FBDQA-课程编号-01510313

iCenter
清华大学iCenter

正态性检验

投资组合优化

智能系统实验室

清华大学iCenter

目录

- 1. 正态性检验——股票对数收益率正态性的检验
- 2. 投资组合优化——找到收益-风险之间的平衡点

- 参考：[1] [德] 伊夫·希尔皮斯科（Yves Hilpisch） 著，姚军 译，Python金融大数据分析，人民邮电出版社，2015.
- notebook: 11_Statistics_a.ipynb

1. 正态性检验

- 许多重要的金融模型，如均值-方差投资组合理论，依赖于证券收益呈正态分布的假设

- 本部分介绍若干用于测试给定数据正态性的方法，并测试了模拟数据集和实际数据集的正态性
  - 图形化方法——频率分布直方图、分位数-分位数图
  - 定量方法——偏度、峰度、假设检验法

1. 正态性检验——频率分布直方图

- 画出待检验数据的频率分布直方图，和正态分布进行对比
- 例：股票的对数收益率的分布——模拟的数据

利用Black-Scholes-Merton公式（几何布朗运动）模拟股票指数的变化

$$dS_t = rS_t dt + \sigma S_t dZ_t$$

将其离散化，并随机生成若干条股票指数变化的样本路径

$$S_t = S_{t-\Delta t} \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} z_t\right)$$

r：恒定无风险短期利率

σ：S的恒定波动率（收益标准差）

Z_t：标准布朗运动（$Z_t - Z_s \sim N(0, t-s)$）

Δt：离散化时间间隔

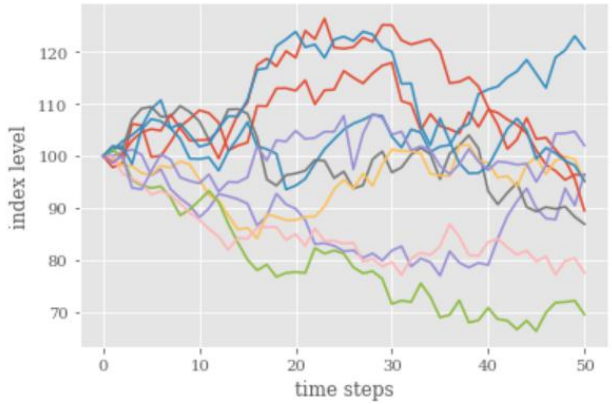
z_t：标准正态分布随机变量

```
S0 = 100.
r = 0.05
sigma = 0.2
T = 1.0
M = 50
I = 250000
```

```
def gen_paths(S0, r, sigma, T, M, I):
    dt = float(T) / M
    paths = np.zeros((M + 1, I), np.float64)
    paths[0] = S0
    for t in range(1, M + 1):
        rand = np.random.standard_normal(I)
        rand = (rand - rand.mean()) / rand.std()
        paths[t] = paths[t - 1] * np.exp((r - 0.5 * sigma ** 2) * dt +
                                         sigma * np.sqrt(dt) * rand)
    return paths
```

```
paths = gen_paths(S0, r, sigma, T, M, I)
```

