



[返回主题列表](#)

(/community?fromPost=702)

【组合管理】——投资组合理论（有效前沿）

 (/user/d975b9103aa2dae30c383146e6020b0e)

陈小米。
发布于2016-02-25 回复 27 浏览 13986  169
(/user/d975b9103aa2dae30c383146e6020b0e)

分享到： 微信  微博 ()  雪球

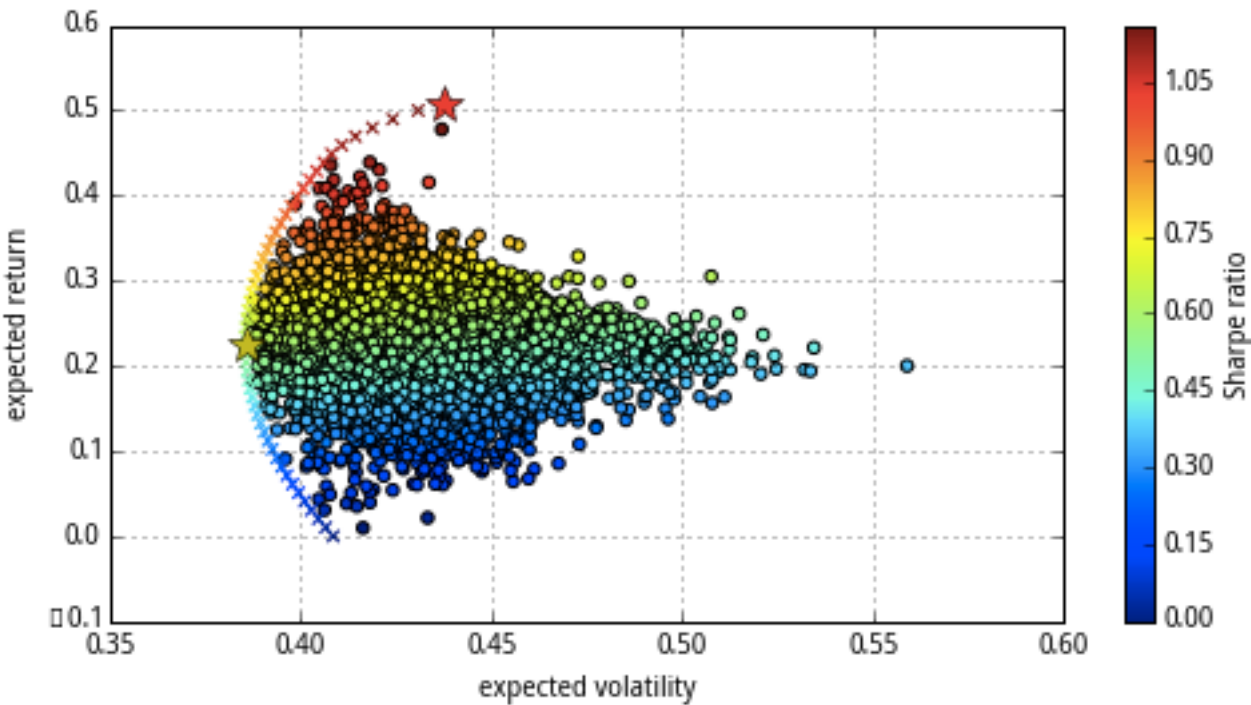
这个算是比较基本的概念了。之前一直思考多股票交易的策略如何分配仓位（weight）。

之前策略的做法都是比较粗糙的等权重买入。

投资组合理论似乎给出了一个比较满意的答案。

- 1. 找到有效前沿。在既定的收益率下使组合的方差最小。
- 2. 找到sharpe最优的组合（收益-风险均衡点）
- 3. 找到风险最小的组合

效果图




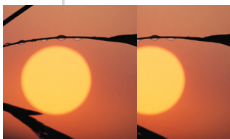
有效前沿在资产配置领域里往后的应用，是结合个人的风险效用曲线来找到合适的投资组合交点。这一步此研究没有涉及。



欢迎关注：**神秘的宽客们**

量化研究，量化学习资料
神策alpha信号每日播报
更多精彩，等你发现

 收起



(/user/d975b9103aa2dae30c383146e6020b0e)
陈小米陈小米ser(d975b9103aa2dae30c383146e6020b0e)

【研究】运用H-60用模
探析资产定价
资金流模型
多因子模型
【研究】量化选股

[查看更多/更多](#)

userId=d975b9103aa2dae30c383146e6020b0e

正态性检验和蒙特卡洛完成投资组合优化

by 陈小米。

最近一直在思考怎样有效的配置资产组合。 很多时候根据条件选好股票池之后， 通常简单粗暴的等分仓位给每只股票。 其实， 这个过程中有很多可以优化的空间。

下面，给大家分享一下如何运用**有效前沿**进行资产组合优化。

PART ONE: 正态性检验

这部分是附赠福利。 只对资产组合优化感兴趣的朋友可以直接跳到**PART TWO**。

1.导入模块

```
In [5]: import pandas as pd
import numpy as np
import statsmodels.api as sm
import scipy.stats as scs
import matplotlib.pyplot as plt
```

