

Portfolio Bandit with Thompson Sampling



김민석 서지완 이유경 이영송 염예진

CONTENTS

프로젝트 개요

데이터셋 생성

포트폴리오 구성

분석 결과

시사점 및 한계

CONTENTS

프로젝트 개요

- 주제 선정 이유
- 포트폴리오란?
- 2-Fund Theorem과 Portfolio 종류
- Thompson Sampling

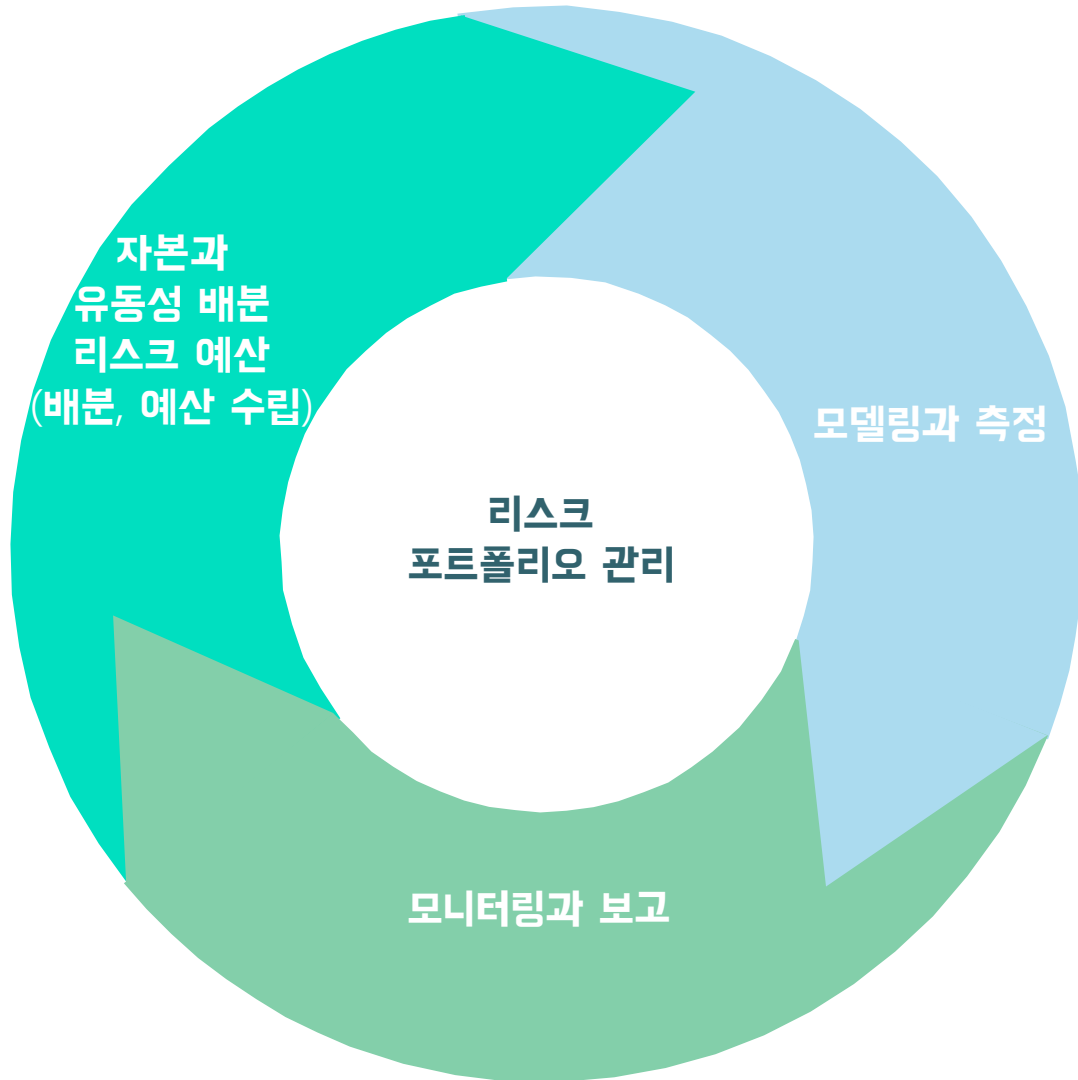
데이터셋 생성

포트폴리오 구성

분석 결과

시사점 및 한계

- 주제 선정 이유



- 전통적 재무이론 (포트폴리오 이론) + 통계학
- 달성 가능한 최적 위험 포트폴리오 모색

<목적>

포트폴리오를 구성하는 자산에 대해
변동성과 수익률을 기준으로

success / fail 베타분포를 학습하기

- 포트폴리오란?

* 포트폴리오란?

동일한 수익에 비해 변동성이 적고, 위험이 적은 방식으로 투자할 수 있게 함

N개의 자산에 대해 N이 충분히 많을 때, 개별 위험은 0에 가까워지고
Covariance만 남게 된다.



