项目结构

```
.
├── alpha_test
│   ├── alpha_test_GRS.txt
│   └── alpha_test.txt
├── beta
│   └── beta.txt
==├── Binary_Tree.py==
├── compare
│   ├── compare_Markowitz.txt
│   ├── compare_MontoCarlo_alpha0.txt
│   └── compare_MontoCarlo.txt
==├── config.py==
==├── draw_tree_picture.py==
==├── Homework1.pdf==
==├── hw1_1.py==
==├── hw1_2.py==
==├── hw1_3.py==
==├── HW1.xlsx==
├── images
│   ├── HS300与Markowitz投资组合收益比较：20150105--20150630.png
│   ├── HS300与Markowitz投资组合收益比较：20150105--20191230.png
│   ├── HS300与Markowitz投资组合收益比较：20150701--20151231.png
│   ├── HS300与Markowitz投资组合收益比较：20160104--20160630.png
│   ├── HS300与Markowitz投资组合收益比较：20160701--20161230.png
│   ├── HS300与Markowitz投资组合收益比较：20170103--20170630.png
│   ├── HS300与Markowitz投资组合收益比较：20170703--20171229.png
│   ├── HS300与Markowitz投资组合收益比较：20180102--20180629.png
│   ├── HS300与Markowitz投资组合收益比较：20180702--20181228.png
│   ├── HS300与Markowitz投资组合收益比较：20190102--20190628.png
│   ├── HS300与Markowitz投资组合收益比较：20190701--20191230.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20150105--20150630.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20150105--20191230.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20150701--20151231.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20160104--20160630.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20160701--20161230.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20170103--20170630.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20170703--20171229.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20180102--20180629.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20180702--20181228.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20190102--20190628.png
│   ├── HS300与MontoCarlo_alpha0投资组合收益比较：20190701--20191230.png
│   ├── HS300与MontoCarlo投资组合收益比较：20150105--20150630.png
│   ├── HS300与MontoCarlo投资组合收益比较：20150105--20191230.png
│   ├── HS300与MontoCarlo投资组合收益比较：20150701--20151231.png
│   ├── HS300与MontoCarlo投资组合收益比较：20160104--20160630.png
│   ├── HS300与MontoCarlo投资组合收益比较：20160701--20161230.png
│   ├── HS300与MontoCarlo投资组合收益比较：20170103--20170630.png
│   ├── HS300与MontoCarlo投资组合收益比较：20170703--20171229.png
│   ├── HS300与MontoCarlo投资组合收益比较：20180102--20180629.png
```

```
|   ├── HS300与MontoCarlo投资组合收益比较：20180702--20181228.png
|   ├── HS300与MontoCarlo投资组合收益比较：20190102--20190628.png
|   ├── HS300与MontoCarlo投资组合收益比较：20190701--20191230.png
|   ├── Montacarlo_CAL_50000_20100104_20141231.png
|   ├── Montacarlo_CAL_50000_20100701_20150630.png
|   ├── Montacarlo_CAL_50000_20110104_20151231.png
|   ├── Montacarlo_CAL_50000_20110701_20160630.png
|   ├── Montacarlo_CAL_50000_20120104_20161230.png
|   ├── Montacarlo_CAL_50000_20120702_20170630.png
|   ├── Montacarlo_CAL_50000_20130104_20171229.png
|   ├── Montacarlo_CAL_50000_20130701_20180629.png
|   ├── Montacarlo_CAL_50000_20140102_20181228.png
|   └── Montacarlo_CAL_50000_20140701_20190628.png
├── option_result
|   ├── binarytree_american_put_100_step.txt
|   ├── binarytree_european_put_100_step.txt
|   ├── binarytree_european_put_10_step
|   ├── binarytree_european_put_10_step.pdf
|   ├── binarytree_european_put_10_step.png
|   ├── binarytree_european_put_10_step.txt
|   ├── binarytree_european_put_50_step.txt
|   └── black_scholes_put.txt
├── project_structure.txt
├── __pycache__
|   ├── Binary_Tree.cpython-36.pyc
|   ├── config.cpython-36.pyc
|   ├── draw_tree_picture.cpython-36.pyc
|   └── hw1_1.cpython-36.pyc
==├── requirements.txt==
└── weights
    ├── weights_Markowitz.pickle
    ├── weights_Markowitz.txt
    ├── weights_MontoCarlo.pickle
    └── weights_MontoCarlo.txt

7 directories, 75 files
```

其中，这些文件是主要文件，config.py是三道题都使用的全局变量文件，HW1.xlsx是前两题使用的数据文件，第一题使用hw1_1.py，第二题使用hw1_1.py和hw1_2.py，第三题使用hw1_3.py、Binary_Tree.py、draw_tree_picture.py（后两个用来画二叉树）

```
pip install -r requirements.txt
```

ubuntu环境下，第三小题使用graphviz需要安装：

```
sudo apt-get install graphviz
```

ubuntu 解决matplotlib中文问题参考：https://www.huuinn.com/archives/533

编程体验

```
.4978242989897134e-05
开始计算组合权重，采用策略：  Markowitz投资组合
进入20110104--20151231权重计算:   0%|完成Markowitz投资组合最优权重二次规划求解，方差最优值为：          | 0/10 [00:00<?, ?it/s]  5
.38988221372181e-05
开始计算组合权重，采用策略：  Markowitz投资组合
进入20110701--20160630权重计算:  30%|███        | 3/10 [00:00<00:00, 28.92it/s] 完成Markowitz投资组合最优权重二次规划求解，方差最优值为：
6.509486052818258e-05
开始计算组合权重，采用策略：  Markowitz投资组合
完成Markowitz投资组合最优权重二次规划求解，方差最优值为：  7.149380252646544e-05
进入20120104--20161230权重计算:  30%|███        | 3/10 [00:00<00:00, 28.92it/s] 开始计算组合权重，采用策略：  Markowitz投资组合
进入20120702--20170630权重计算:  30%|███        完成Markowitz投资组合最优权重二次规划求解，方差最优值为：| 3/10 [00:00<00:00, 28.92it/s]
6.950183840337214e-05
```

第一题

1. 缺失值填补：

股票有停牌等原因会导致存在缺失值，对此使用停牌前一个交易日数据进行填补，而第32只股票从题目给定日期的第一天就有缺失值，因此在使用从前向后填补方法之后，通过从后向前方式对其填补。

```
df_raw = df_raw.fillna(method='ffill')
# 第32只股票第一天就是空缺值，用向前填补方式
df = df_raw.fillna(method='backfill')
```