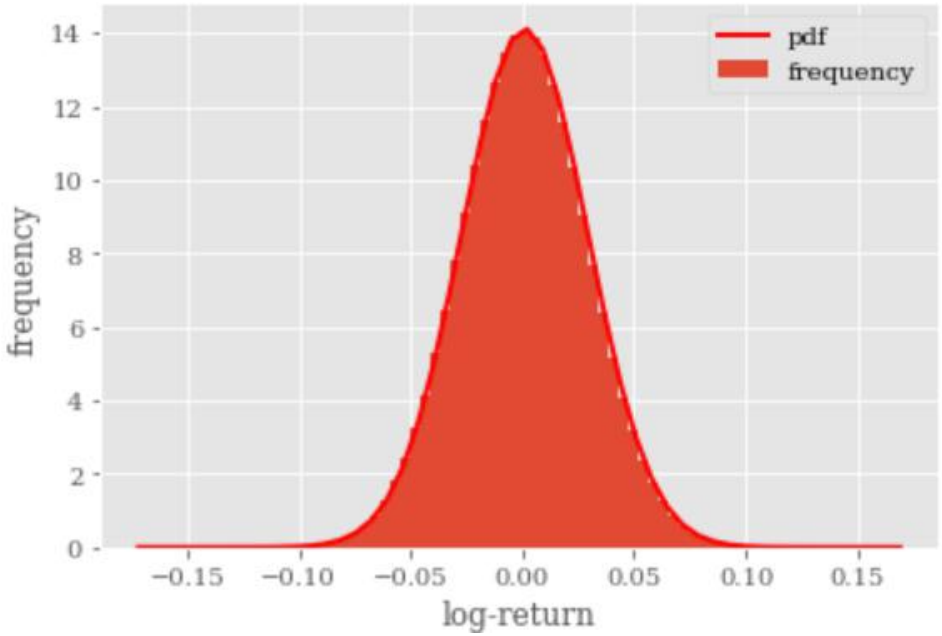
```
plt.plot(paths[:, :10])
```



1. 正态性检验——频率分布直方图

- 计算股票的**对数收益率** $\log \frac{S_{t+\Delta t}}{S_t}$
- 其分布和正态分布接近，说明对数收益率很可能符合正态分布

```
log_returns = np.log(paths[1:] / paths[0:-1])
```




```
plt.hist(log_returns.flatten(), bins=70, normed=True, label='frequency')
plt.grid(True)
plt.xlabel('log-return')
plt.ylabel('frequency')
x = np.linspace(plt.axis()[0], plt.axis()[1])
plt.plot(x, scs.norm.pdf(x, loc=r / M, scale=sigma / np.sqrt(M)),
         'r', lw=2.0, label='pdf')
plt.legend()
```

1. 正态性检验——频率分布直方图

- 计算股票的**对数收益率** $\log \frac{S_{t+\Delta t}}{S_t}$
- 其分布和正态分布接近，说明对数收益率很可能符合正态分布

```
import scipy.stats as scs
def print_statistics(array):
    ''' Prints selected statistics.

    Parameters
    ==========
    array: ndarray
        object to generate statistics on
    '''
    sta = scs.describe(array)
    print("%14s %15s" % ('statistic', 'value'))
    print(30 * "-")
    print("%14s %15.5f" % ('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]))
```

```
print_statistics(log_returns.flatten())
```

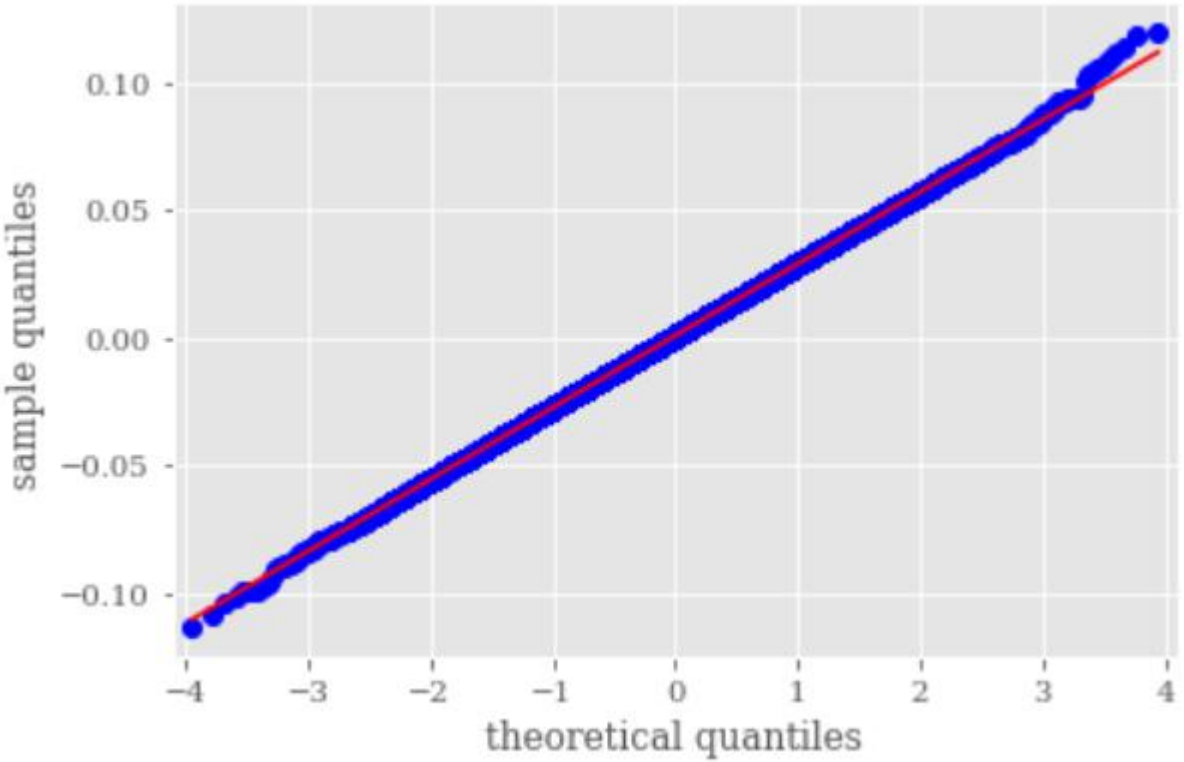
```
     statistic           value
------------------------------
          size  12500000.00000
           min        -0.15664
           max         0.15371
          mean         0.00060
           std         0.02828
          skew         0.00055
      kurtosis         0.00085
```