2.选取几只感兴趣的股票。

002697 红旗连锁, 600783 鲁信创投, 000413 东旭光电, 601588 北辰实业

```
In [94]: stock = ['002697.XSHE','600783.XSHG','000413.XSHE','601588.XSHG']
start_date = '2015-01-01'
end_date = '2015-12-31'
df = get_price(stock, start_date, end_date, 'daily',['close'])
data = df['close']
data.head()
```

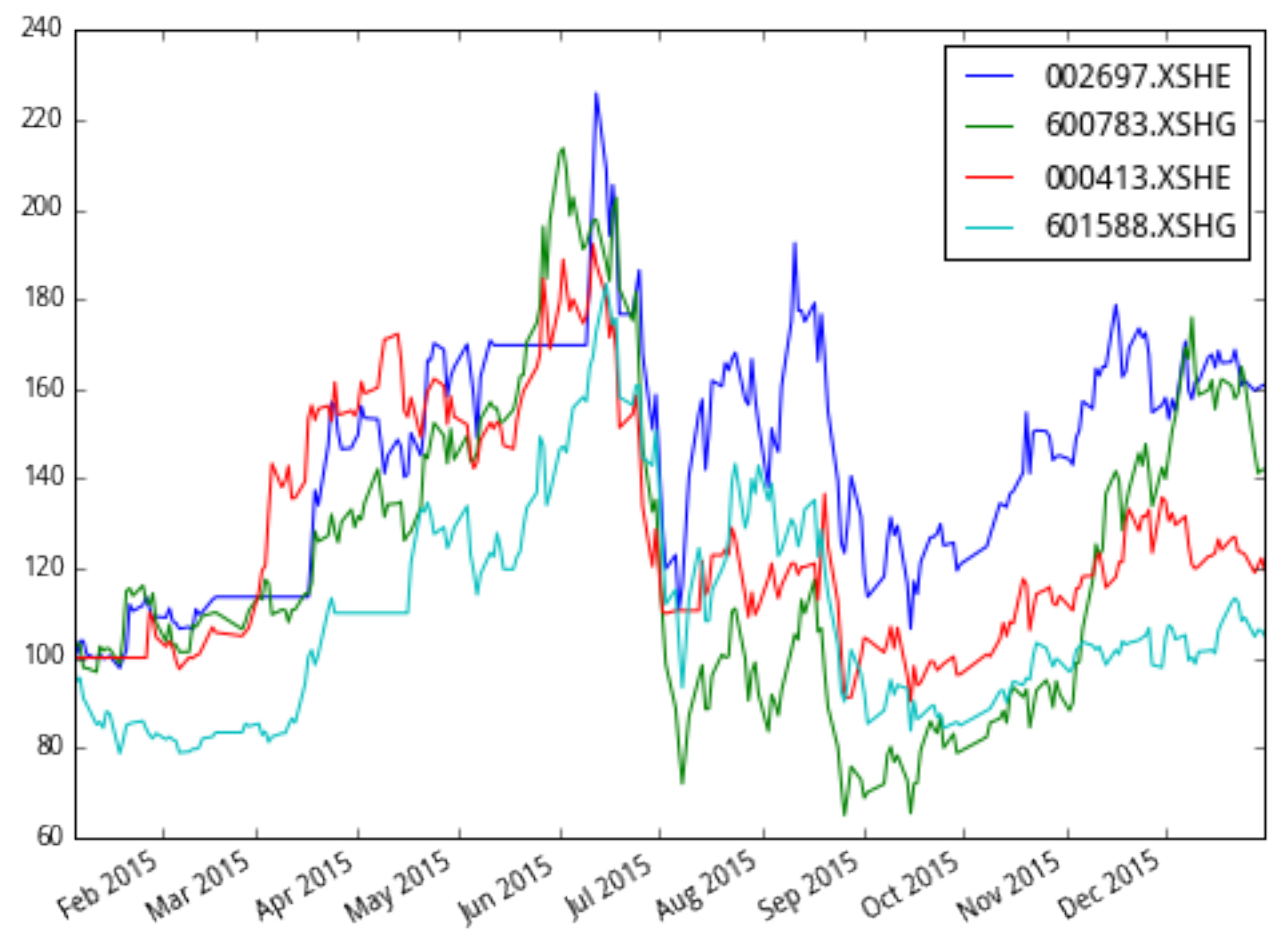
Out[94]:

	002697.XSHE	600783.XSHG	000413.XSHE	601588.XSHG
2015-01-05	4.79	27.53	7.6	5.11
2015-01-06	4.83	28.43	7.6	4.84
2015-01-07	4.97	28.36	7.6	4.88
2015-01-08	4.97	26.93	7.6	4.65
2015-01-09	4.83	26.90	7.6	4.58

3.比较一下机制股票的情况。规范起点为100.

```
In [95]: (data/data.ix[0]*100).plot(figsize = (8,6))
```

Out[95]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd042318990>



4.计算收益率

用pandas计算收益率会比Numpy效率高一些， 可以用shift方法

```
In [96]: log_returns = np.log(data/data.shift(1))
log_returns.head()
```

