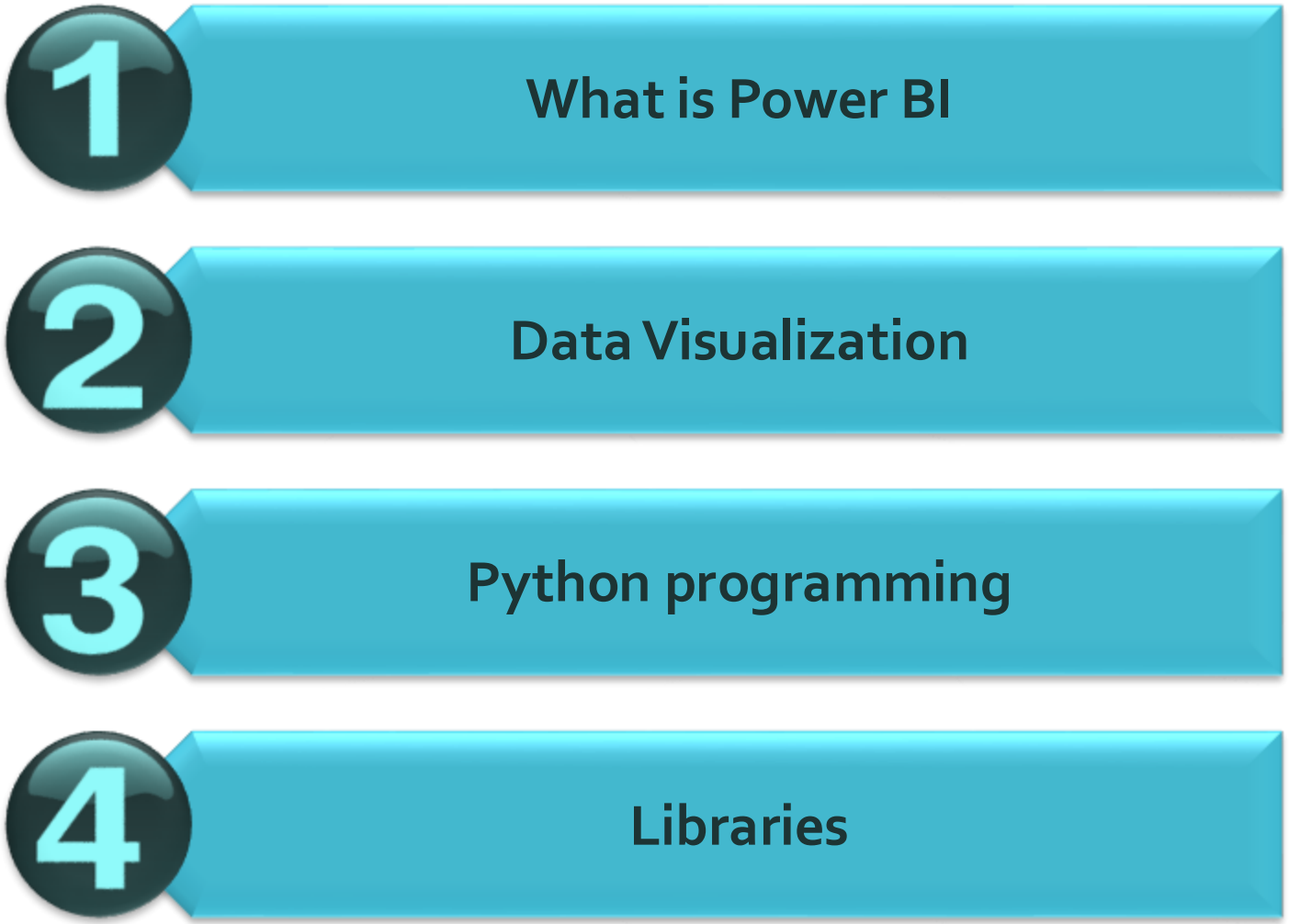
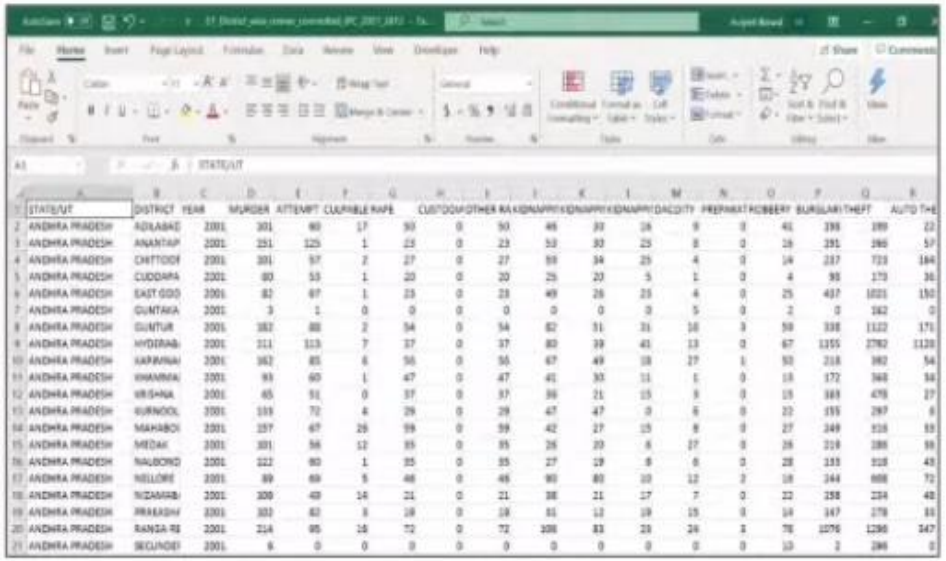
Data Analytics

Dr. Umber Noreen

Outline


1
What is Power BI
2
Data Visualization
3
Python programming
4
Libraries

MS Power BI is the Business Intelligence tool to visualize your data and share insights across your organization



Data

Power BI

Visualization

Power BI

Python Programming

Data Analytics

Python Programming

Why Python ?

