotlib

Run the following command for installing matplotlib –

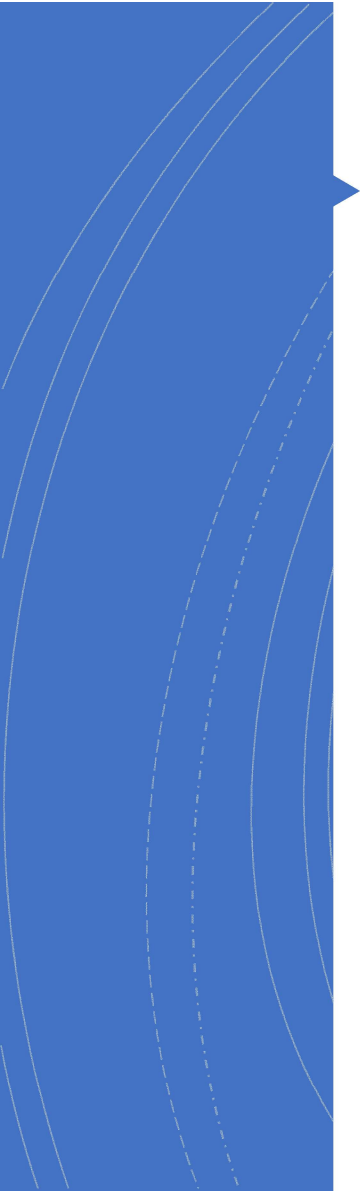
```
1 pip install matplotlib
```

Installing Pandas

```
1 pip install pandas
```

[Pandas](#) helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R.

```
PIP show matplotlib
```



PyPlot

- Python provides many libraries for data visualization like matplotlib, seaborn, ggplot, Bokeh etc. Here i am using the most popular **matplotlib** library.

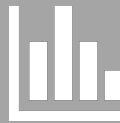
So let's a look on matplotlib.

- [Matplotlib](#) library is a graph plotting library of python.
- Using matplotlib we can plot different scatter plots, line graphs, bar graphs, pie chart and histograms .
- Using these plots we can visualize our data.
- It provides an object-oriented APIs for embedding plots into applications using general-purpose GUI toolkits like Tkinter etc.

Commonly used chart types:



Line Chart



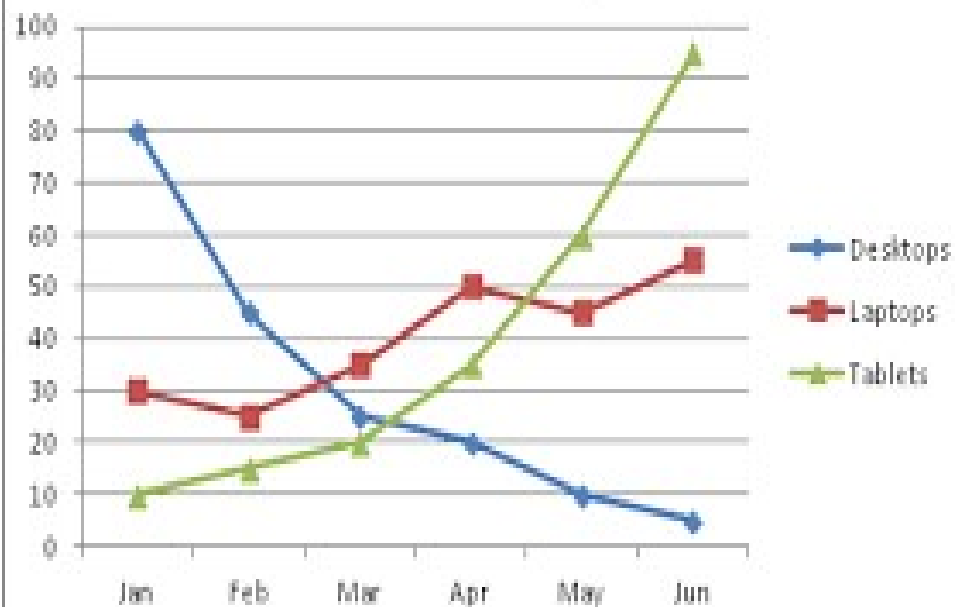
Bar Chart



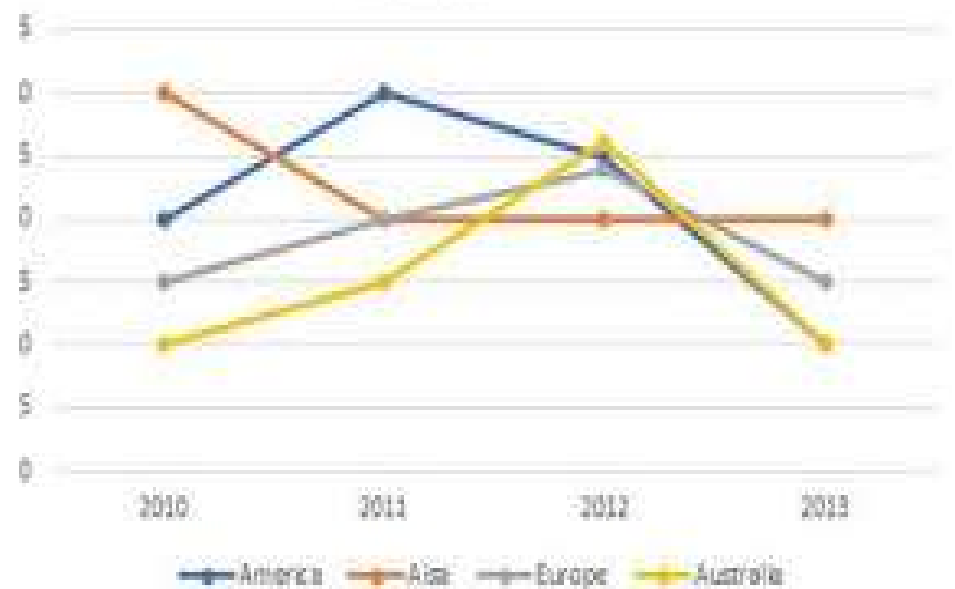
Pie Chart

LINE CHART

Product Trends by Month



Traffic Jam



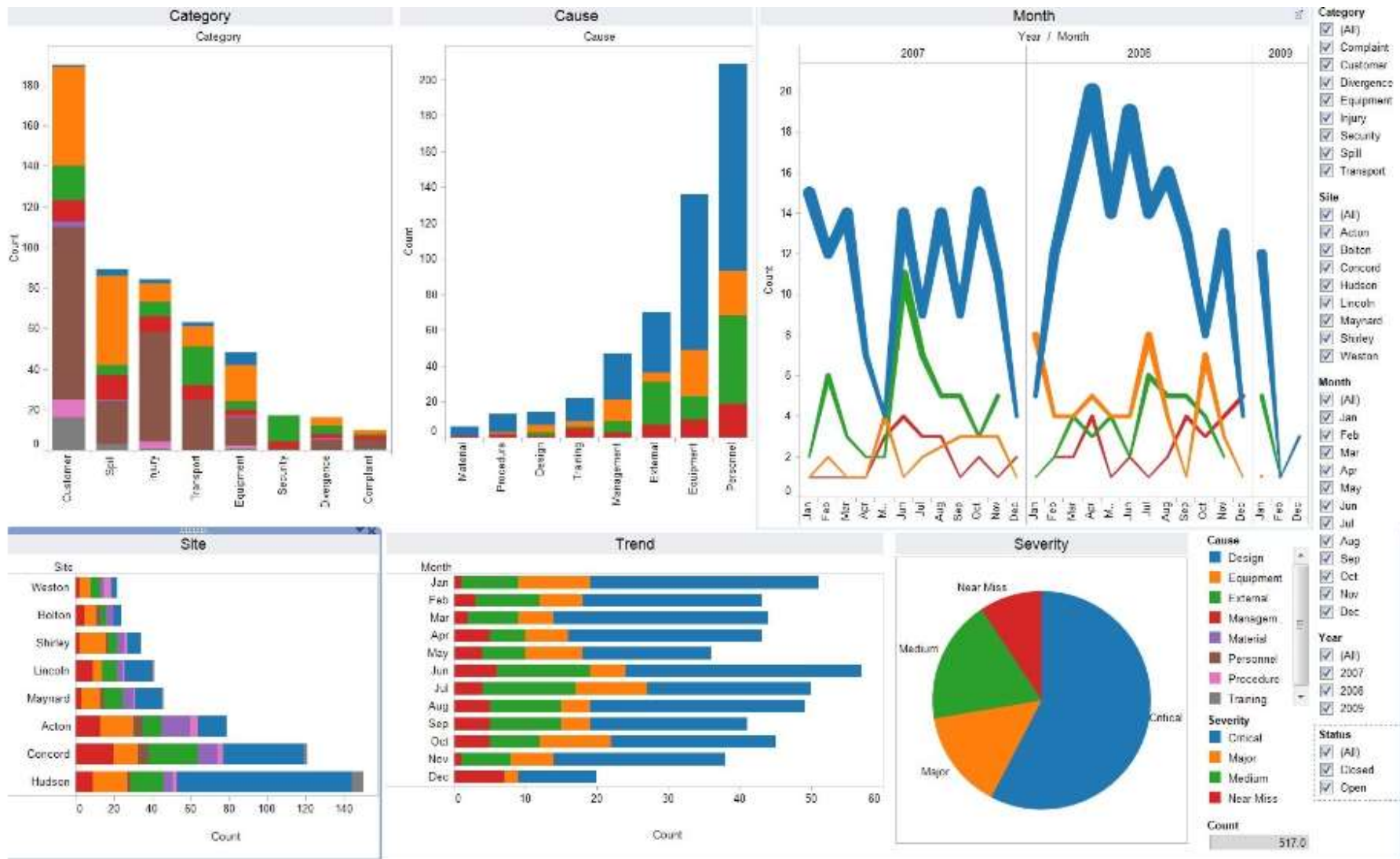
Line Chart

A line chart displays information as a series of data points called “markers”.

Pyplot provides `plot()` function to create line graph

To set the labels for x-axis and y-axis, `xlabel( )` and `ylabel( )` functions are used.

Use `show( )` function to display the line.



Data Visualization Python Tutorial

steps in -

- Importing matplotlib.
- `plt.plot(X, Y , color, others)` plot Y versus X as lines and/OR markers .
- `Plt.xlabel("Your text ")` set the X -Axis label of the current axes.
- `Plt.ylabel("Your text ")` Set the Y axis label of the current axes.
- `Plt.title("Your Title ")` set the title of the current axes.
- `Plt.show()` display a figure

This can be better understood through a practical implementation that follows



Here's some basic code to generating one of the most simple graphs that we can, it will take us only 3 lines.

#Importing pyplot

from matplotlib import pyplot as plt

#Plotting to our canvas

plt.plot([1,2,3],[4,5,1])

#Showing what we plotted

plt.show()

Example:

```
from matplotlib import pyplot as plt
x = [5,8,10]
y = [12,16,6]
plt.plot(x,y)
plt.title('Epic Info')
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.show()
```

Line Graph

A line chart or line graph is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments.

Line graphs are usually used to find relationship between two data sets on different axis; for instance X, Y.

From the above source I got the data of both X and Y axis. year will be on x-axis where population of both countries on Y axis. Since I need two different lines so plot was called twice. The color attribute used to assign the color of the line.

xlabel and ylabel are being used to give some friendly name to the axes and finally .title() for giving the name of the entire graph. When runs it will appear like given below:

MULTIPLE PLOTS

- if we want to plot multiple lines on one chart we can simply call plot() function multiple times.

```
import matplotlib.pyplot as plt
```

```
x=[1,2,3]
```

```
Y =[5,7,4]
```

```
plt.plot(x,y,label='first line')
```

```
x=[1,2,3]
```

```
Y =[10,11,14]
```

```
plt.plot(x2,y2,label='second line')
```

```
Plt.xlabel("Plot Number")
```

```
Plt.ylabel("Important variables")
```

```
Plt.title("New Graph")
```

```
Plt.legend()
```

```
Plt.show()
```