偏度（Skewness）：　描述的是某总体取值分布的对称性
峰度（Kurtosis）：　描述数据分布顶的尖锐程度

1. 正态性检验——分位数-分位数图

- 分位数-分位数图：quantiles-quantiles

```
import statsmodels.api as sm
sm.qqplot(log_returns.flatten()[::500], line='s')
#数据中每500个点取一个点
```



- 画法
  - 将待画的序列从小到大排列，每个数据对应qqplot上的一个点
  - 该点的纵坐标（smaple quantiles）为该点对应数据的数值
  - 该点的横坐标（theoretical quantiles）：
    - 首先根据该点对应数据在数列中的位置来计算其分位数
    - 然后计算标准正态分布中该分位数对应的随机变量值，该值作为横坐标
  - 直线表示由该列数据的均值和标准差对应的正态分布的分位数图
- **解读方法**
  - 若散点和线很接近，说明数据分布接近正态分布；
  - 若左侧散点低于线，右侧散点高于线，说明数据分布有“厚尾”
  - 若左侧散点高于线，右侧散点低于线，说明数据分布比正态分布更“集中”

1. 正态性检验——定量方法

- 虽然图形方法很直观，但是无法定量说明数据的分布是否正态
- 偏度（skewness）、峰度（kurtosis）

$$Skewness(X) = E((\frac{X-\mu}{\sigma})^3) \qquad Kurtosis(X) = E((\frac{X-\mu}{\sigma})^4) - 3$$

- 正态分布的偏度、峰度均为0，若样本的偏度、峰度接近0，则可以认为数据服从正态分布
- 假设检验法——偏度检验、峰度检验、正态性检验：输出p值，若p值高于0.05，则说明数据服从正态分布（p值意义：拒绝“样本服从正态分布”假设的犯错误概率）

```
import scipy.stats as scs
```

```
def normality_tests(arr):
    print("Skew of data set  %14.3f" % scs.skew(arr))
    print("Skew test p-value %14.3f" % scs.skewtest(arr)[1])
    print("Kurt of data set  %14.3f" % scs.kurtosis(arr))
    print("Kurt test p-value %14.3f" % scs.kurtosistest(arr)[1])
    print("Norm test p-value %14.3f" % scs.normaltest(arr)[1])
```

```
normality_tests(log_returns.flatten())
```

```
Skew of data set          0.001
Skew test p-value         0.430
Kurt of data set          0.001
Kurt test p-value         0.541
Norm test p-value         0.607
```

1. 正态性检验——实际数据

- 接下来看看实际的金融数据是否服从正态分布
- 4列数据：美国标普500ETF、黄金ETF、苹果股票、微软股票

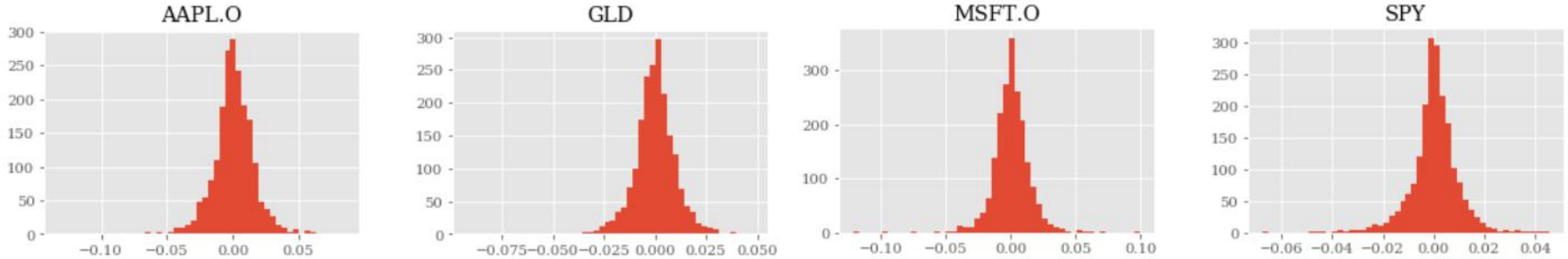
```
import pandas as pd
raw = pd.read_csv('source/tr_eikon_eod_data.csv',
                  index_col=0, parse_dates=True)
symbols = ['SPY', 'GLD', 'AAPL.O', 'MSFT.O']
data = raw[symbols]
data = data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1972 entries, 2010-01-04 to 2017-10-31
Data columns (total 4 columns):
SPY       1972 non-null float64
GLD       1972 non-null float64
AAPL.O    1972 non-null float64
MSFT.O    1972 non-null float64
dtypes: float64(4)
memory usage: 77.0 KB
```