2. 六个月调整投资组合：

对此两种做法，第一种比较简单使用180天为单位做切片，但与题意符合度较差（180个交易日还是和6个月有区别的，6个月有多少交易日也并非固定）。因此本次作业使用月份，即：20150105--20150630、20150701--20151231等，通过时间处理，找到1月和7月的第个交易日进行切片，详情可见 `get_six_month_map` 方法：

```
def get_six_month_map(x_matrix):
```

3. 计算日收益率及日平均收益（用于估计每只股票日期望收益）：

计算公式如公式(1)所示。其中，n代表：每六个月的天数-1，向量 $\vec{r_t}$ 是50维的，每一维度代表一只股票t日收益率

$$\vec{r_t} = \frac{\vec{P_t} - \vec{P_{t-1}}}{\vec{P_{t-1}}}$$

$$\vec{r} = \frac{1}{n}\sum_{t=1}^{n}\vec{r_t} \tag{1}$$

注：也可使用 $r_t = log(\frac{P_t}{P_{t-1}})$ ，两者在 r_t 十分小的是等价无穷小，本次作业使用的是前者，具体可见 `day_yield_compute` 和 `ex_vector_compute` 方法

```
def day_yield_compute(x_matrix):
def ex_vector_compute(x_matrix):
```

4. 协方差矩阵计算：

计算公式如公式(2)所示。其中，协方差矩阵采用无偏估计，n代表：每六个月的天数-1，50代表50支股票，具体可见 `ex_matrix_compute` 和 `cov_matrix_compute` 方法

$$
\begin{aligned}
\Sigma &= E((X-EX)^T(X-EX)) \\
&= \frac{1}{n-1}((X-EX)^T(X-EX))
\end{aligned} \tag{2}
$$

$$
(X-EX)_{n\times 50} = \begin{pmatrix} x_{1,1}-Ex_1 & x_{2,1}-Ex_2 & \cdots & x_{50,1}-Ex_{50} \\ x_{1,2}-Ex_1 & x_{2,2}-Ex_2 & \cdots & x_{50,2}-Ex_{50} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,n}-Ex_1 & x_{2,n}-Ex_2 & \cdots & x_{50,n}-Ex_{50} \end{pmatrix}
$$

```
def ex_matrix_compute(x_matrix, ex_numpy_vector):
def cov_matrix_compute(x_ex_matrix):
```

5. 计算权重：

计算权重有三种方法，相关权重全部都以 *txt* 和 *pickle* 两种格式保存在了weights文件夹中，以下展示Markowitz方法的第一期权重值：

```
[-2.02568126e-02, -1.38108078e-02,  5.55970704e-03, -3.58866925e-02,
         3.86067279e-02,  1.16323980e-01,  1.37263795e-01,  3.62936151e-02,
         8.87808623e-04,  5.30490470e-02, -3.76936523e-02,  2.09016878e-02,
        -2.59852855e-02, -9.22626661e-03, -5.17699372e-02, -2.23733266e-02,
        -8.95150083e-02, -2.04227368e-04,  7.57061094e-02, -2.48383110e-02,
         4.81555255e-03, -1.36931086e-03,  1.76689396e-01,  6.71820232e-03,
        -4.08508405e-02, -4.22821580e-02,  5.18201389e-02,  6.33594037e-02,
        -3.79909286e-02, -7.31253016e-02,  7.69239369e-02, -4.50168414e-02,
        -2.09913657e-02,  3.91885791e-02, -2.16612218e-02, -7.02912166e-04,
         3.18577429e-01, -1.38971134e-02,  7.30523654e-02,  6.70617328e-02,
        -3.20086913e-02,  2.12171342e-02,  3.16190233e-04, -9.34385397e-02,
         6.07676988e-02, -2.80197963e-02,  2.99263338e-01,  8.57752558e-03,
         6.24068580e-02, -3.24326105e-02]
```

具体可见 `save_weights_montocarlo` 和 `save_weights_markowitz` 方法：

```
def save_weights_montocarlo(self):
def save_weights_markowitz(self):
```

