



期权论坛独家PDF

www.optbbs.com

零基础搭建Python 量化研究环境

python研究量化投资的入门教程

自我介绍



东莞何文峰Sam

广东 东莞



扫一扫上面的二维码图案，加我微信

广东东莞人，计算机专业，影子量化团队核心成员。东莞宽客俱乐部负责人。8年投资经验，研究量化投资3年。擅长于python语言。目前专门研究机器学习与量化投资的系统开发。致力于推动各大私募和投资公司的定制化交易系统构建。和推进人工智能技术在量化投资的应用。

讲座大纲

- 搭建python编程环境
- 利用tushare获取行情数据
- 在Python下实现交易策略。
- 交易信号提取与资金曲线展示。
- Python量化研究公用平台介绍

为什么用python

1、具有高度的拓展性。

有各种现成的代码可以使用，写图形界面有Qt，数据统计分析numpy，scipy,pandas，机器学习有pybrain，回测模块有pyalgotrade. 几乎你想干的事都有现成的库可以用

2、学习曲线平缓。

入门容易，现成的教程本多。 非计算机专业入门首选。

3、可以摆脱第三方平台（TB，金字塔等）的局限性。

进行机器学习，事件驱动策略等高级策略的开发。策略的保密性也得到了更高的保障。 还有一点，免费。

4、后续学习对应项目选择多。

可以对接VNPY（连接CTP可实盘），ricequant，优矿等PYTHON量化项目进行后续深入研究。

代码运行环境

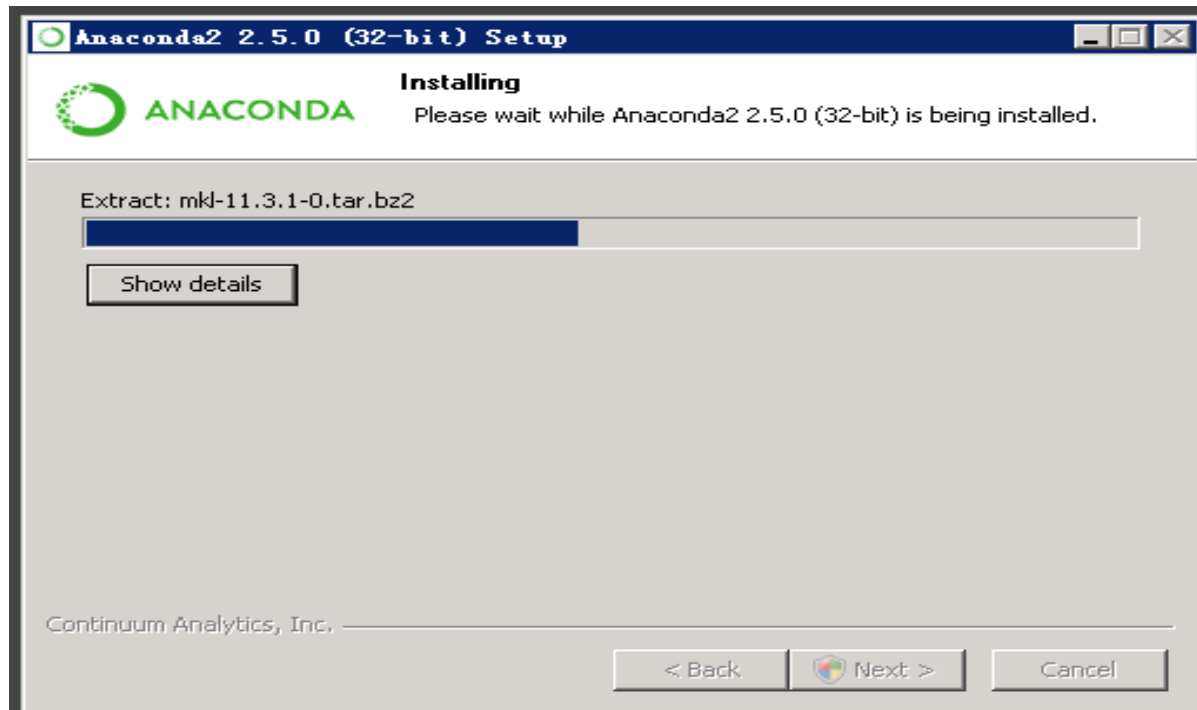
- Windows XP及以上系统。
- Python 2.7

必备软件

- **Anaconda:** python科学计算环境，已经默认安装好python所有组件和常用库（numpy,scipy,pandas等）
- **Pycharm:** 是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。
- **Tushare:** 历史行情数据获取，本教程获取沪深300指数。