1. Beginner friendly
2. Versatile and flexible
3. Most mature package libraries around
4. Career Opportunities
5. Startups and Corporates Python for both
6. Most popular in ML and Data Science

Python Programming

Why Python ?

MAJOR COMPANIES THAT USE python

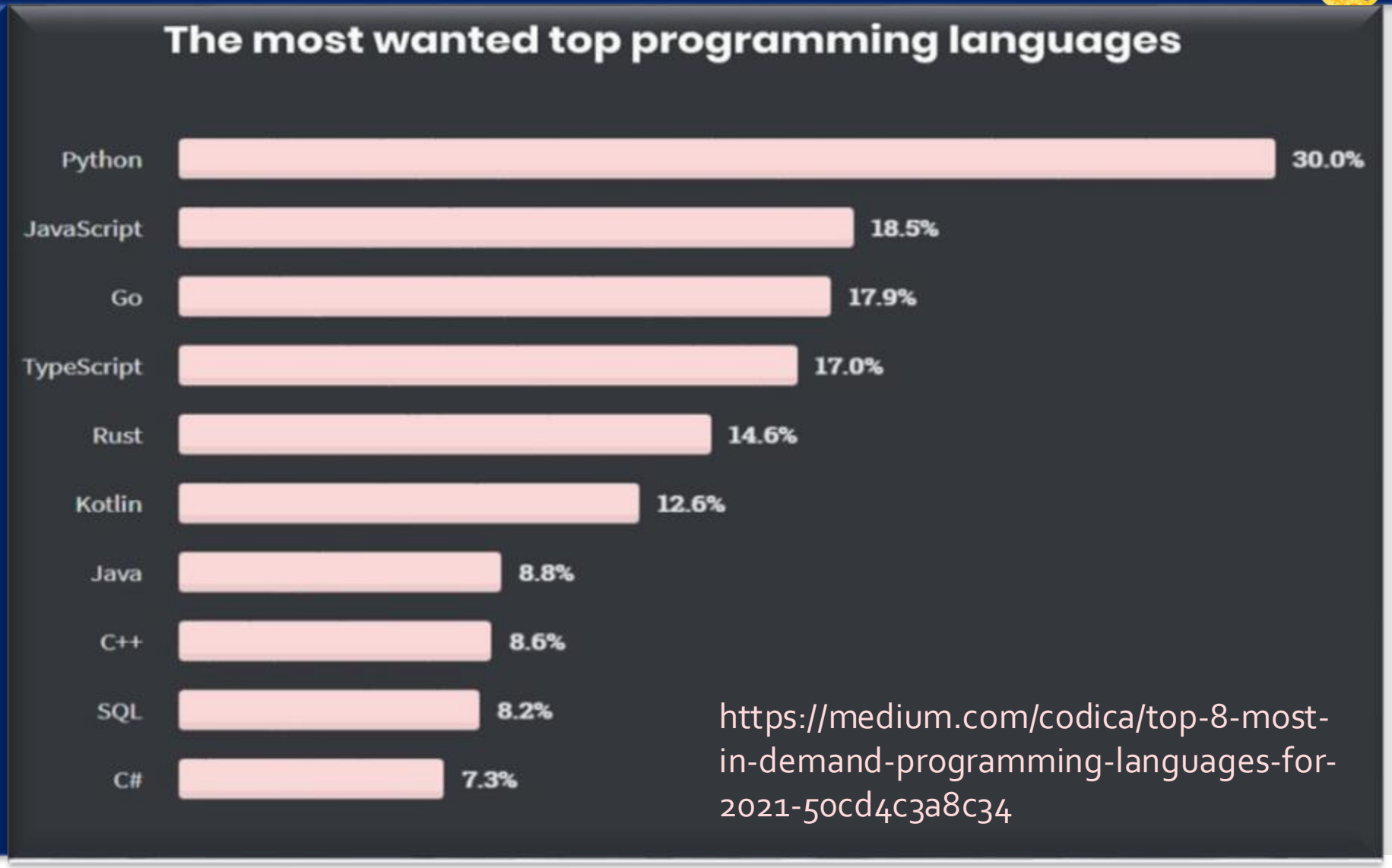