- 2-Fund Theorem과 Portfolio 종류

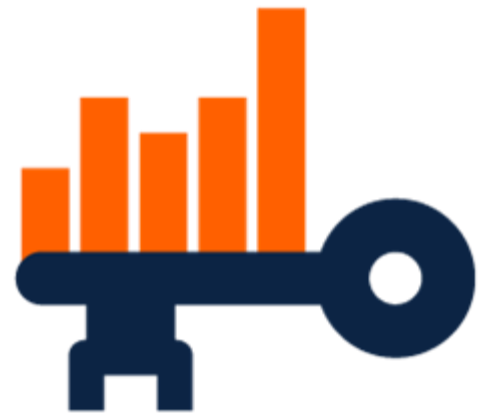
Equally Weighted Portfolio

- ✓ 포트폴리오 구성 자산들에 동일한 weight 부여
- ✓ 소규모 회사와 대규모 회사 모두 동일한 weight 부여
- ✓ 상대적으로 변동성이 크다
- ✓ Small cap 자산들의 상대적으로 큰 변동성 반영



Value Weighted Portfolio

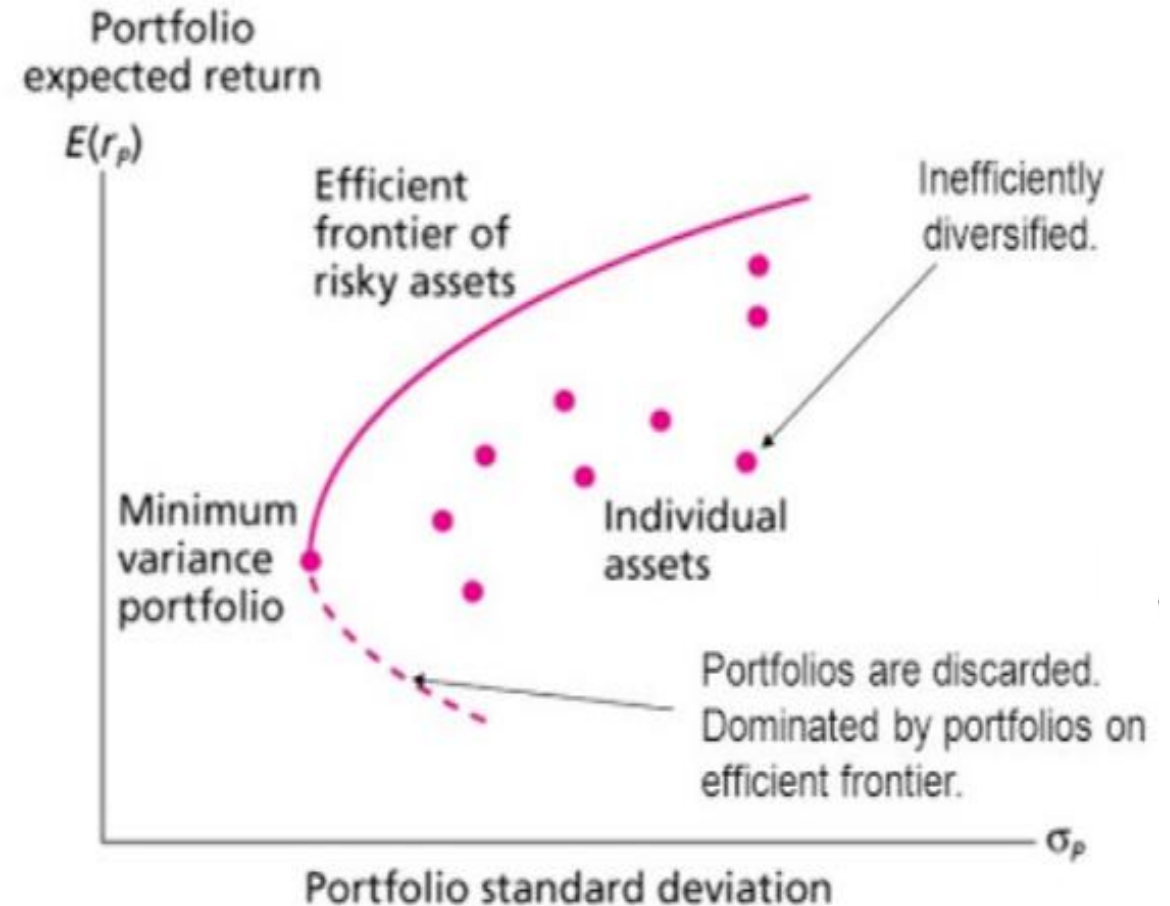
- ✓ $\text{Weight} = \text{기업 1의 자산} / (\text{포트폴리오를 구성하는 자산 시가총액})$
- ✓ 유통 주식수 X 주가
- ✓ 일반적으로 small cap 자산들이 고위험 고수익 자산으로 알려짐