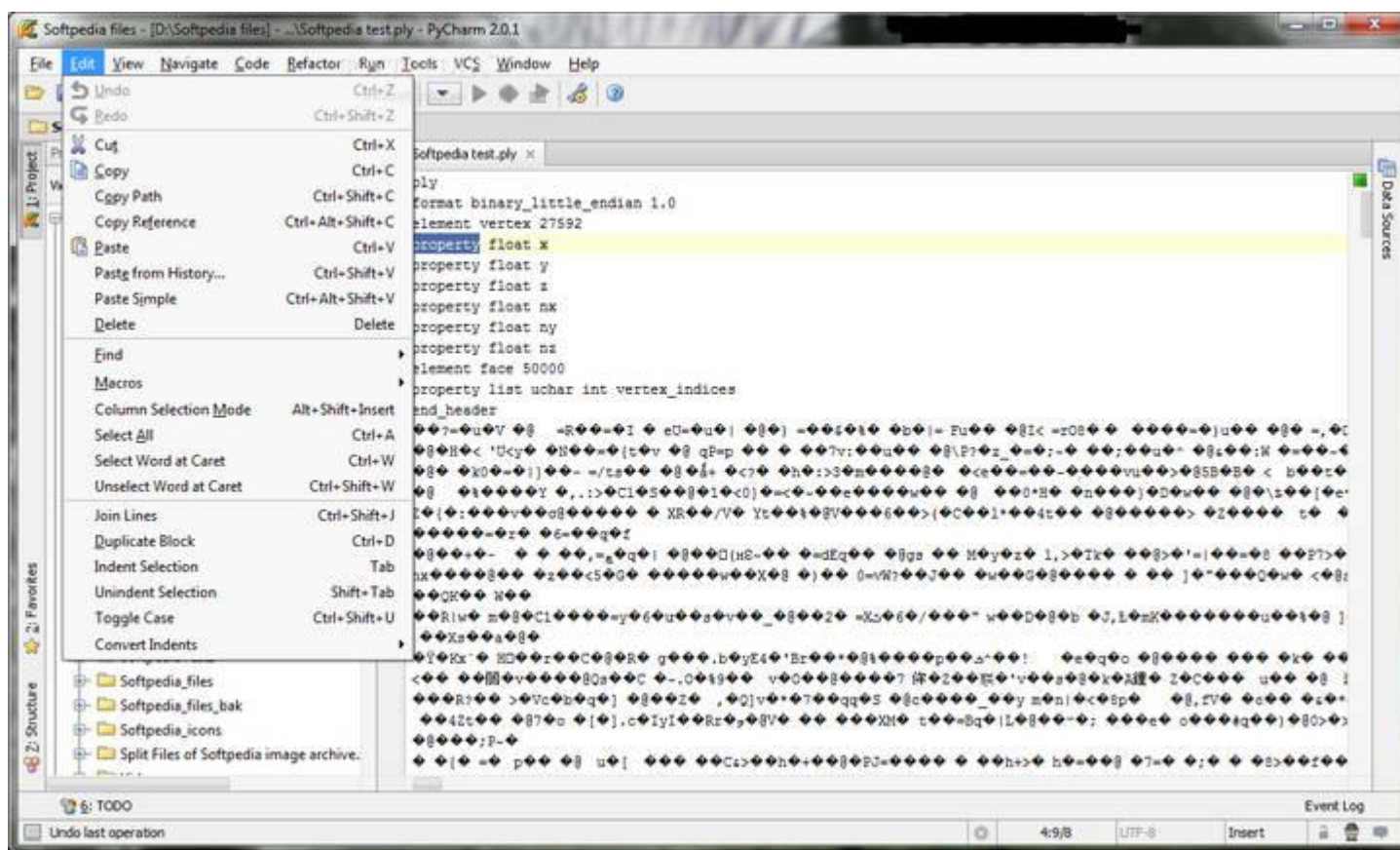
安装Anaconda

下载地址: https://www.continuum.io/downloads#_windows



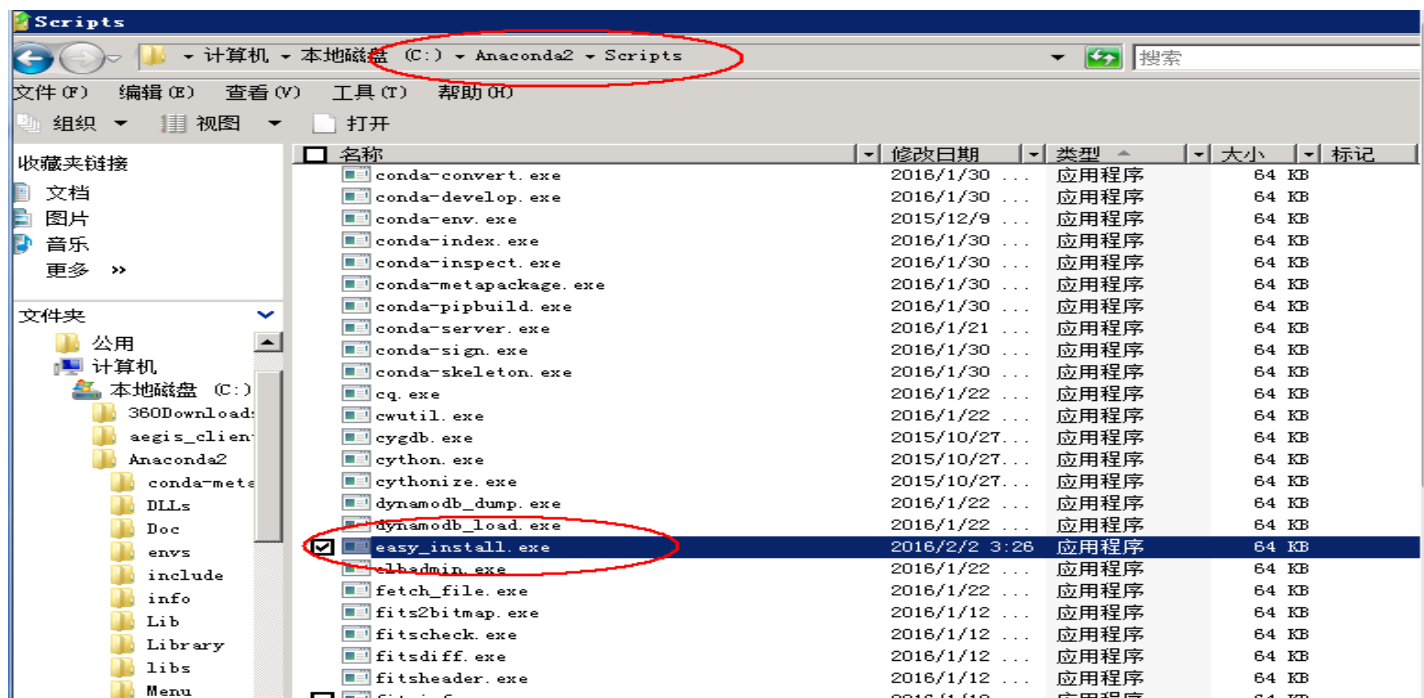
安装Pycharm

- 下载地址<http://www.onlinedown.net/soft/255225.htm>



配置 PIP

PIP是一个安装和管理 Python 包的工具，安装了PIP以后我们可以很方便地安装和管理各种python包。本教程中使用pip安装行情接口tushare。



打开 C:/Anaconda/Scripts / easy-install.exe 点击安装。安装后可以使用

使用pip安装获取数据工具tushare

第一步，点击开始，调出CMD



第二步，输入 pip install tushare 安装行情接口

```
管理员: C:\Windows\system32\cmd.exe
C:\Users\Administrator>pip install tushare
Collecting tushare
  Downloading tushare-0.4.7-py2-none-any.whl (88kB)
    41% |#####| 36kB 2.3MB/s eta 0:00:0
    46% |#####| 40kB 2.6MB/s eta 0:00:0
    51% |#####| 45kB 2.6MB/s eta 0:00:0
    55% |#####| 49kB 2.6MB/s eta 0:00:0
    60% |#####| 53kB 2.6MB/s eta 0:00:0
    65% |#####| 57kB 2.6MB/s eta 0:00:0
    69% |#####| 61kB 2.6MB/s eta 0:00:0
    88% |#####| 77kB ...
    92% |#####| 81kB ..
    97% |#####| 86kB
   100% |#####| 90kB
2.7MB/s
Installing collected packages: tushare
