实验1：常用python操作

廖　军
liaojun@cqu.edu.cn
重庆大学大数据与软件学院

数据科学导论实验课

序号	实验名称	实验内容要求	人数	占比
1	python基础实验	python	每人	30%
2	pythono数据应用实验	python	每人	
3	Linux和Hadoop操作实验	Hadoop及其命令	3-5人	
4	HDFS操作实验	HDFS及其编程	3-5人	

熟悉python的基本库操作

1.Pandas
2.Numpy
3.Scipy
4.Matplotlib

具体请参见：

访问地址：1. https://pandas.pydata.org/docs/user_guide/index.html

2. https://numpy.org/

3. https://www.scipy.org/

4. https://matplotlib.org/stable/index.html

1. Pandas

- It is built on top of NumPy
- Key components provided by Pandas:
  - Series ： 具有均匀数据的一维数组结构。
  - DataFrame： 数据帧是一个具有异构数据的二维数组。DataFrame特点：异构数据、大小可变、数据可变。

From now on: (always put on top of your code)

```
from pandas import Series, DataFrame
import pandas as pd
```

1. Pandas

#读取文件，注意文件的存储路径不能带有中文，否则读取可能出错。

pd.read_excel('data.xls') #读取Excel文件，创建DataFrame。

pd.read_csv('data.csv', encoding = 'utf-8') #Read csv file（or txt）读取文本格式的数据，一般用encoding指定编码。

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’) #HDF file

1. Pandas

```
#List first 5 records
df.head()
```

	rank	discipline	phd	service	sex	salary
0	Prof	B	56	49	Male	186960
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

1. Pandas

- 显示基本统计量(count, mean, std, min, quantiles, max)

```
df.describe()
```

- 如果需要选择行的范围，可以使用“：”指定范围

```
#Select rows by their position:
df[10:20]
```

- 如果我们需要选择一系列行，使用它们的标签，我们可以使用”loc”方法：

```
#Select rows by their labels:
df_sub.loc[10:20,['rank','sex','salary']]
```

- 如果我们需要选择一系列行和/或列，使用它们的位置，我们可以使用”iloc”方法：

```
#Select rows by their labels:
df_sub.iloc[20:50,[0, 3, 4, 5]]
```

1. Pandas

```
In [1]: df = DataFrame(randn(5,2),index=range(0,10,2),columns=list('AB'))

In [2]: df
Out[2]:
          A         B
0  1.068932 -0.794307
2 -0.470056  1.192211
4 -0.284561  0.756029
6  1.037563 -0.267820
8 -0.538478 -0.800654

In [5]: df.iloc[[2]]
Out[5]:
          A         B
4 -0.284561  0.756029

In [6]: df.loc[[2]]
Out[6]:
          A         B
2 -0.470056  1.192211
```

2. Numpy

ndarray 数组的创建

```
data1 = [6, 7.5, 8, 0, 1]
arr1 =  np.array(data1)                    #create 1-d array
from a list

data2 = [[1, 2, 3, 4], [5, 6, 7, 8]]       #list of lists
arr2 = np.array(data2)                     #2-d array
print(arr2.ndim)                           #2
Print(arr2.shape)  # (2,4)
```

2. Numpy

- 定义函数

```
>>> def myfunc(a, b):
...     "Return a-b if a>b, otherwise return a+b"
...     if a > b:
...         return a - b
...     else:
...         return a + b
```

```
>>> vfunc = np.vectorize(myfunc)
>>> vfunc([1, 2, 3, 4], 2)
array([3, 4, 1, 2])
```

2. Numpy

```
array = np.array([[0,1,2],[2,3,4]])
[[0 1 2]
 [2 3 4]]

array = np.zeros((2,3))
[[0. 0. 0.]
 [0. 0. 0.]]

array = np.ones((2,3))
[[1. 1. 1.]
 [1. 1. 1.]]
```

```
array = np.eye(3)
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

array = np.arange(0, 10, 2)
[0, 2, 4, 6, 8]

array = np.random.randint(0, 10, (3,3))
[[6 4 3]
 [1 5 6]
 [9 8 5]]
```

arange is an array-valued version of the built-in Python range function

3. Scipy

- Python中使用scipy.optimize模块的root和fsolve函数进行线性及非线性方程求解
- 例如：求解非线性方程组 $2x_1 - x_2^2 = 1, x_1^2 - x_2 = 2$
- from scipy.optimize import fsolve
- def f(x):

```
    x1 = x[0]
    x2 = x[1]
    return [2*x1 - x2**2 - 1, x1**2 - x2 -2]

result = fsolve(f, [1,1])
print(result)
```

