

Fund Management: Finding the Best Portfolio

1. 문제 제시

이번에 진행한 Term-Project 에 제시된 문제에 따르면, 100 억 원을 관리하는 펀드 매니저의 상황을 가정한다. 그리고, 10 개 이상의 자산 지수를 선택한다. 이 때, 조건이 있다 :

- 공매도 (Short selling)은 금지되어 있다.
- 무위험 이자율(Risk-Free Rate)는 US 13-Week T-bill rate 를 이용한다.

2. 방법 소개 및 결과

세계 각국의 총 11 개의 인덱스를 이용하였다.

- ① S&P500 : 미국의 주요 주식시장 지수 중 하나로, 미국 주식시장에서 가장 큰 500 개 기업을 포함하고 있다. 시장 가치 가중 평균 지수로 계산된다.
- ② KOSPI : 한국의 주요 주식시장 지수로, 한국거래소에 상장된 모든 주식의 가격 변동을 반영한다.
- ③ DAX : 독일의 주요 주식시장 지수로, 독일에서 가장 거래량이 많은 30 개의 기업을 포함하고 있다.
- ④ HANGSENG : 홍콩의 주요 주식시장 지수로, 홍콩 주식시장에서 가장 큰 50 개 기업을 포함하고 있다.
- ⑤ FTSE100 : 영국의 주요 주식시장 지수로, 영국에서 가장 큰 100 개의 기업을 포함하고 있다.
- ⑥ Nikkei : 일본의 주요 주식시장 지수로, 일본에서 가장 큰 225 개의 기업을 포함하고 있다.
- ⑦ BSE Sensex : 인도의 주요 주식시장 지수로, 인도 봄베이 주식거래소에 상장된 가장 큰 30 개 기업을 포함하고 있다.

- ⑧ CAC40 : 프랑스의 주요 주식시장 지수로, 파리 주식거래소에 상장된 가장 큰 40 개 기업을 포함하고 있다.
- ⑨ SSE Composite : 중국의 상하이 종합 지수로, 상하이 증권 거래소에서 가장 널리 추적되는 주가지수이다.
- ⑩ Mexico IPC : 멕시코의 IPC 지수로, 멕시코시티 증권 거래소의 주요 회사들을 포함한다.
- ⑪ BEL20 : 벨기에 주식시장에서 가장 활발하게 거래되는 20 개의 주요 회사들을 포함한다.

이와 같이, 전 세계에 걸쳐 고른 투자를 하려는, 즉, 어떤 한 지역의 이벤트에 휘둘리지 않는 글로벌한 포트폴리오를 세우고자 하는 펀드 매니저, 혹은 투자자를 가정하였다. 또한, 각 국가에서 가장 거래량이 많은 인덱스들을 포트폴리오 구성에 모두 이용함으로써, 각 지역의 정보가 빠르게 전달되도록 구성하였다.

데이터는 현재로부터 총 8 년치의 일일 종가 데이터를 이용하였다.

Efficient frontier 는 여러 투자 포트폴리오 중, 동일한 위험을 가진 것들 중 가장 기대 수익률이 높은 포트폴리오를 나타내거나, 동일한 기대 수익률을 가진 것들 중에서 가장 위험이 낮은 포트폴리오를 나타내는 그래프이다.

현재, 필자는 최근 데이터가 예전 데이터보다 현재 주가에 영향을 크게 미칠 것이라는 견해를 가지고 있기 때문에, 아래와 같은 방식을 이용하였다.

추가적인 구현 사항 : EWMA (Exponentially Weighted Moving Average)

이동 평균 (Moving Average)이 일정 기간 동안의 데이터 평균을 계산하는 반면, EWMA 는 가장 최근의 관측치에 더 큰 가중치를 부여하고, 시간이 지남에 따라 더 이전의 관측치에는 더 적은 가중치를 부여한다. 이런 방식은, 최신 데이터가 더 중요하다는 가정에 기반한다. 그렇기 때문에, 트렌드의 변화를 더 잘 반영할 수 있다.