- 2-Fund Theorem과 Portfolio 종류

Mean-Variance Portfolio

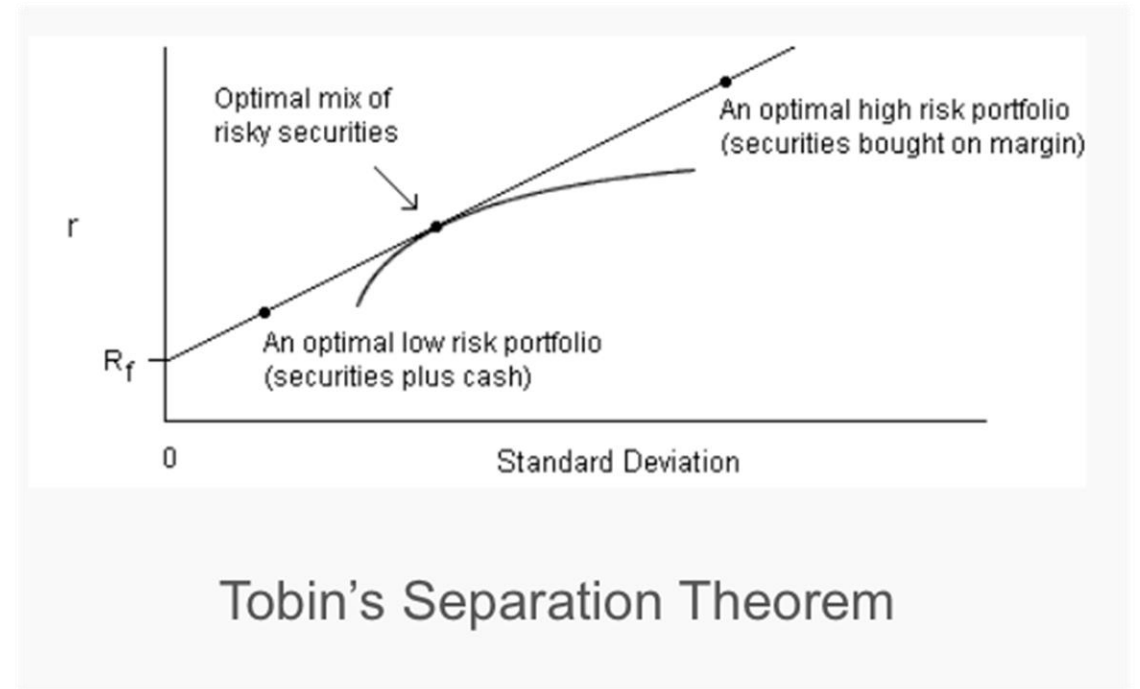
- ✓ 투자자의 선호체계는 평균-분산 기준을 사용하여 포트폴리오들을 비교 가능
- ✓ 최소분산 포트폴리오 (Minimum-Variance Portfolio) 아래에 위치하는 포트폴리오는 비효율적인 것으로 간주
- ✓ 위험 (standard deviation)을 최소화하는 weight 식별



- 2-Fund Theorem과 Portfolio 종류

2-Fund Theorem

- ✓ 미국의 경제학자 '토빈'에 의해 제안된 개념
- ✓ 본래 무위험 자산(Risk Free Asset)과 MV 포트폴리오의 Efficient Portfolio를 결합하기 위해 제안된 개념
- ✓ 무위험 자산이 아닌 2개의 펀드를 결합할 때, 각 모델이 가지는 bias가 줄어들 수 있다는 연구 결과 존재함



>> 2-Fund의 비율을 **Thompson Sampling**을 이용해 결정해보자!!!

- Thompson Sampling

✓ 베이지안 최적화 기법 중 하나인, 톰슨 샘플링 기법을 통해 n 개의 포트폴리오를 혼합

✓ 포트폴리오 블렌딩을 통하여 out-of-sample 데이터에도 좋은 성능 유지

Portfolio Blending via Thompson Sampling

Weiwei Shen^{†,‡} and Jun Wang[†]

[†]School of Computer Science and Software Engineering
East China Normal University, Shanghai, China

[‡]GE Global Research Center, Niskayuna, NY, USA,
realssw@gmail.com, wongjun@gmail.com

Abstract

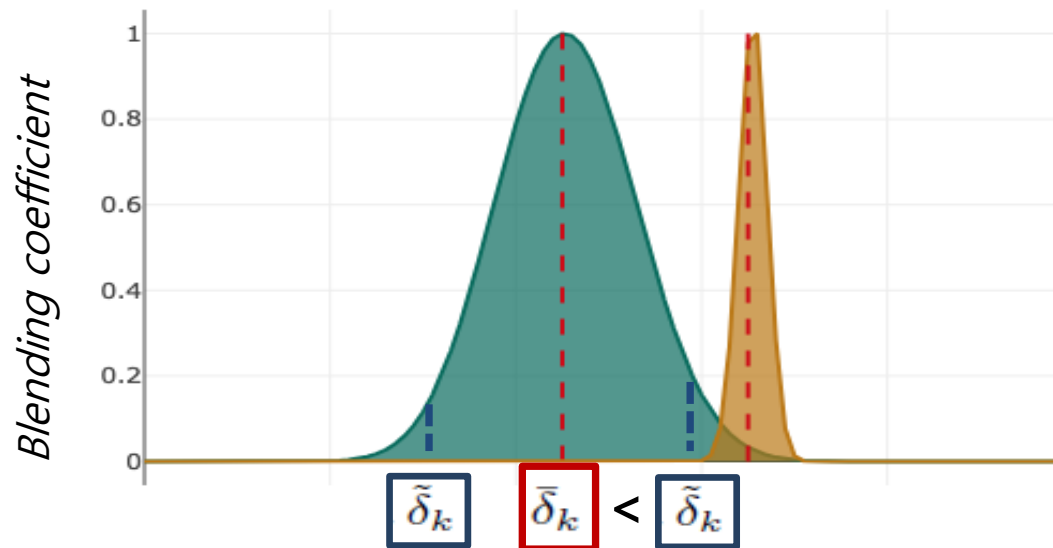
As a definitive investment guideline for institutions and individuals, Markowitz's modern portfolio theory is ubiquitous in financial industry. However, its noticeably poor out-of-sample performance due to the inaccurate estimation of parameters evokes unremitting efforts of investigating effective remedies. One common retrofit that blends portfolios from disparate investment perspectives has received growing attention. While even a naive portfolio blending strategy can be empirically successful, how to effectually and robustly blend portfolios to generate stable performance improvement remains less explored. In this paper, we present a novel online algorithm that leverages Thompson

