

YIG Quant

Portfolio Optimization

Yonsei Investment Group

박성우, 유현우, 황희원

Sep 2023

Table of Contents:

1_Modern Portfolio Theory (MPT)

- # Theory: 이론소개

- # Implementation: 구현

- # Insight: 느낀점

2_Post Modern Portfolio Theory (PMPT)

- # Theory: 이론소개

- # Implementation: 구현

- # Insight: 느낀점

3_Backtesting

Modern Portfolio Theory (MPT)

1

Theory: Introduction to MPT

Harry Markowitz



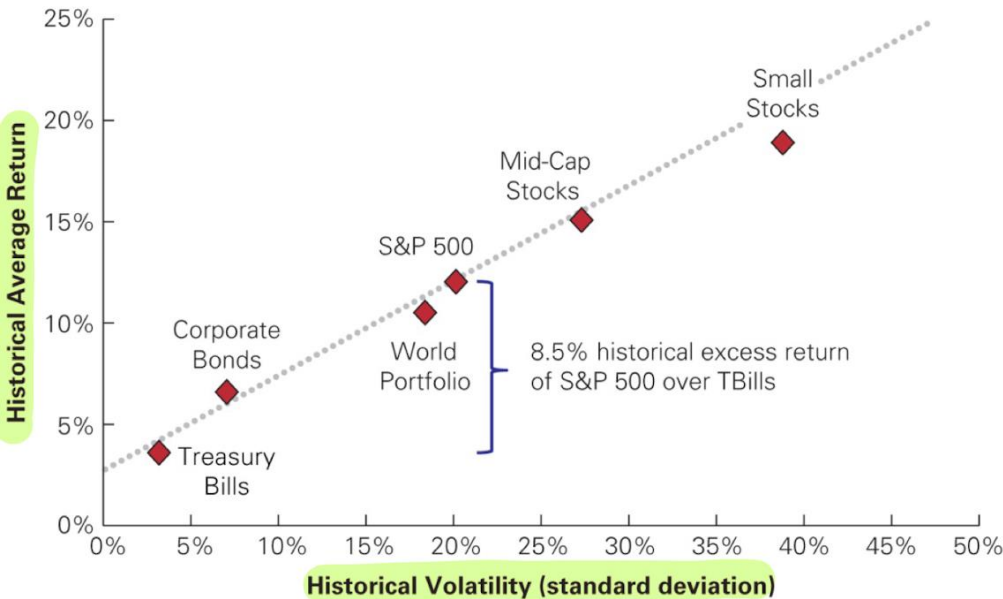
Source: YIG

Thesis of MPT

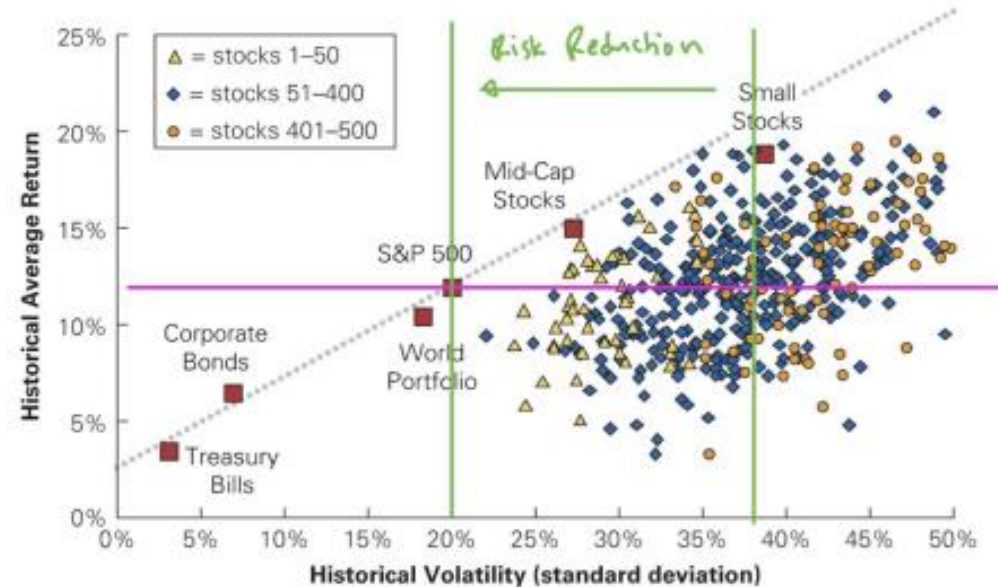
- “Portfolio Selection” by H. Markowitz(1952)
- Risk-averse(위험회피형) 투자자에게
‘최적’의 포트폴리오는?
- 분산투자(diversification)를 통해 risk(variance)를
최소화하고 기대수익률(return)을 극대화할 수 있는
효율적인 포트폴리오
- **효율적인 포트폴리오 : 주어진 리스크 수준에서 얻을 수
있는 최고의 기대 수익률**
- Nobel Prize(경제학상, 1990)

Theory: 핵심 변수 – 1.변동성 2.수익률

변동성 & 수익률의 역사적인 관계



변동성 & 수익률 역사적인 관계 – S&P 500 세분화



- 자산 별 평균 수익률과 변동성(risk)는 양의 상관관계를 가짐 (변동성이 높은 자산일수록 평균 수익률이 높음)
- S&P 500 을 500개의 개별 구성 종목으로 세분화해 분석 시 diversification의 risk 절감 효과 확인할 수 있음
- S&P 500의 평균 수익률 = 500개 개별 종목 평균 수익률, BUT S&P 500 변동성(risk) < 500개 개별 종목 평균 변동성(risk)
- -> 즉, 500개 개별 주식의 분산투자 형태인 S&P 500은 diversification을 통해 risk 절감효과

Implementation: 실전 들어가기 전 알아야 할 것

'포트폴리오의 기대 수익률, 분산 공식

Review: Portfolio Expected Return and Variance

- Notations: N assets, S possible states, $p(s)$ denotes the probability of state s , $r_i(s)$ denotes asset i 's return in state s
- The portfolio weight for asset i : $w_i = \frac{\text{value of asset } i}{\text{value of all assets in portfolio}}$
- The portfolio return r_p in state s : $r_p(s) = w_1 r_1(s) + w_2 r_2(s) + \dots + w_N r_N(s)$
- Expected return of a portfolio P : $E(r_p) = \sum_{s=1}^S p(s) r_p(s)$ or $E(r_p) = \sum_{i=1}^N w_i E(r_i)$
- Variance of a two-asset portfolio P : $\sigma_P^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \text{Cov}(r_A, r_B)$
or
 $\sigma_P^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \rho_{A,B} \sigma_A \sigma_B$

Correlation

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

Source: DB GAPS, YIG

DB GAPS 투자대회 규정

■ 투자가능 국내 상장 ETF

구분	축약명	상품명	추종지수	자산기준		상품기준	
				상한	하한	상한	하한
자산군	국내주식	KOSPI	KODEX 200	40	10	40	0
		KOSDAQ	TIGER 코스닥150			20	0
	해외주식	S&P 500	TIGER 미국S&P500선물(H)	40	10	20	0
		STOXX 50	TIGER 유로스탁스50(합성,H)			20	0
		Nikkei 225	ACE 일본 Nikkei225(H)			20	0
		CSI 300	TIGER 차이나CSI300			20	0
	국내채권	국채 10년	KOSEF 국고채10년	60	20	50	0
		우량회사채	KBSTAR 중기우량회사채(**)			40	0
	해외채권	해외채권	TIGER 단기선진하이일드(합성, H)	20	5	40	5
	원자재	금	KODEX 골드선물(H)			15	0
리스크 관리 수단	Inverse	WTI	TIGER 원유선물Enhanced(H)	20	0	15	0
		KOSPI Short	KODEX 인버스			20	0
	FX	US Long	KOSEF 미국달러선물	20	0	0	20
		US Short	KOSEF 미국달러인버스선물			0	20
	현금	MMF	KOSEF 단기자금	50	1		0
		Cash	현금				1

