

第五篇

金融大数据实战



实训 24

最优投资组合（上）

24.1 实训目的

- ☐ 了解金融相关知识和最优风险资产组合、有效边界。
- ☐ 使用 Python 编写实训代码，利用 Matplotlib 绘制效果图。

24.2 实训要求

- ☐ 掌握 Pandas、NumPy、Matplotlib、xlrd、fix_yahoo_finance 等模块的使用方法。
- ☐ 掌握 Python 读取网络股票信息的方法。
- ☐ 了解最优风险资产组合、有效边界。
- ☐ 掌握利用 Matplotlib 绘制数据图的方法。
- ☐ 本实训需要外网连接。

24.3 实训原理

最优投资组合是指某投资者在可以得到的各种可能的投资组合中，唯一可获得最大效用期望值的投资组合。有效集的上凸性和无差异曲线的下凹性决定了最优组合的唯一性，有效边界是在收益-风险约束条件下能够以最小的风险取得最大的收益的各种证券的集合。

24.4 实训步骤

24.4.1 导入实训需要的模块

代码如下。

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import sys
from pandas_datareader import data as pdr
import yfinance as yf
yf.pdr_override() # 需要调用这个函数
import matplotlib.pyplot as plt
import tushare as ts
import scipy.optimize as sco
import scipy.interpolate as sci
from scipy import stats
import sys
import xlrd
import openpyxl
```

24.4.2 读取数据

数据源是从 Yahoo 得到的苹果、谷歌、亚马逊、微软、甲骨文、高通、百度、新浪、惠普在 2018-01-01 到 2018-07-31 期间的股票数据。

```
s=["AAPL", "GOOG", "AMZN", "MSFT", "ORCL", "QCOM", "BIDU", "SINA", "HPQ"]
px=pd.DataFrame({n: pdr.get_data_yahoo(n, start="2018-01-01", end="2018-07-31")["close"]
for n in s})
```

24.4.3 观察缺失值

如果某支股票某天有缺失值，则去掉这个日期的所有股票信息，这样会让所有股票的日期保持一致，代码如下。

```
px=px.dropna()
print(px.info())
```

结果如图 24-1 所示。

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 145 entries, 2018-01-02 to 2018-07-30
Data columns (total 9 columns):
AAPL      145 non-null float64
GOOG      145 non-null float64
AMZN      145 non-null float64
MSFT      145 non-null float64
ORCL      145 non-null float64
QCOM      145 non-null float64
BIDU      145 non-null float64
SINA      145 non-null float64
HPQ       145 non-null float64
dtypes: float64(9)
memory usage: 11.3 KB
None

```

图 24-1 缺失值情况

24.4.4 数据可视化

如果由于网络原因而无法获得数据，可以直接读取数据，代码如下。

```

# 利用 Pandas 读取 9 支股票的 Excel 数据
data= pd.read_excel('/root/dataset/最优投资组合.xlsx')
data.index = data['Date'].tolist() # 将 date 列复制一份，设为 index 列
data.pop('Date') # 移除 date 列，因为 date 列已经成为 index 列，以后只需看 index 即可
# 将个股价格与其初始值比较并且用 100 标准化，得出个股在相同初始条件下的走势情况
(data / data.ix[0] * 100).plot(figsize=(10, 8), grid=True)
fig = plt.gcf()