Google NETFLIX facebook Instagram

amazon Quora slack intel NASA

Dropbox ebay Spotify CapitalOne

Python Programming

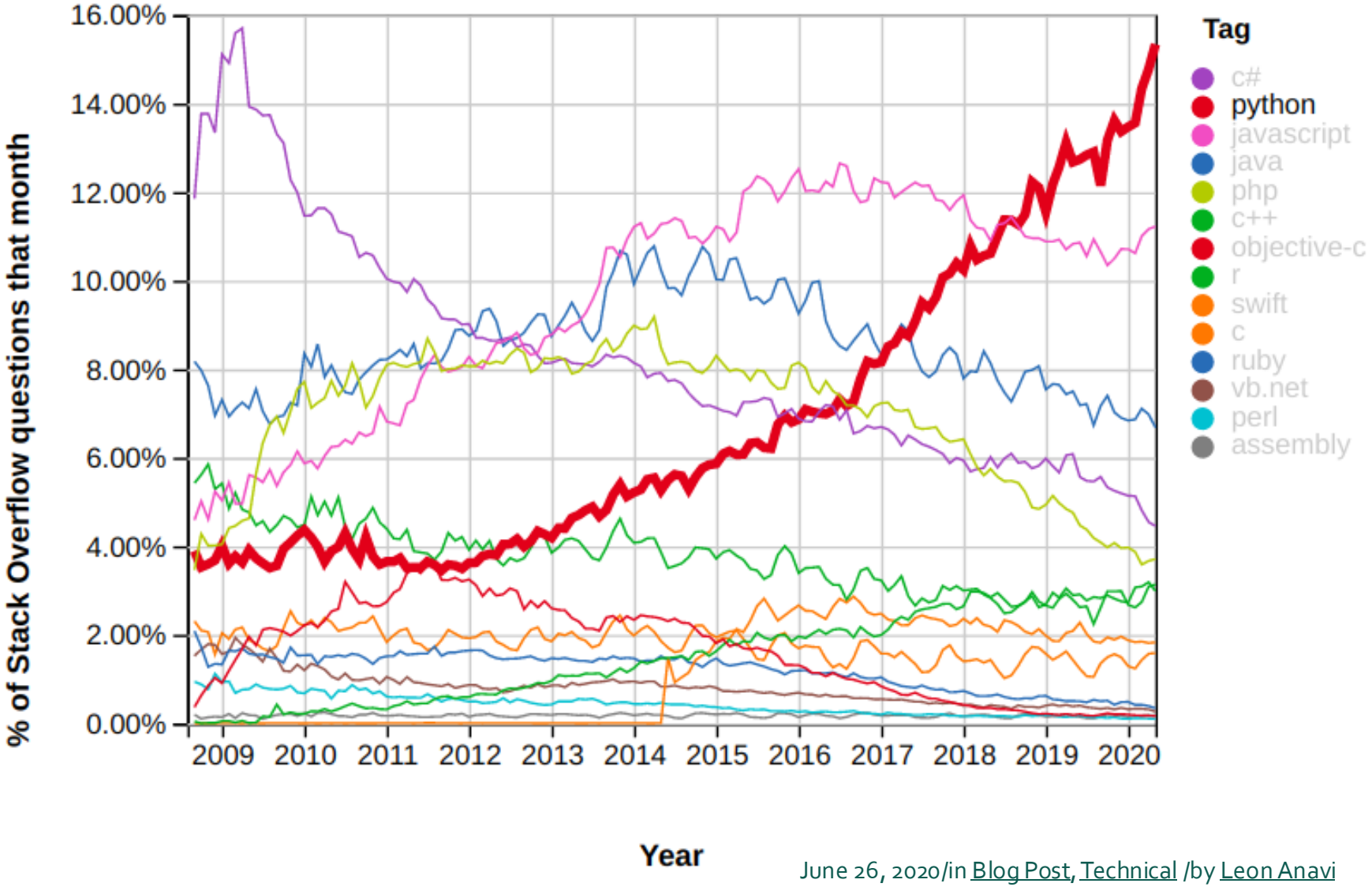
Why Python ?



https://medium.com/codica/top-8-most-in-demand-programming-languages-for-2021-50cd4c3a8c34

Python Programming

Why Python ?