the smallest variance. All other portfolios are "inefficient" in terms of having a higher variance representing a higher risk. However, due to the hurdle of accurately estimating involved parameters, the mean-variance portfolio often performs poorly in out-of-sample settings [Broadie, 1993].

On the other hand, the concept of blending portfolios arising from different investment perspectives to construct a new portfolio can be traced back to the ingenious two-fund separation theorem by [Tobin, 1958]. In the mean-variance framework, the two-fund separation theorem states that the efficient portfolio can be considered as a linear combination of two portfolios. Given the unsatisfactory out-of-sample performance of the mean-variance portfolio, the two-fund separation theorem naturally brings us the opportunity of blending portfolios to achieve better performance than the mean-variance portfolio and other heuristic strategies. However, as

- Thompson Sampling

	Bernoulli Bandit	Portfolio Blending
가치(value)	각 bandit의 성공 확률	분포로부터의 임의의 δ_k = blending coefficient 추출
행동(action)	특정 bandit의 슬롯 당김	MV or EW 포트폴리오 선택 (MV-VW 동일)
보상(reward)	해당 bandit의 성공/실패 결과에 따라 해당 베타 분포 업데이트	아래 비교 후 성공/실패 결과에 따라 베타 분포 업데이트 δ_k vs $\bar{\delta}_k$ δ_k 의 gross return vs $\bar{\delta}_k$ 의 gross return
Exploit & Explore	기대 수익이 낮은 Bandit도 확률적으로 돌려 보며 최고의 Bandit으로 수렴해 감	δ_k 값이 낮은 포트폴리오 구성도 선택해가며 정확한 cut-off point, $\bar{\delta}_k$ 을 찾아 감



$$\bar{\omega}_k^{\text{EM (VM)}} = \bar{\delta}_k \omega_k^{\text{MV}} + (1 - \bar{\delta}_k) \omega_k^{\text{EW (VW)}}$$

$$\begin{cases} \text{Success} & R_k^\top \tilde{\omega}_k^{\text{EM (VM)}} > R_k^\top \bar{\omega}_k^{\text{EM (VM)}} \text{ and } \tilde{\delta}_k > \bar{\delta}_k \\ \text{Success} & R_k^\top \tilde{\omega}_k^{\text{EM (VM)}} < R_k^\top \bar{\omega}_k^{\text{EM (VM)}} \text{ and } \tilde{\delta}_k < \bar{\delta}_k \\ \text{Failure} & R_k^\top \tilde{\omega}_k^{\text{EM (VM)}} > R_k^\top \bar{\omega}_k^{\text{EM (VM)}} \text{ and } \tilde{\delta}_k < \bar{\delta}_k \\ \text{Failure} & R_k^\top \tilde{\omega}_k^{\text{EM (VM)}} < R_k^\top \bar{\omega}_k^{\text{EM (VM)}} \text{ and } \tilde{\delta}_k > \bar{\delta}_k \end{cases}$$

CONTENTS

프로젝트 개요

데이터셋 생성

- 데이터셋 기본 정보
- 데이터셋 구성 순서

포트폴리오 구성

분석 결과

시사점 및 한계

- 데이터셋 기본 정보

- ✓ 데이터셋 3가지 생성 (주간 수익률, 시가 총액, 주별 주가)
- ✓ 15년 ~ 19년 260주, 회사 173개
- ✓ 테이블 shape : 260x173

EX) 주간 수익률 테이블 (469x173)