* ETF 상품별 운용사 : KODEX(삼성자산운용), TIGER(미래에셋자산운용), KBSTAR(KB자산운용), KOSEF(키움자산운용), ACE(한국투자신탁운용)

** KBSTAR 우량회사채 분배금 : 권리기준일 보유비중에 분배금수익률 곱하여 지급기준일에 현금으로 지급

*** KOSPI Short 비중범위 상한은 국내주식 편입비중을 기준으로 하며 최대 20%를 넘을 수 없음

DBGAPS

Modern Portfolio Theory

Implementation: 실전에 들어가자

1. 데이터를 긁어온다 (EX. 1mo_daily_3mo)

구분	국내주식			해외주식			국내채권			해외채권			원자재			Inverse			FX			현금
	KOSPI	KOSDAQ	S&P 500	STOXX 50	Nikkei 225	CSI 300	국채 10년	우량회사채	해외채권	금	WTI	KOSPI Short	US Long	US Short	MMF							
2020/09/14	20.95%	50.97%	12.15%	-4.84%	8.65%	24.58%	0.65%	1.62%	-0.60%	25.56%	-52.48%	-24.05%	0.77%	-1.57%	1.25%							
2020/09/15	21.90%	52.70%	12.81%	-5.56%	8.51%	25.85%	0.73%	1.64%	-0.89%	27.10%	-52.48%	-24.55%	0.34%	-1.16%	1.25%							
2020/09/16	21.18%	48.73%	12.72%	-6.47%	6.37%	23.12%	1.93%	1.92%	-1.15%	25.22%	-52.69%	-24.18%	0.78%	-1.36%	1.25%							
2020/09/17	19.44%	46.28%	9.87%	-5.87%	5.93%	24.19%	1.64%	1.84%	-0.64%	25.02%	-55.03%	-22.98%	-0.05%	-0.86%	1.25%							
2020/09/18	19.17%	48.33%	11.05%	-6.66%	7.00%	26.28%	1.43%	1.81%	-1.02%	25.39%	-50.85%	-22.95%	-1.30%	0.30%	1.25%							
2020/09/21	16.75%	44.12%	8.91%	-7.99%	6.22%	24.21%	1.36%	1.83%	-1.61%	25.10%	-51.16%	-21.32%	-1.21%	0.40%	1.23%							
2020/09/22	14.03%	38.96%	7.72%	-10.13%	6.50%	22.78%	1.57%	1.94%	-2.07%	21.91%	-52.62%	-19.36%	-0.68%	-0.15%	1.24%							
2020/09/23	14.31%	41.09%	9.30%	-10.38%	5.60%	24.41%	1.83%	1.91%	-2.42%	18.97%	-52.92%	-19.68%	-1.25%	0.35%	1.25%							
2020/09/24	11.01%	38.53%	6.40%	-10.86%	5.31%	22.14%	1.86%	2.01%	-2.42%	17.20%	-52.21%	-17.44%	-0.62%	-0.25%	1.23%							
2020/09/25	13.13%	40.50%	8.56%	-10.47%	5.69%	22.81%	1.63%	1.92%	-2.66%	17.52%	-50.13%	-18.93%	-0.91%	0.00%	1.25%							
2020/09/28	15.80%	45.19%	10.39%	-9.39%	7.10%	23.12%	1.94%	2.00%	-1.91%	18.94%	-50.00%	-20.92%	-1.00%	0.10%	1.23%							
2020/09/29	16.89%	46.97%	11.72%	-8.54%	8.71%	23.71%	1.85%	2.04%	-1.70%	20.32%	-49.74%	-21.42%	-1.24%	0.56%	1.24%							
2020/10/05	19.93%	46.66%	14.74%	-6.20%	8.48%	26.30%	0.86%	1.68%	-1.32%	20.79%	-48.64%	-23.46%	-1.58%	0.66%	1.20%							
2020/10/06	20.20%	46.39%	15.97%	-6.03%	9.59%	26.71%	0.51%	1.63%	-1.87%	21.90%	-47.15%	-23.67%	-1.82%	0.66%	1.20%							
2020/10/07	21.16%	42.10%	13.86%	-6.72%	9.82%	25.43%	0.46%	1.64%	-0.64%	20.68%	-46.96%	-24.38%	-1.96%	1.27%	1.20%							
2020/10/08	19.44%	40.72%	15.52%	-6.98%	9.61%	26.08%	0.78%	1.73%	-0.93%	21.99%	-47.32%	-23.39%	-2.11%	1.16%	1.21%							
2020/10/12	19.95%	42.30%	17.42%	-6.65%	7.89%	26.92%	1.24%	1.75%	-0.26%	23.91%	-48.69%	-23.58%	-2.41%	1.36%	1.20%							
2020/10/13	20.04%	40.98%	18.95%	-6.18%	7.71%	27.38%	1.36%	1.81%	-0.13%	23.21%	-48.69%	-23.66%	-2.41%	1.36%	1.19%							
2020/10/14	17.65%	36.19%	17.20%	-8.31%	7.78%	25.47%	1.54%	1.85%	-0.59%	22.58%	-48.82%	-22.03%	-2.08%	1.31%	1.21%							
2020/10/15	16.63%	32.09%	15.62%	-9.02%	6.01%	26.33%	1.87%	1.84%	-0.34%	22.70%	-47.05%	-21.46%	-2.37%	1.46%	1.22%							
2020/10/16	15.30%	29.61%	15.31%	-10.92%	4.19%	25.95%	2.12%	1.89%	-1.06%	23.86%	-47.38%	-20.50%	-2.26%	1.56%	1.21%							
2020/10/19	17.10%	30.70%	15.62%	-9.03%	4.92%	28.41%	2.92%	2.07%	-0.89%	23.50%	-47.88%	-21.74%	-2.23%	1.25%	1.19%							
2020/10/20	17.92%	31.48%	14.04%	-9.98%	5.22%	28.53%	2.79%	2.07%	-0.93%	22.85%	-47.88%	-22.39%	-2.42%	1.55%	1.20%							
2020/10/21	17.63%	29.97%	14.61%	-9.92%	5.13%	28.19%	2.89%	2.11%	-1.48%	23.76%	-47.61%	-22.14%	-2.29%	1.49%	1.19%							
2020/10/22	15.29%	23.25%	12.79%	-11.86%	4.67%	28.15%	3.13%	2.00%	-1.52%	24.11%	-48.33%	-20.48%	-2.00%	1.29%	1.20%							
2020/10/23	16.43%	18.72%	14.20%	-11.82%	3.70%	27.60%	3.18%	2.08%	-1.36%	22.95%	-48.68%	-21.23%	-2.15%	1.29%	1.20%							
2020/10/26	15.79%	18.49%	13.16%	-12.43%	3.34%	24.65%	3.32%	2.23%	-1.39%	21.28%	-51.79%	-20.85%	-2.88%	1.94%	1.20%							
2020/10/27	0.148	0.1978	0.1249	-0.139	0.0317	0.2482	0.0318	0.0223	-0.0144	0.2211	-0.5179	-0.2019	-0.0293	0.0219	0.0121							
2020/10/28	0.1489	0.2382	0.1053	-0.1617	0.0183	0.2552	0.04	0.0241	-0.019	0.2218	-0.5234	-0.2041	-0.023	0.0129	0.0121							
2020/10/29	0.1402	0.2401	0.0818	-0.1763	0.0219	0.2647	0.04	0.0243	-0.0187	0.2155	-0.5251	-0.1971	-0.0157	0.0064	0.0123							
2020/10/30	0.1177	0.1985	0.0594	-0.1891	0.0124	0.2414	0.0363	0.0236	-0.022	0.2079	-0.5416	-0.1817	-0.0181	0.0089	0.0122							
2020/11/02	0.1219	0.2031	0.0664	-0.1829	0.0153	0.2512	0.0287	0.0224	-0.0123	0.1997	-0.5408	-0.1846	-0.0162	0.0074	0.0121							
2020/11/03	0.1443	0.228	0.0823	-0.1571	0.0153	0.2535	0.0286	0.0223	-0.011	0.2053	-0.5171	-0.2007	-0.0167	0.0054	0.0121							
2020/11/04	0.1352	0.2404	0.0945	-0.1514	0.0373	0.2543	0.0389	0.0248	-0.0106	0.2107	-0.5122	-0.1934	-0.0089	-0.001	0.0122							
2020/11/05	0.1524	0.2475	0.1141	-0.1391	0.0307	0.2836	0.0429	0.0252	-0.0072	0.221	-0.5221	-0.2066	-0.0138	0.0039	0.012							
2020/11/06	0.1546	0.246	0.1239	-0.1349	0.0419	0.264	0.0407	0.0249	-0.0034	0.2553	-0.5314	-0.209	-0.0198	0.0103	0.012							
2020/11/09	0.1746	0.2711	0.147	-0.1252	0.062	0.3051	0.0375	0.0243	-0.003	0.281	-0.5209	-0.2221	-0.0272	0.0177	0.0119							