Successfully installed tushare-0.4.7
```

获取历时行情数据

我们已经完成了必备软件的下载和安装，接下来我们将要进行历时行情的获取。主要分为两步

第一步，在python项目开始时导入tushare包。

第二部，生成tushare实例，这里我们生成了一个沪深300指数的实例。

代码如下：

```
In [3]: import tushare as ts ← 导入Tushare包，用于获取历史数据
...: import matplotlib.pyplot as plt ← 导入matplotlib.pyplot包，用于画图
...:
```

```
In [4]: hs300=ts.get_hist_data('hs300') ← 获取沪深300指数的历史数据
```

```
In [5]: hs300['close'].head(10)
```

```
Out[5]:
```

```
date
2016-03-28    3169.728
2016-03-25    3197.817
2016-03-24    3181.851
2016-03-23    3236.092
2016-03-22    3225.790
2016-03-21    3249.437
2016-03-18    3171.963
2016-03-17    3124.204
2016-03-16    3090.033
2016-03-15    3074.780
Name: close, dtype: float64
```

```
In [6]: hs300=hs300.sort_index() ← 将数据的索引反转，方便显示数据
```

```
In [7]: hs300['close'].head(10)
```

```
Out[7]:
```

```
date
2013-04-01    2493.187
2013-04-02    2486.391
2013-04-03    2483.547
2013-04-08    2472.299
2013-04-09    2489.430
2013-04-10    2485.309
2013-04-11    2477.876
2013-04-12    2462.112
2013-04-15    2436.818
```

