

一、Python 学习报告

1. Numpy 学习

Numpy 主要用于科学计算，最重要的一个特点是 N 维数组对象。Numpy 模块中包含着大量对数组进行操作的方法，而对于我们进行的数据分析而言，最关注的主要功能有：

- (1) 快速的矢量化数组运算，如转置、点积等
- (2) 常见的数组算法，如排序、唯一化、集合运算等
- (3) 提供操作数组对象的优化函数和方法，加快数据处理速度

分别针对不同操作进行了示例，选取部分示例列如下：

(1) 创建数组及数组类型

```
In [12]: import numpy as np
          data1=[6,7.5,8,0,1]
          arr1=np.array(data1) #创建数组
          arr1

Out[12]: array([ 6. ,  7.5,  8. ,  0. ,  1. ])
```

```
In [15]: arr1=np.array([[5.1,6.3,7.9],[8.2,9,6,31],[2,2,3,4]])
          arr1 #创建二维数组

Out[15]: array([[5.1, 6.3, 7.9], [8.2, 9, 6, 31], [2, 2, 3, 4]], dtype=object)
```

```
In [10]: arr2=np.arange(6) #顺序数组
          arr2

Out[10]: array([0, 1, 2, 3, 4, 5])
```

```
In [13]: arr3=np.array([1,2,3,4,5,6],dtype=np.float64)
          arr4=np.array([1,2,3,4,5,6],dtype=np.int32) #自定义数据类型
          arr4.dtype

Out[13]: dtype('int32')
```

(2) 特殊数组和数组索引

```
In [50]: from numpy.random import randn
          names=np.array(['Bob','Joe','Will','Bob','Will','Joe','Joe'])
          data=np.arange(28).reshape(7,4)
          data #顺序排列为指定大小的数组

Out[50]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19],
                [20, 21, 22, 23],
                [24, 25, 26, 27]])
```

```
In [51]: data[names='Bob']=3
          data          #数组索引

Out[51]: array([[ 3,  3,  3,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [ 3,  3,  3,  3],
                [16, 17, 18, 19],
                [20, 21, 22, 23],
                [24, 25, 26, 27]])
```

(3) 统计运算与数组存取

```
In [61]: np.unique(names)      #unique函数, 找出数组中唯一值并返回排序后结果

Out[61]: array(['Bob', 'Joe', 'Will'],
              dtype='<S4')

In [67]: data.cumsum()        #累积和

Out[67]: array([  3,  6,  9, 12, 16, 21, 27, 34, 42, 51, 61, 72, 75,
                78, 81, 84, 100, 117, 135, 154, 174, 195, 217, 240, 264, 289,
                315, 342])

In [71]: arr=np.arange(10)
          np.save('some_array',arr)  #数组保存与读取
          np.load('some_array.npy')

Out[71]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

2. pandas 学习

pandas 基于 Numpy 构建, 含有使数据分析工作变得更快更简单的高级数据结构和操作工具。它的两个主要数据结构: Series 和 DataFrame。Series 由一组数据以及一组与之相关的数据标签组成。很多时候, Series 可看成是一个定长的有序字典, 可以直接通过字典来创建 Series。DataFrame 是表格型的数据结构, 有行索引和列索引。DataFrame 构造函数可以接受很多种数据, 如: 二维 ndarray, 由数组、列表或元组组成的字典等。

分别针对不同操作进行了示例, 选取部分示例列如下:

(1) Series 相关操作

```
In [86]: obj1=Series([4,7,-5,3],index=['d','b','a','c'])
          obj1          #创建序列和索引

Out[86]: d      4
         b      7
         a     -5
         c      3
         dtype: int64
```

```
In [93]: import pandas as pd
sdata={'Ohio':35000,'Texas':71000,'Oregon':16000,'Utah':5000}
states=['California','Ohio','Oregon','Texas']
obj2=Series(sdata,index=states)
pd.isnull(obj2)    #匹配并查找空值
```

```
Out[93]: California    True
Ohio                False
Oregon             False
Texas              False
dtype: bool
```

(2) DataFrame 简单操作和与 Series 的变换

```
In [95]: data={'state':['Ohio','Ohio','Ohio','Nevada','nevada'],
              'year':[2000,2001,2002,2001,2002],
              'pop':[1.5,1.7,3.6,2.4,2.9]}
frame1=DataFrame(data,          #生成表格, 按索引和列排序
                  index=['one','two','three','four','five'],
                  columns=['year','state','pop'])
frame1
```

```
Out[95]:
```

	year	state	pop
one	2000	Ohio	1.5
two	2001	Ohio	1.7
three	2002	Ohio	3.6
four	2001	Nevada	2.4
five	2002	nevada	2.9