Python 3.7.1 Shell

File Edit Shell Debug Options Window Help

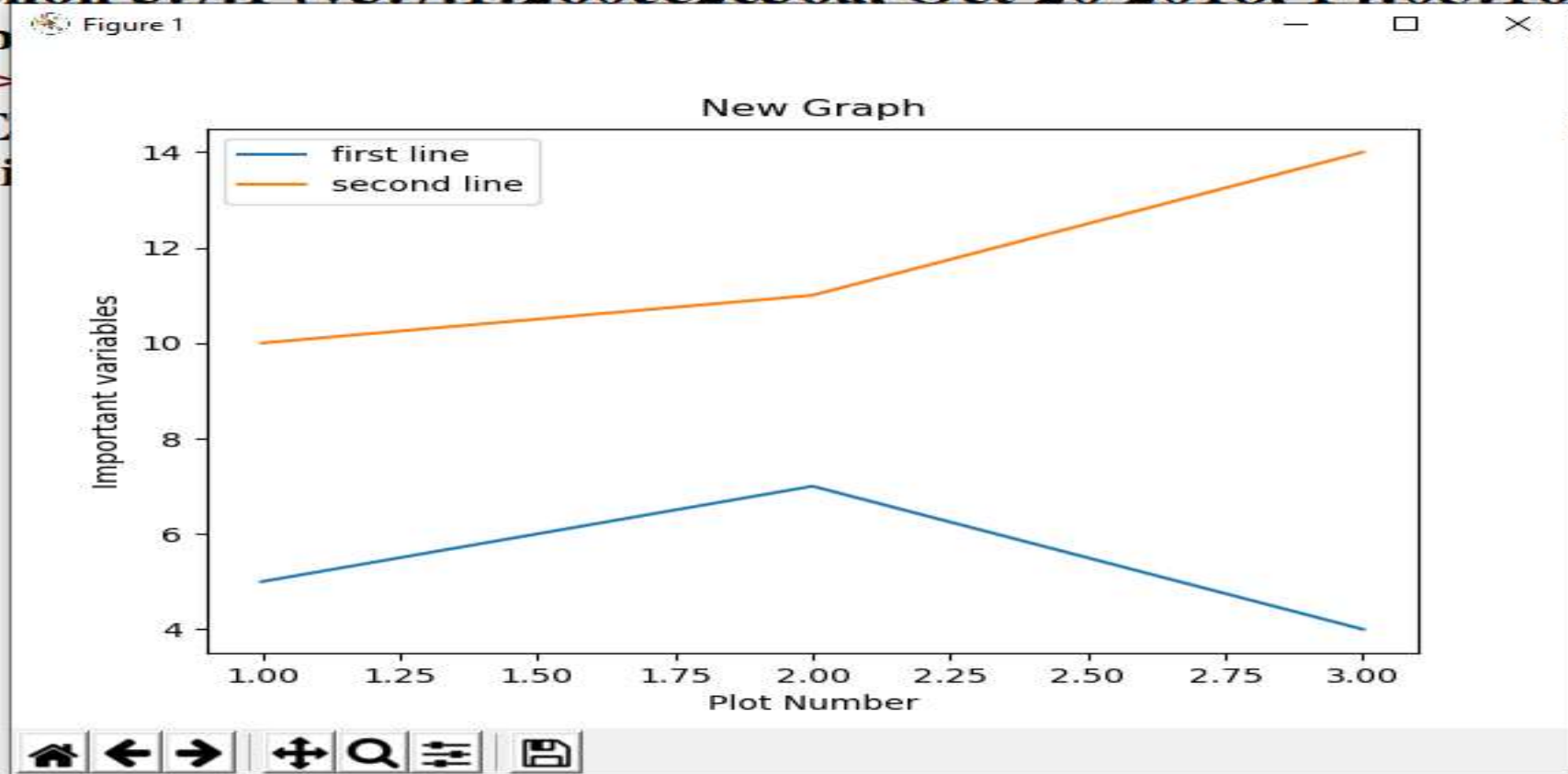
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16)

Type

>>>

RE

a Vi



Type here to search



Multiple views

in case we want to plot lesions in different views in the same window we can use the subplot()function .

```
T=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 , 12 , 13, 14, 15, 16, 17, 18, 19,20 ]
```

```
S=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 , 12 , 13, 14, 15, 16, 17, 18, 19,20 ]
```

```
S2 =[4, 5, 6, 7, 8, 9, 10, 11 , 12, 13, 14, 15, 16, 17, 18 come on 19, 20, 21, 22, 23 ]
```

```
plt.subplot (2, 1, 1)
```

```
plt.plot(t,s)
```

```
plt.xlabel('items')
```

```
plt.ylabel('value')
```

```
plt.title('first chart')
```

```
plt.grid('True')
```

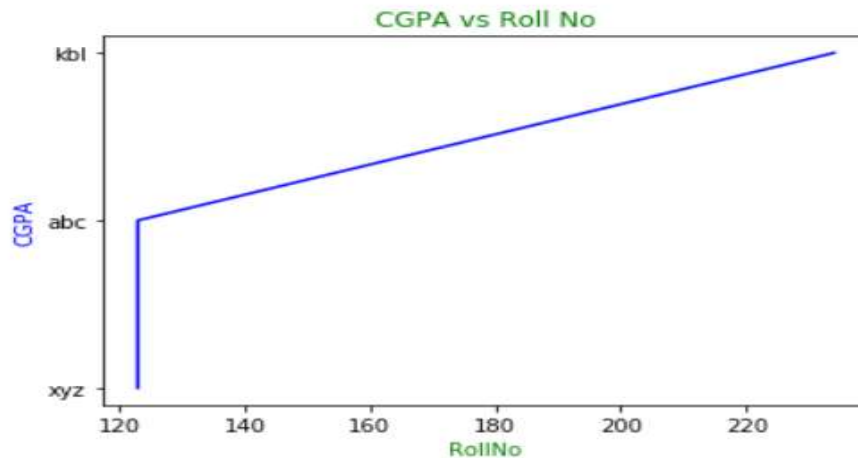
Multiple views code contd.

```
plt.subplot(2, 1, 2)
plt.plot(t,s2)
plt.xlabel('Items')
plt.ylabel('Value')
plt.title('\n\n Second Chart')
plt.grid('True')
```