把数据放在一个盒子里

DataFrame

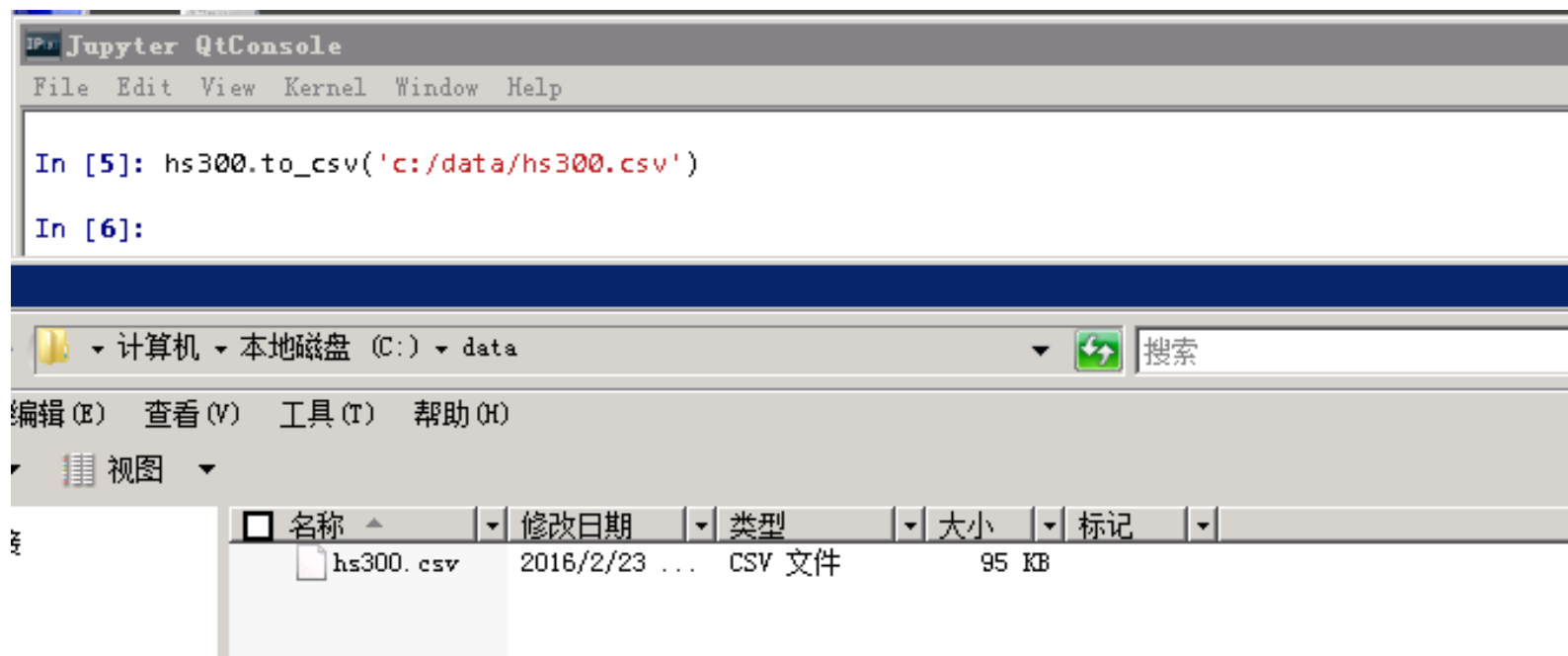
```
• type(hs300)
• Out[11]: pandas.core.frame.DataFrame
• hs300.head(10)
• Out[12]:
```

	open	high	close	low	volume	price_change \
date						
2016-02-22	3083.933	3128.207	3118.868	3071.329	140946976	67.283
2016-02-19	3047.942	3065.968	3051.585	3032.634	87338016	-2.114
2016-02-18	3079.527	3093.513	3053.699	3049.841	124401464	-9.621
2016-02-17	3034.090	3065.785	3063.320	3027.046	128291608	26.285
2016-02-16	2962.187	3042.089	3037.035	2962.187	112579640	90.328
2016-02-15	2888.428	2961.493	2946.707	2886.018	72919432	-17.082
2016-02-05	2982.056	2988.360	2963.789	2962.847	76850808	-20.971
2016-02-04	2960.334	2999.206	2984.760	2960.207	91779728	36.121
2016-02-03	2931.288	2954.710	2948.639	2909.898	77389000	-12.695
2016-02-02	2901.528	2972.426	2961.334	2901.512	88601960	60.286

DataFrame: 是pandas包的一种数据结构, 用于编辑和显示量化数据。我们通过DataFrame编辑数据, 来计算各种指标, 显示交易信号和资金曲线。一切围绕DataFrame而进行。

保存数据

生成了hs300实例以后，我们用to_csv方法对数据进行保存。



显示数据

完成了对数据的获取之后，我们想查看我们得到的数据。

首先我们要确保我们已经导入了画图的包 **matplotlib**.

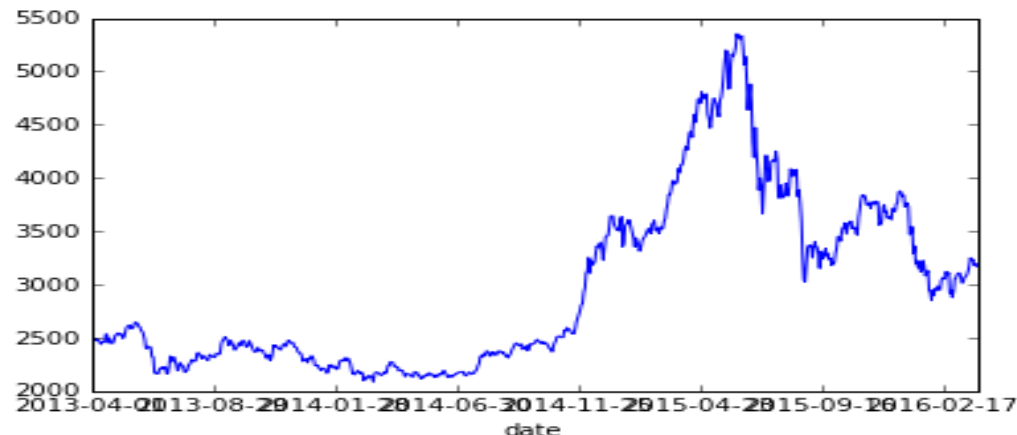
```
import matplotlib.pyplot as plt ← 导入matplotlib.pyplot包，用于画图
```