이는 'pandas' 라이브러리에서 제공하는 'ewm' 함수를 통해 구현할 수 있는데, 함수의 parameter 중, span 매개변수는 지수 가중치의 기간을 결정한다. alpha 는 $2/(span+1)$ 로 결정되며, 결국 알파를 이용하여 아래와 같이 계산된다 :

$$EWMA_t = \alpha * r_t + (1 - \alpha) * EWMA_{t-1}$$

EWMA 를 적용한 Daily return 에 대한 table 은 다음과 같다 :

	Shanghai	BEL20	BSESensex	CAC40	FTSE100	DAX	S&P500	HANG SENG	KOSPI	Mexico	Nikkei
Date											
2015-05-27	0.006275	0.012483	0.001208	0.019473	0.012131	0.012559	0.009163	-0.005970	-0.016795	0.006333	0.001717
2015-05-28	-0.029743	0.005176	-0.000464	0.005283	0.006551	0.002203	0.003896	-0.014229	-0.007501	0.004544	0.002796
2015-05-29	-0.020257	-0.000657	0.003671	-0.005113	0.001595	-0.006226	0.000423	-0.009764	-0.004321	0.002212	0.002039
2015-06-01	-0.002921	-0.001059	0.002916	-0.002900	0.000048	-0.004121	0.000844	-0.005625	-0.004722	0.001968	0.001598
2015-06-02	0.001211	-0.001448	-0.002627	-0.003160	-0.000719	-0.005215	0.000459	-0.005438	-0.006088	0.002355	0.000995
...
2023-05-17	0.000263	-0.000422	0.000580	0.001040	0.000214	0.001261	0.000717	-0.000231	0.000447	0.000936	0.001584
2023-05-18	0.000337	-0.000221	0.000527	0.001147	0.000259	0.001499	0.000890	-0.000058	0.000603	0.000988	0.001869
2023-05-19	0.000248	-0.000113	0.000613	0.001244	0.000291	0.001607	0.000844	-0.000335	0.000767	0.000622	0.001983
2023-05-22	0.000321	-0.000112	0.000676	0.001183	0.000321	0.001513	0.000830	-0.000096	0.000903	0.000307	0.002123
2023-05-23	0.000013	-0.000137	0.000668	0.000896	0.000295	0.001396	0.000592	-0.000343	0.000966	0.000174	0.001999

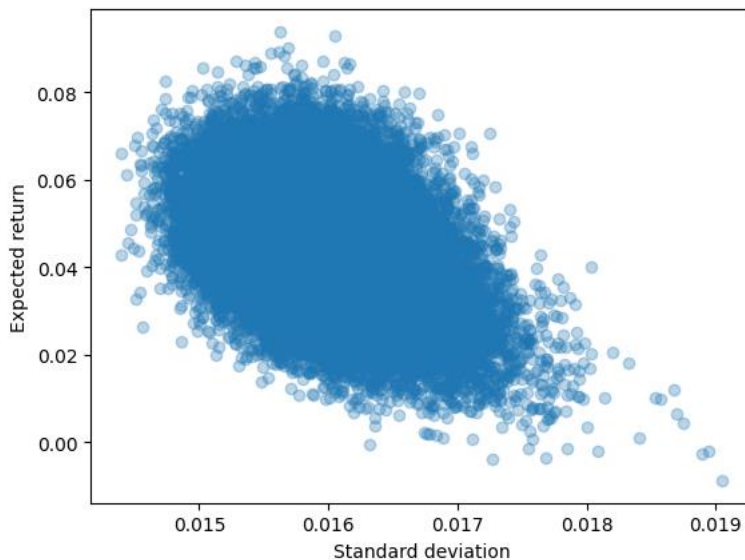
또한, EWMA 를 적용한 annual return 은 다음과 같다 :