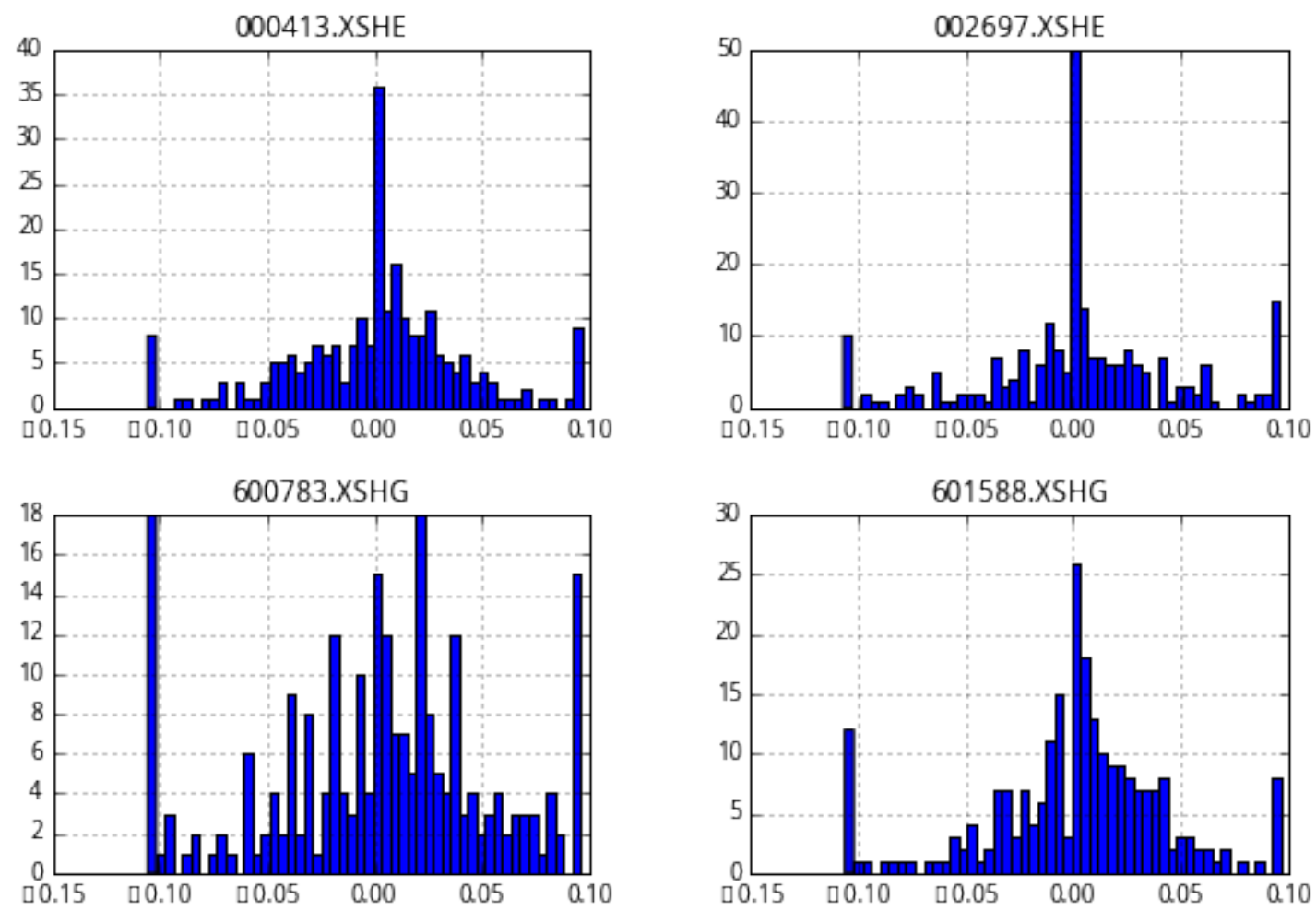
Out[96]:

	002697.XSHE	600783.XSHG	000413.XSHE	601588.XSHG
2015-01-05	0.000000	0.000000	0.000000	0.000000

2015-01-05	NaN	NaN	NaN	NaN
2015-01-06	0.008316	0.032169	0	-0.054285
2015-01-07	0.028573	-0.002465	0	0.008230
2015-01-08	0.000000	-0.051739	0	-0.048278
2015-01-09	-0.028573	-0.001115	0	-0.015168

```
In [97]: log_returns.hist(bins = 50, figsize = (9,6))
```

```
Out[97]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fd0420a5d90>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x7fd04201bc50>],  
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7fd041f9ebd0>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x7fd041f03650>]], dtype=object)
```



从以上结果看到4个数据集都和正态分布的要求相差太多。

5.输出每只股票的统计数据

```
In [98]: #定义print_statistics函数, 为了更加易于理解的方式
#输出给定(历史或者模拟)数据集均值、偏斜度或者峰度等统计数字
def print_statistics(array):
    sta = scs.describe(array)
    print '%14s %15s' %('statistic','value')
    print 30*'- '
    print '%14s %15d' %('size', sta[0])
    print '%14s %15.5f' %('min', sta[1][0])
    print '%14s %15.5f' %('max', sta[1][1])
    print '%14s %15.5f' %('mean', sta[2])
    print '%14s %15.5f' %('std', np.sqrt(sta[3]))
    print '%14s %15.5f' %('skew', sta[4])
    print '%14s %15.5f' %('kurtosis', sta[5])

for st in stock:
    print '\nResults for stock %s' %st
    print 30*'- '
    log_data = np.array(log_returns[st].dropna())
    print_statistics(log_data)
```