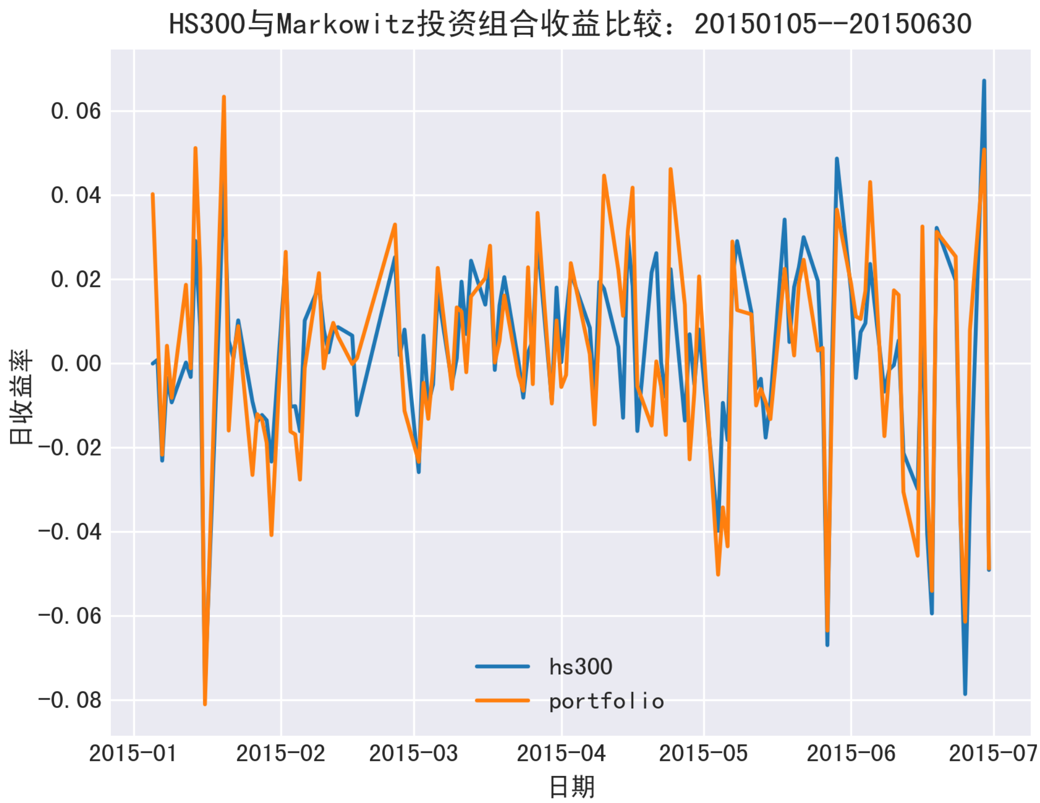
5.1 Markowitz投资组合方法

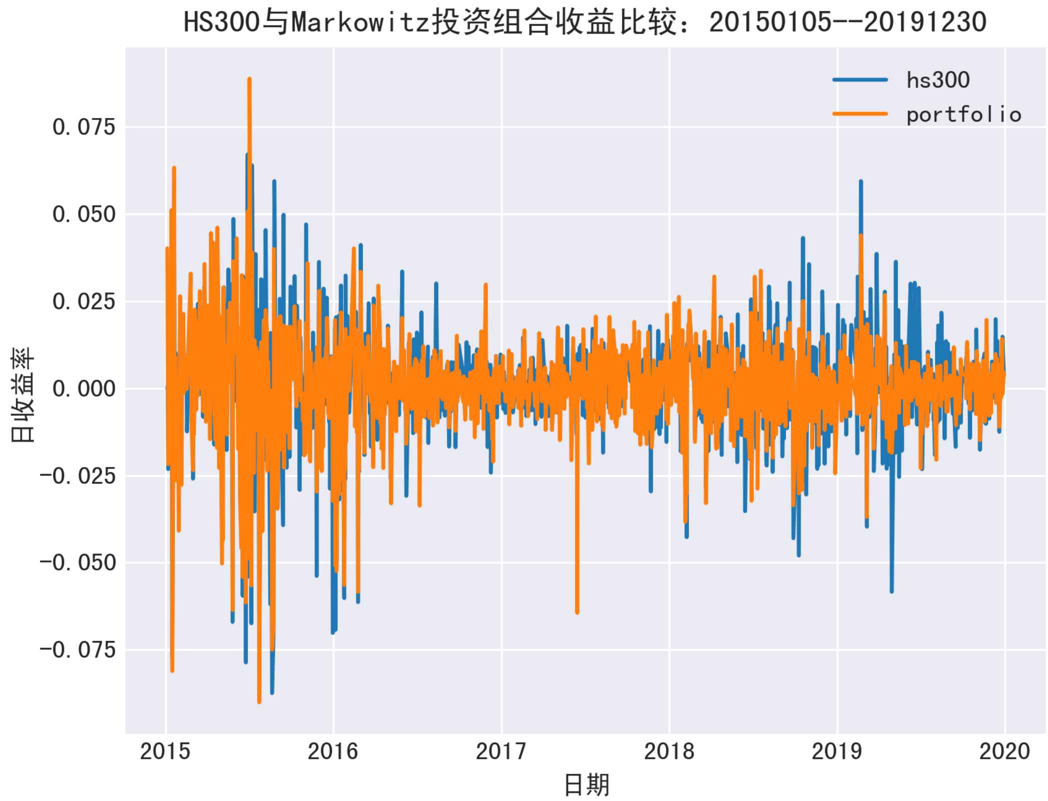
由于Markowitz投资组合理论没有用到无风险利率，因此这种方法并不会用到3%的无风险利率，而 r_{target} 是题中给出的10%期望目标收益，该方法求解如下二次规划问题（题中可以shorting，w可以为负），相关向量和矩阵符号与公式(1)、(2)一致。可通过 **cvxpy** 或 **cvxopt** 两个包实现求解，具体可见 `compute_weight` 方法：

$$
\begin{aligned}
\min_{\vec{w}} \quad & \frac{1}{2}\vec{w}^T \Sigma \vec{w} \\
\text{s.t.} \quad & \vec{w}^T \vec{r} = r_{target} \\
& \vec{1}^T \vec{w} = 1
\end{aligned}
$$

```
def compute_weight(self, x_matrix, total_days=252, method="Markowitz",
starttime=0, endtime=0):
```

该策略与HS300表现比较（仅以第一期20150105--20150630和总投资期20150105--20191230为例，更多结果请见images文件夹）：





从图中可看出，日收益率来看，Markowitz投资组合方法与HS300差不多，但是从2015-2019时间跨度看，日收益率波动情况，橙色portfolio线基本都在蓝色线hs300内部，也就是说Markowitz投资组合方法确实降低了投资组合风险。

计算每期平均收益比较如下表，具体可见 **compare** 文件夹，通过表可看到总体HS300胜出

	开始时间	结束时间	HS300平均日收益	Portfolio平均日收益	win
	20150105	20150630	0.00157	0.001972	Portfolio
	20150701	20151231	-0.00126	-0.00145	HS300
	20160104	20160630	-0.00064	-0.00058	Portfolio
	20160701	20161230	0.000496	8.93E-05	HS300
	20170103	20170630	0.000758	0.000372	HS300
	20170703	20171229	0.000929	0.000961	Portfolio
	20180102	20180629	-0.00146	-0.00056	Portfolio
	20180702	20181228	-0.001	-0.00035	Portfolio
	20190102	20190628	0.002513	0.000695	HS300
	20190701	20191230	0.000354	-0.00035	HS300

	开始时间	结束时间	HS300平均日收益	Portfolio平均日收益	win
全部平均:	20150105	20191230	0.000217	7.30E-05	HS300