Shanghai	-0.068663
BEL20	0.022126
BSESensex	0.151616
CAC40	0.074362
FTSE100	0.027064
DAX	0.058995
S&P500	0.133985
HANG SENG	-0.056646
KOSPI	0.022009
Mexico	0.048401
Nikkei	0.076193

그리고 , EWMA 를 적용한 annual covariance matrix 는 다음과 같다 :

	Shanghai	BEL20	BSESensex	CAC40	FTSE100	DAX	S&P500	HANG SENG	KOSPI	Mexico	Nikkei
Shanghai	0.000924	0.000153	0.000141	0.000222	0.000146	0.000276	0.000191	0.000498	0.000270	0.000060	0.000212
BEL20	0.000153	0.000318	0.000229	0.000338	0.000233	0.000324	0.000226	0.000212	0.000220	0.000207	0.000217
BSESensex	0.000141	0.000229	0.000367	0.000265	0.000198	0.000256	0.000207	0.000223	0.000272	0.000196	0.000225
CAC40	0.000222	0.000338	0.000265	0.000444	0.000294	0.000405	0.000255	0.000255	0.000223	0.000227	0.000254
FTSE100	0.000146	0.000233	0.000198	0.000294	0.000265	0.000265	0.000173	0.000196	0.000139	0.000180	0.000169
DAX	0.000276	0.000324	0.000256	0.000405	0.000265	0.000441	0.000265	0.000310	0.000297	0.000199	0.000271
S&P500	0.000191	0.000226	0.000207	0.000255	0.000173	0.000265	0.000250	0.000196	0.000223	0.000163	0.000202
HANG SENG	0.000498	0.000212	0.000223	0.000255	0.000196	0.000310	0.000196	0.000620	0.000383	0.000118	0.000230
KOSPI	0.000270	0.000220	0.000272	0.000223	0.000139	0.000297	0.000223	0.000383	0.000509	0.000172	0.000252
Mexico	0.000060	0.000207	0.000196	0.000227	0.000180	0.000199	0.000163	0.000118	0.000172	0.000265	0.000133
Nikkei	0.000212	0.000217	0.000225	0.000254	0.000169	0.000271	0.000202	0.000230	0.000252	0.000133	0.000328

이후, Random 하게 50000 개의 portfolio 를 만들어서, standard deviation 을 x 축, expected return 을 y 축으로 하는 그래프를 plotting 해본 결과,



와 같이 나오는 것을 확인할 수 있다.

이후, MVO(Mean-Variance Optimization) problem 을 풀이한 결과, 최적해, 즉 각 index 에 대한 최적 가중치는 아래와 같이 나왔다 (percentage 로 표시하였다.):