June 26, 2020/in Blog Post, Technical /by Leon Anavi

Python Programming

Python IDE

IDLE

SPYDER

PyCharm

eric

Vim

Atom

jupyter

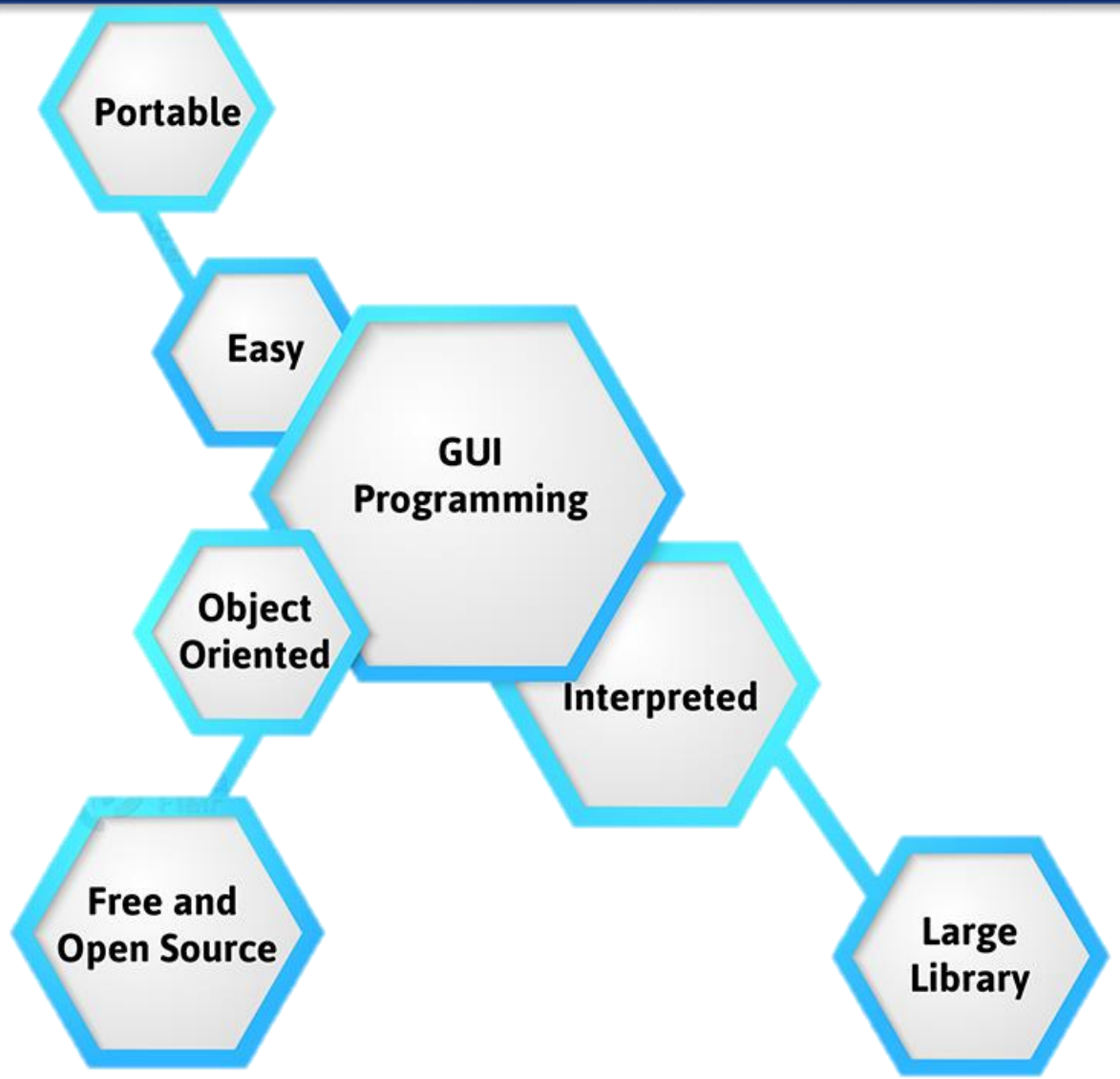
Anaconda

PyDev

Thonny
Python IDE for beginners

Python Programming

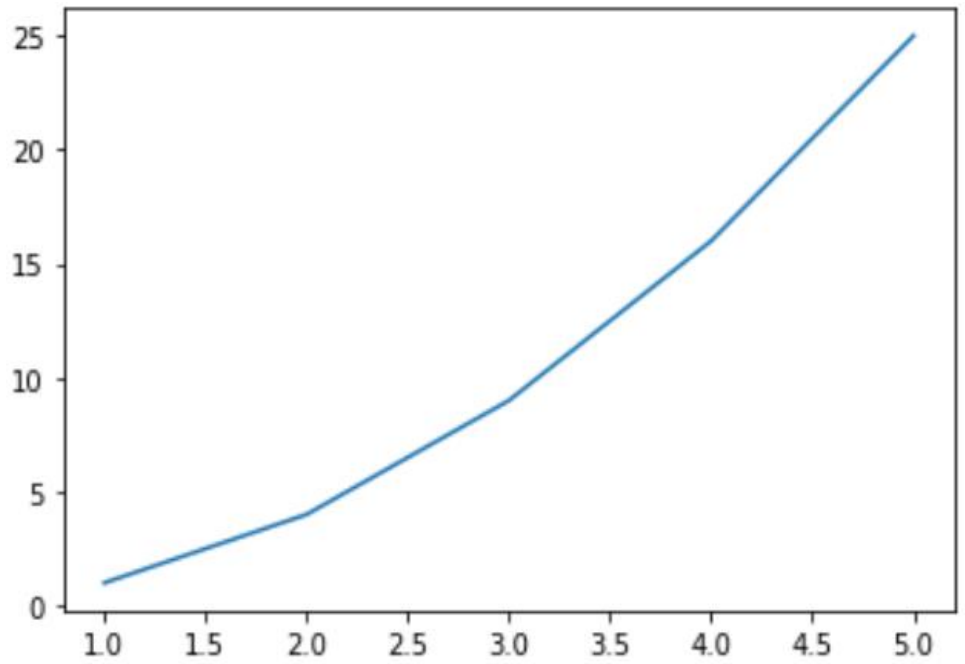
Python Features



Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python

Library Matplotlib

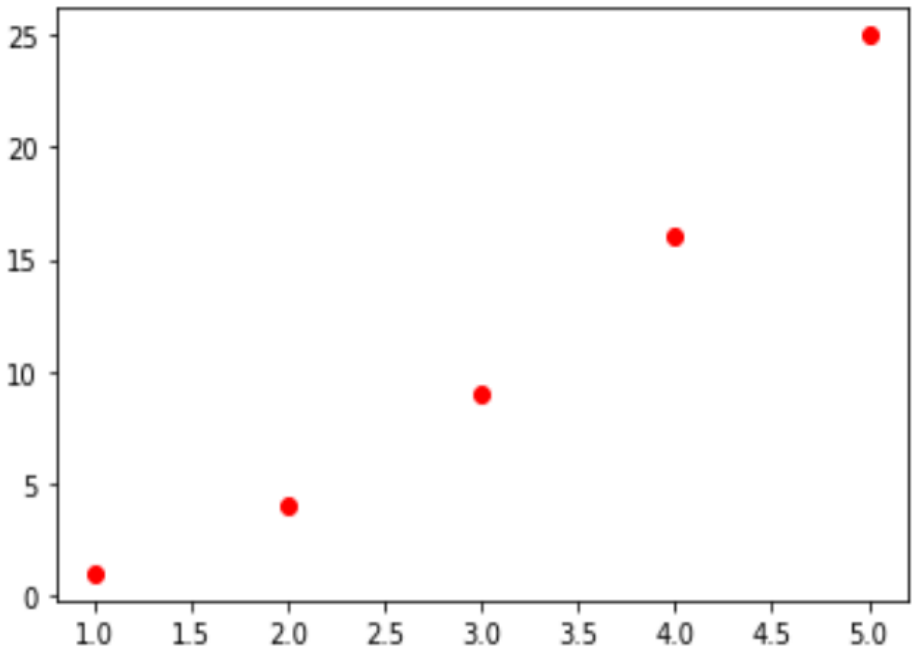
```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.plot(X,Y)

plt.show()
```