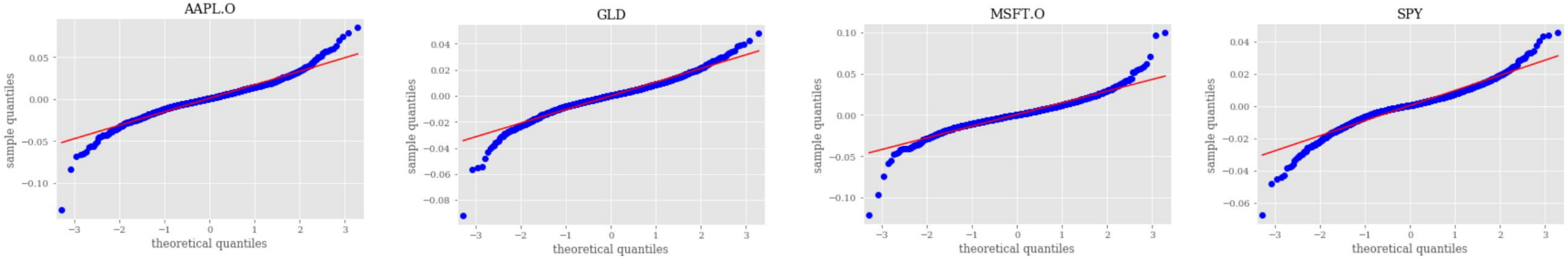
```
(data / data.ix[0] * 100).plot(figsize=(8, 6), grid=True)
```

1. 正态性检验——实际数据

- 分别计算对数收益率 `log_returns = np.log(data / data.shift(1))`
- 画出直方图 `log_returns.hist(bins=50, figsize=(9, 6))`



- 分位数-分位数图：均明显偏离正态分布



1. 正态性检验——实际数据

- 峰度、偏度、假设检验

Results for symbol AAPL.O

Skew of data set	-0.262
Skew test p-value	0.000
Kurt of data set	4.922
Kurt test p-value	0.000
Norm test p-value	0.000

Results for symbol SPY

Skew of data set	-0.469
Skew test p-value	0.000
Kurt of data set	4.543
Kurt test p-value	0.000
Norm test p-value	0.000

Results for symbol GLD

Skew of data set	-0.601
Skew test p-value	0.000
Kurt of data set	5.421
Kurt test p-value	0.000
Norm test p-value	0.000

Results for symbol MSFT.O

Skew of data set	-0.101
Skew test p-value	0.067
Kurt of data set	7.701
Kurt test p-value	0.000
Norm test p-value	0.000

- 4列数据的对数收益率均不符合正态分布，实际金融数据需要用更复杂的模型来建模

目录

- 1. 正态性检验——股票对数收益率正态性的检验
- 2. 投资组合优化——找到收益-风险之间的平衡点
  - 投资组合的收益、风险、夏普比率（第五讲：量化交易策略的检验）
  - 凸优化、插值（第九讲：金融大数据分析中的数学工具）

- 参考：[1] [德] 伊夫·希尔皮斯科（Yves Hilpisch） 著，姚军 译，Python金融大数据分析，人民邮电出版社，2015.
- notebook: 11_Statistics_a.ipynb

目录

- 1. 正态性检验——股票对数收益率正态性的检验
- 2. 投资组合优化——找到收益-风险之间的平衡点
  - 投资组合的收益、风险、夏普比率（第五讲：量化交易策略的检验）
  - 凸优化、插值（第九讲：金融大数据分析中的数学工具）

- 参考：[1] [德] 伊夫·希尔皮斯科（Yves Hilpisch） 著，姚军 译，Python金融大数据分析，人民邮电出版社，2015.
- notebook: 11_Statistics_a.ipynb

2. 投资组合理论

- 现代投资组合理论（Modern Portfolio Theory，MPT）由Markowitz 在1952提出，用数学、统计学方法，代替人们对经验和判断的依赖；

- 对数收益率的正态分布是该理论的基础
  - 只用均值、方差即可完整描述收益的分布
- MPT的基本思路：
  - 通过分散投资，实现投资组合风险最小化或者在指定风险水平下的收益最大化

2. 投资组合理论

- 本部分内容
  - 投资组合的预期收益率（均值）以及方差的计算
  - 有效边界的获取——给定风险水平下取得最大收益的投资组合的集合
  - 在资产中加入无风险投资后，如何配置投资组合