Shanghai : 7.16%

BEL20 : 0.000797%

BSESensex : 0.022%

CAC40 : 0.000202%

FTSE100 : 19.8%

DAX : 0.000329%

S&P500 : 20.4%

HANG SENG : 0.209%

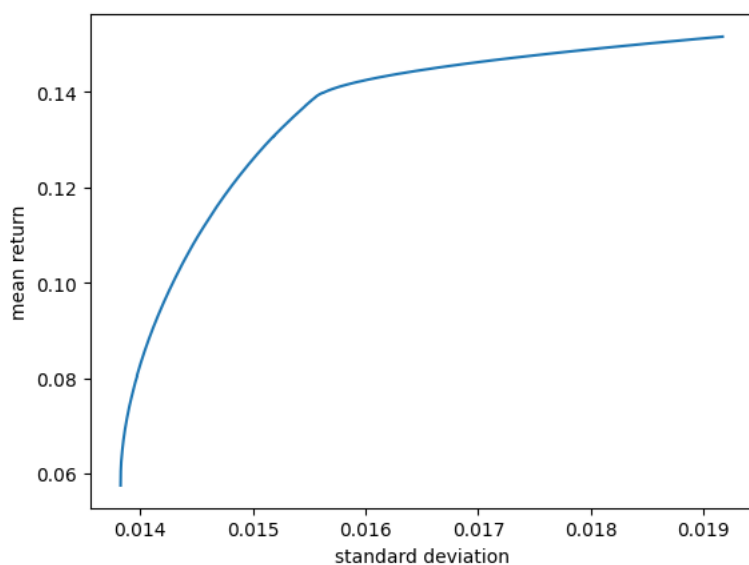
KOSPI : 0.129%

Mexico : 36.3%

Nikkei : 16%

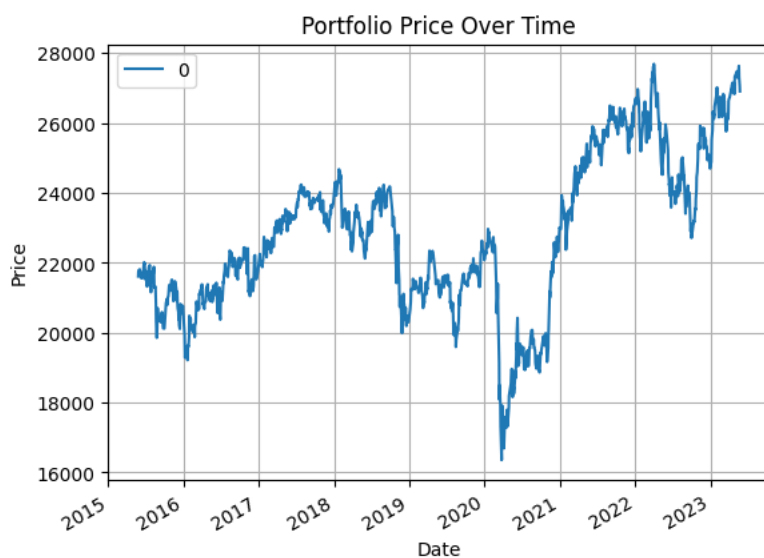
상위 투자 종목 3 가지는 Mexico, S&P500, FTSE100 이다. 이를 토대로, 이렇게 하여 만들어진 포트폴리오는 멕시코, 미국, 영국의 경제 상황에 큰 영향을 받을 것임을 알 수 있다. 나중에 시장 선을 그릴 때는, 이 중 거래량이 제일 많고 전 세계적으로 영향을 미치는 index 인 S&P500 을 이용하여 그릴 예정이다.

결과적으로, Efficient Frontier 는 다음과 같다 :



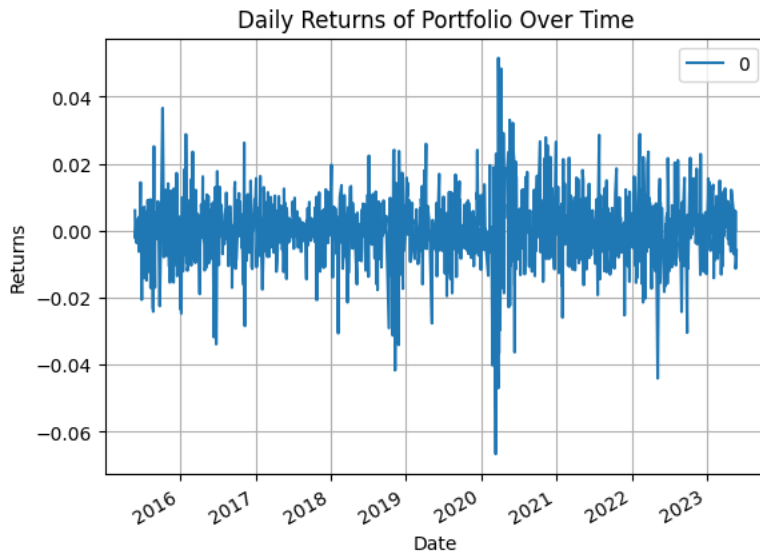
약간 꺾인 모습을 보여주는 Efficient Frontier 이지만, 전체적으로 원하는 형태를 얻을 수 있었다.

