

# Python Libraries

## 1. Pandas-

Pandas is free to use and an open source library, making it one of the most widely used data science libraries in the world. It provides essential data structures like series, dataframes, and panels which help in manipulating data sets and time series.

### Installation of Pandas:

If you have Python and PIP already installed on a system, then installation of NumPy is very easy.

```
pip install pandas
```

### Functionalities:

- **Pandas Series**

```
import pandas as pd  
import numpy as np  
#Create a series with 100 random numbers  
s = pd.Series(np.random.randn(4))  
print s
```

```
0    0.967853  
1   -0.148368  
2   -1.395906  
3   -1.758394  
dtype: float64
```

- **Pandas Dataframe - Create a DataFrame from Lists**

```
import pandas as pd  
data = [1,2,3,4,5]  
df = pd.DataFrame(data)  
print df
```

```
   0  
0  1  
1  2  
2  3  
3  4  
4  5
```

# Python Libraries

- **Head & Tail**

```
import pandas as pd
import numpy as np
#Create a series with 4 random numbers
s = pd.Series(np.random.randn(4))
print ("The original series is:")
print s
print ("The first two rows of the data series:")
print s.head(2)
```

```
The original series is:
0 -0.655091
1 -0.881407
2 -0.608592
3 -2.341413
dtype: float64

The last two rows of the data series:
2 -0.608592
3 -2.341413
dtype: float64
```

# Python Libraries

## 2. NumPy-

NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

### Installation of NumPy:

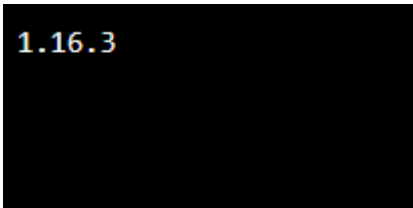
If you have Python and PIP already installed on a system, then installation of NumPy is very easy.

```
pip install numpy
```

### Functionalities:

- **Checking NumPy Version**

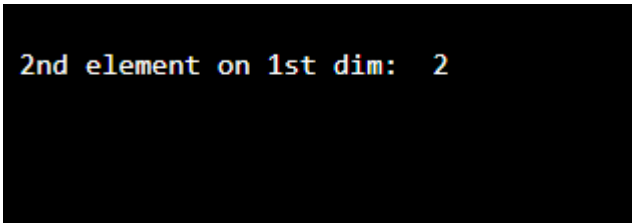
```
import numpy as np  
print(np.__version__)
```



```
1.16.3
```

- **Access 2-D Arrays**

```
import numpy as np  
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])  
print('2nd element on 1st dim: ', arr[0, 1])
```

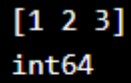


```
2nd element on 1st dim: 2
```

# Python Libraries

- **Converting Data Type on Existing Arrays**

```
import numpy as np  
arr = np.array([1.1, 2.1, 3.1])  
newarr = arr.astype(int)  
print(newarr)  
print(newarr.dtype)
```



```
[1 2 3]  
int64
```

# Python Libraries

## 3. SciPy-

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely. SciPy was created by NumPy's creator Travis Olliphant.

### Installation of SciPy:

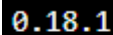
If you have Python and PIP already installed on a system, then installation of SciPy is very easy.

*pip install scipy*

### Functionalities:

- **Checking SciPy Version**

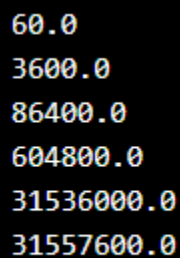
```
import scipy
print(scipy.__version__)
```



```
0.18.1
```

- **Time**

```
from scipy import constants
print(constants.minute)
print(constants.hour)
print(constants.day)
print(constants.week)
print(constants.year)
print(constants.Julian_year)
```



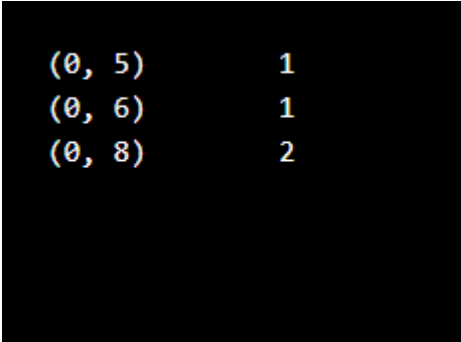
```
60.0
3600.0
86400.0
604800.0
31536000.0
31557600.0
```

# Python Libraries

- **CSR Matrix**

Compressed Sparse Row. For fast row slicing, faster matrix vector products  
import numpy as np

```
from scipy.sparse import csr_matrix  
arr = np.array([0, 0, 0, 0, 0, 1, 1, 0, 2])  
print(csr_matrix(arr))
```



```
(0, 5)      1  
(0, 6)      1  
(0, 8)      2
```

# Python Libraries

## 4. Matplotlib-

Matplotlib is a Python library that uses Python Script to write 2-dimensional graphs and plots. This library helps us to build multiple plots at a time. You can, however, use Matplotlib to manipulate different characteristics of figures as well.

You can use Matplotlib with different toolkits such as Python Scripts, IPython Shells, Jupyter Notebook, and many other graphical user interfaces.