#2 Line Chart

- Finally that is looking pretty cool. Now if we want to include line graph into this we just have to add two line in the above code.

```
In [31]: ▶ pa.plot(data["rollno"],data["cgpa"],color="blue",label="line graph")
pa.xlabel("RollNo",color="green") #xlabel() defines the label of x-axis
pa.ylabel("CGPA",color="blue") #ylabel() defines the label of x-axis
pa.title("CGPA vs Roll No",color="green") #title() is used to give title of this scatter plot
pa.show()
```

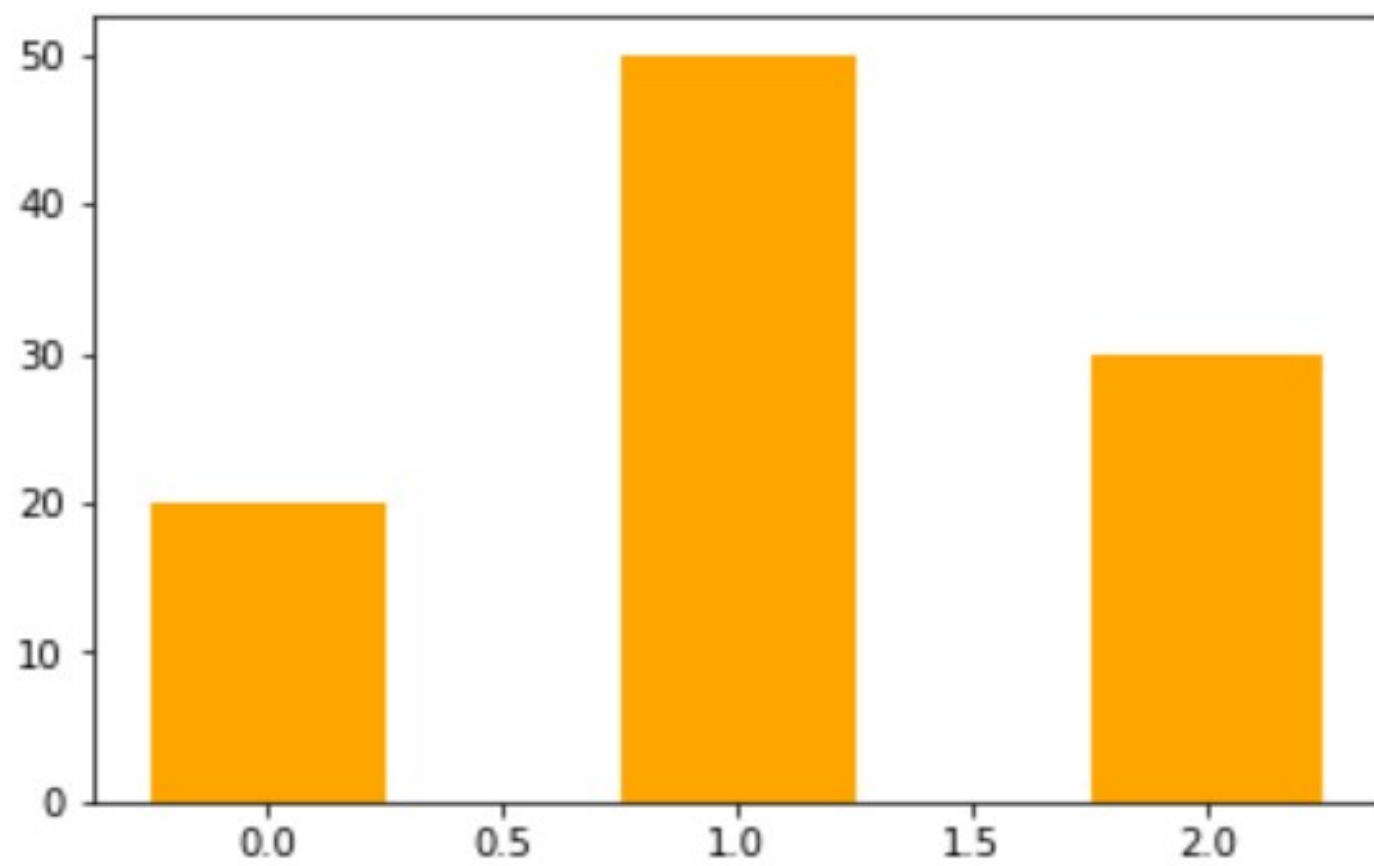


BAR CHART

Another very popular chart on matplotlib is the bar chart. For the bar chart we will use the `bar()` function, where we define the position of the bars on the X axis and their height on the Y axis. Additionally, we can also configure another characteristics for the chart, like width of the bars, color, among others. The X axis will be a range with the same quantity of items as the Y axis. Let's see a simple example, where we will store the configurations we want in variables and then we will pass them to the `bar()` function:

Example

```
import matplotlib.pyplot as plt;  
# Variables for the bar chart  
y_axis = [20,50,30]  
x_axis = range(len(y_axis))  
plt.bar(x_axis, y_axis, width=.5, color='orange')  
plt.show()
```



Bar chart

Market Share for Each Genre 1995-2020

Rank	Genre	Movies	Total Box Office	Tickets	Share
1	Adventure	1,074	\$64,144,171,448	9,173,681,683	27.38%
2	Action	1,036	\$47,641,931,735	6,753,933,433	20.34%
3	Drama	5,353	\$37,381,889,583	5,723,146,752	15.96%
4	Comedy	2,334	\$33,975,999,676	5,234,233,133	14.50%
5	Thriller/Suspense	1,102	\$19,590,920,222	2,856,076,757	8.36%
6	Horror	610	\$11,874,864,870	1,711,914,178	5.07%
7	Romantic Comedy	601	\$9,922,205,116	1,599,534,643	4.24%
8	Musical	186	\$4,105,929,532	542,174,705	1.75%
9	Documentary	2,340	\$2,239,919,045	336,341,986	0.96%
10	Black Comedy	189	\$1,677,461,702	233,209,110	0.72%
11	Western	81	\$1,084,785,441	149,931,395	0.46%
12	Concert/Performance	76	\$396,337,577	54,011,544	0.17%
13	Multiple Genres	44	\$37,024,971	4,437,318	0.02%

data of movies
released from
1995–2017.
The source of
data is .

```
label = ['Adventure', 'Action', 'Drama',  
'Comedy', 'Thriller/Suspense', 'Horror',  
'Romantic Comedy', 'Musical',  
'Documentary', 'Black Comedy', 'Western',  
'Concert/Performance', 'Multiple Genres',  
'Reality']
```

```
no_movies = [ 1074, 854, 4595,  
2125, 942, 509, 548, 149, 1952,  
161, 64, 61, 35, 5]
```