	삼성전자	현대차	POSCO	현대모비스	SK하이닉스	NAVER	한국전력	삼성전자우	신한지주	기아차
week										
15Y_week1	0.003005	-0.020619	0.000000	0.004211	-0.008360	0.035330	-0.024693	-0.009843	-0.048575	-0.019048
15Y_week2	-0.012835	0.055007	0.040392	0.000000	0.054123	0.001334	-0.005970	0.013753	-0.015964	0.019048
15Y_week3	0.020309	-0.028573	-0.063965	0.055172	-0.067858	0.034079	-0.060475	0.005836	-0.008079	-0.028710
15Y_week4	0.033678	-0.047487	-0.014788	0.013821	0.026248	-0.006464	0.082974	0.035259	0.073694	-0.103205
15Y_week5	-0.015234	0.038753	-0.053550	-0.021805	-0.034795	-0.081016	0.018562	-0.007519	-0.037285	0.002151
...
19Y_week48	-0.027399	-0.028399	0.017392	-0.041714	-0.014797	-0.014472	0.014260	-0.017997	-0.017056	-0.046946
19Y_week49	0.015748	-0.016598	0.004301	0.022067	0.002481	0.020203	0.012313	0.014423	0.011403	0.016279
19Y_week50	0.066124	-0.004193	0.039963	0.033174	0.093393	0.025389	-0.041041	0.071377	0.006780	0.025058
19Y_week51	0.014519	0.028988	0.004115	-0.001921	0.065526	0.024761	0.016261	0.002220	0.031045	0.006726
19Y_week52	0.005391	-0.016461	-0.029169	-0.015504	-0.005299	0.013495	-0.003591	0.006630	-0.054977	-0.010107

- ✓ 통근 샘플링 방법을 적용할 기간 : 2015년~2019년
- ✓ 일별 데이터가 아닌 주별 데이터
- ✓ 2017년 : 53주, 나머지 52주
- ✓ 2011년~2019년까지 데이터가 모두 존재하는 회사는 173개

- 데이터셋 구성 순서

1) 한국거래소(KRX)에서
KOSPI시장 시가총액 기준
으로 내림차순으로 정렬한
상위 231개 기업을
선정 및 고정
(기준 : 2015년 1월 6일 월요일)

Marketdata

화면번호/화면명검색

검색

시장동향

지수

주식

채권

파생

일반상품

ETF/ETN/ELW

해외연계시장 정보

통계

온라인간행물

데이터상품

[30001] 실시간 거래현황

종목정보

투자참고

순위정보

기타상품

상장현황

[80163] 공모도 현황

30015 시가총액 상/하위

시장구분

전체 KOSPI KOSDAQ KONEX

업종 구분

코스피

조회일자

20180201

조회

차트

컬럼

Excel

CSV

2020/03/17 PM 09:30:21 (20분 지연 정보)

순위	종목코드	종목명	시가	고가	저가	종가	대비
1	005930	삼성전자	2,531,000	2,548,000	2,486,000	2,491,000	▼ 4,000
2	000660	SK하이닉스	74,900	75,200	74,000	74,400	▲ 900
3	005935	삼성전자우	2,137,000	2,137,000	2,051,000	2,059,000	▼ 50,000
4	005380	현대차	161,500	162,000	158,500	160,500	▼ 1,500
5	005490	POSCO	385,000	400,000	381,500	395,000	▲ 14,500
6	035420	NAVER	910,000	932,000	909,000	923,000	▲ 13,000
7	051910	LG화학	434,500	440,000	418,500	418,500	▼ 13,500
8	207940	삼성바이오...	449,000	449,500	427,000	443,500	▲ 5,000
9	105560	KB금융	66,000	67,000	65,800	66,100	▼ 1,200
10	028260	삼성물산	143,000	144,000	141,000	141,500	▼ 500