1-1. 각각의 항목 정의

- Period (1mo,3mo,12mo,3yr) : 전체 데이터의 기간
(3년'치' 데이터, 6개월'치' 데이터 등)
- Data interval(daily, weekly) : 데이터 사이 간격
(daily: 9/18, 9/19, ... // weekly: 9/18, 9/25, ...)
- Return interval(1mo,3mo,12mo): “어떤 기간”의 수익률
의미하는지 (1mo = 월별 수익률, 12mo = 연간 수익률)

- Period : ‘overfitting’과 관련있음.
 - 미래의 시장 상황을 예측 가능하다면, 가장 비슷한 period 찾는 것이 BEST (근데 누가 그걸 알겠냐..)
- Data interval: ‘rebalancing’과 관련없음 (헛갈림 주의).
 - 1주 리밸런싱 한다고 해서, 1주 간격으로 DATA 모을 필요 없음
- Return interval : ‘momentum’과 관련있음.
 - 1개월 대비 수익률 = 단기 모멘텀 3개월 대비 수익률 = 중기 모멘텀 1년 대비 수익률 = 장기 모멘텀
 - 1개월이 단기 모멘텀? 1일, 1주가 단기 아니냐?
 - ▶ 1일, 1주는 오히려 “역” 모멘텀. 즉, 추세와 반대로 향한다

Source: YIG

[illegible]

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

- 'Sharpe Ratio' 를 최대화 하는 개별 종목의 '비중'을 구한다.
 - 대회 규정에 맞게 비중의 제약들을 감안한다.
- 참고, Expected Return (0.320%) : '수익률'의 산술 평균이기 때문에 실제 0.320% 번다는 의미 아님.
 - 실제 기대 수익률은 후에 '백테스팅'을 통해 알 수 있음
- Sharpe Ratio는 후에 Annualized를 통해 비교 가능하도록 통일함

8

Implementation: 실전에 들어가자

3. BEST 'Sharpe Ratio' 찾기

	Period 기준 정렬	Daily	Weekly	Daily-Weekly	Period avg		RI 기준 정렬 (Daily)	RI avg
Overfitting	1mo_1mo	9.651676675	5.658628686	3.993047989		단기 모멘텀	1mo_1mo	9.651676675
	1mo_3mo	21.57681177	22.22667081	-0.649859042			3mo_1mo	8.98591359
	1mo_12mo	6.565616127	5.20877167	1.356844456	12.5980349		12mo_1mo	1.609593034
	3mo_1mo	8.98591359	6.681359224	2.304554366			3yr_1mo	3.770221595
	3mo_3mo	10.09756316	9.013335059	1.084228102			1mo_3mo	21.57681177
	3mo_12mo	7.470672713	6.750107187	0.720565526	8.85138315		3mo_3mo	10.09756316
	12mo_1mo	1.609593034	1.166887175	0.442705858			12mo_3mo	2.782561504
	12mo_3mo	2.782561504	2.216571789	0.565989715			3yr_3mo	5.032914255
	12mo_12mo	1.603682976	0.475159758	1.128523219	1.9986125		1mo_12mo	6.565616127
	3yr_1mo	3.770221595	1.223464919	2.546756675			3mo_12mo	7.470672713
	3yr_3mo	5.032914255	1.496471841	3.536442414			12mo_12mo	1.603682976
Underfitting	3yr_12mo	3.858095512	1.227792227	2.630303286	4.22041045	장기 모멘텀	3yr_12mo	3.858095512

- **Daily vs Weekly 논쟁** : '그냥 daily로 가는게 맞는듯 하다'
- 1mo_3mo 빼고 전부 daily Sharpe가 더 높음
- "데이터가 더 많아서 정밀한 추정이 가능한 것" 논리는 항상 유효
- OR "데이터가 더 세분화 되므로 변동성(risk)가 더 작음"

- **Period 설정 관련 (의도적 overfitting을 할 것인지)**
- Period를 짧게 잡을수록 Sharpe는 올라가는 추세 보임 (3yr 제외)
- 즉, 오버피팅 효과는 분명히 존재한다
- 반면, RI(return interval)을 짧게 잡는다고 Sharpe가 올라가진 않음
- 즉, "단기" 모멘텀이 최고는 아님