Results for stock 002697.XSHE

statistic	value
size	243
min	-0.10697
max	0.09596
mean	0.00196
std	0.04652
skew	-0.13239
kurtosis	0.33900

Results for stock 600783.XSHG

statistic	value
size	243
min	-0.10563
max	0.09554
mean	0.00144
std	0.05250
skew	-0.30149
kurtosis	-0.30247

Results for stock 000413.XSHE

statistic	value
size	243
min	-0.10569
max	0.09651
mean	0.00073
std	0.04084
skew	-0.20718
kurtosis	0.78082

Results for stock 601588.XSHG

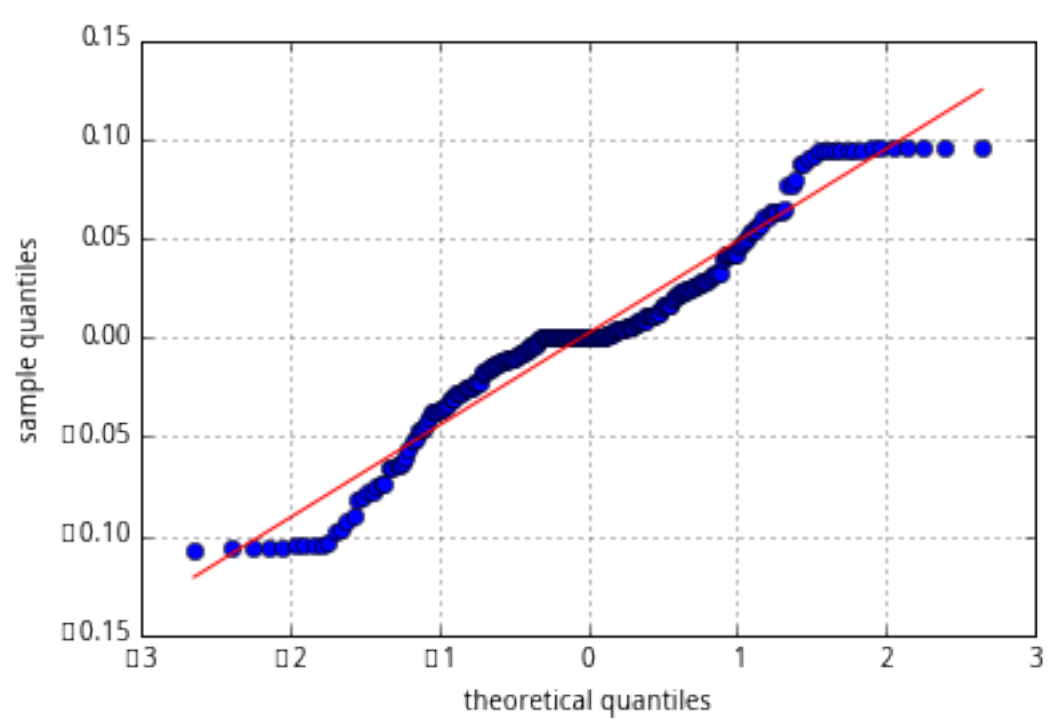
statistic	value
size	243
min	-0.10629
max	0.09660
mean	0.00020
std	0.04250
skew	-0.46498
kurtosis	0.80946

6.画qq图观察数据

下面是002697.XSHE 对数收益率 分位数-分位数图

```
In [99]: sm.qqplot(log_returns['002697.XSHE'].dropna(),line = 's')
plt.grid(True)
plt.xlabel('theoretical quantiles')
plt.ylabel('sample quantiles')
```

Out[99]: <matplotlib.text.Text at 0x7fd041b9ff90>



很显然，样本的分位数值不在一条直线上，表明“非正态性”。左侧和右侧分别有许多值远低于和远高于直线。这是典型的Fat tails。Fat tails是频数分布中观察到的两端的异常值。

7.进行正态性检验

```
In [100]: def normality_test(array):
'''
对给定的数据集进行正态性检验
组合了3中统计学测试
偏度测试 (Skewtest) —足够接近0
峰度测试 (Kurtosistest)—足够接近0
正态性测试
'''

print 'Skew of data set %15.3f' % scs.skew(array)
print 'Skew test p-value %14.3f' % scs.skewtest(array)[1]
print 'Kurt of data set %15.3f' % scs.kurtosis(array)
print 'Kurt test p-value %14.3f' % scs.kurtosistest(array)[1]
print 'Norm test p-value %14.3f' % scs.normaltest(array)[1]

for st in stock:
    print '\nResults for st %s' %st
    print 32*'- '
    log_data = np.array(log_returns[st].dropna())
    normality_test(log_data)
```

Results for st 002697.XSHE

Skew of data set	-0.132
Skew test p-value	0.388
Kurt of data set	0.339
Kurt test p-value	0.233
Norm test p-value	0.338