1

2

3

4

5

6

7

8

9

10

>

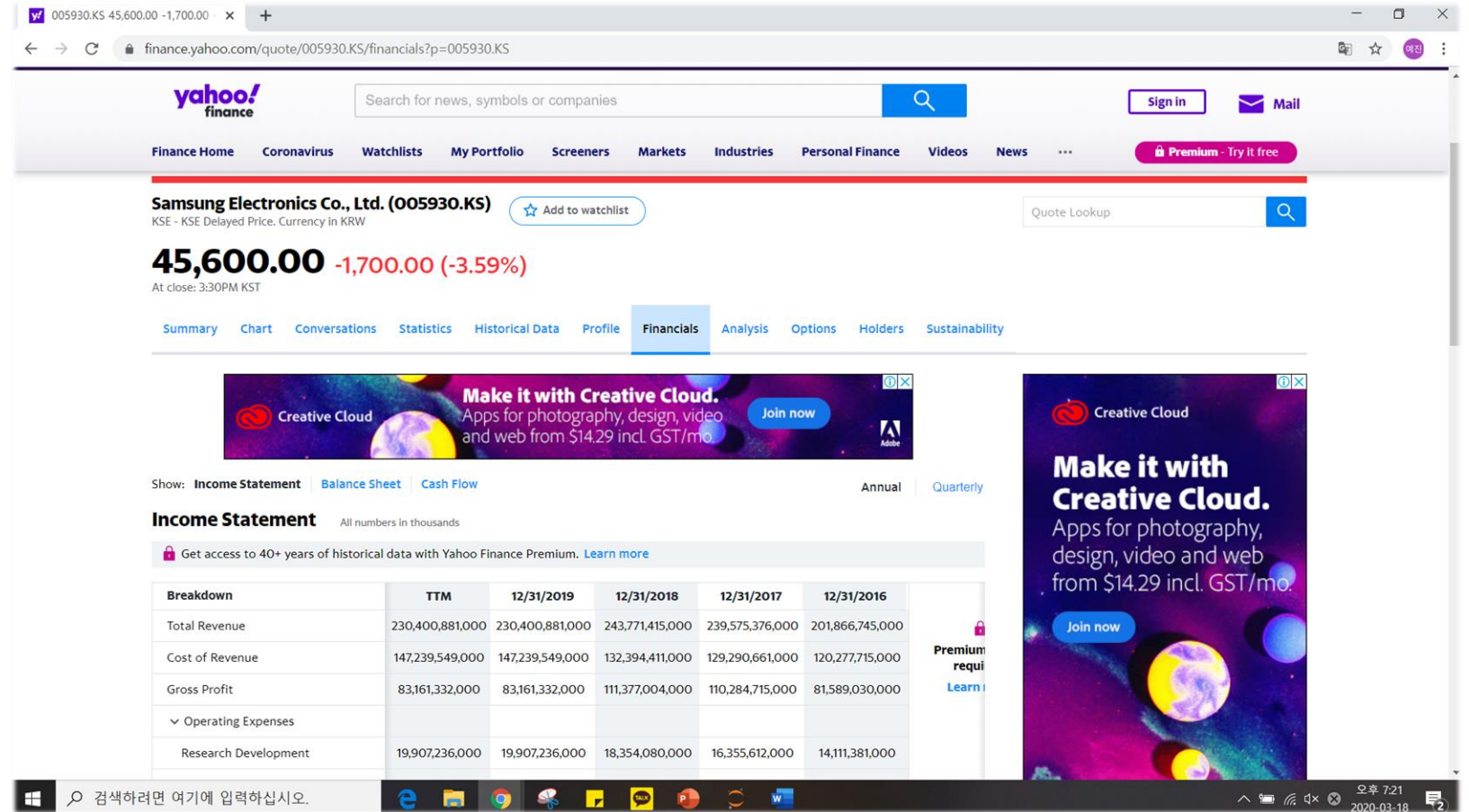
>>

2) 한국거래소에서 크롤링한 데이터셋에서 231개의 종목코드, 종목명, 시가총액 column만 사용

[illegible]

- 데이터셋 구성 순서

3) Yahoo Finance에서 231개의 기업에 해당하는 주가 데이터를 크롤링함



005930.KS 45,600.00 -1,700.00 x +

finance.yahoo.com/quote/005930.KS/financials?p=005930.KS

yahoo! finance

Search for news, symbols or companies

Sign in Mail

Finance Home Coronavirus Watchlists My Portfolio Screeners Markets Industries Personal Finance Videos News Premium - Try it free

Samsung Electronics Co., Ltd. (005930.KS) Add to watchlist

KSE - KSE Delayed Price. Currency in KRW

45,600.00 -1,700.00 (-3.59%)

At close: 3:30PM KST

Summary Chart Conversations Statistics Historical Data Profile Financials Analysis Options Holders Sustainability

Make it with Creative Cloud. Apps for photography, design, video and web from \$14.29 incl. GST/mo. Join now

Show: Income Statement Balance Sheet Cash Flow Annual Quarterly

Income Statement All numbers in thousands

Get access to 40+ years of historical data with Yahoo Finance Premium. Learn more

Breakdown	TTM	12/31/2019	12/31/2018	12/31/2017	12/31/2016
Total Revenue	230,400,881,000	230,400,881,000	243,771,415,000	239,575,376,000	201,866,745,000
Cost of Revenue	147,239,549,000	147,239,549,000	132,394,411,000	129,290,661,000	120,277,715,000
Gross Profit	83,161,332,000	83,161,332,000	111,377,004,000	110,284,715,000	81,589,030,000
Operating Expenses					
Research Development	19,907,236,000	19,907,236,000	18,354,080,000	16,355,612,000	14,111,381,000

Premium required Learn more

Make it with Creative Cloud. Apps for photography, design, video and web from \$14.29 incl. GST/mo. Join now

오후 7:21 2020-03-18

- 데이터셋 구성 순서

4) 2017년 10월 5일
기간에는 일주일 추석 연휴로
주가 데이터가 존재하지 않음
따라서 해당 주는 삭제

- 기업의 결측값 多

⇒ 기업 합병 및 상장 폐지 원인 Column 삭제

- **요일 결정값 약간 있는 경우**

⇒ 목요일 주가 데이터로 대체함

```
nan_list = ['14Y_week18', '14Y_week36', '15Y_week21', '15Y_week39', '16Y_week6',
            '16Y_week23', '16Y_week33', '16Y_week40', '17Y_week5', '17Y_week18',
            '17Y_week40', '17Y_week41', '17Y_week52', '18Y_week1', '18Y_week19',
            '18Y_week39', '18Y_week53', '19Y_week5', '19Y_week18']
```

[illegible]

- 데이터셋 구성 순서

5) Price 데이터로 로그 수익률을 생성했음

이를 통해 15년 ~ 19년의 주별
covariance matrix를 생성
(mean-variance portfolio
구성을 위한 것)

Weekreturn (주간 수익률)