2. 投资组合理论——均值/方差的计算

- 选择5种资产：苹果、微软、亚马逊股票，德国GDX指数，黄金ETF

```
raw = pd.read_csv('source/tr_eikon_eod_data.csv',
                  index_col=0, parse_dates=True)
symbols = ['AAPL.O', 'MSFT.O', 'AMZN.O', 'GDX', 'GLD']
noa = len(symbols)
```

```
(data / data.ix[0] * 100).plot(figsize=(8, 5), grid=True)
```

2. 投资组合理论——均值/方差的计算

- 均值-方差指的是对数收益率的均值、方差

```
rets = np.log(data / data.shift(1))
```

- 设1年有252个交易日，可从每类资产的每日对数收益率计算年化对数收益率

```
rets.mean() * 252
```

```
AAPL.O     0.218633
MSFT.O     0.126401
AMZN.O     0.269869
GDX       -0.096212
GLD        0.012069
dtype: float64
```

- 对数收益率的协方差矩阵（风险以及相关性）

设n维随机变量（n种资产）为$X = (X_1, \ldots, X_n)$，则其协方差矩阵为

$$\Sigma = \begin{bmatrix} Cov(X_1, X_1) & \ldots & Cov(X_1, X_n) \\ \ldots & \ldots & \ldots \\ Cov(X_n, X_1) & \ldots & Cov(X_n, X_n) \end{bmatrix}$$

其中$Cov(X_i, X_j) = E((X_i - EX_i)(X_j - EX_j))$为$X_i$和$X_j$的协方差

```
rets.cov() * 252
```

	AAPL.O	MSFT.O	AMZN.O	GDX	GLD
AAPL.O	0.064899	0.022504	0.026932	0.014669	0.001510
MSFT.O	0.022504	0.050234	0.029146	0.010995	-0.000426
AMZN.O	0.026932	0.029146	0.097792	0.009917	-0.001584
GDX	0.014669	0.010995	0.009917	0.150716	0.048760
GLD	0.001510	-0.000426	-0.001584	0.048760	0.027666

2. 投资组合理论——均值/方差的计算

- 假定不允许建立空头头寸，所有头寸均为多头(正)头寸，且头寸的总和为100%
- 随机生成5种资产的投资权重 $w_i, i = 1, \ldots, 5$， 满足 $\sum_{i=1}^{5} w_i = 1$

```
weights = np.random.random(noa)
weights /= np.sum(weights)
```

```
weights
```

```
array([0.0346395 , 0.02726489, 0.2868883 , 0.10396806, 0.54723926])
```

- 投资组合的预期收益（均值）这样计算

设X_i为i资产的收益率，$\mu_i = E(X_i)$为i资产的预期收益率，那么在固定权重下的预期组合收益为

$$\mu_P = E(\sum_{i=1}^{5} w_i X_i) = \sum_{i=1}^{5} w_i E(X_i) = \sum_{i=1}^{5} w_i \mu_i = w^T \mu$$

其中$w = (w_1, \ldots, w_5)^T$是权重向量，$\mu = (\mu_1, \ldots, \mu_5)^T$是预期收益向量

```
np.sum(rets.mean() * weights) * 252
  # expected portfolio return
```

```
0.08504370198230825
```

2. 投资组合理论——均值/方差的计算

- 投资组合的方差、标准差这样计算

$$\sigma_P^2 = E(\sum_{i=1}^{5} w_i X_i - \sum_{i=1}^{5} w_i \mu_i)^2$$
$$= E(\sum_{i=1}^{5} w_i (X_i - \mu_i))^2$$
$$= \sum_{i=1}^{5}\sum_{j=1}^{5} w_i w_j Cov(X_i, X_j)$$
$$= w^T \Sigma w$$
$$\sigma_P = \sqrt{w^T \Sigma w}$$

```
np.dot(weights.T, np.dot(rets.cov() * 252, weights))
  # expected portfolio variance
```

```
0.024966990290115794
```

```
np.sqrt(np.dot(weights.T, np.dot(rets.cov() * 252, weights)))
  # expected portfolio standard deviation/volatility
```

```
0.15800946266004387
```