然后我们直接调用hs300实例的**plot（）**方法，就可以对我们的数据进行画图显示。

如下展示：

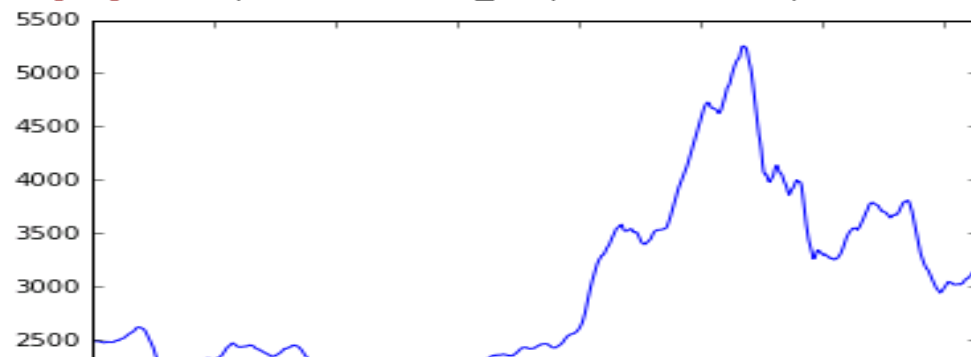
基本交互式讲解—数据展示

```
In [9]: hs300['close'].plot()  
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7847d90>
```



显示收盘价

```
In [10]: hs300['ma10'].plot()  
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7a593f0>
```

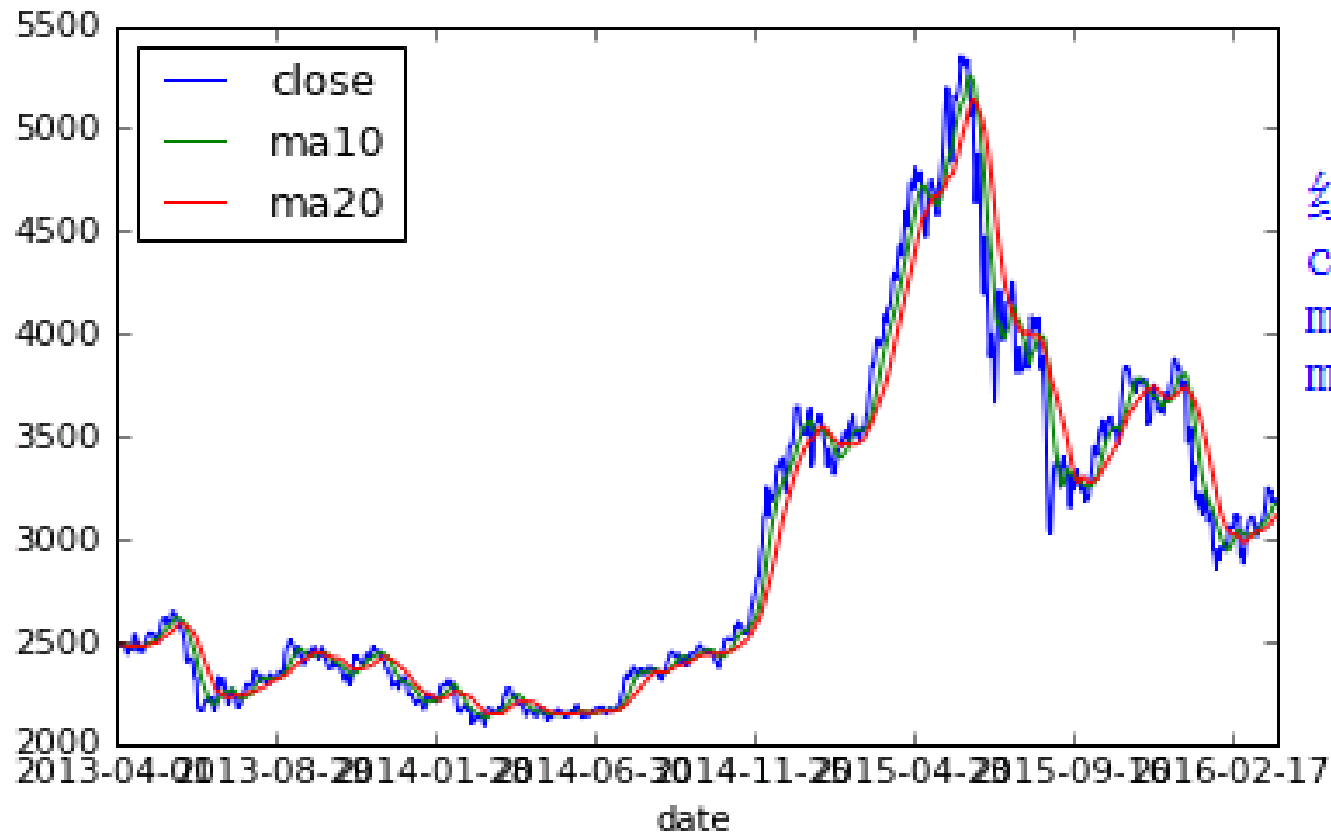


显示10日均线

均线组合显示

```
In [15]: hs300[['close', 'ma10', 'ma20']].plot()
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7e5d170>
```