```

结果如图 24-2 所示。

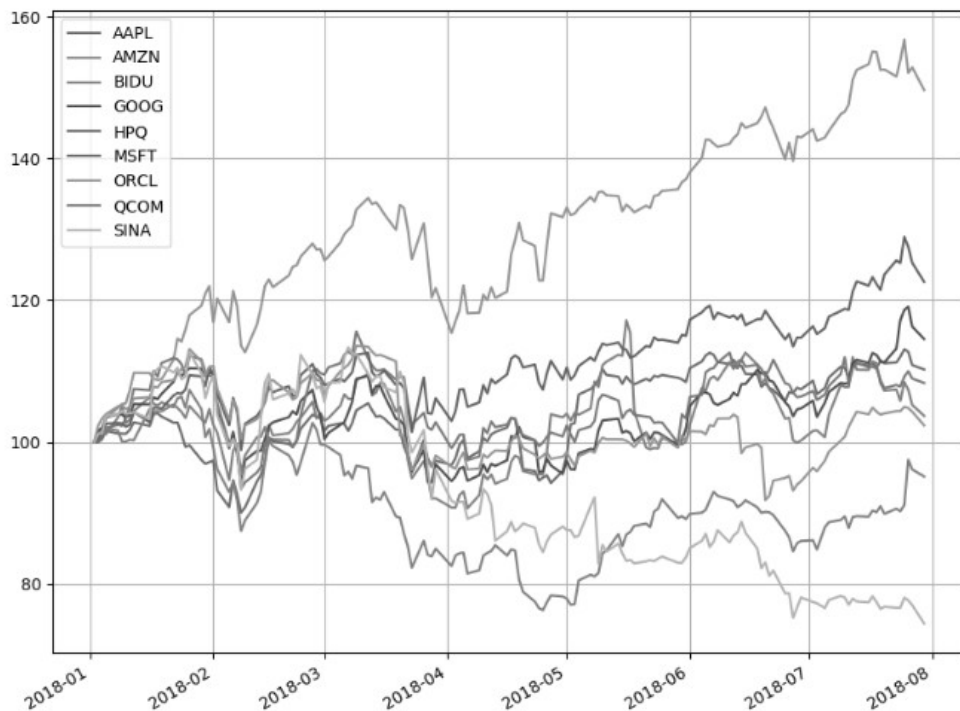


图 24-2 股票走势情况图

24.4.5 初步统计分析

现在需要计算个股的收益率。金融计算收益率时大部分使用对数收益率（log return）而不是用算数收益率。用当天的收盘价与前一天相比较。代码如下。

```
# 计算对数收益率
log_returns = np.log(data / data.shift(1))
log_returns.hist(bins=50, figsize=(12, 9))
fig = plt.gcf()# 获取当前图片
plt.savefig('2.png')
```