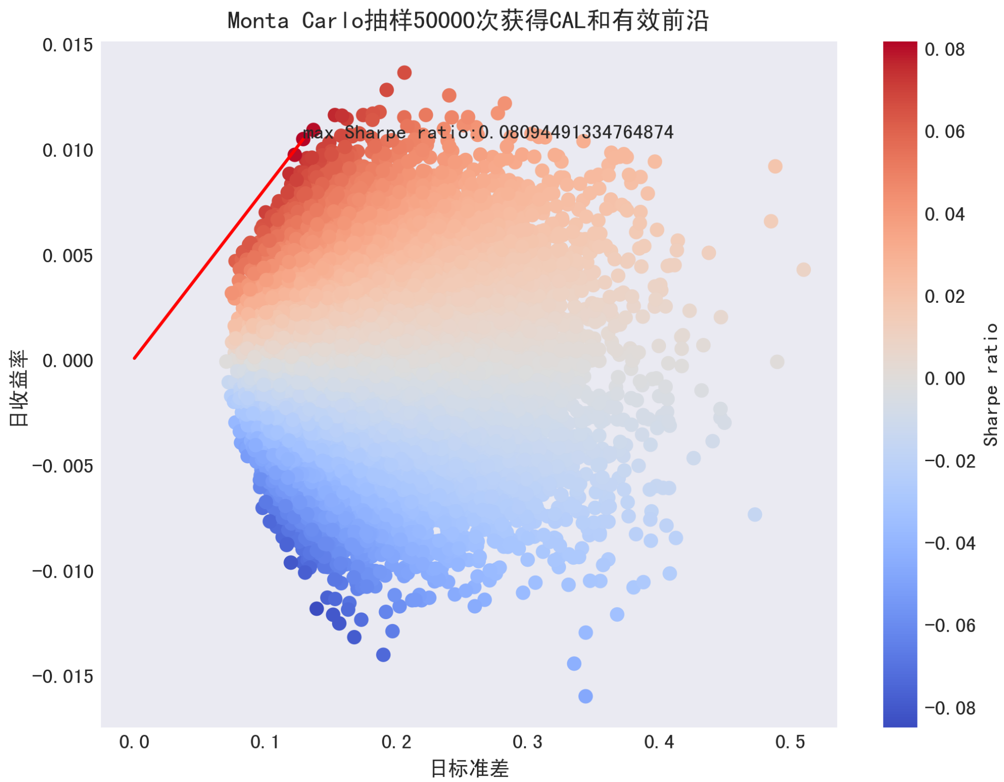
5.2 Monto Carlo方法

这个方法通过求解下式最优化问题获取权重，由于分母有w的二次项，目前只能通过蒙特卡洛数值方法逼近最优解。具体抽样方法为：从 $N(1/50, 1)$ 中随机抽取49个权重，最后一个权重通过1减去前49个之和得到。

需要注意的是这里的 $r_{f_{day}}$ 不再是3%，因为 $\bar{r_p}$，σ_p 都是日度单位，此处采用 平均每年天数=5年交易日总天数/5，无风险日利率=3%/平均每年天数。获取最优市场组合权重之后，通过结合无风险日利率制作资本市场线，

$$
\begin{aligned}
\max_{\vec{w}} \quad & Sharpe\ ratio = tan\theta = \frac{\bar{r_p} - r_{f_{day}}}{\sigma_p} \\
\text{s.t.} \quad & \vec{1}^T \vec{w} = 1 \\
& \bar{r_p} = \vec{w}^T \vec{r} \\
& \sigma_p = \sqrt{\vec{w}^T \Sigma \vec{w}}
\end{aligned}
$$

相关资本市场线和有效前沿（仅以第一期20100104_20141231为例，更多结果请见images文件夹）：



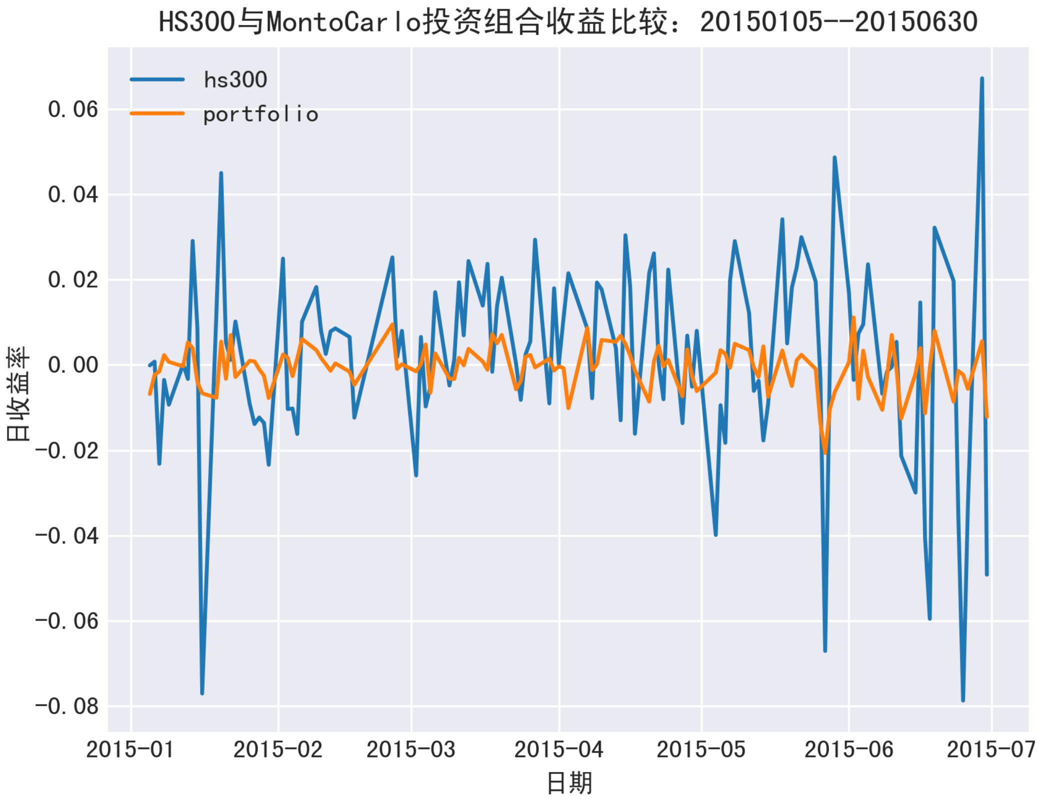
得到市场组合权重w之后，再通过下式解得无风险资产投资权重 α：

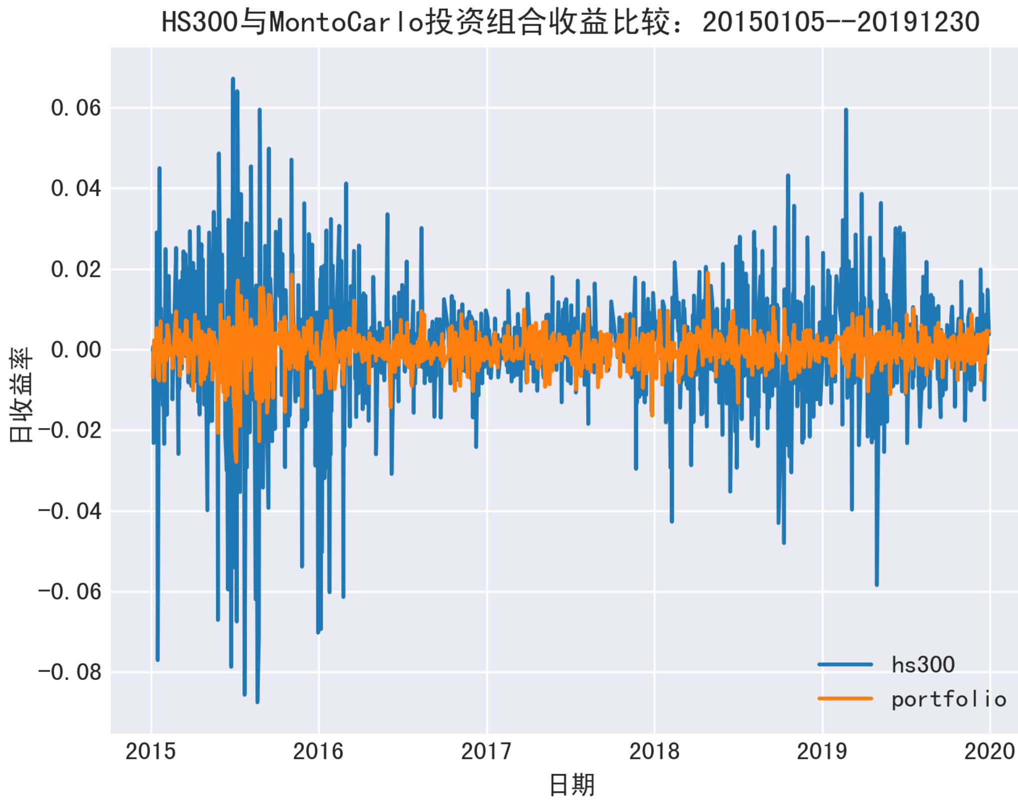
$$\alpha r_{f_{day}} + (1-\alpha)\bar{r_p} = r_{target_{day}}$$