How to Annualize a Sharpe Ratio

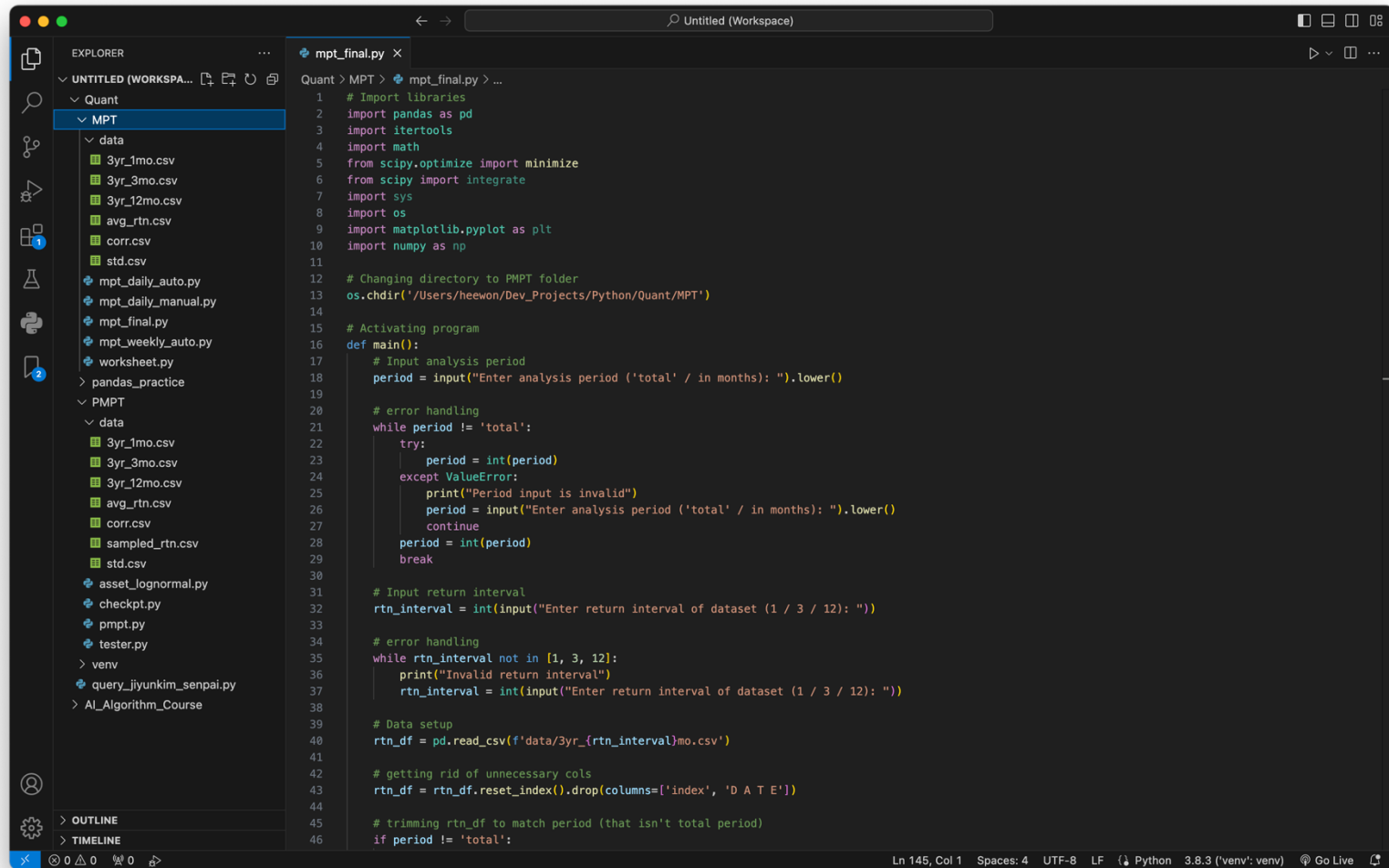
Sharpe Ratios are equal to the effective return divided by the standard deviation. Commonly, Sharpe Ratios on a daily, weekly or monthly basis are annualized by multiplying by the square root of the higher frequency time period. This is because

- The effective return is proportional to time.
- Assuming a Weiner process governs stock prices, variance is proportional to time. Hence standard deviation is proportional to the square root of time.

So you would scale a Sharpe Ratio by multiplying by $t/\sqrt{t} = \sqrt{t}$, where t is the frequency you are annualizing from. To summarize,

- Monthly Sharpe Ratios are annualized by multiplying by $\sqrt{12}$
- Daily Sharpe Ratios are annualized by multiplying by $\sqrt{252}$ (assuming 252 trading days in a year)

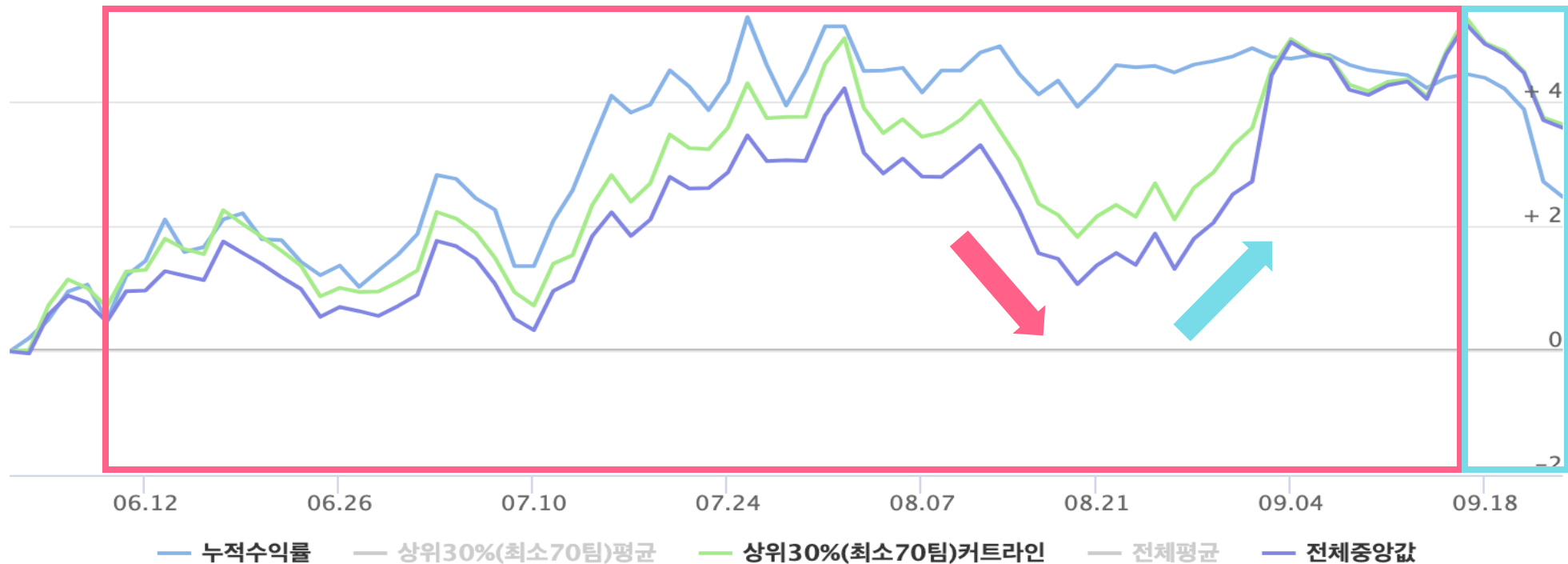
But (and this is a big but), a [paper](#) has demonstrated that this is misleading, and can often overestimate the actual Sharpe Ratio. The Sharpe Ratio is calculated from estimated quantities, and subject to errors. You have to take into account the specific properties of the returns distribution. For example, the Sharpe Ratio of hedge funds can be overestimated by 65% or more if you do not correctly model their serial autocorrelation.



The screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders 'Quant' and 'MPT', and various CSV files and Python scripts. The code editor displays the content of 'mpt_final.py'.

```
1 # Import libraries
2 import pandas as pd
3 import itertools
4 import math
5 from scipy.optimize import minimize
6 from scipy import integrate
7 import sys
8 import os
9 import matplotlib.pyplot as plt
10 import numpy as np
11
12 # Changing directory to PMPT folder
13 os.chdir('/Users/heewon/Dev_Projects/Python/Quant/MPT')
14
15 # Activating program
16 def main():
17     # Input analysis period
18     period = input("Enter analysis period ('total' / in months): ").lower()
19
20     # error handling
21     while period != 'total':
22         try:
23             period = int(period)
24         except ValueError:
25             print("Period input is invalid")
26             period = input("Enter analysis period ('total' / in months): ").lower()
27             continue
28         period = int(period)
29         break
30
31     # Input return interval
32     rtn_interval = int(input("Enter return interval of dataset (1 / 3 / 12): "))
33
34     # error handling
35     while rtn_interval not in [1, 3, 12]:
36         print("Invalid return interval")
37         rtn_interval = int(input("Enter return interval of dataset (1 / 3 / 12): "))
38
39     # Data setup
40     rtn_df = pd.read_csv(f'data/3yr_{rtn_interval}mo.csv')
41
42     # getting rid of unnecessary cols
43     rtn_df = rtn_df.reset_index().drop(columns=['index', 'DATE'])
44
45     # trimming rtn_df to match period (that isn't total period)
46     if period != 'total':
```

DB GAPS 투자대회 수익률 추이



- 데이터 선정 논리 정교화 과정을 거쳐 계속된 업데이트 끝에 MPT 모델 완성.
 - 따라서 하나의 전략으로 꼭 대회 진행한 것이 아니기에 완벽한 분석은 불가
- 1) 전 기간에 걸쳐, 전반적으로 일반적인 전략들 대비 Outperform**
- 2) 상승장에서는 상대적으로 underperform / 하락장에서는 Outperform**
- 파란색 영역은 다른 전략을 사용 (feat. 하락장을 대비해라)

퀀트적 투자결정을 '처음' 해보면서 느낀점

1. '정성'적인 변수를 고려 X '정량'변수만을 고려하여, 투자 결정을 내려보니 **마음이 비교적 더 편하다.** (완전히 편하지는 않더라..)
2. **기초** 수준의 '정량'적 투자 결정이지만,
그래도 웬만한 (90%) 사람들 보다는 낫더라
3. '정성'적인 느낌이 매우 강하게 올 때가 있을 때(ex. 원유가 더 오를 것 같다), 그 느낌을 퀀트 방식에 **일부 반영하는 것이 좋을 것 같다**
(실제로 원유 많이 오름)
4. '**데이터**'는 '논리'의 핵심이다. '**데이터**'가 있으니 비로소 '논리'적인 이라는 말이 어울린다
 - => 곧 자신감으로 연결되기 때문에 결국 **투자 행복도와** 연결됨
5. 결국 '오버피팅', '확증편향' 등 투자자의 심리적인 '**왜곡**' 요인이 들어갈 유혹이 너무 많다.
그러한 유혹에 넘어가는 순간 이도저도 아니게 된다.
6. **'하락장'에 대비해라. 괜히 변동성이 risk가 아니다.**

Post Modern Portfolio Theory (PMPT)

2

Assumption #1: 모든 투자수익이 정규분포를 따른다는 가정 → **Lognormal Distribution**

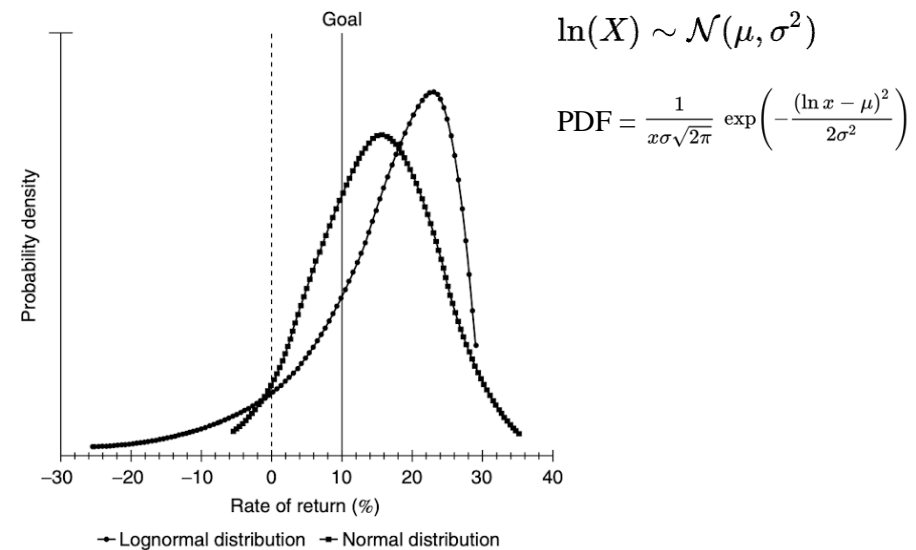
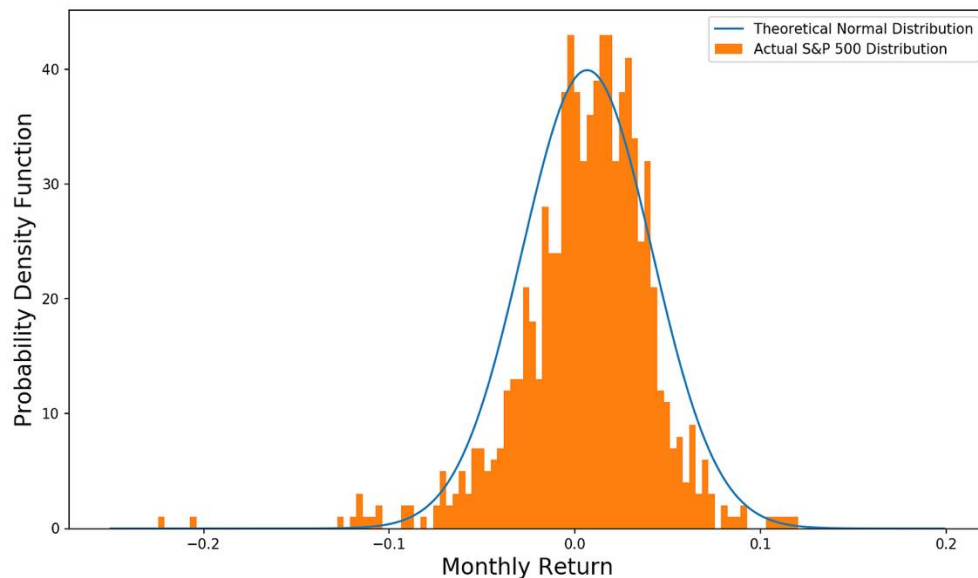


Figure 5.2 Distribution of S&P 500 lognormal fit to monthly returns, 1992-96

Table 5.3 Skewness statistics for five years, 1992-96

Index	Volatility skewness	% of total variance from returns above the mean	% of total variance from returns below the mean	Statistical skewness*
Lehman aggregate	0.48	32.35	67.65	-0.18
Russell 2000	0.59	37.19	62.81	0.59
S&P 500	0.63	38.63	61.37	-0.28
90-day T-bill	0.93	48.26	51.74	-0.01
MSCI EAFE	1.21	54.67	45.33	0.13

*This is the usual statistical measure of skewness (the third moment of the distribution). Zero skewness represents symmetry while positive and negative values indicate positive and negative skewness, respectively.