Line Graph

Library
Matplotlib

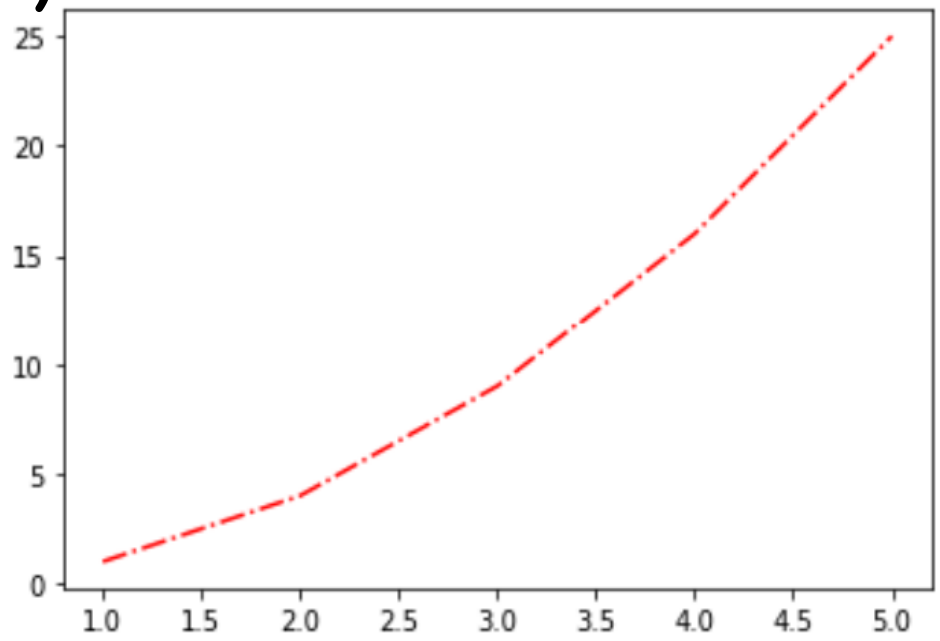
```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.plot(X,Y, 'ro')

plt.show()
```



Line Graph

Library Matplotlib

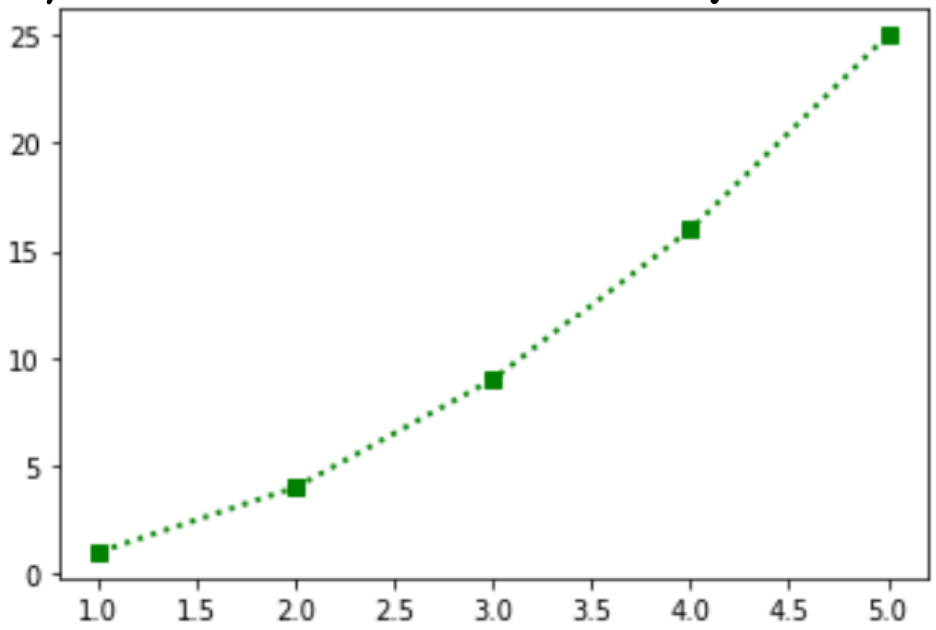
```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.plot(X,Y, 'r-.')

plt.show()
```



Line Graph

Library
Matplotlib

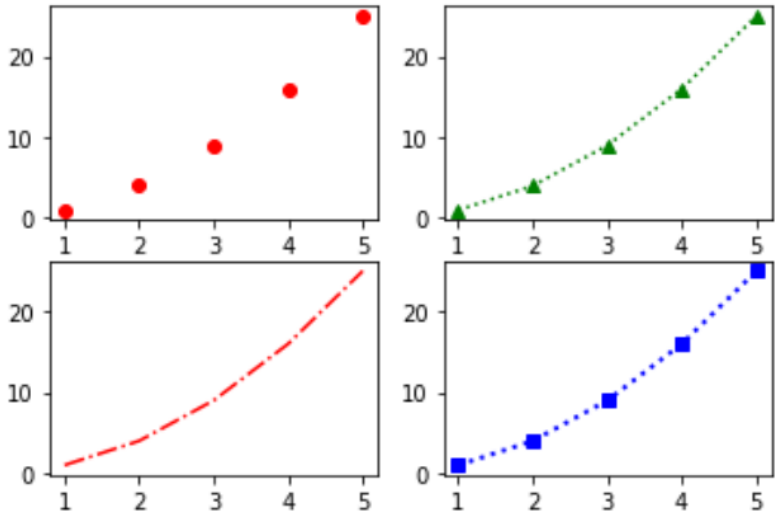
```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.plot(X,Y, 'r-.', linewidth=2.0)

plt.show()
```



Line Graph

Library
Matplotlib

```
import matplotlib.pyplot as plt
X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.subplot(221)
plt.plot(X,Y,'ro')
plt.subplot(222)
plt.plot(X,Y,'g:^')
plt.subplot(223)
plt.plot(X,Y,'r-.')
plt.subplot(224)
plt.plot(X,Y,'b:s',linewidth=2.0)
plt.show()
```



Line Graph

Library
Matplotlib

```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.title("My First plot")
plt.xlabel("Counting")
plt.ylabel("Square Values")
plt.plot(X,Y, 'r-.', linewidth=2.0)
plt.show()
```

Line Graph

Library Matplotlib

```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.title("My First plot", fontsize=20)
plt.xlabel("Counting", color="r")
plt.ylabel("Square Values", color="g")
plt.plot(X,Y,'b-o',linewidth=2.0)

plt.show()
```