```
In [98]: frame1['pop']    #从DataFrame获取列的Series
```

```
Out[98]: one        1.5
two         1.7
three       3.6
four        2.4
five        2.9
Name: pop, dtype: float64
```

```
In [99]: frame1.ix['three']    #从DataFrame获取行的Series
```

```
Out[99]: year        2002
state        Ohio
pop          3.6
Name: three, dtype: object
```

(3) DataFrame 的索引

```
In [125]: data.ix['Colorado', ['two', 'three']] #选取行和列
```

```
Out[125]: two      5
          three    6
          Name: Colorado, dtype: int32
```

```
In [123]: frame3=frame2.drop('state', axis=1)
          frame3 #删除指定列
```

```
Out[123]:
```

	year	pop
one	2000	1.5
two	2001	1.7

(4) 缺失数据处理

```
In [131]: from numpy import nan as NA
          data=DataFrame([[1., 6.5, 3.], [1., NA, NA], [NA, NA, NA], [NA, 6.5, 3.]])
          data
```

```
Out[131]:
```

	0	1	2
0	1.0	6.5	3.0
1	1.0	NaN	NaN
2	NaN	NaN	NaN
3	NaN	6.5	3.0

```
In [132]: data1=data.dropna()
          data1 #删除缺失数据
```

```
Out[132]:
```

	0	1	2
0	1.0	6.5	3.0

```
In [134]: data2=data.fillna(method = 'ffill')
          data2 #指定方法填充数据
```

```
Out[134]:
```

	0	1	2
0	1.0	6.5	3.0
1	1.0	6.5	3.0
2	1.0	6.5	3.0
3	1.0	6.5	3.0

(5) 层次化索引

```
In [137]: frame=DataFrame(np.arange(12).reshape(4,3),
                           index=[['a','a','b','b'],[1,2,1,2]],
                           columns=[['Ohio','Ohio','Colorado'],
                                     ['Green','Red','Green']])
frame.index.names=['key1','key2']
frame.columns.names=['state','color']
frame
```

Out[137]:

	state	Ohio		Colorado
	color	Green	Red	Green
key1	key2			
a	1	0	1	2
	2	3	4	5
b	1	6	7	8
	2	9	10	11

```
In [139]: frame.sum(level='color',axis=1)  #根据级别进行汇总统计
```

Out[139]:

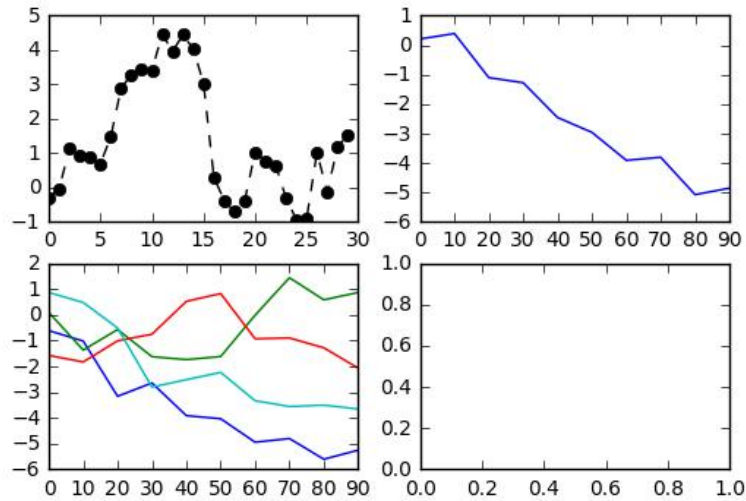
	color	Green	Red
key1	key2		
a	1	2	1
	2	8	4
b	1	14	7
	2	20	10

3. Matplotlib 的学习

绘图是数据分析工作中最重要的任务之一。Matplotlib 是一种功能丰富且高效的绘图环境。分别针对不同操作进行了示例，选取部分示例列如下：

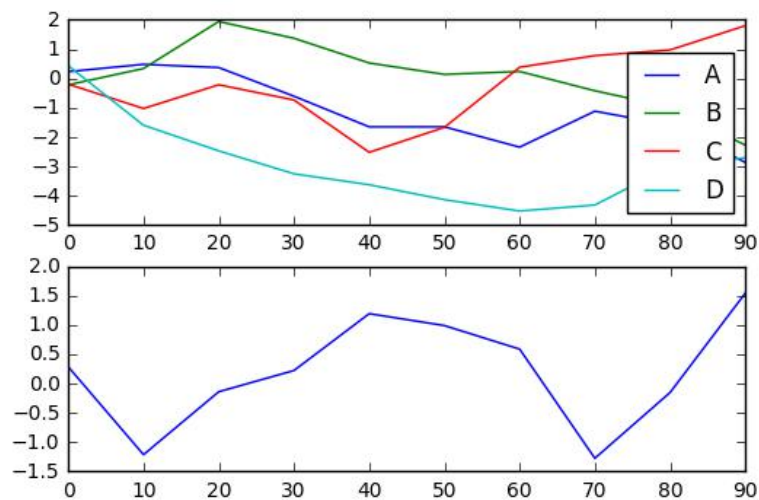
(1) Matplotlib 画图

