ED5340 - Data Science: Theory and Practise

L13 - Libraries

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Prominent Libraries

used in data science / analysis

- Pandas
- Numpy
- matplotlib
- Scikit-Learn
- scipy
- Tensorflow (https://www.tensorflow.org/)
- PyTorch (https://pytorch.org/)

Pandas

L13_CSV.py

- Handling data (particularly for large scale data)
- csv, sql etc.
- import pandas as pd
- https://pandas.pydata.org/

Numpy

Numpy1.py

- Numerical python
- Arrays (similar to lists, but works 50 times faster!)
- Contigous memory locations (‘C’ concept of array)
- Access and manipulate easier
- Written in C / C++
- ndarray object
- import numpy as np

Numpy

Numpy1.py

- Numerical python
- Arrays (similar to lists, but works 50 times faster!)
- Contigous memory locations (‘C’ concept of array)
- Access and manipulate easier
- Written in C / C++
- ndarray object

Numpy

Numpy1.py

- Numerical python
- Arrays (similar to lists, but works 50 times faster!)
- Contigous memory locations (‘C’ concept of array)
- Access and manipulate easier
- Written in C / C++
- ndarray object

Numpy

Numpy2.py

- Creating arrays of any dimension
- Array indexing / slicing
- Array reshape
- arange, linspace functions
- Operations on array

matplotlib

L13_plottingfns.py

- similar to plotting using matlab
- plot 2D/3D curves, surfaces etc.
- importantly, contour plots
- import matplotlib.pyplot as plt
- import numpy as np

matplotlib

Simple plotting - L13_plottingfns.py

```
x = np.linspace(-5, 5, 100)

y = x**2

plt.plot(x,y)

# Default size figure
```

matplotlib

Simple plotting - L13_plottingfns.py

```
plt.plot(x, y, ‘o--')

plt.plot(x, y, 'o')

plt.plot(x, y, 'o--', color='red', lw = 2.5, ms = 2)
```

Adding figure

labelling, legend etc. - L13_plottingfns.py

```
#adding figure

plt.figure(figsize=(6,3))

plt.plot(x, y, 'o-', color='red', lw = 1.5, ms = 2, label = 'par. crv')

plt.xlabel('parameter')

plt.ylabel('curve')

plt.legend(loc='upper right', fontsize = 10)
```

Using subplots

L13_plottingfns.py

```
#using subplots
x = np.arange(-2.0, 2.0, 0.01)
y = x ** 2
#Default is single figure
#fig, ax = plt.subplots()
#single axis
fig, ax = plt.subplots(1, 1, figsize = (4,4))
ax.plot(x, y)
ax.set_xlabel('new x-label')
ax.set_ylabel('new y-label')
ax.set_title('Single axis plot')
# ax.set(xlabel='x vlues', ylabel='y values',
#        title='Explicit function y = f(x)')
ax.grid()
plt.show()
```

Using subplots

Multiple plotting - L13_plottingfns.py

```
#multiple axis figure
m_fig, m_axes = plt.subplots(2, 2, figsize = (8,4))
ax = m_axes[0][0]
ax.plot(x,y)
ax.set(xlabel='x vlues', ylabel='y values',
       title='Explicit function y = f(x)')
ax = m_axes[0][1]
ax.plot(x1,y1)
ax.set(xlabel='x vlues', ylabel='y values',
       title='Explicit function y = f(x)')
plt.show()
```

Plotting a curve

https://matplotlib.org/stable/gallery/mplot3d/lines3d.html - L13_plottingfns.py

```
# Parametric space curve t, t**2, t**3


t = np.arange(-2.0, 2.0, 0.1)
x = t
y = t ** 2
z = t ** 3

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.plot(x, y, z)

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
ax.set_title('Parametric space curve t, t**2, t**3')
```

Plotting a surface

https://matplotlib.org/stable/gallery/mplot3d/lines3d.html - L13_plottingfns.py

```
x = np.arange(-2.0, 2.0, 0.1)
y = np.arange(-2.0, 2.0, 0.1)
# The following will print a 3D surface
X,Y=np.meshgrid(x,y) #Forming MeshGrid
Z = X **2 + Y ** 2
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X, Y, Z)

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

plt.show()
```

Contour plot

L13_plottingfns.py

```
fig = plt.figure()
#ax = fig.add_subplot(111, projection='3d')


#ax.plot_surface(X, Y, Z)
cp  = plt.contour(x, y, Z)
plt.clabel(cp, fontsize=8)



plt.show()
```