结果如图 24-3 所示。

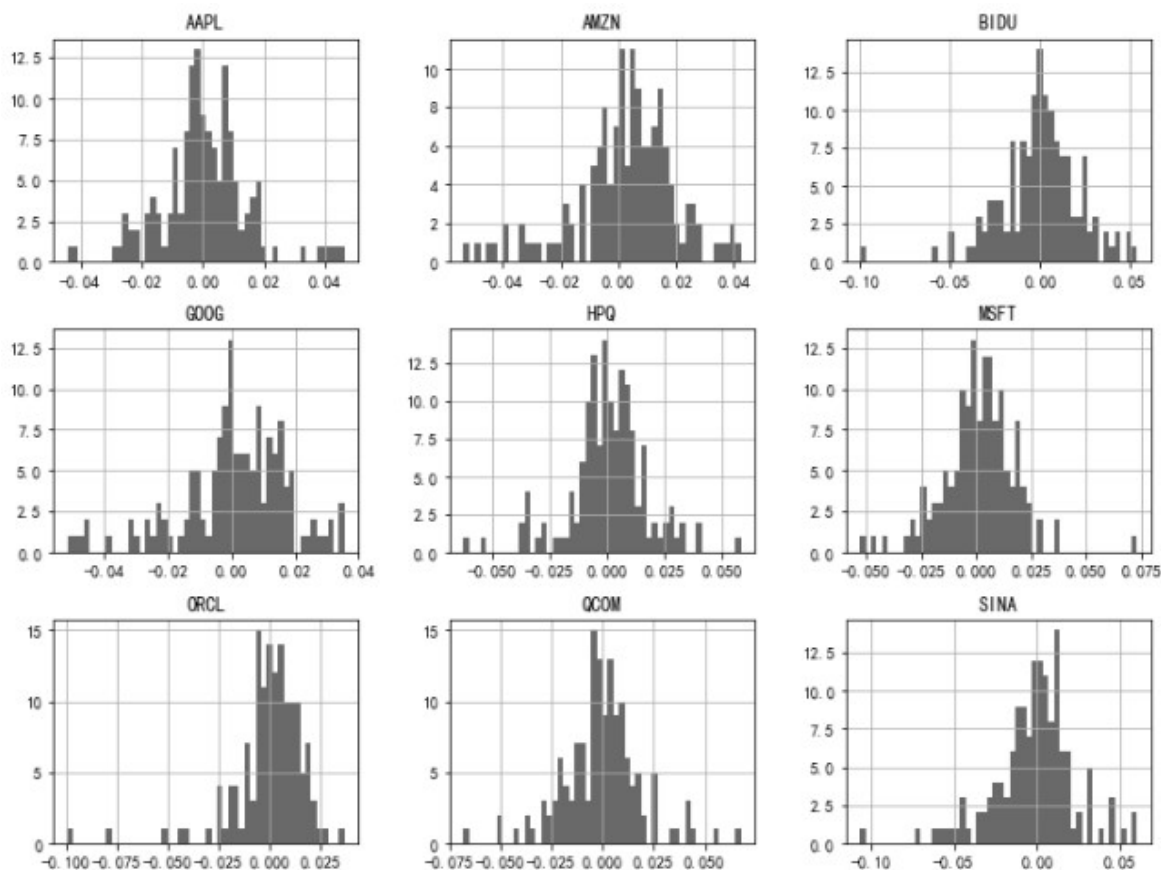


图 24-3 计算收益率

可以看到每支股票的分布形状与正态分布较为类似。

24.4.6 投资组合优化

Markowitz 均值-方差投资组合理论需要假设正态分布收益率，而投资组合的风险取决于投资各组合中资产收益率的相关性。基于此，年化收益率和协方差矩阵就是我们需要的内容。代码如下。

```
# 使用对数收益率
rets = log_returns
# 计算年化收益率
year_ret = rets.mean() * 252
# 计算协方差矩阵
year_volatility = rets.cov() * 252

print(year_ret)
print(year_volatility)
```

结果如图 24-4 所示。

AAPL	0.170704
GOOG	0.237410
AMZN	0.705345
MSFT	0.356497
ORCL	0.040803
QCOM	-0.086940
BIDU	0.063798
SINA	-0.517108
HPQ	0.140327
dtype: float64	

图 24-4 年化收益率

24.4.7 计算组合均值收益率

根据金融投资相关理论，风险需要分散，每个股票都会有一定比例的投资权重。一个资产组合的收益率（均值）为组合中个股收益率（均值）的权重之和，组合均值收益率如图 24-5 所示。

	AAPL	GOOG	AMZN	MSFT	ORCL	QCOM	BIDU
AAPL	0.053196	0.040882	0.029860	0.039739	0.030070	0.038931	0.040511
GOOG	0.040882	0.072982	0.047174	0.056969	0.039722	0.046215	0.051358
AMZN	0.029860	0.047174	0.076375	0.049296	0.024109	0.034821	0.052378
MSFT	0.039739	0.056969	0.049296	0.066933	0.038260	0.046153	0.048991
ORCL	0.030070	0.039722	0.024109	0.038260	0.074030	0.034689	0.034763
QCOM	0.038931	0.046215	0.034821	0.046153	0.034689	0.094533	0.043442
BIDU	0.040511	0.051358	0.052378	0.048991	0.034763	0.043442	0.121545
SINA	0.039337	0.053110	0.050851	0.056828	0.032723	0.039460	0.067032
HPQ	0.035714	0.047952	0.037136	0.045776	0.036813	0.049078	0.049120
	SINA	HPQ					
AAPL	0.039337	0.035714					
GOOG	0.053110	0.047952					
AMZN	0.050851	0.037136					
MSFT	0.056828	0.045776					
ORCL	0.032723	0.036813					
QCOM	0.039460	0.049078					
BIDU	0.067032	0.049120					
SINA	0.149023	0.049974					
HPQ	0.049974	0.072325					

图 24-5 组合均值收益率

1. 模拟权重分配

使用随机数来随机模拟权重分配，并将模拟权重分配给 9 个股票，代码如下。

```
# 一共有 9 支股票
number_of_assets = 9
# 生成 9 个随机数
weights = np.random.random(number_of_assets)
# 将 9 个随机数归一化，每一份就是一个权重，权重之和为 1
weights /= np.sum(weights)
```