또한, 포트폴리오 가격의 시간에 따른 변화는 아래 그래프와 같다 :



또한, 이 기간 내의 포트폴리오 연간 수익률을 구해본 결과, **4.8839%**로 구해졌다.
S&P500 의 이 기간 내의 수익률보다는 낮은 수치이지만, 다른 인덱스들과 함께 만들어진 포트폴리오기 때문에, volatility 가 낮아졌다는 결론을 얻을 수 있다.

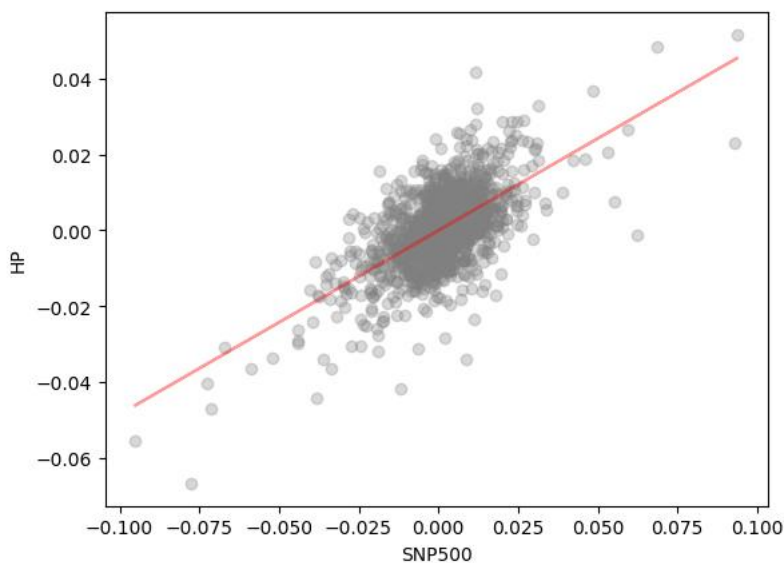
일별 수익률은 아래의 그래프와 같다 :



여기서, 이제 Python 에서 linear regression 을 수행해주는 함수인 'ols' 함수를 이용하여, S&P500 으로 대표되는 시장 에서의 linear regression 을 수행한다.

```
# Linear regression
model = ols("Portfolio ~ SNP500", daily_return_array).fit()

# Get model parameters (alpha and beta)
p = model.params
# Draw scatter plot and linear regression line
plt.scatter(x = daily_return_array['SNP500'], y = daily_return_array['Portfolio'], alpha = 0.3, color = 'gray')
plt.plot(daily_return_array['SNP500'], p.Intercept + p.SNP500 * daily_return_array['SNP500'], 'r-', alpha = 0.4)
plt.xlabel("SNP500")
plt.ylabel("HP")
plt.show()
```



이와 같이 Linear regression 이 된다.

아래는 OLS Regression 을 한 결과, 즉, Predictability 와 Statistical measure 이다:

```

>
      OLS Regression Results
Dep. Variable: Portfolio      R-squared:    0.396
Model: OLS                  Adj. R-squared: 0.395
Method: Least Squares       F-statistic: 1018.
Date: Wed, 24 May 2023      Prob (F-statistic): 3.72e-172
Time: 12:50:22             Log-Likelihood: 5376.2
No. Observations: 1555      AIC: -1.075e+04
Df Residuals: 1553         BIC: -1.074e+04
Df Model: 1
Covariance Type: nonrobust

      coef    std err   t    P>|t| [0.025 0.975]
Intercept -6.144e-05  0.000   -0.317  0.751 -0.000  0.000
SNP500    0.4843     0.015  31.899  0.000  0.455  0.514
Omnibus:    75.163   Durbin-Watson: 1.988
Prob(Omnibus): 0.000   Jarque-Bera (JB): 226.637
Skew:       -0.158     Prob(JB): 6.12e-50
Kurtosis:    4.843     Cond. No.    78.5

