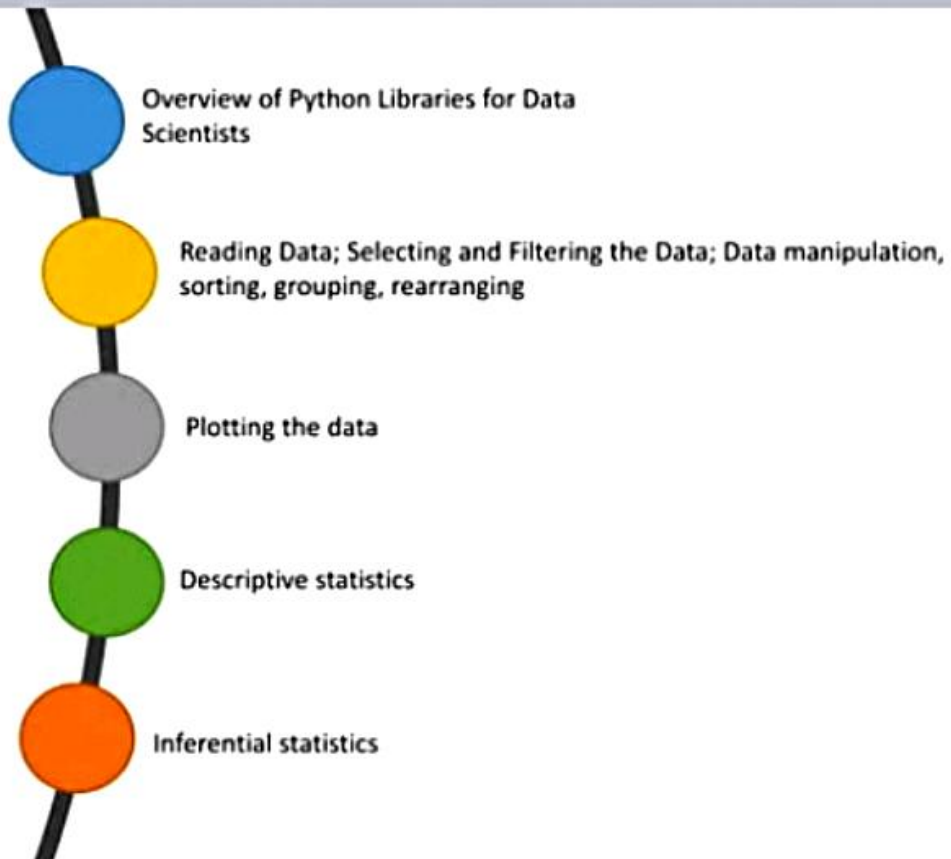


Tutorial Content



Python Libraries for Data Science

NumPy:

- introduces objects for multidimensional arrays and matrices, as well as functions that allow to easily perform advanced mathematical and statistical operations on those objects
- provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance
- many other python libraries are built on NumPy

Link: <http://www.numpy.org/>

Python Libraries for Data Science

SciPy:

- collection of algorithms for linear algebra, differential equations, numerical integration, optimization, statistics and more
- part of SciPy Stack
- built on NumPy

Link: <https://www.scipy.org/scipylib/>



Python Libraries for Data Science

Pandas:

- adds data structures and tools designed to work with table-like data (similar to Series and Data Frames in R)
- provides tools for data manipulation: reshaping, merging, sorting, slicing, aggregation etc.
- allows handling missing data

Link: <http://pandas.pydata.org/>



Python Libraries for Data Science

SciKit-Learn:

- provides machine learning algorithms: classification, regression, clustering, model validation etc.
- built on NumPy, SciPy and matplotlib

Link: <http://scikit-learn.org/>

Python Libraries for Data Science

matplotlib:

- python 2D plotting library which produces publication quality figures in a variety of hardcopy formats
- a set of functionalities similar to those of MATLAB
- line plots, scatter plots, barcharts, histograms, pie charts etc.
- relatively low-level; some effort needed to create advanced visualization

Link: <https://matplotlib.org/>

Login to the Shared Computing Cluster

- Use your SCC login information if you have SCC account
- If you are using tutorial accounts see info on the blackboard

Note: Your password will not be displayed while you enter it.

Selecting Python Version on the SCC

view available python versions on the SCC

```
[scc1 ~] module avail python
```

load python 3 version

```
[scc1 ~] module load python/3.6.2
```


Start Jupyter nootebook

On the Shared Computing Cluster

```
[scrl ~] jupyter notebook
```

 jupyter

Logout

Files **Running** Clusters

Select items to perform actions on them

Upload Item + 

<input type="checkbox"/>	Name	Last Modified
<input type="checkbox"/>	dataScience.pyth	8 minutes ago
<input type="checkbox"/>	figma.csv	2 minutes ago
<input type="checkbox"/>	Saleses.csv	a minute ago

Loading Python Libraries

```
In [ ]: #Import Python Libraries  
import numpy as np  
import scipy as sp  
import pandas as pd  
import matplotlib as mpl  
import seaborn as sns
```

Press Shift+Enter to execute the *jupyter* cell

Reading data using pandas

```
In [ ]: #Read csv file
        df = pd.read_csv("http://rds.bu.edu/examples/python/data_analysis/Salaries.csv")
```

Note: The above command has many optional arguments to fine-tune the data import process.

There is a number of pandas commands to read other data formats:

```
pd.read_excel('myfile.xlsx', sheet_name='Sheet1', index_col=None, na_values=['NA'])
pd.read_stata('myfile.dta')
pd.read_sas('myfile.sas7bdat')
pd.read_hdf('myfile.h5', 'df')
```

Exploring data frames

```
In [3]: #List first 5 records  
df.head()
```

```
Out[3]:
```

	rank	discipline	phd	service	sex	salary
0	Prof	B	56	49	Male	188980
1	Prof	A	12	6	Male	93000
2	Prof	A	23	20	Male	110515
3	Prof	A	40	31	Male	131205
4	Prof	B	20	18	Male	104800

Data Frame data types

Pandas Type	Native Python Type	Description
object	string	The most general dtype. Will be assigned to your column if column has mixed types (numbers and strings).
int64	int	Numeric characters. 64 refers to the memory allocated to hold this character.
float64	float	Numeric characters with decimals. If a column contains numbers and NaNs(see below), pandas will default to float64, in case your missing value has a decimal.
datetime64, timedelta[ns]	N/A (but see the datetime module in Python's standard library)	Values meant to hold time data. Look into these for time series experiments.

Data Frame data types

```
In [4]: #Check a particular column type  
df['salary'].dtype
```

```
Out[4]: dtype('int64')
```

```
In [5]: #Check types for all the columns  
df.dtypes
```

```
Out[4]: rank          object  
discipline  object  
phd          int64  
service     int64  
sex         object  
salary      int64  
dtype: object
```

Data Frame data types

```
In [4]: #Check a particular column type  
df['salary'].dtype
```

```
Out[4]: dtype('int64')
```

```
In [5]: #Check types for all the columns  
df.dtypes
```

```
Out[4]: rank          object  
discipline  object  
phd          int64  
service     int64  
sex         object  
salary      int64  
dtype: object
```


Data Frames attributes

Python objects have *attributes* and *methods*.

df.attribute	description
dtypes	list the types of the columns
columns	list the column names
axes	list the row labels and column names
ndim	number of dimensions
size	number of elements
shape	return a tuple representing the dimensionality
values	numpy representation of the data