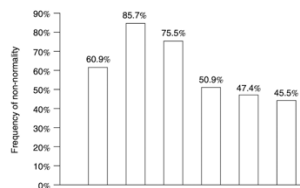


Figure 5.3 Frequency of non-normal returns for major asset classes

- 모든 투자수익이 정규분포를 따르지 않는다 (ex: S&P 500은 neg. skew)
- 5개 주요 index가 60% 확률로 정규분포를 따르지 않음 (Rom & Ferguson)
- PMPT는 투자수익에 더 적합한 fitting을 제공하는 **lognormal distribution** 활용 (3 parameter: μ, σ, τ - extreme value)

Theory: Limitations of MPT

Assumption #2: risk = 변동성 (variance, standard deviation) → **MAR & Downside Risk (Downside Deviation)**

MPT



PMPT



- “Risk = 변동성”으로 가정하는 MPT의 경우 평균에서 벗어난 모든 움직임이 리스크로 고려됨 → upside 변동성이 큰 것이 패널티로 적용
- PMPT는 “MAR(Minimum Acceptable Return) 아래로 생기는 변동성”을 리스크로 정의하며 실제 투자자에게 유의미한 Downside Risk 도출
- Downside Risk는 Downside Deviation 을 활용해 측정
- MAR을 조정해 투자자의 리스크 선호도 및 시장 전망 반영 가능 (bull market: higher MAR, bear market: lower MAR)

Theory: Limitations of MPT

성과지표의 업그레이드: Sharpe ratio → Sortino ratio

MPT

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

where:

R_p = return of portfolio

R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

PMPT

$$\text{Sortino Ratio} = \frac{R_p - r_f}{\sigma_d}$$

where:

R_p = Actual or expected portfolio return

r_f = Risk-free rate

σ_d = Standard deviation of the downside

6-step Implementation Process

1. 포트폴리오 구성 자산 별 τ (extreme val.) 구하기 (optional)
2. 포트폴리오의 lognormal function 구하기
3. MAR 선정
4. Downside Deviation 구하기
5. Sortino ratio 구하기
6. Sortino ratio maximizing optimization을 통해 최적 포트폴리오 비중 도출

Implementation #1: 포트폴리오 구성 자산 별 τ 구하기 (optional)

Random Sampling: Bootstrap Procedure

5.52, 1.62, -7.47, 5.6, -7.69, -1.02, 0.87, 4.57, 6.42, 3.08, 0.79, -0.49
 0.76, 1.01, -12.4, -3.07, 10.42, 5.12, 3.95, -0.3, 4.17, -2.36, 4.05, -5.15
 2.97, 3.9, 0.37, 1.01, 3.75, 3.47, 8.98, -6.35, 2.69, 3.88, -5.26, -2.44

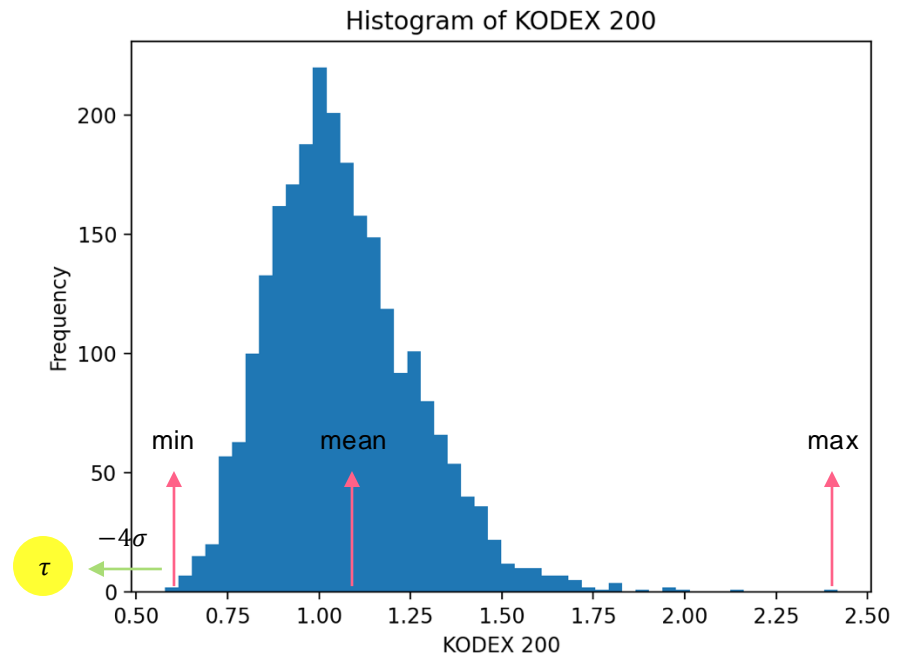
1.01, -7.69 -7.69

pmpt.py sampled_rtn.csv

Quant > PMPT > data > sampled_rtn.csv

KODEX 200	TIGER KOSDAQ	TIGER SAMP	TIGER SYNTH	ACE Japan	TIGER CHINA	KOSEF 10YK1	KBSTAR CRED	TIGER SYNTH	KODEX Gold	TIGER WTI	KODEX INVE	KODEX USD	KODEX USD	KOSEF Enh
1.55	1.41	1.33	1.32	1.02	0.91	0.96	0.97	0.97	0.78	1.35	0.76	1.18	1.11	1.02
1.26	1.1	0.86	1.35	1.03	0.96	0.92	0.99	0.99	1.05	1.55	0.84	1.23	0.98	1.02
1.15	1.34	1.1	1.01	1.12	1.06	1.07	1	1	1.3	1.58	0.72	1.05	0.88	1.02
1.01	1.26	0.93	1.68	1.31	0.78	0.99	0.99	1.02	0.83	1.09	0.68	1.26	1.04	1.01
0.8	0.82	0.93	0.87	1.18	0.9	0.89	1	1.06	0.82	1.49	0.96	0.96	1.12	1.01
1.11	0.93	1.13	1.29	1.76	1	0.98	1.01	0.95	1.17	1.16	0.83	1.17	0.94	1.01
1.83	0.84	1.87	1.4	1.83	1.05	0.88	1	1	1.35	1.08	1.01	0.99	1.04	1.01
1.19	1.13	1.07	1.2	1.43	0.81	0.99	0.99	1.06	0.93	1.05	0.66	1.14	0.89	1.02
1.3	1.04	1.04	1.5	1.11	1.02	0.91	0.97	1.13	0.95	1.45	1.07	0.92	0.92	1.02
1.22	1.63	1.21	1.24	1.47	0.75	1.03	1.03	1.03	1.17	1.18	0.89	1.18	1.09	1.02
1.4	1.14	1.2	1.13	1.07	0.86	0.99	1.03	1.09	0.95	1.33	0.73	1.19	0.86	1.01
1.17	1.44	1.22	1.35	0.84	0.86	1.12	1	1.05	0.84	1.28	1.08	1.11	1.03	1.02
0.72	0.94	0.74	0.94	1.07	1.37	0.88	1.02	1.03	1	1.66	0.76	1.05	0.82	1.01
1.15	0.99	1.08	1.18	1.12	0.85	0.93	1	1.02	1.11	1.53	1.1	1.09	0.97	1.02
1.29	1.25	1.03	1.24	1.03	1.07	0.91	1	0.93	0.78	1.11	0.76	0.99	1.13	1.01
1.27	0.84	1.02	1.16	0.97	0.75	0.98	0.97	0.97	0.93	1.06	0.87	1.14	1	1.02
1.06	0.58	0.92	0.92	1.59	0.81	0.93	1.02	1.08	0.99	1.17	0.89	1.06	1.07	1.02
1.24	0.68	1.19	1.18	0.79	0.92	0.92	1.02	1.03	1.03	1.07	0.9	1.12	1.15	1.02

Simulation for attaining τ



- Base data: 3 years, monthly returns data (3개년치 월 수익률)
- 랜덤으로 12개 월 수익률 sampling → compound → 랜덤 연 수익률 데이터 확보
- 위 random sampling 과정을 여러 번 (2,500 번) 반복해 large sample data 확보
- (optional) 시각화를 위해 히스토그램 도출