Results for st 600783.XSHG

Skew of data set	-0.301
Skew test p-value	0.053

```
Kurt of data set      -0.302
Kurt test p-value      0.346
Norm test p-value      0.099

Results for st 000413.XSHE
-----
Skew of data set      -0.207
Skew test p-value      0.179
Kurt of data set      0.781
Kurt test p-value      0.032
Norm test p-value      0.041

Results for st 601588.XSHG
-----
Skew of data set      -0.465
Skew test p-value      0.004
Kurt of data set      0.809
Kurt test p-value      0.028
Norm test p-value      0.001
```

从上述测试的p值来看，否定了数据集呈正态分布的测试假设。这说明，**股票市场收益率的正态假设不成立。**

PART TWO：均值-方差投资组合理论

该理论基于用均值和方差来表述组合的优劣的前提。将选取几只股票，用蒙特卡洛模拟初步探究组合的有效前沿。

通过最大Sharpe和最小方差两种优化来找到最优的资产组合配置权重参数。

最后，刻画出可能的分布，两种最优以及组合的有效前沿。

1.选取几只感兴趣的股票

000413 东旭光电，000063 中兴通讯，002007 华兰生物，000001 平安银行，000002 万科A

并比较一下数据（2015-01-01至2015-12-31）

```
In [102]: stock_set = ['000413.XSHE','000063.XSHE','002007.XSHE','000001.XSHE','000002.XSHE']
noa = len(stock_set)
df = get_price(stock_set, start_date, end_date, 'daily', ['close'])
data = df['close']
#规范化后时序数据
(data/data.ix[0]*100).plot(figsize = (8,5))
```

Out[102]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd041958810>



2.计算不同证券的均值、协方差

每年252个交易日，用每日收益得到年化收益。

计算投资资产的协方差是构建资产组合过程的核心部分。运用pandas内置方法生产协方差矩阵。

```
In [103]: returns = np.log(data / data.shift(1))
returns.mean()*252
```

Out[103]: 000413.XSHE 0.184516
000063.XSHE 0.176790
002007.XSHE 0.309077
000001.XSHE -0.102059
000002.XSHE 0.547441
dtype: float64

```
In [104]: returns.cov()*252
```

Out[104]:

	000413.XSHE	000063.XSHE	002007.XSHE	000001.XSHE	000002.XSHE
000413.XSHE	0.420215	0.206469	0.190350	0.096231	0.081533
000063.XSHE	0.206469	0.381515	0.151234	0.051234	0.041234
002007.XSHE	0.190350	0.151234	0.541234	0.011234	0.021234
000001.XSHE	0.096231	0.051234	0.011234	0.211234	0.031234
000002.XSHE	0.081533	0.041234	0.021234	0.031234	0.411234

000063.XSHE	0.206469	0.373006	0.212459	0.150631	0.144360
002007.XSHE	0.190350	0.212459	0.326912	0.096578	0.075142
000001.XSHE	0.096231	0.150631	0.096578	0.205620	0.138199
000002.XSHE	0.081533	0.144360	0.075142	0.138199	0.231636

3.给不同资产随机分配初始权重

由于A股不允许建立空头头寸，所有的权重系数均在0-1之间

```
In [105]: weights = np.random.random(noa)
          weights /= np.sum(weights)
          weights

Out[105]: array([ 0.37505798,  0.21652754,  0.31590981,  0.06087709,  0.03162758])
```

4.计算预期组合年化收益、组合方差和组合标准差

```
In [106]: np.sum(returns.mean()*weights)*252

Out[106]: 0.21622558669017816

In [107]: np.dot(weights.T, np.dot(returns.cov()*252,weights))

Out[107]: 0.23595133640121463

In [108]: np.sqrt(np.dot(weights.T, np.dot(returns.cov()* 252,weights)))

Out[108]: 0.4857482232609962
```