```

또한, ANOVA 결과는 다음과 같다 :

	df	sum_sq	mean_sq	F	PR(>F)
SNP500	1.0	0.059238	0.059238	1017.518828	3.720410e-172
Residual	1553.0	0.090413	0.000058	NaN	NaN

3. 결과 분석

현재 운용중인 자산이 100 억 원이기 때문에, 위에서 구한 최적 weight 에 의하면

Shanghai : 716000000 원

BEL20 : 79700 원

BSESensex : 2200000 원

CAC40 : 20200 원

FTSE100 : 1980000000 원

DAX : 32900 원

S&P500 : 2040000000 원

HANG SENG : 2090000 원

KOSPI : 12900000 원

Mexico : 3630000000 원

Nikkei : 1600000000 원

을 투자하는 것이 최적의 포트폴리오 구성이다.

이 포트폴리오 구성은 앞에서 설명했듯이 투자자의 견해를 반영하고 있는데, 최근 100 일간의 데이터에 더 큰 가중치, 즉 EWMA 에서의 span 을 100 으로 만들어 parameter 들을 구함으로서 견해를 직접적으로 반영하였다.

MVO 최적화 문제를 풀이하며 말 그대로 평균-분산 에 있어 최적인 각 인덱스에 대한 가중치를 찾을 수 있었다.

마지막으로는, 이렇게 하여 만들어진 포트폴리오가 시장을 잘 따라가는지 알아보기 위해 S&P500 으로 대변되는 시장에 대해 SCL 을 그려보았는데, OLS Regression 의 결과와 ANOVA 결과를 통해 나온 measure 변수들의 크기를 보았을 때, 시장선에 아주 가까운 데이터들이 모였다는 것을 알 수 있었으며, 이를 통하여 꽤 성공적인 데이터를 얻었음을 알 수 있었다.

4. 문제 해결 사항 및 제언

처음에는 MVO 최적화 문제가 아닌 Black-Litterman 방식을 이용하여 최적화를 구현해보고자 했으나, 문제 특성 상 Index 를 이용하기 때문에 시가 총액을 얻을 수 없는 관계로 그와 같은 방식으로 구현하는 데 실패하였다. 이후에 단일 주식들을 이용한 포트폴리오를 구성할 때는 MVO 의 overfitting, 입력 값에 따른 불안정성과 같은 문제점을 해결하기 위해 Black-Litterman, Factor Investment strategy 등의 다른 방법들을 이용하여 최적 weight 를 구할 수 있을 것이다.

문제 해결 사항으로는, 주어진 코드에서 각 인덱스에 대한 정보가 옳지 않게 들어가는 것을 파악하고, 그 순서에 맞게 데이터 위치를 재조정했다.

```
ticker_list = ['^GSPC', '^KS11', '^GDAXI', '^HSI', '^FTSE', '^N225', '^BSESX', '^FCHI', '000001.SS', '^BFX', '^MXM']
index_name = ['Shanghai', 'BEL20', 'BSESensex', 'CAC40', 'FTSE100', 'DAX', 'S&P500', 'HANG SENG', 'KOSPI', 'Mexico', 'Nikkei']

data = yf.download(tickers = ticker_list, period = "8y", interval = "1d")
close = data['Close']
close.columns = index_name
close.dropna(inplace=True)

close
```

이 부분이 상술한 부분이며, ticker_list 와 index_name 의 순서가 맞지 않는 것이 아니라, 이 순서로 해야 데이터의 각 column 에 값이 옳게 들어가는 것을 확인할 수 있다.

<코드를 zip 파일에 함께 첨부하겠습니다.>