同样地，$r_{target_{day}}$ 也不能用10%，计算方法同无风险日利率。具体可见 `compute_weight` 方法：

```
def compute_weight(self, x_matrix, total_days=252, method="Markowitz",
starttime=0, endtime=0):
```

该策略与HS300表现比较（仅以第一期20150105--20150630和总投资期20150105--20191230为例，更多结果请见images文件夹）：





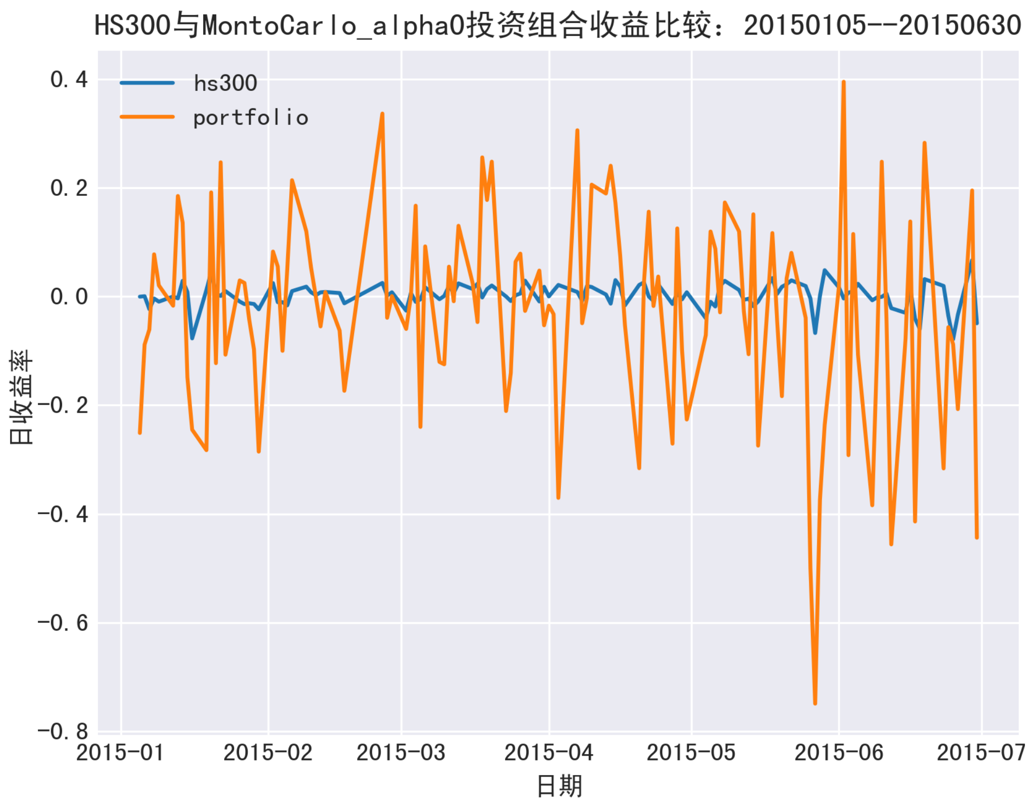
计算每期平均收益比较如下表，具体可见 **compare** 文件夹，通过表可看到整体HS300胜出

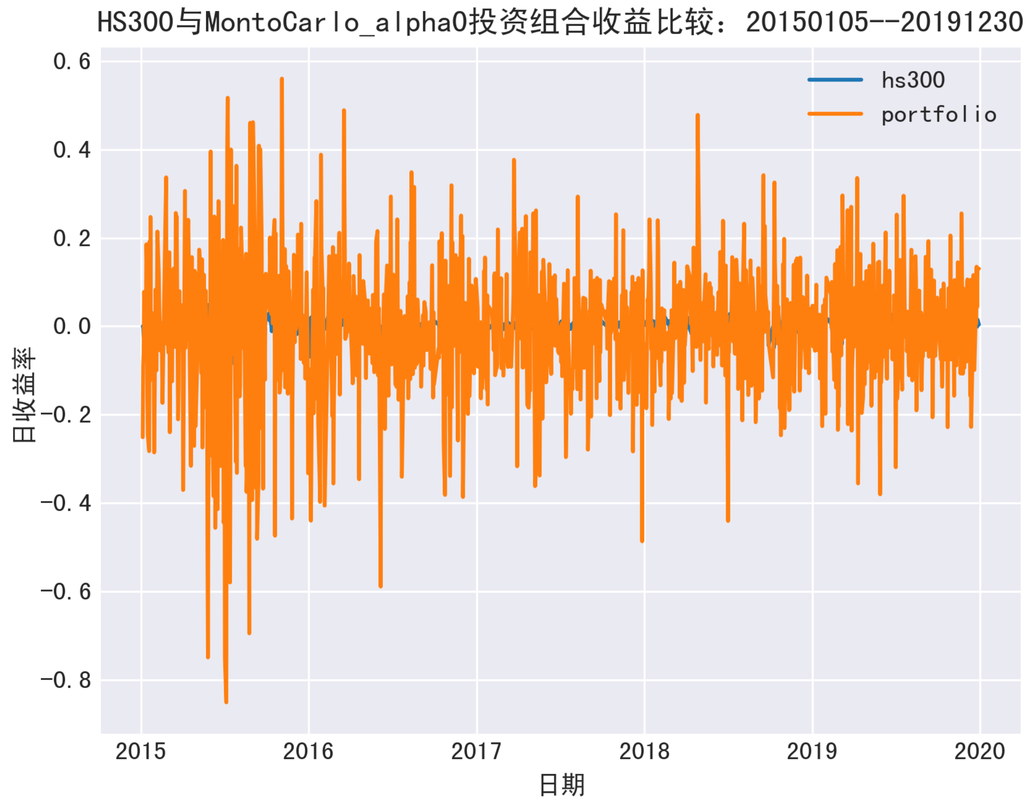
	开始时间	**结束时间**	**HS300平均日收益**	**Portfolio平均日收益**	win
	20150105	20150630	0.00157	-0.000653946	HS300
	20150701	20151231	-0.00126	-0.000365245	Portfolio
	20160104	20160630	-0.00064	-0.000298277	Portfolio
	20160701	20161230	0.000496	-0.000134006	HS300
	20170103	20170630	0.000758	-7.22E-06	HS300
	20170703	20171229	0.000929	-0.000703249	HS300
	20180102	20180629	-0.00146	0.00038444	Portfolio
	20180702	20181228	-0.001	7.94E-06	Portfolio
	20190102	20190628	0.002513	-0.000139735	HS300
	20190701	20191230	0.000354	5.86E-05	HS300
全部时间:	20150105	20191230	0.000217	-0.000186443	HS300