	삼성전자	현대차	POSCO	현대모비스	SK하이닉스	NAVER	한국전력	삼성전자우
14Y_week1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14Y_week2	-0.009224	-0.002191	-0.019139	0.000000	0.013193	-0.042373	0.001417	-0.027737
14Y_week3	0.016086	0.013072	0.001609	0.026480	-0.034670	0.067798	-0.014266	-0.023183
14Y_week4	-0.018406	-0.033006	-0.039349	0.024098	-0.044390	-0.084352	-0.041062	0.011659
14Y_week5	-0.015601	0.035168	-0.013468	0.023530	0.073804	0.025470	0.062384	0.018790
...
19Y_week48	-0.027399	-0.028399	0.017392	-0.041714	-0.014797	-0.014472	0.014260	-0.017997
19Y_week49	0.015748	-0.016598	0.004301	0.022067	0.002481	0.020203	0.012313	0.014423
19Y_week50	0.066124	-0.004193	0.039963	0.033174	0.093393	0.025389	-0.041041	0.071377
19Y_week51	0.014519	0.028988	0.004115	-0.001921	0.065526	0.024761	0.016261	0.002220
19Y_week52	0.005391	-0.016461	-0.029169	-0.015504	-0.005299	0.013495	-0.003591	0.006630

- 데이터셋 구성 순서

6) **시가 총액의 비중**을 구하기
위해 연 초 유통 주식수를
연도별로 고정하여
주가와 곱함

	삼성전자	현대차	SK하이닉스	한국전력	NAVER	POSCO	삼성생명
14Y_week1	1.02157e+10	4.41012e+13	2.48112e+13	1.96735e+13	4.46196e+12	2.33601e+13	1.85401e+13
14Y_week2	1.01219e+10	4.40047e+13	2.51407e+13	1.97014e+13	4.27685e+12	2.29172e+13	1.83574e+13
14Y_week3	1.0286e+10	4.45837e+13	2.42841e+13	1.94223e+13	4.57686e+12	2.29541e+13	1.84487e+13
14Y_week4	1.00984e+10	4.31361e+13	2.32297e+13	1.8641e+13	4.20663e+12	2.20685e+13	1.82113e+13
14Y_week5	9.94209e+09	4.46802e+13	2.50089e+13	1.98409e+13	4.31515e+12	2.17732e+13	1.8814e+13
...
19Y_week48	7.42389e+12	2.67636e+13	5.86042e+13	1.81355e+13	5.6531e+12	2.02273e+13	1.45e+13
19Y_week49	7.54173e+12	2.6323e+13	5.87498e+13	1.83602e+13	5.76847e+12	2.03145e+13	1.46e+13
19Y_week50	8.05727e+12	2.62129e+13	6.4501e+13	1.76219e+13	5.9168e+12	2.11428e+13	1.516e+13
19Y_week51	8.17511e+12	2.69839e+13	6.8869e+13	1.79108e+13	6.06513e+12	2.123e+13	1.524e+13
19Y_week52	8.2193e+12	2.65433e+13	6.8505e+13	1.78466e+13	6.14754e+12	2.06197e+13	1.49e+13

- 데이터셋 구성

	기업 (173개)
주 별 날 짜 11~ 19 년	Price (주별 주가)

SHAPE : (470 X 173)

	기업 (173개)
주 별 날 짜 11~ 19 년	Return (주간 수익률)

SHAPE : (469 X 173)

	기업 (173개)
주 별 날 짜 15~ 19 년	Mkt cap (시가 총액)

SHAPE : (260 X 173)

CONTENTS

프로젝트 개요

데이터셋 생성

포트폴리오 구성

- EW
- VW
- MV (min variance)
- MV (max sharpe)
- Thompson(EW + MV min variance)
- Thompson(VW + MV min variance)
- 2019년 1주~50주에 가중치의 변화 PLOT

분석 결과

시사점 및 한계

- EW (Equally-Weighted Portfolio)

Weight : 1/N

삼성전자 X 1/173

현대차 X 1/173

.

.

.

두산건설 X 1/173

EW 포트폴리오 구성 :

가중치 X 기대 수익률

1) 가중치 생성

	삼성전자	현대차	POSCO	현대모비스	SK하이닉스	...	한화손해보험
week							
14Y_week2	-4.66355e-06	-7.94852e-05	-0.000332661	0	0.000454109	...	9.86578e-05
14Y_week3	5.15042e-06	0.000306291	3.08397e-05	0.00055494	-0.00190213	...	-1.49843e-05
14Y_week4	-5.44074e-06	-0.00121379	-0.000862333	0.000756899	-0.00178733	...	9.84557e-06
14Y_week5	-6.18675e-06	0.00146746	-0.000317059	0.000824894	0.00306058	...	8.99094e-06
14Y_week6	8.82036e-07	0.000287891	-0.000591912	0.000688618	0.0011552	...	-7.083e-06
...
19Y_week48	-0.00923491	-0.00105388	0.000304302	-0.000882299	-0.000821172	...	3.93458e-07
19Y_week49	0.00274881	-0.000720121	6.29452e-05	0.000547118	0.000184563	...	-9.3188e-06
19Y_week50	0.0136583	-0.000166122	0.000677892	0.000663842	0.00719976	...	2.49068e-06
19Y_week51	0.00277395	0.00181744	4.9788e-05	-3.56669e-05	0.00373077	...	1.98851e-06
19Y_week52	0.0010129	-0.000373888	-0.000420261	-0.000170722	-0.000374268	...	-1.46753e-06