```
In [158]: import matplotlib.pyplot as plt
fig= plt.figure() #创建空表
ax1=fig.add_subplot(2,2,1) #创建第一个方格
plt.plot(randn(30).cumsum(), 'ko--') #画图
ax2=fig.add_subplot(2,2,2)
plt.plot(Series(randn(10).cumsum(), #画序列图
               index=np.arange(0,100,10)))
ax3=fig.add_subplot(2,2,3)
plt.plot(DataFrame(randn(10,4).cumsum(0), #画表格图
                  columns=['A', 'B', 'C', 'D'],
                  index=np.arange(0,100,10)))
ax4=fig.add_subplot(2,2,4)
plt.show()
```



(2) pandas 中的画图工具

```
In [159]: fig, axes = plt.subplots(2,1)
s=Series(np.random.randn(10).cumsum(),
         index=np.arange(0,100,10))
s.plot(ax=axes[1])
df = DataFrame(np.random.randn(10,4).cumsum(0),
               columns=['A', 'B', 'C', 'D'],
               index=np.arange(0,100,10))
df.plot(ax=axes[0])
plt.show()
```



二、股指期货学习报告

1. 股指期货定义

(1) 期货

期货就是按照约定价格超前进行买卖的交易合约。期货主要不是货，而是以某种大众产品如棉花、大豆、石油等及金融资产如股票、债券等为标的标准化可交易合约。交收期货的日子可以是一星期之后，一个月之后，三个月之后，甚至一年之后。

通俗地讲，现货是指一手交钱，一手交货；期货是指定下未来交割。比如现在 11 月 22 号，我们约定 12 月 30 买卖，这就是期货。

(2) 股指

股指是股票价格指数简称，就是把一些股票按照权重编制，衡量股市走势的一种指数。一些金融服务机构编制出股票价格指数，作为市场价格变动的指标。投资者据此就可以检验自己投资的效果，并用以预测股票市场的动向。

(3) 股指期货

期货有很多种，标的物可以是某种商品（例如黄金、原油、农产品），也可以是金融工具。股指期货就是其中一种，是指以股指为标的物的标准化期货合约，双方约定在未来的某个特定日期，可以按照事先确定的股指的大小，进行标的指数的买卖，到期后通过现金结算差价来进行交割。

2. 股指期货的产品

目前国内有三种股指期货合约，分别为沪深 300 股指期货 IF，中证 500 股指期货 IC，上证 50 股指期货 IH，每一种股指期货根据时间不同各有四个合约，一个当月合约，一个下月合约和两个季月合约。沪深 300 指数是股指期货最重要的标的指数，比如目前上市的沪深 300 股指期货合约分别为 IF1612，IF1701，IF1703，IF1706。IF1612 是当月合约，指的是 2016 年 12 月第三个星期五进行交割的合约，IF1701 是下月合约，指的是 2017 年 1 月第三个星期五进行交割的合约，IF1703 是当季合约，指的是 2017 年 3 月第三个星期五进行交割的合约，IF1706 是下季合约，指的是 2017 年 6 月第三个星期五进行交割的合约。

3. 股指期货的交易流程

我们的盈利方式主要有两种：做多和做空。

做多：看好未来的上涨前景而进行买入持有等待上涨获利。所以做多就是买入股票、外汇或期货等。

做空：预期未来行情下跌，将手中借入的股票按目前价格卖出，待行情跌后买进再归还，获取差价利润。其交易行为特点为先卖后买。比如预计某一股票未来会跌，就在当期价位高时借入此股票（实际交易是买入看跌的合约）卖出，再到股价跌到一定程度时买进，以现价还给卖方，产生的差价就是利润。

多头和空头各支付 10%-40% 的保证金买卖股指期货合约，然后按照每一个指数点位 300 元的价格计算盈亏，指数每上涨一个点多头盈利 300 元，而空头亏损 300 元，指数每下跌一个点空头盈利 300 元，而多头亏损 300 元。这个 300 元/点叫价格乘数，世界各国的价格乘数是不一致的。