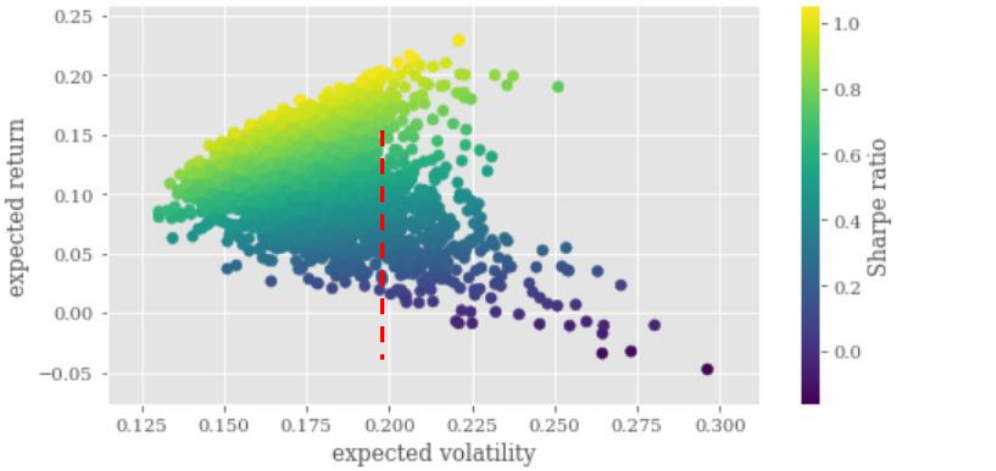
投资组合理论——有效边界的获取

- **有效边界**：给定风险水平下最大化收益的投资组合构成的集合

- 蒙特卡洛模拟，随机生成较大规模的投资组合权重向量w
- 对于每一种模拟的权重，计算其预期投资组合收益μ_P、标准差σ_P、夏普比率$SR = \frac{\mu_P - r_f}{\sigma_P}$，$r_f$为无风险短期利率，设$r_f = 0$

```
prets = []
pvols = []
for p in range (2500):
    weights = np.random.random(noa)
    weights /= np.sum(weights)
    prets.append(np.sum(rets.mean() * weights) * 252)
    pvols.append(np.sqrt(np.dot(weights.T,
                        np.dot(rets.cov() * 252, weights))))
prets = np.array(prets)
pvols = np.array(pvols)
```

```
plt.figure(figsize=(8, 4))
plt.scatter(pvols, prets, c=prets / pvols, marker='o')
```



2. 投资组合理论——有效边界的获取

- 投资组合优化是一个约束最优化问题，可以使用最优化方法来求有效边界。具体可以用scipy.optimize库的minimize函数求解
- 以下求解3类最优化的投资组合：最大化夏普、最小化方差、给定收益率下最小化方差

首先定义一个函数，传入为权重，返回投资组合的平均收益、标准差、夏普

```
def statistics(weights):
    weights = np.array(weights)
    pret = np.sum(rets.mean() * weights) * 252
    pvol = np.sqrt(np.dot(weights.T, np.dot(rets.cov() * 252, weights)))
    return np.array([pret, pvol, pret / pvol])
```

2. 投资组合理论——有效边界的获取

- 1. 最大化夏普比率的投资组合

定义目标函数（需要最小/大化的函数）这里最大化夏普

```
def min_func_sharpe(weights):
    return -statistics(weights)[2]
```

等式约束为所有权重之和为1

```
cons = ({'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
```

优化算法需要设置初始值

```
noa * [1. / noa,]
```

决策变量（权重）上下界约束

```
bnds = tuple((0, 1) for x in range(noa))
```

```
import scipy.optimize as sco
opts = sco.minimize(min_func_sharpe, noa * [1. / noa,], method='SLSQP',
                    bounds=bnds, constraints=cons)
```

```
opts
```

```
     fun: -1.055661702049552
     jac: array([ 4.79370356e-05, -6.97374344e-05,  5.30928373e-05,  8.51
281360e-01,
       -4.90382314e-04])
 message: 'Optimization terminated successfully.'
    nfev: 49
     nit: 7
    njev: 7
  status: 0
 success: True
       x: array([0.47403302, 0.050359  , 0.39381482, 0.        , 0.081793
16])
```

最优权重下的结果，预期收益率约为21.7%，预期波动率约为20.6%，最优夏普比率为1.06

```
statistics(opts['x']).round(3)
```

```
array([0.217, 0.206, 1.056])
```

2. 投资组合理论——有效边界的获取

- 2. 最小化方差的投资组合

将最小化方差作为优化目标

```
def min_func_variance(weights):
    return statistics(weights)[1] ** 2
```

等式约束为所有权重之和为1

```
cons = ({'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
```

优化算法需要设置初始值

```
noa * [1. / noa,]
```

决策变量（权重）上下界约束

```
bnds = tuple((0, 1) for x in range(noa))
```

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

```
optv
```

```
     fun: 0.015978878561032903
     jac: array([0.03154033, 0.03221243, 0.03183019, 0.06642421, 0.03195841])
 message: 'Optimization terminated successfully.'
    nfev: 70
     nit: 10
    njev: 10
  status: 0
 success: True
       x: array([0.1203471 , 0.23091281, 0.07015531, 0.        , 0.57858478])
```