5.3 Monto Carlo alpha 0方法

由于 2. Monto Carlo方法 中 $\bar{r_p}$ 通常接近10%（日收益率10%与$r_{f_{day}}$相差较大），α 数值通常在97%左右（基本全投资无风险资产），因此为了直截了当查看最优市场组合权重表现，这个方法将2中的 α 直接设为0，即只考虑市场组合，不考虑无风险利率

该策略与HS300表现比较（仅以第一期20150105--20150630和总投资期20150105--20191230为例，更多结果请见images文件夹）：





计算每期平均收益比较如下表，具体可见 **compare** 文件夹，通过表可看到整体HS300胜出

	开始时间	**结束时间**	**HS300平均日收益**	**Portfolio平均日收益**	win
	20150105	20150630	0.00157	-0.02793	HS300
	20150701	20151231	-0.00126	-0.01474	HS300
	20160104	20160630	-0.00064	-0.01711	HS300
	20160701	20161230	0.000496	-0.00942	HS300
	20170103	20170630	0.000758	-0.00493	HS300
	20170703	20171229	0.000929	-0.02424	HS300
	20180102	20180629	-0.00146	0.006716	Portfolio
	20180702	20181228	-0.001	-0.00371	HS300
	20190102	20190628	0.002513	-0.00882	HS300
	20190701	20191230	0.000354	-0.00181	HS300
全部时间:	20150105	20191230	0.000217	-0.01062	HS300

第二题

1. 随机抽样5只股票

使用Dataframe.sample，为保证可重复性使用random_state=1属性（第一小题Monto Carlo那里也设置了随机种子保证可重复性），通过随机种子设置，随机抽到：22 2 49 26 33这五只股票，具体可见 `random_sample_stock` 函数

```
def random_sample_stock():
```

2. 计算Beta系数

Beta系数计算公式：

$$\beta_i = \frac{\sigma_{i,M}}{\sigma_M^2}$$

可以看出，计算 β 的要素全在协方差矩阵之中，将HS300加入数据框之后，再利用第一题的协方差矩阵计算方法，直接可求得5只股票同市场组合的协方差阵：

$$\Sigma_{5,M} = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,M} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{M,1} & \sigma_{M,2} & \cdots & \sigma_M^2 \end{pmatrix}$$

通过上式很容易发现要求得 β_i，所有数据都在协方差阵的最后一行（列）

求得这五只股票beta为

```
[0.98212886 1.20705893 0.89710279 1.09145141 0.99050688]
```

3. alpha显著性判断

3.1 alpha检验

alpha 检验有两层含义，既可以用于检验定价模型，也可以用于检验因子/策略是否有显著的超额收益。

在CAPM条件下，对下式进行OLS回归后检验截距项：

$$R_{i,t} = \alpha_{i,t} + \beta_i R_{M,t} + \varepsilon_{i,t}$$

其中，$R_{i,t} = r_{i,t} - r_{f_{day}}$，$r_{i,t}$是公式(1)中 $\vec{r_t}$ 的一个维度，$R_{M,t} = r_{M,t} - r_{f_{day}}$，$r_{M,t}$ 是HS300日收益率，β_i 为上面求得的Beta（假设10年内每只股票各自Beta一致）。检验零假设：$H_0 \colon \alpha_{i,t} = 0$，使用样本数量为12145（五只股票，样本期为20100105—20191231，2430天，日收益计算公式20191231日收益率无法获得，因此：12145 = 2430 * 5 - 5），检验结果为：

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.411
Model:                            OLS   Adj. R-squared:                  0.411
```

```
Method:                 Least Squares   F-statistic:                     8487.
Date:                Wed, 21 Oct 2020   Prob (F-statistic):               0.00
Time:                        08:33:12   Log-Likelihood:                 31487.
No. Observations:               12145   AIC:                        -6.297e+04
Df Residuals:                   12143   BIC:                        -6.295e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0002      0.000     -1.322      0.186      -0.001       0.000
x1             1.0000      0.011     92.127      0.000       0.979       1.021
==============================================================================
Omnibus:                     5078.263   Durbin-Watson:                   1.928
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1698553.225
Skew:                          -0.733   Prob(JB):                         0.00
Kurtosis:                      60.917   Cond. No.                         66.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

从结果中可以看出：$\alpha_{i,t}$ 估计值为-0.0002，P值为0.186>0.05应该无法拒绝原假设，即：$\alpha_{i,t}$ 并不显著不为零。另外，从自变量前系数和显著性来看，这段时间CAPM模型几乎是完美反映了这几只股票收益率。

3.2 GRS检验

Reference:

> Gibbons, Ross, Shanken, 1989. A test of the efficiency of a given portfolio, Econometrica, 57,1121-1152. DOI:10.2307/1913625

由于每只股票 α 可能有所不同，故统一通过3.1的检验会有不妥，GRS检验通过对一系列股票联合检验。检验股票的联合 α 为0的原假设是否成立，具体可参考上述文献。

相关实现参考：finance_byu

检验结果：

```
grsstat: 0.935236087740272
pval: 0.45687218432061016


|           | stock1   | stock2   | stock3   | stock4   | stock5   |
|-----------|----------|----------|----------|----------|----------|
| Intercept | 0.000    | -0.001   | -0.000   | 0.000    | -0.000   |
|           | (0.19)   | (-1.60)  | (-1.47)  | (0.74)   | (-0.40)  |
| Market    | 0.982    | 1.207    | 0.897    | 1.091    | 0.991    |
|           | (40.71)  | (39.28)  | (39.50)  | (51.99)  | (37.48)  |
| Obs       | 2429     | 2429     | 2429     | 2429     | 2429     |
| Rsq       | 0.41     | 0.39     | 0.39     | 0.53     | 0.37     |
```