举个例子，假如 20160301 10:38 的沪深 300 指数为 2900 点，预期价格会下跌，于是我卖出一手股指期货合约，保证金比例为 10%，则占用的保证金为 $2900 \times 300 \times 10\% = 87000$ 元，10:59 价格跌至 2880 点，我选择买入平仓，那么盈利为 $(2900 \text{ 点} - 2880 \text{ 点}) \times 300 \text{ 元/点} = 6000$ 元，收益率为 $6000 / 87000 = 6.9\%$ ，20160301 13:32 的沪深 300 指数为 2870 点，预期价格会上涨，于是我

买入一手股指期货合约,占用的保证金为 $2870 \times 300 \times 10\% = 86100$ 元, 14:18 价格达到 2920 点, 我选择卖出平仓, 那么盈利为 $(2920 \text{ 点} - 2870 \text{ 点}) \times 300 \text{ 元/点} = 15000$ 元, 收益率为 17.4%。

4. 股指期货与股票的区别

(1) 期货合约有到期日, 不能无限期持有

股票买入后可以一直持有, 正常情况下股票数量不会减少。但股指期货都有固定的到期日, 到期就要进行平仓或者交割。

(2) 期货合约是保证金交易, 必须每日结算

股票买入后, 在卖出以前, 账面盈亏都是不结算的。但股指期货不同, 交易后每天要按照结算价对持有在手的合约进行结算, 账面盈利可以提走, 但账面亏损第二天开盘前必须补足(即追加保证金)。而且由于是保证金交易, 亏损额甚至可能超过你的投资本金, 这一点和股票交易不同。

(3) 期货合约可以卖空

股指期货合约可以十分方便地卖空, 等价格回落后再买回。股票融券交易也可以卖空, 但难度相对较大。当然一旦卖空后价格不跌反涨, 投资者会面临损失。

(4) 股指期货关注宏观经济

一般说来, 股指期货市场是专注于根据宏观经济资料进行的买卖, 而现货市场则专注于根据个别公司状况进行的买卖。

(5) 股指期货实行 T+0 交易, 而股票实行 T+1 交易

T+0 即当日买进当日卖出, 没有时间和次数限制, 而 T+1 即当日买进, 次日卖出, 买进的当日不能当日卖出。

5. 主力合约

所谓主力合约指的是**成交量最大的合约**。也有说法是主力合约是持仓量最大的合约, 因为通常来讲, 持仓量最大的合约也是成交量最大的合约。

因为它是市场上最活跃的合约, 所有投机者基本上都在参与这个合约。

6. 日内交易

(1) 概念

股指期货日内交易是一种交易模式。主要是指持仓时间短, **不留过夜持仓**的交易方式。日内短线交易捕捉入市后能够马上脱离入市成本的交易机会, 入市之后如果不能马上获利, 就准备迅速离场。因为这种交易方式在市时间短, 所以承受的市场波动的风险较低。

(2) 交易方法

日内交易的经典交易方法主要有以下几种:

A. **区间突破**。区间突破基于昨日振幅与今日开盘价的关系; $\text{昨日振幅} = \text{昨日最高价} - \text{昨日最低价}$; $\text{上轨} = \text{今日收盘价} + N \times \text{昨日振幅}$; $\text{下轨} = \text{今日收盘价} - N \times \text{昨日振幅}$; 当价格突破上轨, 买入开仓; 当价格跌穿下轨, 卖出开仓。

B. **菲阿里四价**。昨天高点、昨天低点、昨日收盘价、今天开盘价, 可并称为菲阿里四价。上轨 = 昨日高点; 下轨 = 昨日低点; 当价格突破上轨, 买入开仓; 当价格跌穿下轨, 卖出开仓。

C. **空中花园**。空中花园在当天高开或低开时使用, 即当 $\text{开盘价} \geq \text{昨天收盘价} \times 1.01$ 或 $\text{开盘价} \leq \text{昨天收盘价} \times 0.99$ 时; $\text{上轨} = \text{第一根 K 线的最高价}$; $\text{下轨} = \text{第一根 K 线的最低价}$; 当价格突破上轨, 买入开仓; 当价格跌穿下轨, 卖出开仓。

D. **横盘突破**。横盘突破在过去 30 根 K 线的高低点围绕中轴上下 0.5% 的范围内波动时；上轨 = 过去 30 根 K 线的最高价；下轨 = 过去 30 根 K 线的最低价；当价格突破上轨，买入开仓；当价格跌穿下轨，卖出开仓。

E. **转向交易**。转向交易基于今日开盘价；上轨 = 今日开盘价 + 今日开盘价 * 0.01；下轨 = 今日开盘价 - 今日开盘价 * 0.01；当价格突破上轨，买入开仓；当价格跌穿下轨，卖出开仓。