Line Graph

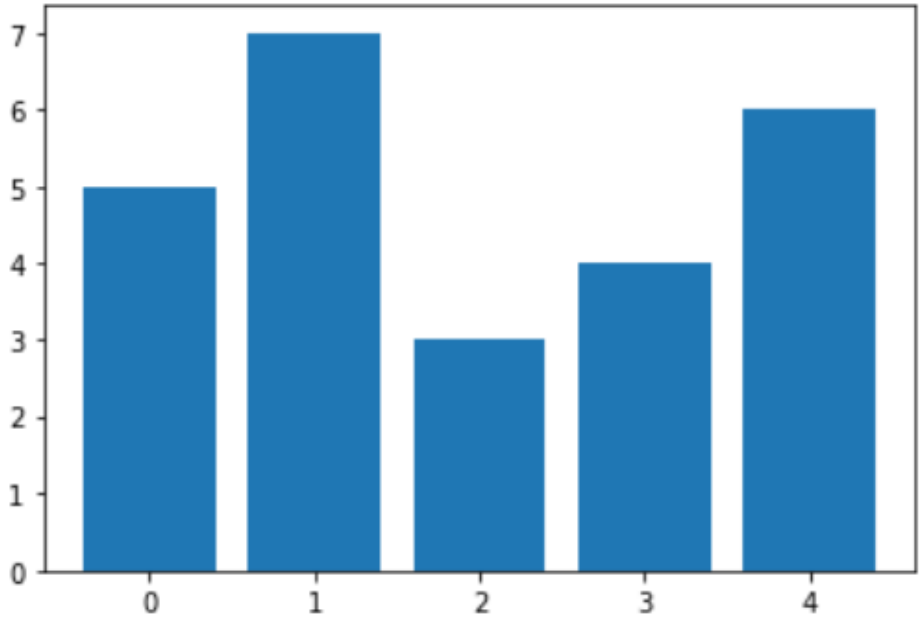
Library
Matplotlib

```
import matplotlib.pyplot as plt

X=[1,2,3,4,5]
Y=[1,4,9,16,25]
plt.title("My First plot", fontsize=20)
plt.xlabel("Counting", color="r")
plt.ylabel("Square Values", color="g")
plt.plot(X,Y,'b-o',linewidth=2.0)
plt.legend(['First series'])
plt.show()
```

Bar Charts

Library
Matplotlib

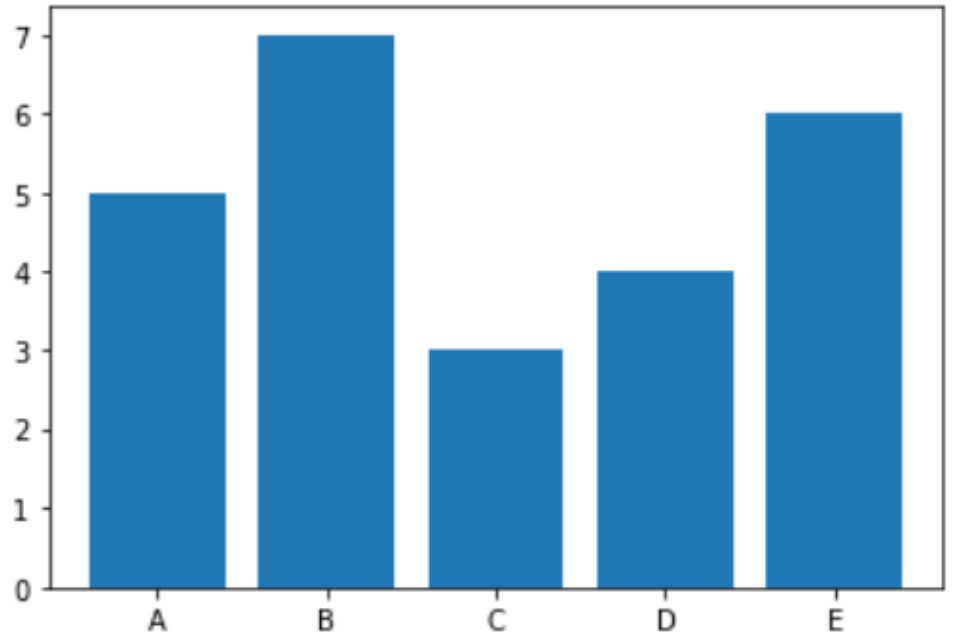
```
import matplotlib.pyplot as plt

index = [0,1,2,3,4]
values = [5,7,3,4,6]
plt.bar(index,values)
plt.show()
```



Bar Charts

Library
Matplotlib

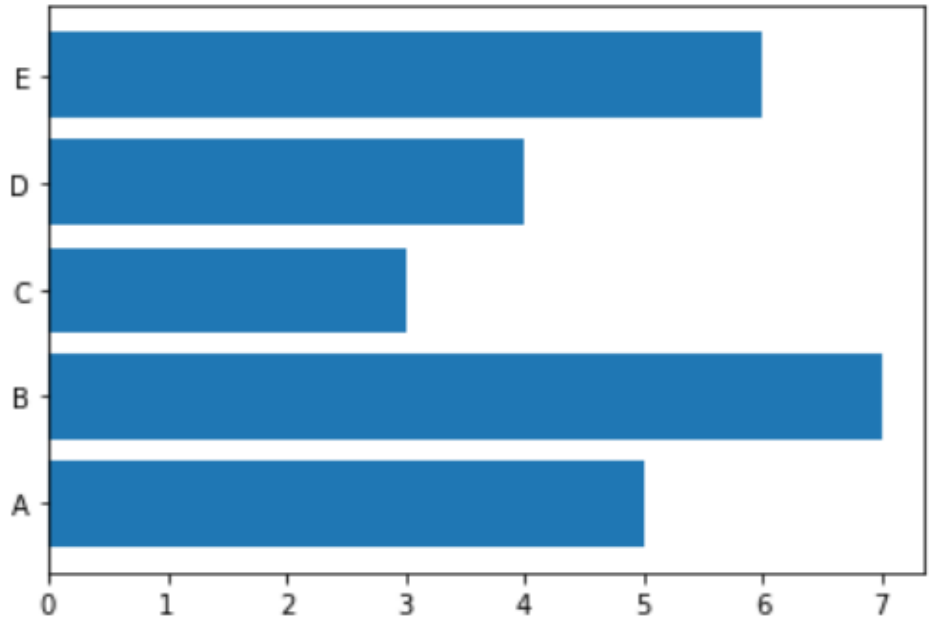
```
import matplotlib.pyplot as plt

index = ['A','B','C','D','E']
values = [5,7,3,4,6]
plt.bar(index,values)
plt.show()
```