最优权重下的结果，预期收益率约为8.1%，最优波动率约为12.6%，夏普比率为0.644

```
statistics(optv['x']).round(3)
```

```
array([0.081, 0.126, 0.644])
```

2. 投资组合理论——有效边界的获取

- 3. 给定收益率时最小化方差的投资组合——最优投资组合
- x——最优投资组合；红星——最优夏普比投资组合；黄星——最小方差投资组合
- **有效边界**——给定风险水平下最大化预期收益的投资组合构成的集合
- 等价于由所有收益率高于绝对最小方差投资组合的最优投资组合构成的集合

定义目标函数（需要最小/大化的函数）这里最小化标准差

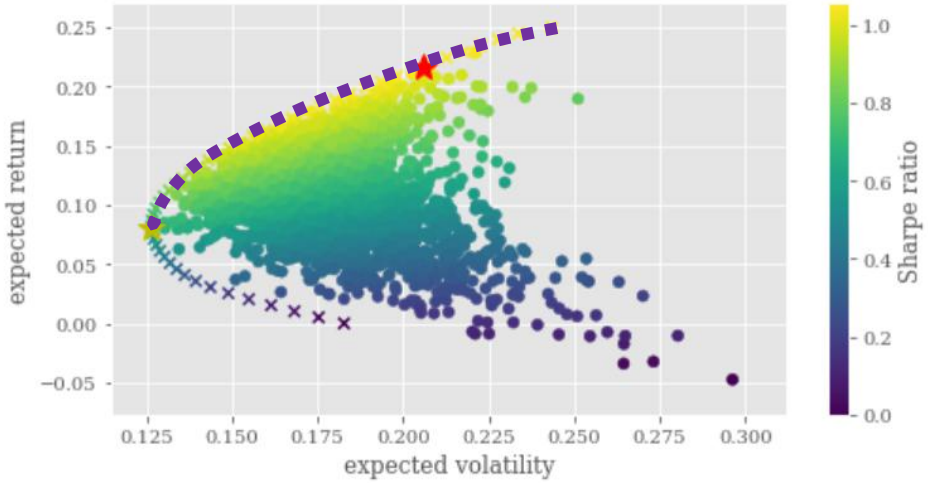
```
def min_func_port(weights):
    return statistics(weights)[1]
```

求收益率为0~0.25时的最优投资组合（波动率最小）

```
import scipy.optimize as sco
trets = np.linspace(0.0, 0.25, 50)
tvols = []
for tret in trets:
    cons = ({'type': 'eq', 'fun': lambda x:  statistics(x)[0] - tret},
            {'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
    res = sco.minimize(min_func_port, noa * [1. / noa,], method='SLSQP',
                       bounds=bnds, constraints=cons)
    tvols.append(res['fun'])
tvols = np.array(tvols)
```

约束包括：1. 收益率为给定值；2. 权重之和为1；3. 权重的上下界

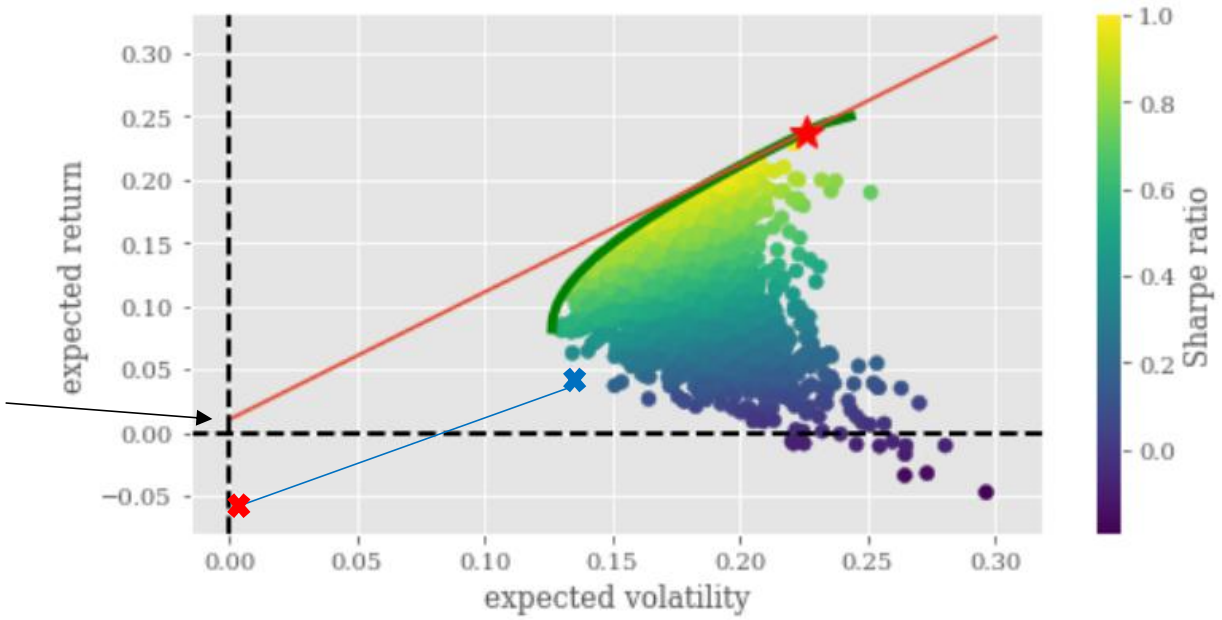
```
cons = ({'type': 'eq', 'fun': lambda x:  statistics(x)[0] - tret},
        {'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
bnds = tuple((0, 1) for x in weights)
```