组合显示
close
ma10
ma20

```
In [16]:
```

用python实现一个简单策略

- 策略逻辑：

空仓：10日均线在40日均线的上下3点范围内。

多头：10日均线 高于40日均线势3点。

空头：10日均线低于40日均线3点。

（此次略非多即空）

用python实现一个简单策略

In [20]:

In [21]: spread=3 设置参数, 阈值为3

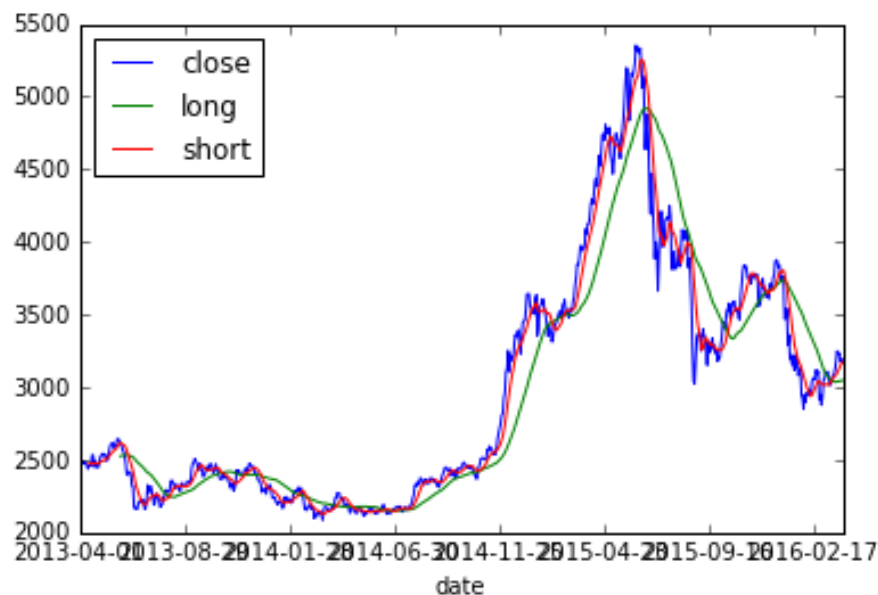
In [22]: hs300['short']=np.round(pd.rolling_mean(hs300['close'],window=10),2) 增加短线列, 10日均线

In [23]: hs300['long']=np.round(pd.rolling_mean(hs300['close'],window=40),2) 增加长线列, 40日均线

In [24]: hs300['short-long']=hs300['short']-hs300['long'] 增加差值列, 用于跟参数3比较

In [25]: hs300[['close','long','short']].plot()

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x82d9290>



将收盘价, 长短均线组合展示。

In [26]: |

用python实现信号提取

- 完成均线的描述以后，我们需要对策略的多空信号进行提取。
- 具体做法是通过数据统计库numpy的where函数对逻辑进行判定。在hs300实例中增加【signal】列，从而产生交易信号。

代码如下：

用python实现信号提取

```
In [26]: hs300['signal']=np.where(hs300['short-long']>spread,1,0)
```

```
In [27]: hs300['signal']=np.where(hs300['short-long']<spread,-1,hs300['signal'])
```

```
In [28]: print hs300['signal'].value_counts()
```

```
1      385
```

```
-1     305
```

```
0       39
```

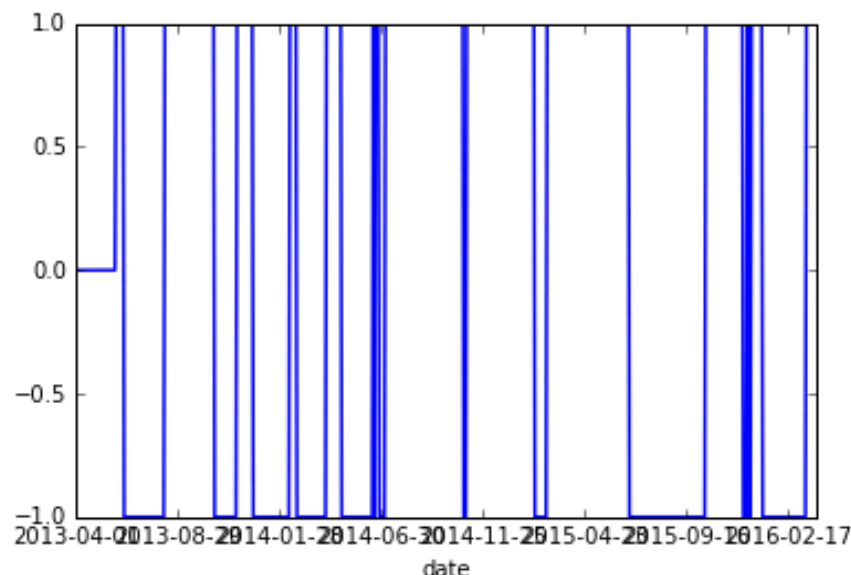
```
Name: signal, dtype: int64
```

对产生信号进行统计

增加[SIGNAL]列，通过np的where函数进行逻辑判断，从而把0：空仓信号； 1：多仓信号； -1：空仓信号产生记录。

```
In [29]: hs300['signal'].plot(lw=1.5)
```

```
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x82e42d0>
```