假设有 6000 种投资组合，每一种投资组合由一组随机权重组成，这样可以产生 6000 组投资组合。代码如下。

```
portfolio_returns = []
portfolio_volatilities = []
for p in range(6000):

    weights = np.random.random(number_of_assets)
    weights /= np.sum(weights)
    portfolio_returns.append(np.sum(rets.mean() * weights) * 252)
    portfolio_volatilities.append(np.sqrt(np.dot(weights.T, np.dot(rets.cov() * 252, weights))))
```

2. 收益率和波动率可视化

可以得到 6000 个组合收益率和波动率，代码如下。

```
portfolio_returns = np.array(portfolio_returns)
portfolio_volatilities = np.array(portfolio_volatilities)
plt.figure(figsize=(9, 5)) # 作图大小
plt.scatter(portfolio_volatilities, portfolio_returns, c=portfolio_returns / portfolio_volatilities,
            marker='o') # 画散点图
plt.grid(True)
plt.xlabel('expected volatility')
plt.ylabel('expected return')
plt.colorbar(label='Sharpe ratio')
fig = plt.gcf() # 获取当前图片
plt.savefig('3.png')
```

24.5 实训结果

实训结果如图 24-6 所示。

每个点对应某个投资组合，该点有其对应的收益率和波动率（标准差），其颜色对应不同的夏普率。可见，越往左上方，夏普率越高。

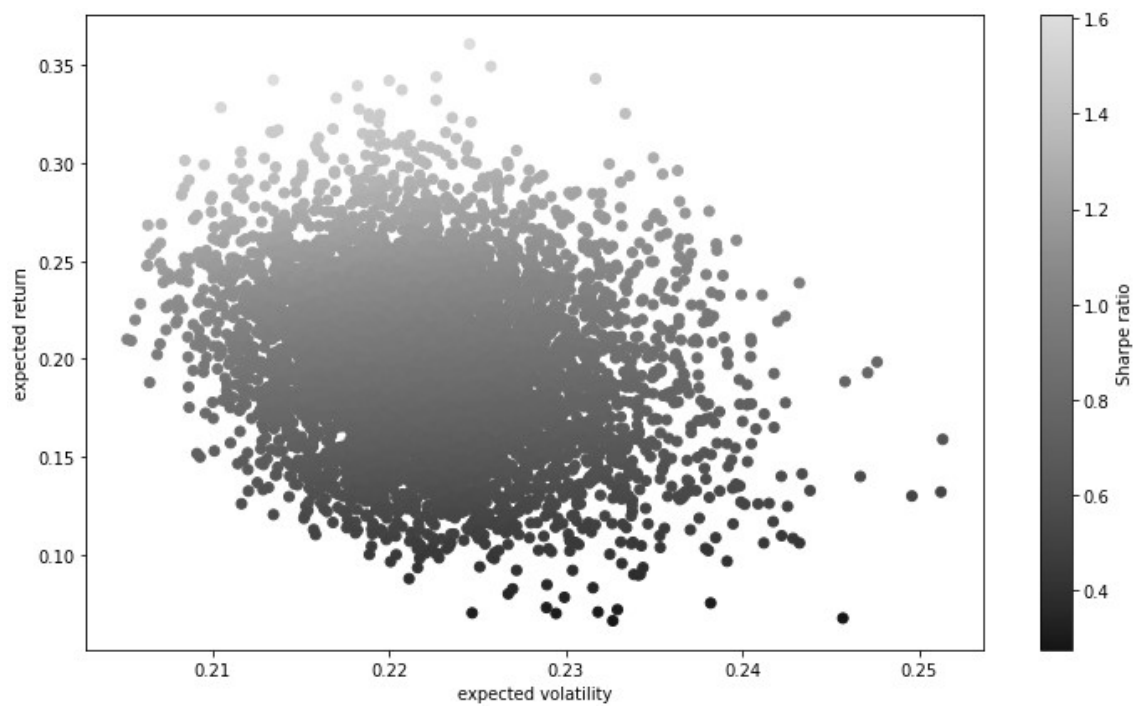


图 24-6 6000 个组合收益率、波动率和夏普率散点图