Bar Charts

Library
Matplotlib

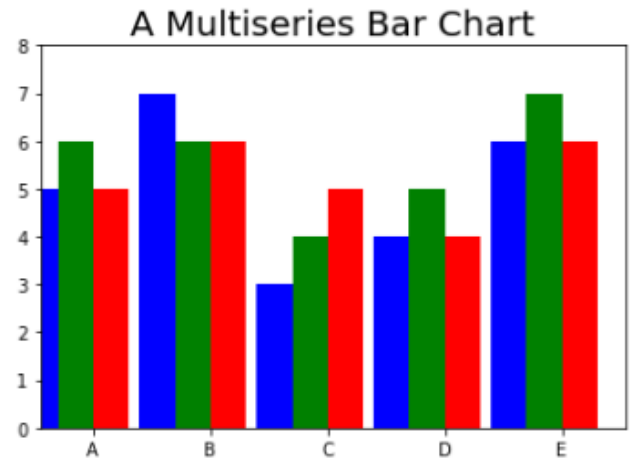
```
import matplotlib.pyplot as plt

index = ['A','B','C','D','E’]
values = [5,7,3,4,6]
plt.barh(index,values)
plt.show()
```



Multiple Bar Charts

Library
Matplotlib

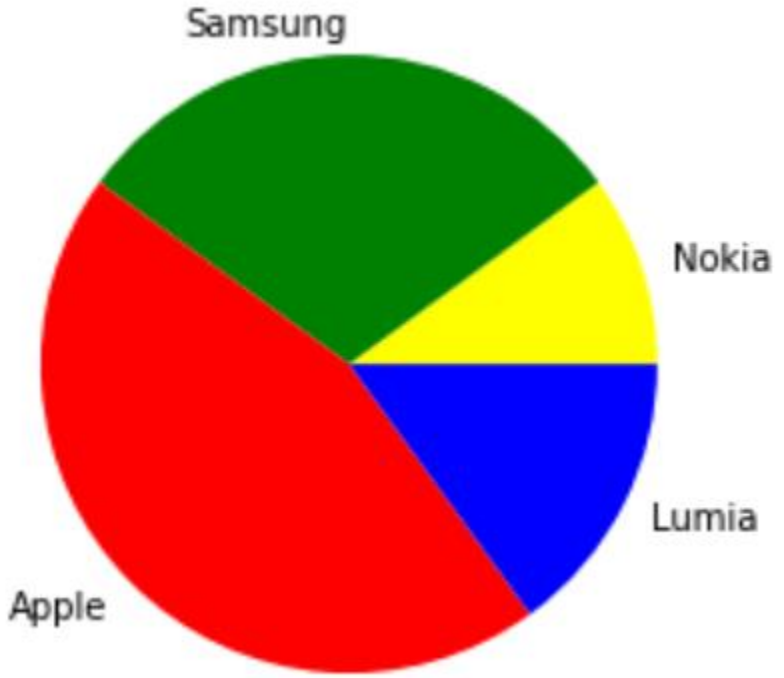
```
import matplotlib.pyplot as plt
import numpy as np

index = np.arange(5)
values1 = [5,7,3,4,6]
values2 = [6,6,4,5,7]
values3 = [5,6,5,4,6]
plt.axis([0,5,0,8])
plt.title('A Multiseries Bar
Chart',fontsize=20)
bw=0.3
plt.bar(index,values1,bw,color='b')
plt.bar(index+bw,values2,bw,color='g')
plt.bar(index+2*bw,values3,bw,color='r')
plt.xticks(index+1.5*bw,['A','B','C','D','E'])
plt.show()
```



Pie Charts

Library Matplotlib

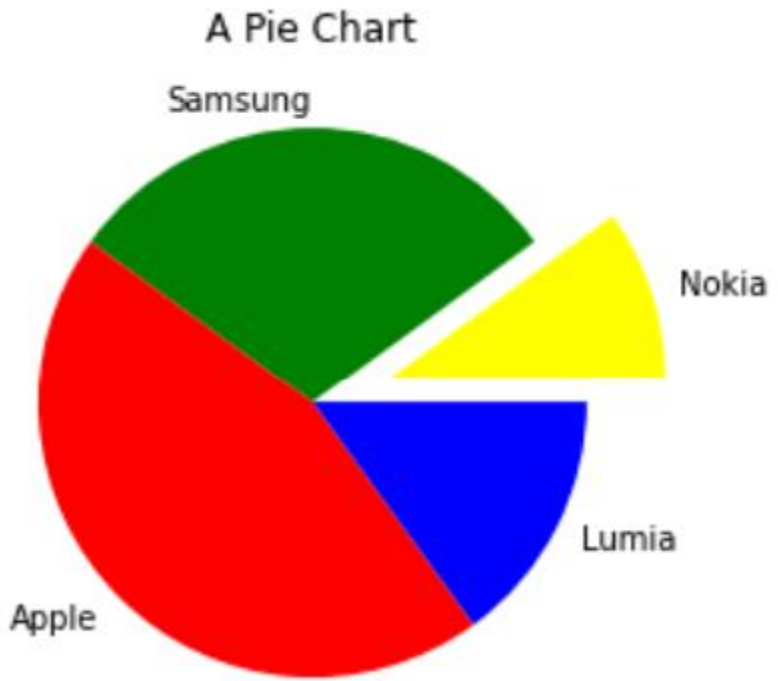
```
import matplotlib.pyplot as plt

labels = ['Nokia','Samsung','Apple','Lumia']
values = [10,30,45,15]
colors = ['yellow','green','red','blue']
plt.pie(values,labels=labels,colors=colors)

plt.show()
```