将产生的信号可视化展示

用python实现资金曲线

好了，所有数据跟信号已经准备完毕，策略严格上说已经完成了。但是我们需要做一个简单的回测模块来看看我们的交易策略的运行情况。

具体的思路是通过取对数的方式归一化，将价格走势和资金曲线同步显示在一张图标当中，代码如下：

用python实现资金曲线

In [30]: `hs300['market']=np.log(hs300['close']/hs300['close'].shift(1))` 增加market列用取对数的形式记录走势

In [31]: `hs300['straegy']=hs300['signal'].shift(1)*hs300['market']` 增加straegy列用于记录相对于走势的资金曲线

In [32]: `hs300[['market','straegy']].cumsum().apply(np.exp).plot()`

Out[32]: `<matplotlib.axes._subplots.AxesSubplot at 0x82e4590>`



组合显示。

In [33]:

一切的背后.....

```
In [33]: hs300.count()
```

```
Out[33]:
```

open	729
high	729
close	729
low	729
volume	729
price_change	729
p_change	729
ma5	729
ma10	729
ma20	729
v_ma5	729
v_ma10	729
v_ma20	729
short	720
long	690
short-long	690
signal	729
market	728
straegy	728

dtype: int64

所有的运算其实就在矩阵里面完成了。包括指标，差值，信号，资金曲线。

聪明的你现在明白了吗？

把数据放在一个盒子里

DataFrame

- `type(hs300)`

- `Out[11]: pandas.core.frame.DataFrame`

- `hs300.head(10)`

- `Out[12]:`

	open	high	close	low	volume	<u>price_change</u>	\
date							
2016-02-22	3083.933	3128.207	3118.868	3071.329	140946976	67.283	
2016-02-19	3047.942	3065.968	3051.585	3032.634	87338016	-2.114	
2016-02-18	3079.527	3093.513	3053.699	3049.841	124401464	-9.621	
2016-02-17	3034.090	3065.785	3063.320	3027.046	128291608	26.285	
2016-02-16	2962.187	3042.089	3037.035	2962.187	112579640	90.328	
2016-02-15	2888.428	2961.493	2946.707	2886.018	72919432	-17.082	
2016-02-05	2982.056	2988.360	2963.789	2962.847	76850808	-20.971	
2016-02-04	2960.334	2999.206	2984.760	2960.207	91779728	36.121	
2016-02-03	2931.288	2954.710	2948.639	2909.898	77389000	-12.695	
2016-02-02	2901.528	2972.426	2961.334	2901.512	88601960	60.286	

DataFrame: 是pandas包的一种数据结构, 用于编辑和显示量化数据。我们通过DataFrame编辑数据, 来计算各种指标, 显示交易信号和资金曲线。一切围绕DataFrame而进行。

Python,站在巨人的肩膀上

- 通过交互式的代码，我们看到了从数据获取，策略构建，交易信号提取，资金曲线显示等量化研究全过程，让大家对如何在python下做量化有个大概的认识。
- 但是，我们平常用python做量化，除了自己写代码以外会不会有更方便的方法呢？

我们一开始就介绍 python具有高度的拓展性，其实很多代码别人都已经帮我们写好了，并且整合到了成熟的代码框架当中，我们直接使用即可，下面我给大家介绍几个常用的架构。

优矿 (<http://uqer.io>)

优矿是海通量化实验室的一个线上量化研究平台。在上面能够做股票的量化阿尔法策略。通过股票池和选股多因子计算出阿尔法收益。简单，方便，直观。而且社区里面有很多现成的代码可以参考。适合初学者。