由P-value得不能推翻原假设，即不能证明联合 α 显著不为0。

第三题

1. 前提假设：

第三题给的无风险利率3%是连续复利的无风险利率，否则需要通过 $log(1+r_f)$ 换算。

2. 二叉树

2.1 解释

给定条件：S=30，K=30，r=3%，σ=35%，t=1，m_steps

相关解释：

```
S:当前标的资产价格；
K:期权的执行价格；
r:年化无风险利率；
sigma:标的资产连续复利收益率的标准差；
t:以年表示的时间长度；
m_steps:二叉树的步长。
```

1. 计算u，d，P：

$$\Delta t = t/m\_steps$$
$$u = e^{\sigma\sqrt{\Delta t}},\ d = 1/u$$
$$P = \frac{e^{r\Delta t} - d}{u-d}$$

2. 再通过下式计算最后一期二叉树标的资产价格：

$$S_{d^m} = S * d^{m\_steps}$$
$$S_{d^{m-1}u} = S_{d^m} * u^2$$
$$S_{d^{m-2}u^2} = S_{d^{m-1}u} * u^2$$
$$\cdots$$

3. 通过与执行价格K比较计算最后一期期权价值：（例如看涨期权）

$$f_{d^m} = max(S_{d^m} - K, 0)$$
$$f_{d^{m-1}u} = max(S_{d^{m-1}u} - K, 0)$$
$$\cdots$$

4. 最终通过下式一步步往前推的第一期

$$f_{d^{m-1}} = e^{-r\Delta t}((1-P)f_{d^m} + Pf_{d^{m-1}u})$$
$$\cdots$$

美式看跌期权在上一步增加一个比较环节：

$$\hat{f_{d^{m-1}}} = e^{-r\Delta t}((1-P)f_{d^m} + Pf_{d^{m-1}u})$$
$$S_{d^{m-1}} = S_{d^{m-1}u} * d$$
$$f_{d^{m-1}} = max(\hat{f_{d^{m-1}}}, S_{d^{m-1}} - K)$$
$$\cdots$$