Pie Charts

Library Matplotlib

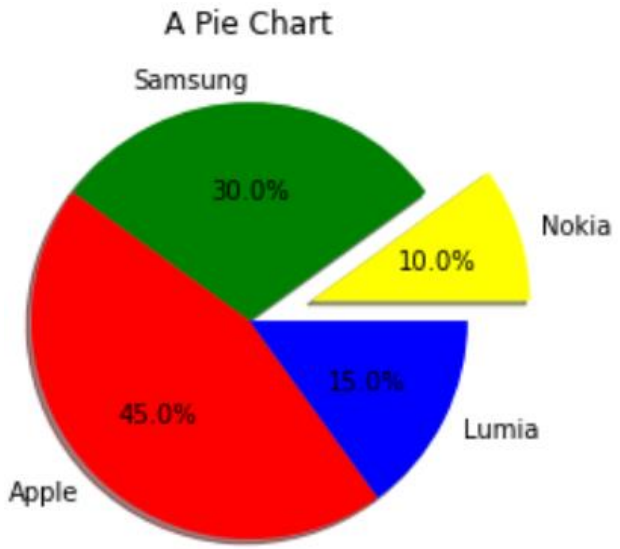
```
import matplotlib.pyplot as plt

labels = ['Nokia','Samsung','Apple','Lumia']
values = [10,30,45,15]
colors = ['yellow','green','red','blue']

explode = [0.3,0,0,0]
plt.title('A Pie Chart')
plt.pie(values,labels=labels,colors=colors,explode=explode)
plt.show()
```



Pie Charts

Library
Matplotlib

```
import matplotlib.pyplot as plt

labels = ['Nokia','Samsung','Apple','Lumia']
values = [10,30,45,15]
colors = ['yellow','green','red','blue']

explode = [0.3,0,0,0]
plt.title('A Pie Chart')

plt.pie(values,labels=labels,colors=colors,explode=explode,sh
adow=True,autopct='%1.1f%%')
plt.show()
```



Data Frame: A data frame is a multi-dimensional array where the rows and the columns can be labeled.

```
1  import pandas as pd
2  import numpy as np
3
4  myarray = np.array([[1, 2, 3], [4, 5, 6]])
5  rownames = ['a', 'b']
6  colnames = ['one', 'two', 'three']
7  mydataframe = pd.DataFrame(myarray, index=rownames, columns=colnames)
8
9  print(mydataframe)
```

```
   one  two  three
a    1    2      3
b    4    5      6
```

Load CSV File with Pandas: You can load your CSV data using Pandas and the pandas.read csv() function.

How To Load Machine Learning Data

```
import pandas as pd
# import pandas library to read dataset
df = pd.read_csv('salary_data.csv')
# if program file and Dataset both are in Same folder
df = pd.read_csv('D:\\03 LECTURES\\000 WCC\\0 FALL 2021\\Machine Learning\\Week-02-D-21-10-2021\\salary_data.csv')
# Windows format for file path
df = pd.read_csv('D:/03 LECTURES/000 WCC/0 FALL 2021/Machine Learning/Week-02-D-21-10-2021/salary_data.csv')
# Linux/MacOS format for file path
```

Read Dataset and display it

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df)
```

View the First and last 6 rows of your data using the head() and tail() function on the Pandas Data Frame

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
#print(df)
print(df.head(6))
print(df.tail(6))
print(df.sample(6))
```

Dimensions of Your Dataset: View the shape and size of your dataset by printing the shape property on the Pandas DataFrame

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df.shape)
```

Data Type For Each Attribute: list the data types used by the DataFrame to characterize each attribute using the dtypes property

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df. dtypes)
```

Descriptive Statistics

Understanding Your Data With Descriptive Statistics

The describe() function on the Pandas DataFrame lists 8 statistical properties of each attribute. They are:

- Count
- Mean
- Standard Deviation
- Minimum Value
- 25^{th} Percentile
- 50^{th} Percentile (Median)
- 75^{th} Percentile
- Maximum Value

Descriptive Statistics

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df. describe())
```

Descriptive Statistics

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df. describe())
```

Information about dataset

Understanding Your Data With Descriptive Statistics

- import pandas as pd
- df = pd.read_csv('salary_data.csv')
- print(df. info())

Columns/attributes of dataset

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df. columns)
```

Size(number of values) of dataset

Understanding Your Data With Descriptive Statistics

```
import pandas as pd
df = pd.read_csv('salary_data.csv')
print(df. size)
print(df. values)
```

Visualizing the dataset: The library matplotlib.pyplot is used for Visualizing the Dataset

```
import pandas as pd

df = pd.read_csv('salary_data.csv')

import matplotlib.pyplot as plt

plt.xlabel('Years Experience', fontsize=16, color='g')
plt.ylabel('Salary', fontsize=16, color='b')
plt.title("Salary of Employees \n with Experience", fontsize=20, color='r')
plt.scatter(df.YearsExperience,df.Salary,color='b',marker='s')
plt.axis([0,14, 20000, 140000])
plt.grid(True)

plt.show()
```