社区

高质量开源策略
激发投资灵感

去社区发现



研究

海量金融大数据
专业量化分析平台

开始研究



回测

专注交易算法逻辑
快速获得策略表现评估

回测一下



交易

一键实盘模拟
云端托管，自动交易

模拟交易

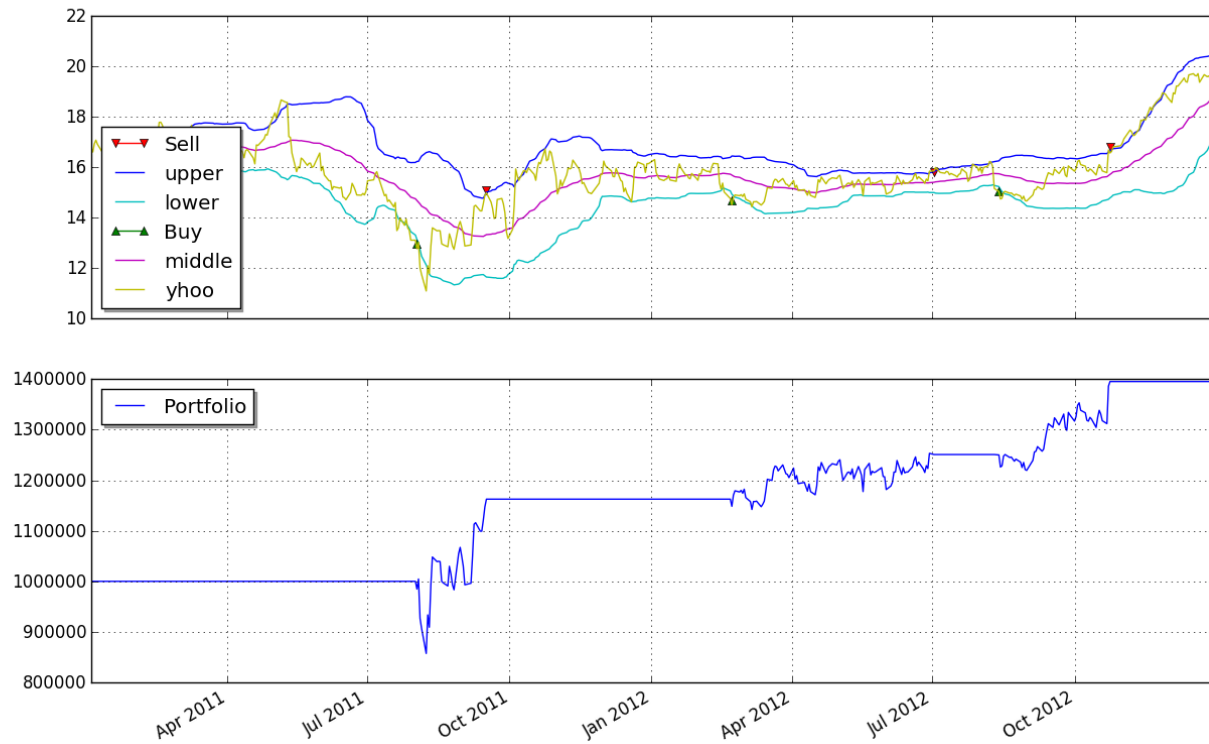
优矿 (http://uqer.io)



pyalgotrade

- 在github上的开发框架，有中文版本pyalgotrade-cn版本。
- 这个框架其实就是类似于把我交互式的代码封装起来再优化一下，中文版的pyalgotrade-cn支持直接通过下载tushare数据，而且回测界面也相对完善，可以看到信号的发生位置。
- 如果你想在本地构建交易系统，可以选择pyalgotrade
- 目前还不能实盘国内市场。可用于回测。

pyalgotrader



VN.PY

- 是国内宽客‘用python的交易员’开发的一套量化系统，在github上面长期维护更新。
- 机构交易员开发量化系统，无论架构跟拓展性都非常优秀，里面包含了CTP模块，研究透了之后可以直接用VNPY进行实盘交易。有很多个人或者机构都在使用。
- 官方网站：<http://vnpy.org>

VN.PY

系列内容安排

整个系列文章目前预计包含的内容如下：

1. 类CTP交易API的工作原理
2. 类CTP交易API的Python封装设计
3. vn.py项目中API封装的编译，其他交易API的封装
4. 事件驱动引擎原理和使用
5. 华宝LTS交易平台开发：登陆、订阅市场行情
6. 华宝LTS交易平台开发：交易功能
7. 华宝LTS交易平台开发：持仓、交易、下单监控

联系作者

本人的知乎号：[用Python的交易员](#)，可以通过知乎私信联系到我，一上。

下一步，学什么？

- 构建自己的一套python实盘交易系统
- 学习网络爬虫，与python交易系统相结合。进行简单的事件驱动交易。
- 学习人工神经网络，数据挖掘等工具，进行人工智能量化交易系统的开发。

- 教程到此结束，希望各位看完教程之后能大概了解python研究量化的思路。我们会在后续的课程当中进行更深入的讲解。
- 对课程有疑问，请联系我：
电话：13925568090
QQ ： 350658395
- 祝愿大家的量化交易系统高收益，低回撤。

附：程序源码

```
• import tushare as ts
• import matplotlib.pyplot as plt
• import numpy as np
• import pandas as pd

•
• spread=3
• hs300=ts.get_hist_data('hs300')
• hs300=hs300.sort_index()
• print hs300.head(10)
• #print hs300.tail(10)
• #hs300.to_csv("path")
• #hs300['close'].plot()
• #hs300['ma10'].plot()
• #hs300['ma20'].plot()
• #plt.show()
• #print type(hs300)

•
• hs300['short']=np.round(pd.rolling_mean(hs300['close'],window=10),2)
• hs300['long']=np.round(pd.rolling_mean(hs300['close'],window=40),2)
• hs300['short-long']=hs300['short']-hs300['long']

•
• hs300['signal']=np.where(hs300['short-long']>spread,1,0)
• hs300['signal']=np.where(hs300['short-long']<spread,-1,hs300['signal'])
• print hs300['signal'].value_counts()

•
• #hs300['signal'].plot(lw=1.5)
• #plt.ylim([-1.1,1.1])
• #plt.show()

•
• hs300['market']=np.log(hs300['close']/hs300['close'].shift(1))
• hs300['straegy']=hs300['signal'].shift(1)*hs300['market']
• print hs300['straegy']
• hs300[['market','straegy']].cumsum().apply(np.exp).plot()
• plt.show()
```