5.用蒙特卡洛模拟产生大量随机组合

进行到此，我们最想知道的是给定的一个股票池（证券组合）如何找到风险和收益平衡的位置。

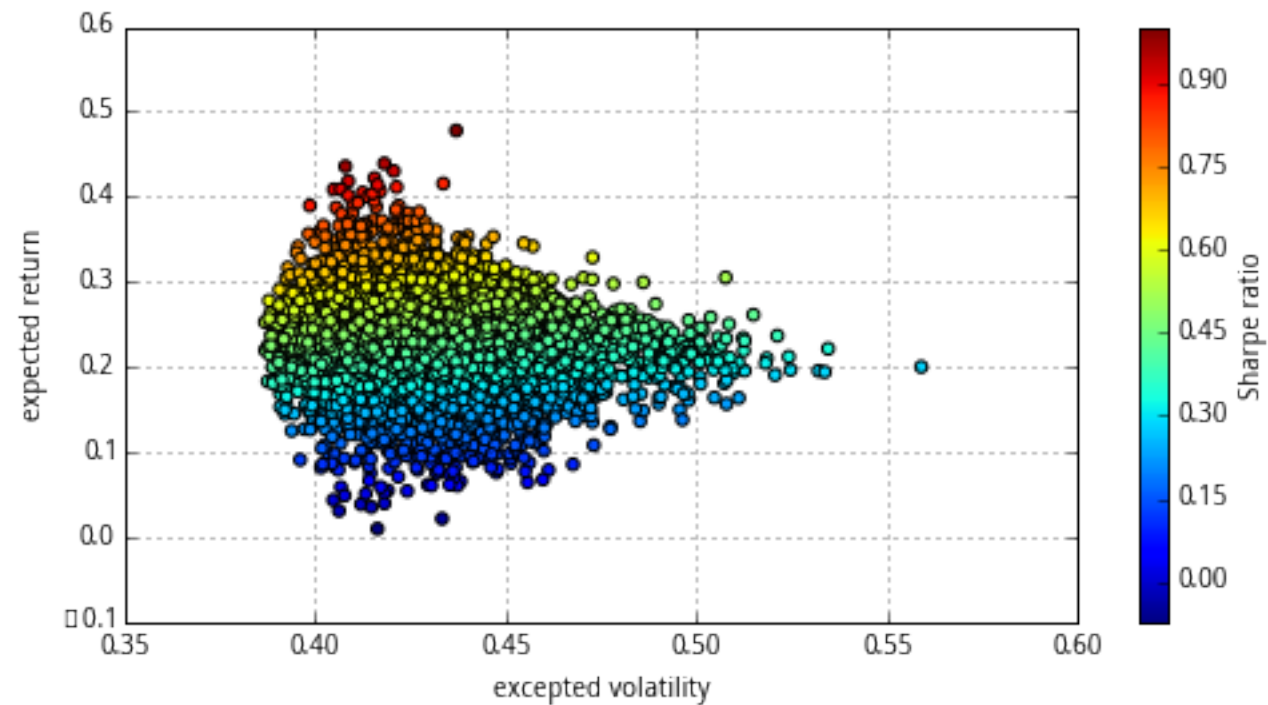
下面通过一次蒙特卡洛模拟，产生大量随机的权重向量，并记录随机组合的预期收益和方差。

```
In [111]: port_returns = []
          port_variance = []
          for p in range(4000):
              weights = np.random.random(noa)
              weights /=np.sum(weights)
              port_returns.append(np.sum(returns.mean()*252*weights))
              port_variance.append(np.sqrt(np.dot(weights.T, np.dot(returns.cov()*252, weights)
              )))

          port_returns = np.array(port_returns)
          port_variance = np.array(port_variance)

          #无风险利率设定为4%
          risk_free = 0.04
          plt.figure(figsize = (8,4))
          plt.scatter(port_variance, port_returns, c=(port_returns-risk_free)/port_variance, ma
          rker = 'o')
          plt.grid(True)
          plt.xlabel('excepted volatility')
          plt.ylabel('expected return')
          plt.colorbar(label = 'Sharpe ratio')

Out[111]: <matplotlib.colorbar.Colorbar instance at 0x7fd04155e638>
```



6.投资组合优化1——sharpe最大

建立statistics函数来记录重要的投资组合统计数据（收益，方差和夏普比）

通过对约市最优问题的求解，得到最优解，其中约市最权重总和为1

通过对约束最优问题的求解，得到最优解。其中约束是权重之和为1。

```
In [115]: def statistics(weights):
    weights = np.array(weights)
    port_returns = np.sum(returns.mean()*weights)*252
    port_variance = np.sqrt(np.dot(weights.T, np.dot(returns.cov()*252,weights)))
    return np.array([port_returns, port_variance, port_returns/port_variance])

#最优投资组合的推导是一个约束最优化问题
import scipy.optimize as sco

#最小化夏普指数的负值
def min_sharpe(weights):
    return -statistics(weights)[2]

#约束是所有参数(权重)的总和为1。这可以用minimize函数的约定表达如下
cons = ({'type':'eq', 'fun':lambda x: np.sum(x)-1})

#我们还将参数值(权重)限制在0和1之间。这些值以多个元组组成的一个元组形式提供给最小化函数
bnds = tuple((0,1) for x in range(noa))

#优化函数调用中忽略的唯一输入是起始参数列表(对权重的初始猜测)。我们简单的使用平均分布。
opts = sco.minimize(min_sharpe, noa*[1./noa,], method = 'SLSQP', bounds = bnds, constraints = cons)
opts
```

```
Out[115]: status: 0
success: True
njev: 4
nfev: 28
fun: -1.1623048291871221
x: array([ -3.60840218e-16,  2.24626781e-16,  1.63619563e-01,
        -2.27085639e-16,  8.36380437e-01])
message: 'Optimization terminated successfully.'
jac: array([ 1.81575805e-01,  5.40387481e-01,  8.18073750e-05,
        1.03137662e+00, -1.60038471e-05,  0.00000000e+00])
nit: 4
```

得到的最优组合权重向量为：

```
In [116]: opts['x'].round(3)
```

```
Out[116]: array([-0.    ,  0.    ,  0.164, -0.    ,  0.836])
```

sharpe最大的组合3个统计数据分别为：

```
In [117]: #预期收益率、预期波动率、最优夏普指数
statistics(opts['x']).round(3)
```

```
Out[117]: array([ 0.508,  0.437,  1.162])
```