- Sample data의 mean, stdev, min, max 계산
- Min, max 중 mean과 더 가까운 값 선정
- 해당 값을 mean으로부터 4 * stdev 만큼 멀리 이동 → τ
- Ex) mean = 12%, std = 8%, min = -15%, max = 70% → extreme value = -15% - 4(8%) = -47%

Implementation #2: 포트폴리오의 lognormal function 구하기

포트폴리오 mean, stdev, τ 구하기

Lognormal function 공식

Review: Portfolio Expected Return and Variance

- Notations: N assets, S possible states, $p(s)$ denotes the probability of state s , $r_i(s)$ denotes asset i 's return in state s

- The portfolio weight for asset i : $w_i = \frac{\text{value of asset } i}{\text{value of all assets in portfolio}}$

- The portfolio return r_p in state s : $r_p(s) = w_1 r_1(s) + w_2 r_2(s) + \dots + w_N r_N(s)$

- Expected return of a portfolio P : $E(r_p) = \sum_{s=1}^S p(s) r_p(s)$ or $E(r_p) = \sum_{i=1}^N w_i E(r_i)$

- Variance of a two-asset portfolio P : $\sigma_p^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \text{Cov}(r_A, r_B)$

or

$$\sigma_p^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \rho_{A,B} \sigma_A \sigma_B$$

Correlation

Portfolio τ :

- Min = weighted combination of the min of all assets in portfolio
- Max = weighted combination of the max of all assets in portfolio

The three basic parameters estimated from the sample

Mean = sample mean

SD = sample standard deviation

τ = extreme value computed as described above

Some auxiliary parameters

$$\text{Dif} = |\text{Mean} - \tau|$$

$$\sigma = \ln \left(\left(\frac{\text{SD}}{\text{Dif}} \right)^2 + 1 \right)$$

$$\mu = \ln(\text{Dif}) - \sigma^2$$

$$\alpha = \frac{1}{(\sqrt{2\pi} \cdot \sigma)}$$

$$\beta = -\frac{1}{(2\sigma^2)}$$

Formula for the lognormal curve $f(x)$

If the extreme value is a minimum and x is greater than the extreme value then

$$f(x) = \frac{\alpha}{x - \tau} \cdot \exp(\beta \cdot (\ln(x - \tau) - \mu))$$

If the extreme value is a maximum and x is less than the extreme value then

$$f(x) = \frac{\alpha}{\tau - x} \cdot \exp(\beta \cdot (\ln(\tau - x) - \mu))$$

Implementation #3, 4: MAR 선정, Downside Deviation 구하기

MAR Selection

MAR을 조정해 투자자의 리스크 선호도 및 시장 전망 반영 가능 (bull market: higher MAR, bear market: lower MAR)

- 투자자 리스크 선호도:** 우리의 경우 DB GAPS 상위권이 목표이기 때문에 top 20를 달성하기 위한 MAR 채택 가능 (현재 대략 5%)

19	하락도 락이다	10,507,192,737	5.072
20	셋미나	10,507,034,220	5.070
21	전희진	10,504,347,799	5.043

- 시장 전망:** 9월 FOMC 이후, 미국 기준금리 추가인상 가능성이 높아진 환경에서 bearish 한 시장 예상 → lower MAR

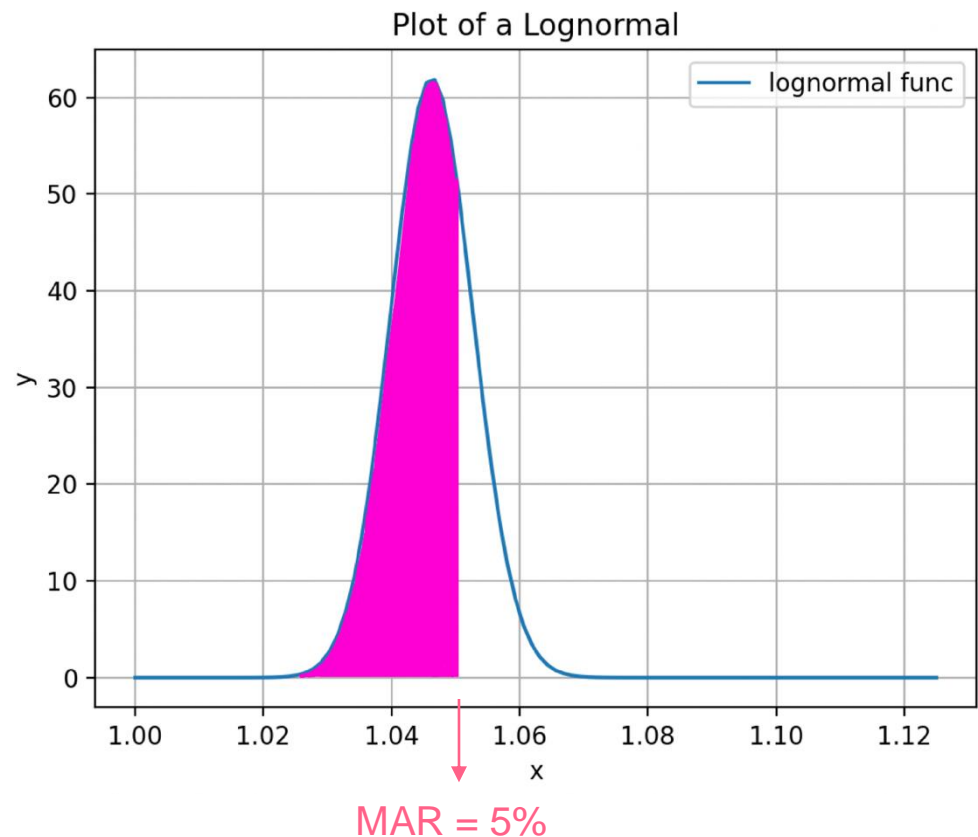


MAR = 5%

Source: Forsey, YIG

Downside Deviation Calculation

$$\text{Downside deviation} = \sqrt{\int_{-\infty}^{\text{MAR}} (\text{MAR} - x)^2 f(x) dx}$$



Implementation #5, 6: Sortino ratio 구하기, Optimization

Sortino Ratio Optimization for DB GAPS

```

message: Optimization terminated successfully
success: True
status: 0
  fun: -10.731924791734984
   x: [-8.773e-10  1.885e-01 ... -8.302e-10 -8.668e-10]
  nit: 9
  jac: [ 7.073e+00  4.126e+00 ...  8.578e+00 -1.411e+00]
 nfev: 149
 njev: 9
[-8.77262712e-10  1.88495230e-01  5.18827967e-02 -9.38581817e-10
 4.81172030e-02 -1.04462193e-09 -9.43660172e-10 -3.87800395e-10
 2.00000001e-01 -9.66323143e-10  1.13009544e-01  1.88495230e-01
 2.00000002e-01 -8.30186443e-10 -8.66802225e-10]