A large orange shape on the left side of the slide, consisting of a rectangle with a quarter-circle cutout from its top-right corner.

numpy

arange is a numpy method that generates an array of sequential numbers. For instance `numpy.arange(5)` will generate a `numpy.ndarray` like `[0,1,2,3,4,5]`

A yellow dashed line in the bottom right corner, consisting of four short, curved segments arranged in a diagonal pattern.

A large orange shape on the left side of the slide, consisting of a rectangle with a quarter-circle cutout on its right side.

Why is it needed?

We need some data for X-axis and right now we only have labels that can't be used for plotting purpose. So we will generate an array of length of label and use it on X-axis. Code for the graph is given below:

A yellow dashed line in the bottom right corner, consisting of four short, curved segments arranged in an upward-curving path.

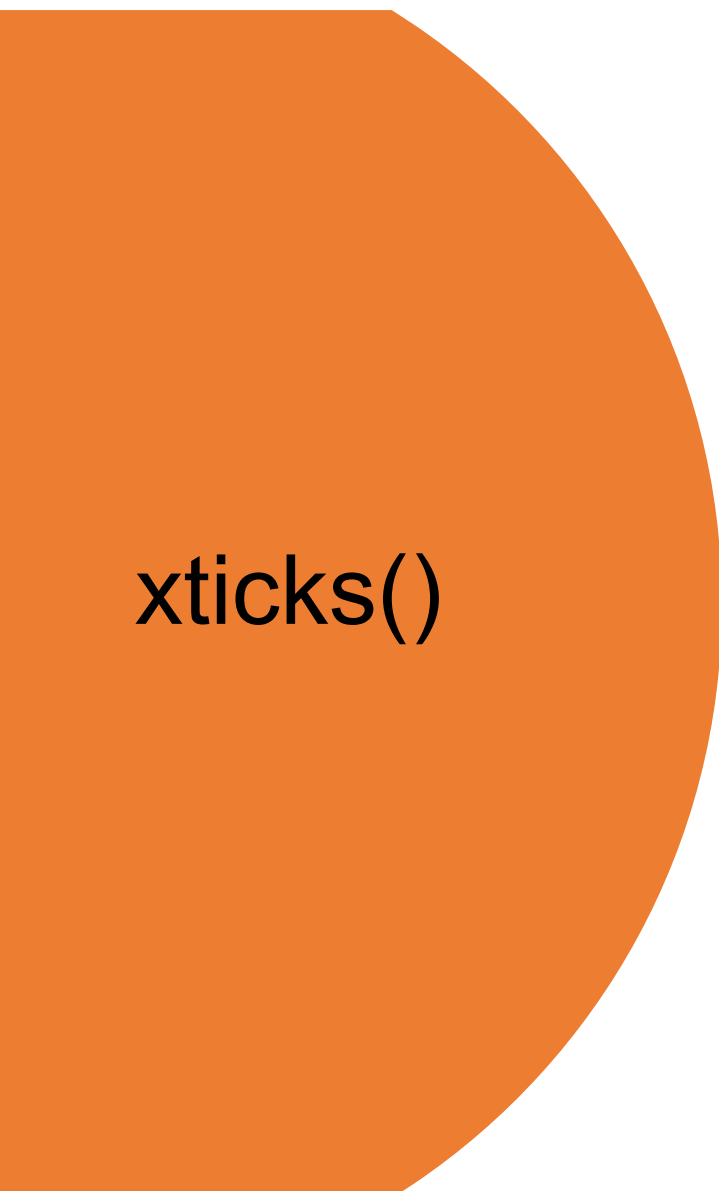
CODE

```
import matplotlib.pyplot as plt
import numpy as np

label = ['Adventure', 'Action', 'Drama', 'Comedy', 'Thriller/Suspense', 'Horror',
        'Romantic Comedy', 'Musical',
        'Documentary', 'Black Comedy', 'Western', 'Concert/Performance', 'Multiple
        Genres', 'Reality']

no_movies = [ 941, 854, 4595, 2125, 942, 509, 548, 149, 1952, 161,
64, 61, 35, 5]

def plot_bar_x():
    # this is for plotting purpose
    index = np.arange(len(label))
    plt.bar(index, no_movies)
    plt.xlabel('Genre', fontsize=5)
    plt.ylabel('No of Movies', fontsize=5)
    plt.xticks(index, label, fontsize=5, rotation=30)
    plt.title('Market Share for Each Genre 1995-2017')
    plt.show()
plot_bar_x()
```

A large orange shape on the left side of the slide, consisting of a rectangle with a quarter-circle cutout at the top right corner.

`xticks()`

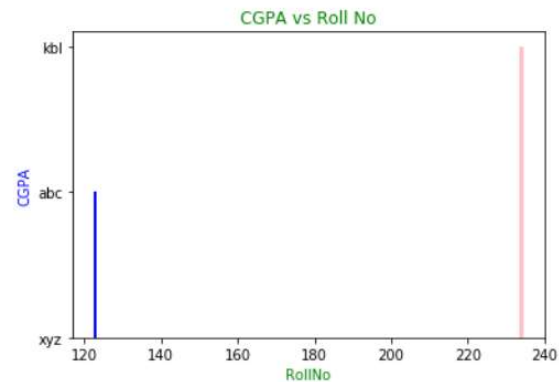
you see for `xticks` I used both index and label. Labels will be placed on each tick that is generated due to index sequence. I also give a bit of room for x-Axis labels by rotating them. When run something like below will be generated.