7.投资组合优化2——方差最小

接下来，我们通过方差最小来选出最优投资组合。

```
In [118]: #但是我们定义一个函数对 方差进行最小化
def min_variance(weights):
    return statistics(weights)[1]

optv = sco.minimize(min_variance, noa*[1./noa,],method = 'SLSQP', bounds = bnds, constraints = cons)
optv
```

```
Out[118]: status: 0
success: True
njev: 7
nfev: 50
fun: 0.38542969450547221
x: array([ 1.14787640e-01,  3.28089742e-17,  2.09584008e-01,
        3.53487044e-01,  3.22141307e-01])
message: 'Optimization terminated successfully.'
jac: array([ 0.3851725 ,  0.43591119,  0.3861807 ,  0.3849672 ,  0.38553924,  0.
        .
        ])
nit: 7
```

方差最小的最优组合权重向量及组合的统计数据分别为：

```
In [119]: optv['x'].round(3)
```

```
Out[119]: array([ 0.115,  0.    ,  0.21 ,  0.353,  0.322])
```

```
In [120]: #得到的预期收益率、波动率和夏普指数
statistics(optv['x']).round(3)
```

```
Out[120]: array([ 0.226,  0.385,  0.587])
```


8.组合的有效前沿

有效前沿有既定的目标收益率下方差最小的投资组合构成。

在最优化时采用两个约束， 1.给定目标收益率， 2.投资组合权重和为1。

```
In [138]: def min_variance(weights):
          return statistics(weights)[1]

          #在不同目标收益率水平 (target_returns) 循环时，最小化的一个约束条件会变化。
          target_returns = np.linspace(0.0,0.5,50)
          target_variance = []
          for tar in target_returns:
              cons = ({'type':'eq','fun':lambda x:statistics(x)[0]-tar},{ 'type':'eq','fun':lambda x:np.sum(x)-1})
              res = sco.minimize(min_variance, noa*[1./noa,],method = 'SLSQP', bounds = bnds, constraints = cons)
              target_variance.append(res['fun'])

          target_variance = np.array(target_variance)
```

下面是最优化结果的展示。

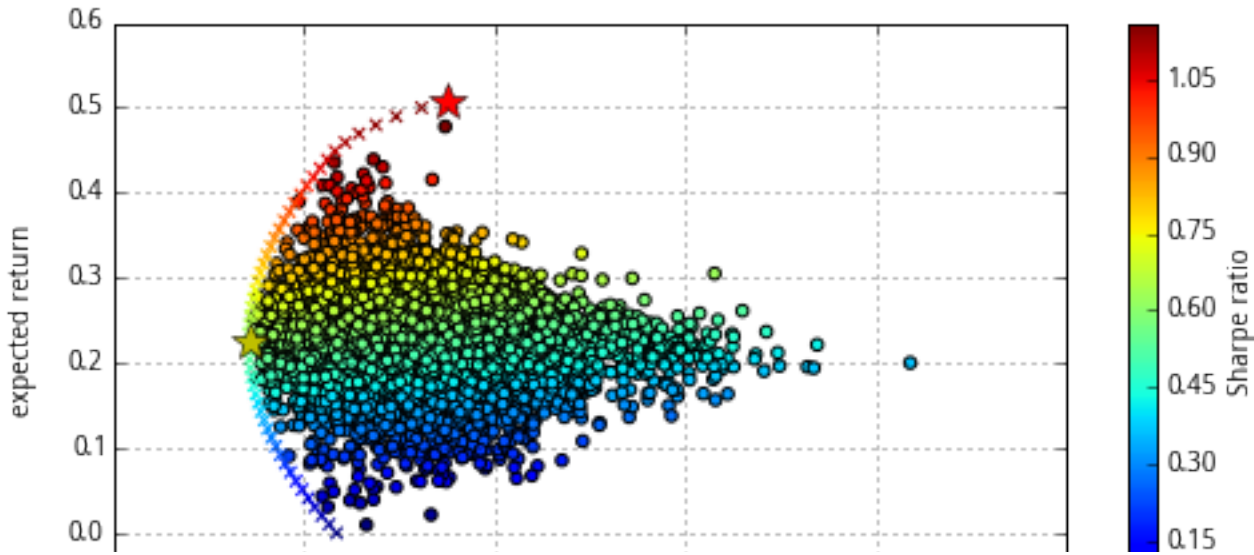
叉号：构成的曲线是有效前沿（目标收益率下最优的投资组合）

红星：sharpe最大的投资组合

黄星：方差最小的投资组合

```
In [139]: plt.figure(figsize = (8,4))
          #圆圈：蒙特卡洛随机产生的组合分布
          plt.scatter(port_variance, port_returns, c = port_returns/port_variance,marker = 'o')
          #叉号：有效前沿
          plt.scatter(target_variance,target_returns, c = target_returns/target_variance, marker = 'x')
          #红星：标记最高sharpe组合
          plt.plot(statistics(opts['x'])[1], statistics(opts['x'])[0], 'r*', markersize = 15.0)
          #黄星：标记最小方差组合
          plt.plot(statistics(optv['x'])[1], statistics(optv['x'])[0], 'y*', markersize = 15.0)
          plt.grid(True)
          plt.xlabel('expected volatility')
          plt.ylabel('expected return')
          plt.colorbar(label = 'Sharpe ratio')
```

Out[139]: <matplotlib.colorbar.Colorbar instance at 0x7fd040d72518>



感谢作者




向导式策略生成器

无需编程，1分钟生成策略

[立即体验](#)

(/algorithm/index/wizard?f=community&m=ad)