2. 投资组合理论——无风险投资的加入

- 无风险投资组合配置：无风险收益与投资组合的连线
- 选哪一个有效投资组合？——代表该投资组合的点处，有效边界的切线恰好通过无风险投资组合的风险-收益点（设无风险利率$r_f = 0.01$）
- 资本市场线：无风险资产和最高夏普比率的连线

无风险投资组合的风险-收益点



无风险投资的加入——资本市场线

- 方法一——插值+最优化：

首先找到有效边界（之前求得的点）

```
ind = np.argmin(tvols)
evols = tvols[ind:]
erets = trets[ind:]
```

进行3次样条插值，并求其1阶导数

```
import scipy.interpolate as sci
tck = sci.splrep(evols, erets)
```

```
def f(x):
    ''' Efficient frontier function (splines approximation). '''
    return sci.splev(x, tck, der=0)
def df(x):
    ''' First derivative of efficient frontier function. '''
    return sci.splev(x, tck, der=1)
```

定义目标函数（需要最小/大化的函数）这里最小化标准差

```
def min_func_port(weights):
    return statistics(weights)[1]
```

设切点为(σ, r)，则资本-市场线$t(x) = a + bx$符合以下数学条件

$t(0) = a = r_f$

$t(\sigma) = a + b\sigma = f(\sigma)$

$t'(\sigma) = b = f'(\sigma)$

需要求解(a, b, σ)

```
def equations(p, rf=0.01):
    eq1 = rf - p[0]
    eq2 = rf + p[1] * p[2] - f(p[2])
    eq3 = p[1] - df(p[2])
    return eq1, eq2, eq3
```

利用fsolve函数求解该方程组，需要给定(a, b, σ)的初始值

```
opt = sco.fsolve(equations, [0.01, 0.5, 0.15])
opt
```

```
array([0.01      , 1.0090905 , 0.22552992])
```

```
cons = ({'type': 'eq', 'fun': lambda x:  statistics(x)[0] - f(opt[2])},
        {'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
res = sco.minimize(min_func_port, noa * [1. / noa,], method='SLSQP',
                   bounds=bnds, constraints=cons)
```

```
res['x'].round(3)
```

```
array([0.525, 0.025, 0.443, 0.   , 0.007])
```

想一想

- 资本市场线还可以如何得到?
- 提示:

$$\mu = r_f + S_M \cdot \sigma.$$

  - S_M： 最大夏普比率

设切点为(σ, r)，则资本-市场线$t(x) = a + bx$符合以下数学条件

$t(0) = a = r_f$

$t(\sigma) = a + b\sigma = f(\sigma)$

$t'(\sigma) = b = f'(\sigma)$

需要求解(a, b, σ)

投资组合理论——无风险投资的加入

- 方法二

$SR = \frac{\mu_P - 0.01}{\sigma_P}$达到最大值

```
def min_func_sharpe(weights):
    return -(statistics(weights)[0]-0.01)/statistics(weights)[1]
cons = ({'type': 'eq', 'fun': lambda x:  np.sum(x) - 1})
res = sco.minimize(min_func_sharpe, noa * [1. / noa,], method='SLSQP',
                   bounds=bnds, constraints=cons)
```

```
res['x'].round(3)
```

```
array([0.522, 0.026, 0.438, 0.   , 0.013])
```

```
res['x'].round(3)
```

```
array([0.525, 0.025, 0.443, 0.   , 0.007])
```

总结

- 正态性检验
  - 很多金融模型要求股票对数收益率呈正态分布，因此需要对实际金融数据进行正态性检验——图形化方法（频率分布直方图、分位数-分位数图）、定量方法（偏度、峰度、假设检验）
- 投资组合优化
  - 通过调整不同资产的投资比例，实现收益-风险之间的平衡

谢谢指正！