2.2 计算结果

美式看跌100步： Option price: 3.7557436745895885

欧式看跌100步： Option price: 3.667340740092775

欧式看跌50步： Option price: 3.6570804703496114

欧式看跌10步： Option price: 3.5760697183104884

代码详情可见 `hw1_3.py` ，结果详情可见 **option_result** 文件夹，该文件夹中有每一步的股票价格和期权价值，在步数小的情况下（其实最好5步以内）可以做出二叉树图，下面仅展示 **欧式看跌10步** 中间结果：

```
[[(30.0, 3.58)], [(26.86, 4.88), (33.51, 2.22)], [(26.86, 4.88), (30.0, 3.23)],
[(24.04, 6.47), (30.0, 3.23), (37.43, 1.17)], [(24.04, 6.47), (30.0, 3.23),
(33.51, 1.83)], [(24.04, 6.47), (26.86, 4.57), (33.51, 1.83)], [(21.52, 8.3),
(26.86, 4.57), (33.51, 1.83), (41.81, 0.47)], [(21.52, 8.3), (26.86, 4.57),
(33.51, 1.83), (37.43, 0.81)], [(21.52, 8.3), (26.86, 4.57), (30.0, 2.81), (37.43,
0.81)], [(21.52, 8.3), (24.04, 6.25), (30.0, 2.81), (37.43, 0.81)], [(19.27,
10.29), (24.04, 6.25), (30.0, 2.81), (37.43, 0.81), (46.71, 0.11)], [(19.27,
10.29), (24.04, 6.25), (30.0, 2.81), (37.43, 0.81), (41.81, 0.21)], [(19.27,
10.29), (24.04, 6.25), (30.0, 2.81), (33.51, 1.38), (41.81, 0.21)], [(19.27,
10.29), (24.04, 6.25), (26.86, 4.19), (33.51, 1.38), (41.81, 0.21)], [(19.27,
10.29), (21.52, 8.23), (26.86, 4.19), (33.51, 1.38), (41.81, 0.21)], [(17.25,
12.3), (21.52, 8.23), (26.86, 4.19), (33.51, 1.38), (41.81, 0.21), (52.17, 0.0)],
[(17.25, 12.3), (21.52, 8.23), (26.86, 4.19), (33.51, 1.38), (41.81, 0.21),
(46.71, 0.0)], [(17.25, 12.3), (21.52, 8.23), (26.86, 4.19), (33.51, 1.38),
(37.43, 0.41), (46.71, 0.0)], [(17.25, 12.3), (21.52, 8.23), (26.86, 4.19), (30.0,
2.3), (37.43, 0.41), (46.71, 0.0)], [(17.25, 12.3), (21.52, 8.23), (24.04, 6.01),
(30.0, 2.3), (37.43, 0.41), (46.71, 0.0)], [(17.25, 12.3), (19.27, 10.37), (24.04,
6.01), (30.0, 2.3), (37.43, 0.41), (46.71, 0.0)], [(15.44, 14.2), (19.27, 10.37),
(24.04, 6.01), (30.0, 2.3), (37.43, 0.41), (46.71, 0.0), (58.28, 0.0)], [(15.44,
14.2), (19.27, 10.37), (24.04, 6.01), (30.0, 2.3), (37.43, 0.41), (46.71, 0.0),
(52.17, 0.0)], [(15.44, 14.2), (19.27, 10.37), (24.04, 6.01), (30.0, 2.3), (37.43,
0.41), (41.81, 0.0), (52.17, 0.0)], [(15.44, 14.2), (19.27, 10.37), (24.04, 6.01),
(30.0, 2.3), (33.51, 0.8), (41.81, 0.0), (52.17, 0.0)], [(15.44, 14.2), (19.27,
10.37), (24.04, 6.01), (26.86, 3.72), (33.51, 0.8), (41.81, 0.0), (52.17, 0.0)],
[(15.44, 14.2), (19.27, 10.37), (21.52, 8.21), (26.86, 3.72), (33.51, 0.8),
(41.81, 0.0), (52.17, 0.0)], [(15.44, 14.2), (17.25, 12.48), (21.52, 8.21),
(26.86, 3.72), (33.51, 0.8), (41.81, 0.0), (52.17, 0.0)], [(13.82, 15.91), (17.25,
12.48), (21.52, 8.21), (26.86, 3.72), (33.51, 0.8), (41.81, 0.0), (52.17, 0.0),
(65.1, 0.0)], [(13.82, 15.91), (17.25, 12.48), (21.52, 8.21), (26.86, 3.72),
(33.51, 0.8), (41.81, 0.0), (52.17, 0.0), (58.28, 0.0)], [(13.82, 15.91), (17.25,
12.48), (21.52, 8.21), (26.86, 3.72), (33.51, 0.8), (41.81, 0.0), (46.71, 0.0),
(58.28, 0.0)], [(13.82, 15.91), (17.25, 12.48), (21.52, 8.21), (26.86, 3.72),
(33.51, 0.8), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0)], [(13.82, 15.91), (17.25,
12.48), (21.52, 8.21), (26.86, 3.72), (30.0, 1.57), (37.43, 0.0), (46.71, 0.0),
(58.28, 0.0)], [(13.82, 15.91), (17.25, 12.48), (21.52, 8.21), (24.04, 5.78),
(30.0, 1.57), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0)], [(13.82, 15.91), (17.25,
12.48), (19.27, 10.55), (24.04, 5.78), (30.0, 1.57), (37.43, 0.0), (46.71, 0.0),
(58.28, 0.0)], [(13.82, 15.91), (15.44, 14.38), (19.27, 10.55), (24.04, 5.78),
(30.0, 1.57), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0)], [(12.38, 17.44), (15.44,
14.38), (19.27, 10.55), (24.04, 5.78), (30.0, 1.57), (37.43, 0.0), (46.71, 0.0),
(58.28, 0.0), (72.72, 0.0)], [(12.38, 17.44), (15.44, 14.38), (19.27, 10.55),
(24.04, 5.78), (30.0, 1.57), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0), (65.1,
0.0)], [(12.38, 17.44), (15.44, 14.38), (19.27, 10.55), (24.04, 5.78), (30.0,
1.57), (37.43, 0.0), (46.71, 0.0), (52.17, 0.0), (65.1, 0.0)], [(12.38, 17.44),
(15.44, 14.38), (19.27, 10.55), (24.04, 5.78), (30.0, 1.57), (37.43, 0.0), (41.81,
0.0), (52.17, 0.0), (65.1, 0.0)], [(12.38, 17.44), (15.44, 14.38), (19.27, 10.55),
(24.04, 5.78), (30.0, 1.57), (33.51, 0.0), (41.81, 0.0), (52.17, 0.0), (65.1,
0.0)], [(12.38, 17.44), (15.44, 14.38), (19.27, 10.55), (24.04, 5.78), (26.86,
3.05), (33.51, 0.0), (41.81, 0.0), (52.17, 0.0), (65.1, 0.0)], [(12.38, 17.44),
(15.44, 14.38), (19.27, 10.55), (21.52, 8.39), (26.86, 3.05), (33.51, 0.0),
(41.81, 0.0), (52.17, 0.0), (65.1, 0.0)], [(12.38, 17.44), (15.44, 14.38), (17.25,
12.66), (21.52, 8.39), (26.86, 3.05), (33.51, 0.0), (41.81, 0.0), (52.17, 0.0),
(65.1, 0.0)], [(12.38, 17.44), (13.82, 16.09), (17.25, 12.66), (21.52, 8.39),
(26.86, 3.05), (33.51, 0.0), (41.81, 0.0), (52.17, 0.0), (65.1, 0.0)], [(11.08,
18.83), (13.82, 16.09), (17.25, 12.66), (21.52, 8.39), (26.86, 3.05), (33.51,
0.0), (41.81, 0.0), (52.17, 0.0), (65.1, 0.0), (81.23, 0.0)], [(11.08, 18.83),
(13.82, 16.09), (17.25, 12.66), (21.52, 8.39), (26.86, 3.05), (33.51, 0.0),
(41.81, 0.0), (52.17, 0.0), (65.1, 0.0), (72.72, 0.0)], [(11.08, 18.83), (13.82,
```

```
16.09), (17.25, 12.66), (21.52, 8.39), (26.86, 3.05), (33.51, 0.0), (41.81, 0.0),
(52.17, 0.0), (58.28, 0.0), (72.72, 0.0)], [(11.08, 18.83), (13.82, 16.09),
(17.25, 12.66), (21.52, 8.39), (26.86, 3.05), (33.51, 0.0), (41.81, 0.0), (46.71,
0.0), (58.28, 0.0), (72.72, 0.0)], [(11.08, 18.83), (13.82, 16.09), (17.25,
12.66), (21.52, 8.39), (26.86, 3.05), (33.51, 0.0), (37.43, 0.0), (46.71, 0.0),
(58.28, 0.0), (72.72, 0.0)], [(11.08, 18.83), (13.82, 16.09), (17.25, 12.66),
(21.52, 8.39), (26.86, 3.05), (30.0, 0.0), (37.43, 0.0), (46.71, 0.0), (58.28,
0.0), (72.72, 0.0)], [(11.08, 18.83), (13.82, 16.09), (17.25, 12.66), (21.52,
8.39), (24.04, 5.96), (30.0, 0.0), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0),
(72.72, 0.0)], [(11.08, 18.83), (13.82, 16.09), (17.25, 12.66), (19.27, 10.73),
(24.04, 5.96), (30.0, 0.0), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0), (72.72,
0.0)], [(11.08, 18.83), (13.82, 16.09), (15.44, 14.56), (19.27, 10.73), (24.04,
5.96), (30.0, 0.0), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0), (72.72, 0.0)],
[(11.08, 18.83), (12.38, 17.62), (15.44, 14.56), (19.27, 10.73), (24.04, 5.96),
(30.0, 0.0), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0), (72.72, 0.0)], [(9.92,
20.08), (12.38, 17.62), (15.44, 14.56), (19.27, 10.73), (24.04, 5.96), (30.0,
0.0), (37.43, 0.0), (46.71, 0.0), (58.28, 0.0), (72.72, 0.0), (90.74, 0.0)]]
```

3. Black Scholes公式

参数与二叉树一致（无需m_steps）

3.1 公式

$$
\begin{aligned}
CALL &= SN(d_1) - Ke^{-rt}N(d_2) \\
PUT &= Ke^{-rt}N(-d_2) - SN(-d_1)
\end{aligned}
$$

其中，

$$
\begin{aligned}
d_1 &= \frac{ln(S/K) + (r + \sigma^2/2)t}{\sigma t} \\
d_2 = d_1 - \sigma t &= \frac{ln(S/K) + (r - \sigma^2/2)t}{\sigma t} \\
N(d) &= \frac{1}{\sqrt{2\pi}} \int_{\infty}^{d} e^{-\frac{1}{2}x^2} \mathrm{d}x
\end{aligned}
$$

3.2 结果

BS公式欧式看跌： Option price: 3.677627713214079

可以看出二叉树结果在步数增大时逐步逼近了BS公式结果