3. Scipy

- 线性插值
- Scipy中的interpolate函数
- from scipy import interpolate
- 下面函数是一维线性函数用法

```
scipy.interpolate.interp1d(x, y, kind='linear',
axis=-1, copy=True, bounds_error=None,
fill_value=nan, assume_sorted=False)
```

4. Matplotlib

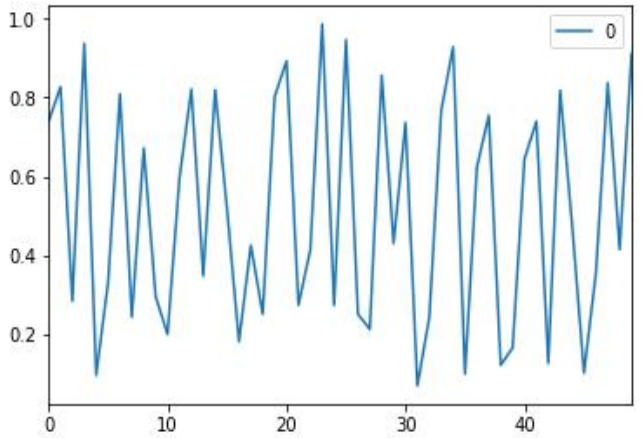
- 基本的导入包：

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

4. Matplotlib

- 基本的plot

```
df= pd.Series(np.random.randn(1000),
              index=pd.date_range('1/1/2000', periods=1000))
df =df.cumsum() #cumulative sum
df.plot()
```



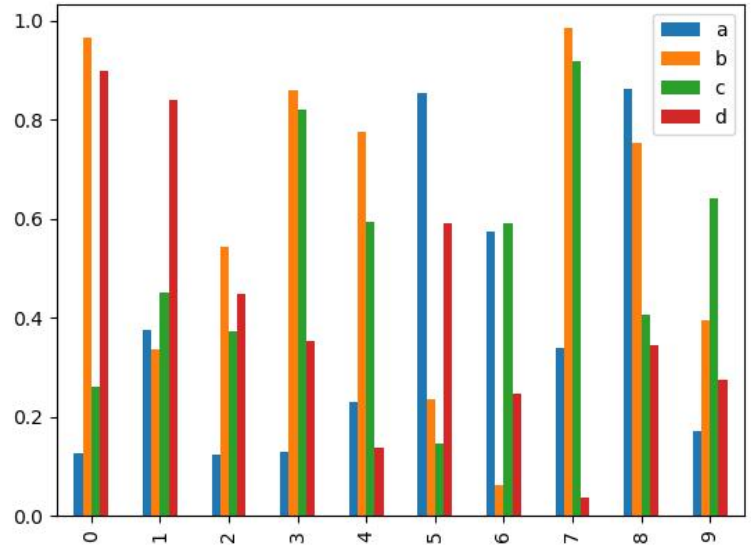
- 线条line

```
df = pd.DataFrame(np.random.rand(50))
df.plot.line()
```

4. Matplotlib

- 条形bar

```
df2 = pd.DataFrame(np.random.rand(10, 4),
                   columns=['a', 'b', 'c', 'd'])
df2.plot.bar()
```

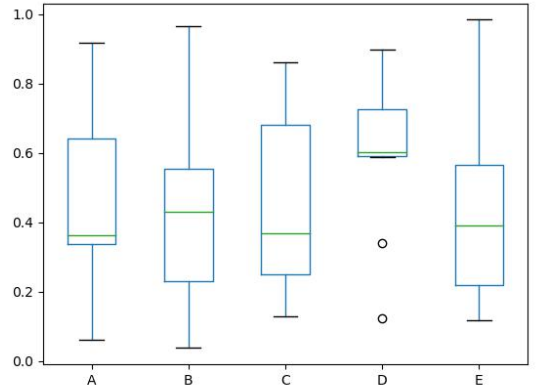


- 直方图hist

```
df = pd.DataFrame(np.random.rand(50))
df.plot.hist()
```

Histograms can be drawn by using the DataFrame.plot.hist() and Series.plot.hist() methods.