```

Max Sortino Ratio

Number of iterations
performed by optimizer

Optimal Portfolio Weight

['KODEX 200', 'TIGER KOSDAQ150', ... , 'KOSEF Enhanced Cash']

```

1  # Import libraries
2  import pandas as pd
3  import itertools
4  import math
5  from scipy.optimize import minimize
6  from scipy import integrate
7  import sys
8  import os
9  import matplotlib.pyplot as plt
10 import numpy as np
11
12 # Changing directory to PMPT folder
13 os.chdir('/Users/heewon/Dev_Projects/Python/Quant/PMPT')
14
15 # Activating program
16 def main():
17     # Input analysis period
18     period = input("Enter analysis period ('total' / in months): ").lower()
19
20     # error handling
21     while period != 'total':
22         try:
23             period = int(period)
24         except ValueError:
25             print("Period input is invalid")
26             period = input("Enter analysis period ('total' / in months): ").lower()
27             continue
28         period = int(period)
29         break
30
31     # Input return interval
32     rtn_interval = int(input("Enter return interval of dataset (1 / 3 / 12): "))
33
34     # error handling
35     while rtn_interval not in [1, 3, 12]:
36         print("Invalid return interval")
37         rtn_interval = int(input("Enter return interval of dataset (1 / 3 / 12): "))
38
39     # DB GAPS port
40     port = ['KODEX 200',
41             'TIGER KOSDAQ150',
42             'TIGER S&P500',
43             'TIGER SYNTH-EURO STOXX 50(H)',
44             'ACE Japan Nikkei225(H)',
45             'TIGER CHINA A300',
46             'KOSPI 10YKTB',

```

1. 포트폴리오 투자 수익률은 정규분포를 잘 따른다..! (특히 DB GAPS)
2. → 따라서 PMPT가 MPT 대비 갖는 edge가 크진 않았다
3. Random Sampling과 확률분포함수(PDF)를 사용하는 방법론이다 보니 Sample Size에 따라 결과값의 변동폭이 컸다
4. Lognormal로 특정 'market scenario'의 수익률 분포를 도출해 미래 시장 수익률을 전망하는 lognormal을 구할 수 있다

4.6 EXTENSION USING SCENARIOS

We know that next year's returns are dependent on economic and market forces that are changing. What can be done to include these changing conditions into our model? We briefly describe one approach, based on market scenarios. The idea is to divide past returns into a handful of groups based on the market scenario existing when they were generated. We then use our bootstrap approach to fit a lognormal curve to each asset for each scenario. Finally, we obtain a probability model for next year's return by using a mixture of these lognormal models with weights chosen according to our beliefs about next year's scenario.

Back-testing

3

결과 그래프

MPT 년도별 수치

1. 1mon-3mon

년도	2021	2022	2023
수익률	7.3%	5.7%	7.8%
고점 대비 하락률	-10.36%	-7.47%	-12.14%
변동성	0.55%	0.55%	0.55%

2. 3mon-3mon

년도	2021	2022	2023
수익률	9.6%	2.8%	8.0%
고점 대비 하락률	-11.38%	-5.5%	-11.79%
변동성	0.41%	0.41%	0.41%

PMPT 년도별 수치

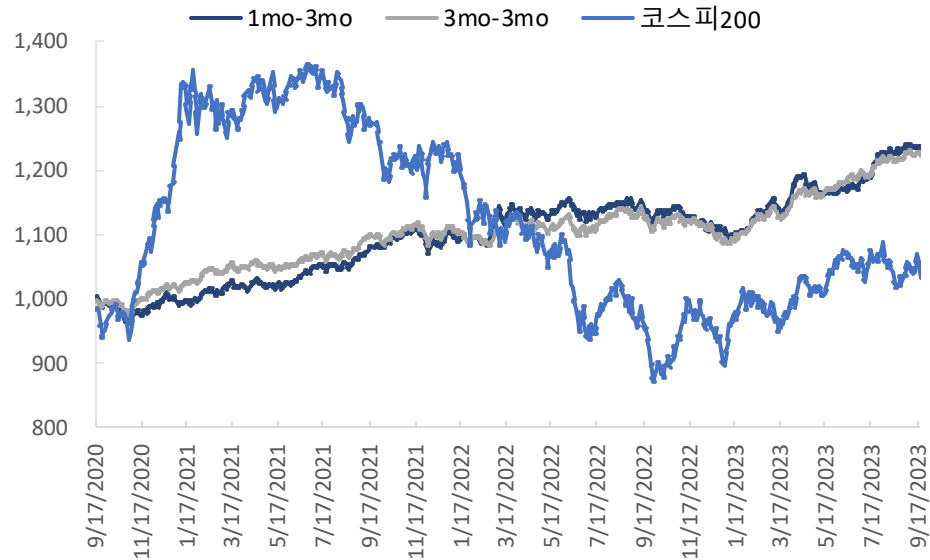
년도	2021	2022	2023
수익률	23.6%	0.3%	14.9%
고점 대비 하락률	-24%	-8.54%	-18.34%
변동성	1.11%	0.01%	0.03%

- Mpt의 경우 비교적 일정한 변동성을 가짐으로써, 안정적인 수익률을 창출했음을 추론할 수 있음
- Pmpt의 경우 2021년 다른 모델과 년도에 비해 높은 변동성을 가져 왔는데 이는 pmpt의 모델이 변동성을 고려하기 보다는 하락에 대한 변동성을 고려해서 생긴 거라고 추론함

결과 그래프

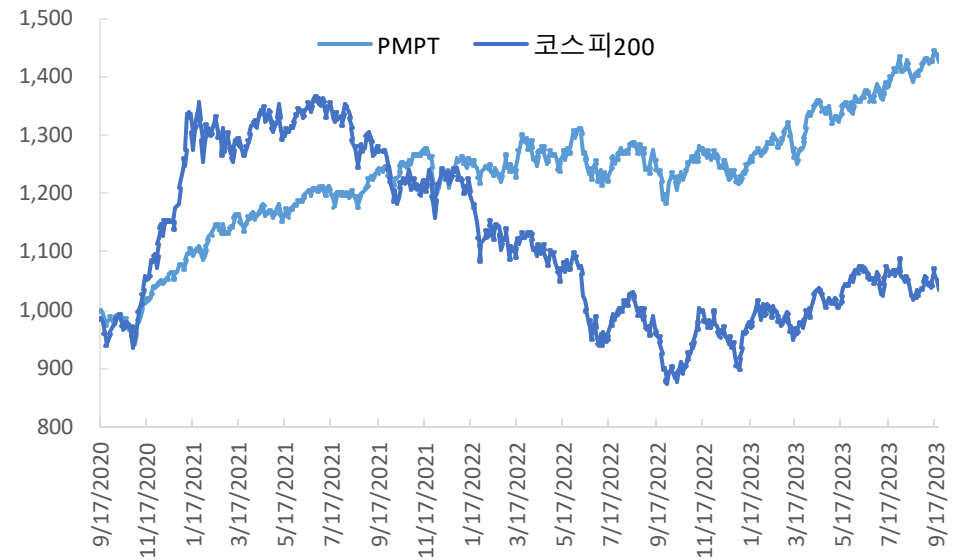
MPT 년도별 수익률 그래프

단위: x



PMPT 년도별 수익률 그래프

단위: x



- 코스피 200 지수가 코로나, 중국 재봉쇄 등 매크로에 의해 높은 등락폭(숫자)에 비해 MPT, PMPT 이론을 적용한 포트폴리오들은 각각 3년 최종 수익률 24%, 22%, 43%을 얻을 수 있었음
- 해당 모델들은 코스피 200 지수를 각각 202, 190, 392bp 상회하고 있으며 이를 통해 해당 모델은 현재 시점에서 유용한 전략임을 알 수 있음

1. 한 주식에 전 재산으로 투자 하거나 비과학적인 방법으로 투자할 경우(코스피 200에 올인), 손실을 입거나 낮은 수익률을 얻을 수 있음
2. 다양한 포트폴리오 최적화 모델을 통한다면 비효율적인 배분에 비해 높은 수익률을 기록할 수 있음
3. 하지만 다양한 모델이 있고, 각각의 한계점이 존재하며 모델 간의 순위가 없기에 다양한 모델을 통해 투자자 스스로 판단하여 투자하길 바람