A yellow dashed line in the bottom right corner, consisting of four short, curved segments arranged in a roughly diagonal path.

#3 Bar Graph

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 data = pd.read_csv("cgpa.csv")
5 data = data.head(30)
6 #bar() method is used to plot a bar graph
7 #Here i am taking a list of colors to showing graph attractive
```

```
In [33]: ▶ pa.bar(data["rollno"], data["cgpa"], color=["green", "blue", "pink", "red"])
          pa.xlabel("RollNo", color="green")
          pa.ylabel("CGPA", color="blue")
          pa.title("CGPA vs Roll No", color="green")
          pa.show()
```



Problem 1

Draw a bar and line graph based on the following data.

Subjects	Science	English	History	Maths	Music
Number of students passed	60	40	80	40	90

Problem 2

Draw a bar and line graph based on the following data.

TV	Onida	Sony	Omega	Jolly	Prince
No. of sets manufactured on a certain day	250	300	150	200	100

PIE CHART

```
import matplotlib.pyplot as plt
```

```
# Data to plot
```

```
labels = 'Python', 'C++', 'Ruby', 'Java'
```

```
sizes = [215, 130, 245, 210]
```

```
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
```

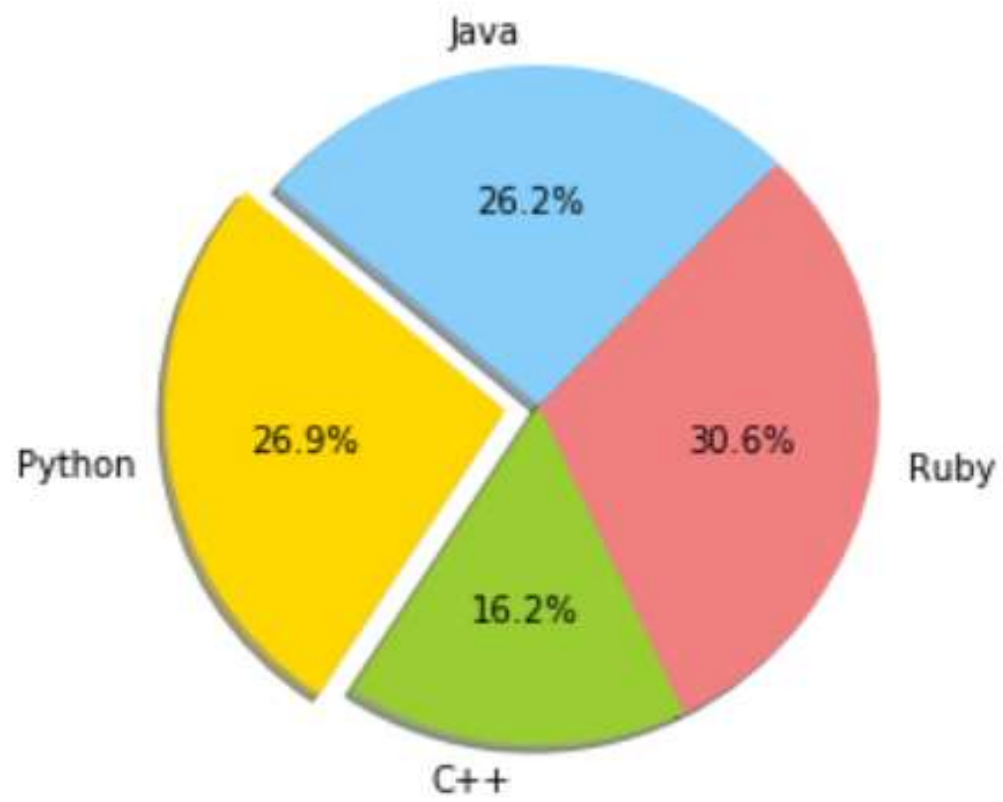
```
explode = (0.1, 0, 0, 0) # explode 1st slice
```

```
# Plot
```

```
plt.pie(sizes, explode=explode, labels=labels, colors=colors,  
autopct='%1.1f%%', shadow=True, startangle=140)
```

```
plt.axis('equal')
```

```
plt.show()
```



Within the `plt.pie`, we specify the "slices," which are the relevant sizes for each part. Then, we specify the color list for the corresponding slices. Next, we can optionally specify the "Start angle" for the graph. This lets you start the line where you want. In our case, we chose a 90 degree angle for the pie chart, which means the first division will be a vertical line. Next, we can optionally add a shadow to the plot for a bit of character, and then we can even use "explode" to pull out a slice a bit.

pie() parameters

- **explode** : array-like, optional, default: None
If not None, is a len(x) array which specifies the fraction of the radius with which to offset each wedge.
- **labels** : list, optional, default: None
A sequence of strings providing the labels for each wedge
- **colors** : array-like, optional, default: None
A sequence of matplotlib color args through which the pie chart will cycle. If None, will use the colors in the currently active cycle.
- **autopct** : None (default), string, or function, optional
If not None, 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. If it is a function, it will be called.

pie() parameters contd.

To add percentages to each of the constituents of the pie chart, we add in the line, `autopct='%1.1f%%'`, to the `plt.pie()` function.

This formats the percentage to the tenth place.

If you want to format the percentage to the hundredths place, you would use the statement, `autopct='%1.2f%%'`

If you want to format the percentage to the thousandths place, you would use the statement, `autopct='%1.3f%%'`

shadow : bool, optional, default: False

Draw a shadow beneath the pie.

startangle : float, optional, default: None

If not None, rotates the start of the pie chart by angle degrees counterclockwise from the x-axis.

radius : float, optional, default: None

The radius of the pie, if radius is None it will be set to 1.

Let's Practice

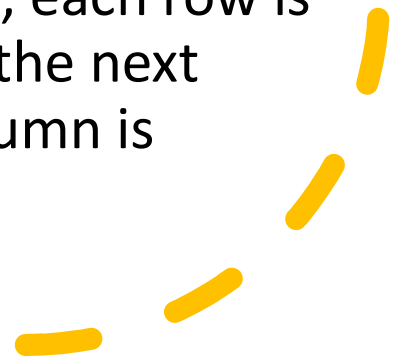
- Write a program to create a CSV file and save the records of 10 students (Name , Roll No. , Total Marks out of 500) and then plot all three graphs (Line, Bar and Pie) to show who has scored highest marks.



WORKING WITH CSV FILE

What is a CSV file?

A CSV file is a type of plain text file that uses specific structuring to arrange tabular data. CSV is a common format for data interchange as it's compact, simple and general. Many online services allow its users to export tabular data from the website into a CSV file. Files of CSV will open into Excel, and nearly all databases have a tool to allow import from CSV file. The standard format is defined by rows and columns data. Moreover, each row is terminated by a newline to begin the next row. Also within the row, each column is separated by a comma.



SAMPLE CSV file



The image shows a Notepad window titled 'contacts.csv - Notepad'. The window contains a CSV file with the following content:

```
File Edit Format View Help
First Name,Middle Name,Last Name,Title,Suffix,Initials,Web Page,Gender,Birthda
by,Billing Information,Directory Server,Sensitivity,Priority,Private,Categorie
Bob,,Smith,,,,,,,,,,,,,bob@example.com,,,,,,,,,123-456-7890,,,,,,,,,,,,,
Mike,,Jones,,,,,,,,,,,,,mike@example.com,,,,,,,,,098-7654-321,,,321 Fake Avenue,32
```

- CSV stands for "comma-separated values". Its data fields are most often separated, or [delimited](#), by a [comma](#). For example, let's say you had a spreadsheet containing the following data.

Name	Class	Dorm	Room	GPA
Sally Whittaker	2018	McCarren House	312	3.75
Belinda Jameson	2017	Cushing House	148	3.52
Jeff Smith	2018	Prescott House	17-D	3.20
Sandy Allen	2019	Oliver House	108	3.48

The above data could be represented in a CSV-formatted file as follows:

```
Sally Whittaker,2018,McCarren House,312,3.75  
Belinda Jameson,2017,Cushing House,148,3.52  
Jeff Smith,2018,Prescott House,17-D,3.20  
Sandy Allen,2019,Oliver House,108,3.48
```

Here, the fields of data in each row are delimited with a comma and individual rows are separated by a [newline](#).

Notepad (or any text editor)

- To create a CSV file with a text editor, first choose your favorite text editor, such as [Notepad](#) or [vim](#), and open a new file. Then enter the text data you want the file to contain, separating each value with a comma and each [row](#) with a new line.

```
Title1,Title2,Title3  
one,two,three  
example1,example2,example3
```

Save this file with the [extension](#) **.csv**. You can then open the file using Microsoft Excel or another spreadsheet program. It would create a table of data similar to the following:

CSV FILE contd.

Title1	Title2	Title3
one	two	three
example1	example2	example3

- In the **CSV** file you created, individual fields of data were separated by commas. But what if the data itself has commas in it?
- If the fields of data in your **CSV** file contain commas, you can protect them by enclosing those data fields in double [quotes](#) ("). The commas that are part of your data are kept separate from the commas which delimit the fields themselves.

CSV FILE contd.

- For example, let's say that one of our text fields is a user-created description that allows commas in the description. If our data looked like this:

Lead	Title	Phone	Notes
Jim Grayson	Senior Manager	(555)761-2385	Spoke Tuesday, he's interested
Prescilla Winston	Development Director	(555)218-3981	said to call again next week
Melissa Potter	Head of Accounts	(555)791-3471	Not interested, gave referral

NOTEPAD

- To retain the commas in our "Notes" column, we can enclose those fields in quotation marks. For instance:

```
Lead,Title,Phone,Notes
```

```
Jim Grayson,Senior Manager,(555)761-2385,"Spoke Tuesday, he's interested"
```

```
Prescilla Winston,Development Director,(555)218-3981,said to call again next week
```

```
Melissa Potter,Head of Accounts,(555)791-3471,"Not interested, gave referral"
```

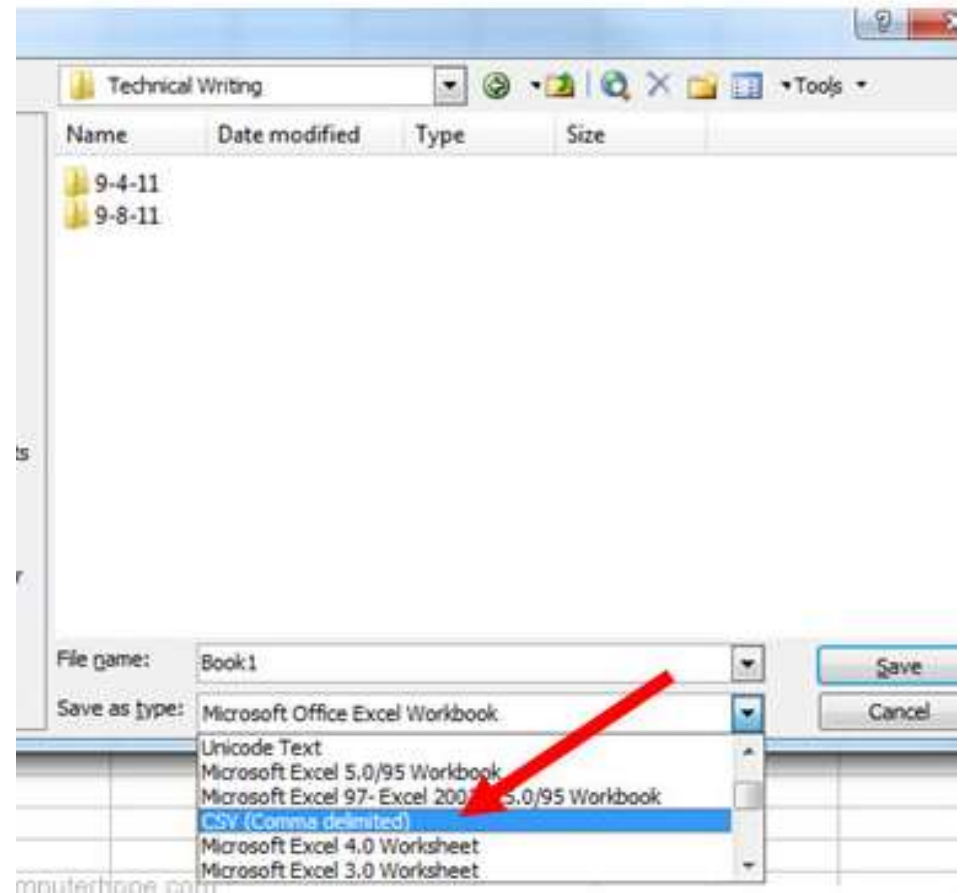
Microsoft Excel

- To create a **CSV** file using Microsoft Excel, launch Excel and then open the file you want to save in **CSV** format. For example, below is the data contained in our example Excel worksheet:

Item	Cost	Sold	Profit
Keyboard	\$10.00	\$16.00	\$6.00
Monitor	\$80.00	\$120.00	\$40.00
Mouse	\$5.00	\$7.00	\$2.00
		Total	\$48.00

EXCEL

- Once open, click **File** and choose **Save As**. Under **Save as type**, select **CSV (Comma delimited)** or **CSV (Comma delimited) (*.csv)**, depending on your version of Microsoft Excel.

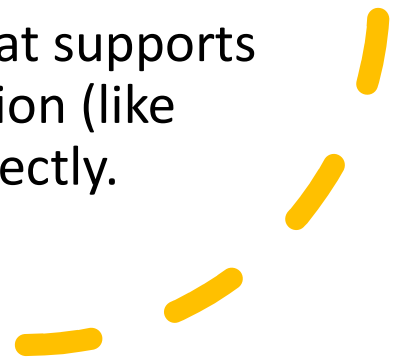


- After you save the file, you are free to open it up in a text editor to view it or edit it manually. Its contents will resemble the following:

```
Item, Cost, Sold, Profit
Keyboard, $10.00, $16.00, $6.00
Monitor, $80.00, $120.00, $40.00
Mouse, $5.00, $7.00, $2.00
, , Total, $48.00
```

Where Do CSV Files Come From?

- CSV files are normally created by programs that handle large amounts of data. They are a convenient way to export data from spreadsheets and databases as well as import or use it in other programs. For example, you might export the results of a data mining program to a CSV file and then import that into a spreadsheet to analyze the data, generate graphs for a presentation, or prepare a report for publication.
- CSV files are very easy to work with programmatically. Any language that supports text file input and string manipulation (like Python) can work with CSV files directly.



Parsing CSV Files With Python's Built-in CSV Library

The [csv library](#) provides functionality to both read from and write to CSV files. Designed to work out of the box with Excel-generated CSV files, it is easily adapted to work with a variety of CSV formats. The csv library contains objects and other code to read, write, and process data from and to CSV files.



Reader and Writer Modules

- The CSV module has several functions and classes available for reading and writing CSVs, and they include:
- `csv.reader` function
- `csv.writer` function
- `csv.DictReader` class
- `csv.Dictwriter` class



How to Read a CSV File

```
import csv
with open('innovators.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```



A large orange shape on the left side of the slide, consisting of a rectangle with a quarter-circle cutout at the top right corner.

`csv.reader()`

we are using the `csv.reader()` function in default mode for CSV files having comma delimiter.



Using csv.Reader():

```
import csv
f= open('example.csv')
readCSV = csv.reader(f,delimiter=',')
for row in readCSV:
print(row) print(row[0])
print(row[0],row[1],row[2],)
```



Using csv.Reader() contd.

Suppose our CSV file was using **tab** as a delimiter. To read such files, we can pass optional parameters to the `csv.reader()` function. Let's take an example.

```
import csv
file= open('people.csv', 'r',)
reader = csv.reader(file, delimiter = '\t')
for row in reader:
    print(row)
```

Using DictReader()

```
import csv
f=open("emp.csv")
reader = csv.DictReader(f, delimiter=',')
for line in reader:
    print(line["EMPNO"])
    print(line["ENAME"])
```

In the example above, we open a file and pass the file object to our function as we did before. The function passes the file object to our DictReader class. We tell the DictReader that the delimiter is a comma. This isn't actually required as the code will still work without that keyword argument. However, it's a good idea to be explicit so you know what's going on here. Next we loop over the reader object and discover that each line in the reader object is a dictionary. This makes printing out specific pieces of the line very easy.

Writing CSV files Using `csv.writer()`

To write to a CSV file in Python, we can use the `csv.writer()` function.

The `csv.writer()` function returns a writer object that converts the user's data into a delimited string. This string can later be used to write into CSV files using the `writerow()` function. Let's take an example.



The image features a white background with several yellow decorative elements. On the left side, there are two vertical bars of different heights. On the right side, there is a large, thick yellow L-shaped graphic that frames the central text area.

WRITING INTO CSV FILE

Writing into CSV

```
import csv
f= open('data.csv', 'a')
writer = csv.writer(f)
writer.writerow(["Anu", "35", "Doctor"])
writer.writerow(["Vijay", "28", "Analyst"])
f.close()
```

NOTE: MODE OF FILE CAN BE "w" OR "a"
DEPENDING UPON THE REQUIREMENT

Writing Multiple rows

```
import csv
csv_rowlist = [{"SN": "Movie",
"Protagonist"}, [1, "Lord of the Rings",
"Frodo Baggins"], [2, "Harry Potter", "Harry
Potter"]]
F= open('protagonist.csv', 'w')
writer = csv.writer(f)
writer.writerows(csv_rowlist)
F.close()
```



Writing to a CSV File with Tab Delimiter

```
import csv
F= open('protagonist.csv', 'w')
writer = csv.writer(F, delimiter = '\t')
writer.writerow(["SN", "Movie",
"Protagonist"])

writer.writerow([1, "Lord of the Rings",
"Frodo Baggins"])
writer.writerow([2, "Harry Potter", "Harry
Potter"])
F.close()
```



Using the Pandas library to Handle CSV files

- Pandas is a popular data science library in Python for data manipulation and analysis. If we are working with huge chunks of data, it's better to use pandas to handle CSV files for ease and efficiency.
- Before we can use pandas, we need to install it.
- Once we install it, we can import Pandas as:

```
import pandas as pd
```