4. Matplotlib



- 箱图 *Box plots*

Series.plot.box()

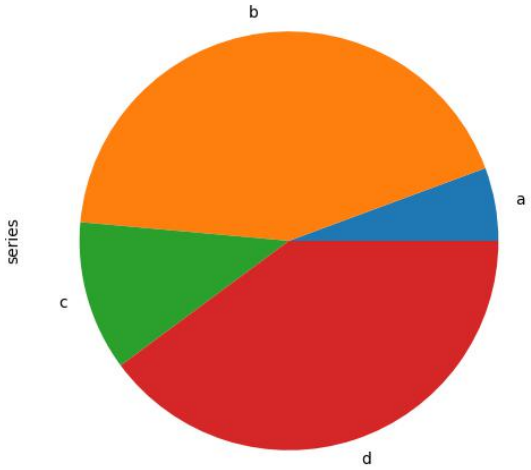
DataFrame.plot.box(),

或 DataFrame.boxplot()

```
df = pd.DataFrame(np.random.rand(10, 5), columns=["A", "B", "C", "D", "E"])
df.plot.box();
```

- 饼图pie
- DataFrame.plot.pie() or Series.plot.pie()
- series = pd.Series(3 * np.random.rand(4), index=["a", "b", "c", "d"], name="series")

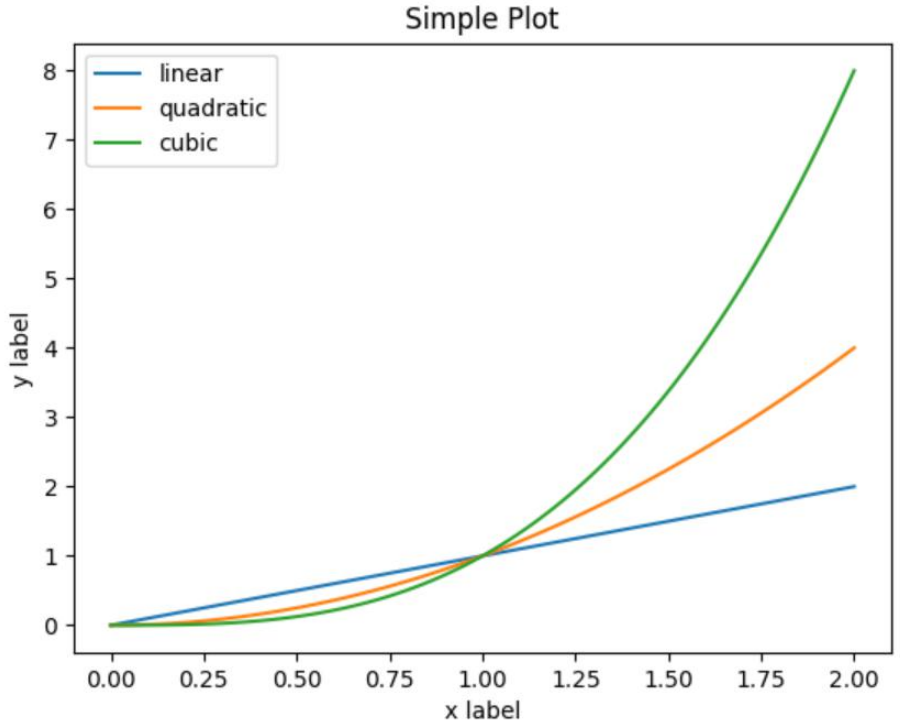
```
series.plot.pie(figsize=(6, 6));
```



4. Matplotlib

```
x = np.linspace(0, 2, 100)

plt.plot(x, x, label='linear')  # Plot some data on the (implicit) axes.
plt.plot(x, x**2, label='quadratic')  # etc.
plt.plot(x, x**3, label='cubic')
plt.xlabel('x label')
plt.ylabel('y label')
plt.title("Simple Plot")
plt.legend()
```



4. Matplotlib

- 介绍基本的可视化方法：

① 更多的方法：自学

② 参考相应网站： https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

③ https://matplotlib.org/stable/tutorials/index.html