### Installation of Matplotlib:

If you have Python and PIP already installed on a system, install it using this command:

```
pip install matplotlib
```

### Functionalities:

- `add_subplot()`
- `show()`
- `bar()`
- `barh()`

```
# Import `pyplot` from `matplotlib`  
import matplotlib.pyplot as plt
```

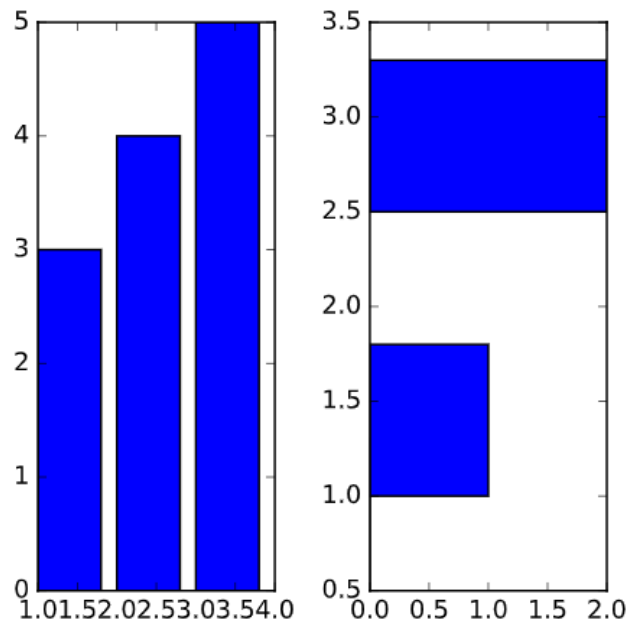
```
# Initialize the plot  
fig = plt.figure(figsize=(20,10))  
ax1 = fig.add_subplot(121)  
ax2 = fig.add_subplot(122)
```

```
# or replace the three lines of code above by the following line:  
#fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,10))
```

```
# Plot the data  
ax1.bar([1,2,3],[3,4,5])  
ax2.barh([0.5,1,2.5],[0,1,2])
```

```
# Show the plot  
plt.show()
```

# Python Libraries





# Python Libraries

## 5. Seaborn-

Seaborn is a library that uses Matplotlib underneath to plot graphs. It can be used to visualize random distributions.

### Installation of Seaborn:

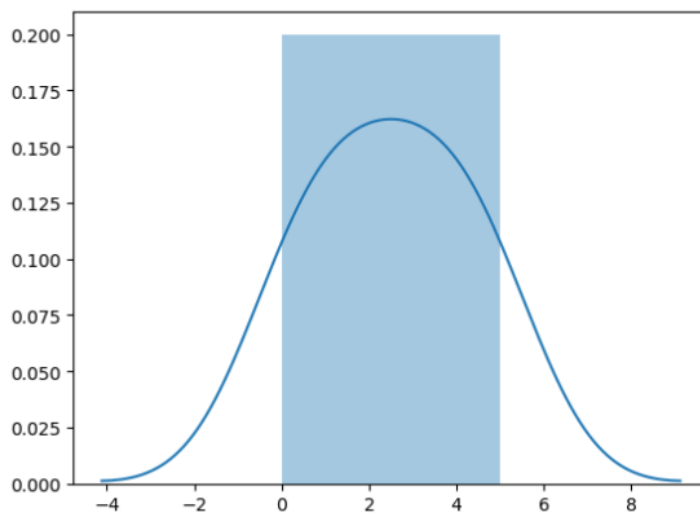
If you have Python and PIP already installed on a system, install it using this command:

```
pip install seaborn
```

### Functionalities:

- `distplot()`
- `show()`

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot([0, 1, 2, 3, 4, 5])
plt.show()
```



# Python Libraries

## 6. NLTK (Natural Language Toolkit)-

NLTK is a set of language processing libraries and other programs that cumulatively provide a numerical and symbolic language processing solution for English only. It is written in Python. With NLTK, natural language processing with python has become more standard and ideal.

### Installation of NLTK:

If you have Python and PIP already installed on a system, install it using this command:

```
pip install nltk
```

### Functionalities:

- **NLTK Word Tokenizer**

```
>>> nltk.word_tokenize("Last night, I went to Mrs. Martinez's housewarming.  
It was a disaster.")
```

```
['Last', 'night', ',', 'I', 'went', 'to', 'Mrs.', 'Martinez', "'s", 'housewarming', '.',  
'It', 'was', 'a', 'disaster', '.']
```

- **Stopwords**

```
from nltk.corpus import stopwords  
stop_words=set(stopwords.words("english"))  
print(stop_words)
```

```
{'their', 'then', 'not', 'ma', 'here', 'other', 'won', 'up', 'weren', 'being', 'we', 'those', 'an',
```

- **Stemming**

```
from nltk.stem import PorterStemmer  
from nltk.tokenize import sent_tokenize, word_tokenize  
ps = PorterStemmer()  
stemmed_words=[]  
for w in filtered_sent:  
    stemmed_words.append(ps.stem(w))  
  
print("Filtered Sentence:",filtered_sent)  
print("Stemmed Sentence:",stemmed_words)
```

# Python Libraries

```
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?']  
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?']
```