评论




(/user/179b1b0056b901e816d968d28fa8e81e)

kuhn (/user/179b1b0056b901e816d968d28fa8e81e)

收藏一下，太专业了，很需要


2016-02-25

 (/user/534bc264d42b91280e38e069102f6c2e)

天大hero (/user/534bc264d42b91280e38e069102f6c2e)

虽然看不懂 感觉好牛叉 膜拜一下


2016-02-25

 (/user/4606cbae1743f0d543aef9f69b51a759)

landmine (/user/4606cbae1743f0d543aef9f69b51a759)

学习~


2016-02-26

 (/user/888188b0dd2449b1975346db142d65e9)

cjhren (/user/888188b0dd2449b1975346db142d65e9)

哈哈。把书本上的知识A股化了。
最近也在看《python 金融大数据分析》


2016-02-28

 (/user/0f914195a39242d21c29934566e18396)

韭菜Hulk (/user/0f914195a39242d21c29934566e18396)

哈哈 这是我之前在现代资产组合的回测结果
<https://www.joinquant.com/post/353> (<https://www.joinquant.com/post/353>)


2016-02-28

 (/user/d975b9103aa2dae30c383146e6020b0e)

陈小米。 (/user/d975b9103aa2dae30c383146e6020b0e)

挺好挺好， 互动补充~


2016-02-28

 (/user/a2f85828d3a8e48ffc61764cdbeff56e)

岚逸 (/user/a2f85828d3a8e48ffc61764cdbeff56e)

干货~太专业实用了， 谢谢无私分享~

2016-02-29

 (/user/4338c519589ea0eb5a5205ad11730ce6)


东极岛 (/user/4338c519589ea0eb5a5205ad11730ce6)

研究中， 请问

```
np.dot(weights.T, np.dot(returns.cov()*252,weights))  
np.sqrt(np.dot(weights.T, np.dot(returns.cov()* 252,weights)))
```

是什么意思

2016-03-22

 (/user/48c6200451d21c4b3fe6093a39fbd9df)

软猫克鲁 (/user/48c6200451d21c4b3fe6093a39fbd9df)

这货太干，需要就水消化消化

2016-03-22



(/user/78c764caa0f970c1467cb3daf042f734)

Hua (/user/78c764caa0f970c1467cb3daf042f734)

楼主，为什么我找到的股票收盘价与你一开头给出的完全不同。000413在1月份几乎没有交易啊

2016-03-22



(/user/d975b9103aa2dae30c383146e6020b0e)

陈小米。 (/user/d975b9103aa2dae30c383146e6020b0e)

请问你是在哪找到的股票价？ BTW, get_price这个api取出的是前复权价。

2016-03-22



(/user/d975b9103aa2dae30c383146e6020b0e)

陈小米。 (/user/d975b9103aa2dae30c383146e6020b0e)

求组合的方差、组合的标准差。
求方差时是收益率序列的协方差矩阵加权出来，并年化（*252）的。

2016-03-22



(/user/78c764caa0f970c1467cb3daf042f734)

Hua (/user/78c764caa0f970c1467cb3daf042f734)

谢楼主回复，我用tushare导的，也是前复权。于是我还特地去查了下东方财富网的股价。差别之大应该不是前后赋权导致的

2016-03-24



(/user/d975b9103aa2dae30c383146e6020b0e)

陈小米。 (/user/d975b9103aa2dae30c383146e6020b0e)

方便把你的结果写在研究里发出来看一下吗？ 有代码的话比较好找原因。

2016-03-25



(/user/78c764caa0f970c1467cb3daf042f734)

Hua (/user/78c764caa0f970c1467cb3daf042f734)

不好意思楼主，我弄错了，没有问题。谢谢

2016-03-25



(/user/a2f85828d3a8e48ffc61764cdbeff56e)

岚逸 (/user/a2f85828d3a8e48ffc61764cdbeff56e)

请问，得出的有效前沿曲线上的某一点对应的组合配比怎么求出？

2016-04-09



(/user/e09346f3f9e5cff35c9c7e548bdbbf782)

星乐 (/user/e09346f3f9e5cff35c9c7e548bdbf782)

@陈小米。请教一下，收益率log_returns = np.log(data/data.shift(1))，为什么求对数而不用差值求取收益率。会对求解结果有影响吗？

2016-04-27



(/user/d975b9103aa2dae30c383146e6020b0e)

陈小米。 (/user/d975b9103aa2dae30c383146e6020b0e)

@星乐 研究中收益率都是用对数收益率，理论上对数收益率才是正态分布。

2016-05-10



(/user/805c0c3a4db4fe3c4c0b096169cc517e)

quying (/user/805c0c3a4db4fe3c4c0b096169cc517e)

这部分只能做股票市场的投资组合管理么？基金市场的无法做么？谢谢

2016-10-21



(/user/0272b27b4f5d607965a363da0cf6cf75)

雨泽 (/user/0272b27b4f5d607965a363da0cf6cf75)

PART 2后面怎么没了

2016-10-27

您尚未登录，请 [登录 \(/user/login/index?redirect=/post/702\)](/user/login/index?redirect=/post/702) 或者 [注册 \(/user/register/index\)](/user/register/index) 聚宽发表回复。

关于

数据

帮助