2) 포트폴리오 구성

	return_EW
week	
15Y_week1	-0.007394
15Y_week2	0.001896
15Y_week3	-0.003627
15Y_week4	0.011634
15Y_week5	0.024299
...	...
19Y_week48	-0.008233
19Y_week49	-0.014826
19Y_week50	0.026661
19Y_week51	0.010754
19Y_week52	-0.003587

- VW (Value-Weighted Portfolio)

Weight :
시가총액/ 포트폴리오
구성하는 기업의 전체
시가총액

VW 포트폴리오 구성 :

Weight x 기대수익률

1) 가중치 생성

	삼성전자	현대차	POSCO	현대모비스	SK하이닉스	...	한화손해보험
week							
14Y_week2	-4.25059e-05	-1.00945e-05	-8.81983e-05	0	6.07966e-05	...	0.000270024
14Y_week3	7.41297e-05	6.02398e-05	7.41441e-06	0.000122028	-0.00015977	...	-0.000113796
14Y_week4	-8.48181e-05	-0.000152103	-0.000181332	0.000111049	-0.000204563	...	0.000141809
14Y_week5	-7.18943e-05	0.000162067	-6.2066e-05	0.000108435	0.00034011	...	9.21716e-05
14Y_week6	1.80798e-05	3.96416e-05	-0.000134732	0.000180152	0.000125771	...	-0.000124874
...
19Y_week48	-0.000126263	-0.000130873	8.01463e-05	-0.000192231	-6.81881e-05	...	2.44472e-05
19Y_week49	7.25731e-05	-7.6488e-05	1.98207e-05	0.000101692	1.1435e-05	...	-0.000199309
19Y_week50	0.00030472	-1.9322e-05	0.000184162	0.000152875	0.000430384	...	5.06411e-05
19Y_week51	6.69093e-05	0.000133583	1.89642e-05	-8.8536e-06	0.000301961	...	0.000172986
19Y_week52	2.48426e-05	-7.58584e-05	-0.000134418	-7.14479e-05	-2.44213e-05	...	-4.0602e-05

2) 포트폴리오 구성

	return_VW
week	
15Y_week1	-0.012904
15Y_week2	0.008866
15Y_week3	-0.013704
15Y_week4	0.019657
15Y_week5	0.016394
...	...
19Y_week48	-0.011578
19Y_week49	-0.002344
19Y_week50	0.032153
19Y_week51	0.016586
19Y_week52	-0.010228

- MV (Mean-Variance Portfolio) -> min variance

공식 $\min_{\pi} \pi' \Sigma \pi,$

제약조건 $\pi' \mu = c$

$$\pi' \mathbf{1} = 1$$

$$\pi \geq 0$$

- 제약조건을 만족하면서 해당 공식을 최소화하는 WEIGHT를 찾기

1) 가중치 생성

	삼성전자	현대차	POSCO	현대모비스	SK하이닉스	...	한화손해보험	JB금융지주	부광약품	부산가스	크라운제과
week											
19Y_week1	0.57	0	1.06	0	2.43	...	0	0	0	14.04	0
19Y_week2	0.55	0	1.08	0	2.49	...	0	0	0	13.75	0
19Y_week3	0.74	0	0.67	0	2.41	...	0	0	0	14.5	0
19Y_week4	0.5	0	0.6	0	2.09	...	0	0	0	15.09	0
19Y_week5	0.46	0	0.55	0	2.02	...	0	0	0	14.99	0
...
19Y_week48	0	0.43	0.54	0	3.13	...	0	0	0	18.14	0
19Y_week49	0	0	0.3	0	3.11	...	0	0	0	17.93	0
19Y_week50	0	0	0.29	0	3.08	...	0	0	0	18.03	0
19Y_week51	0	0.27	0.08	0	3.46	...	0	0	0	16.25	0
19Y_week52	0	0.48	0.14	0	3.45	...	0	0	0	16.47	0

2) 포트폴리오 구성

	EW	VW	MV_sha	MV_var
week				
15Y_week1	-0.007394	-0.012904	0.029309	0.003129
15Y_week2	0.001896	0.008866	0.019637	0.008332
15Y_week3	-0.003627	-0.013704	0.011890	-0.017004
15Y_week4	0.011634	0.019657	0.021816	0.012279
15Y_week5	0.024299	0.016394	0.013613	0.010659
...
19Y_week48	-0.008233	-0.011578	-0.017323	-0.008585
19Y_week49	-0.014826	-0.002344	0.023013	-0.007494
19Y_week50	0.026661	0.032153	0.034755	0.012831
19Y_week51	0.010754	0.016586	0.025721	0.015851
19Y_week52	-0.003587	-0.010228	-0.008010	-0.020156

- MV (Mean-Variance Portfolio) -> max sharpe

공식 $\min_{\pi} \pi' \Sigma \pi,$

제약조건 $\pi' \mu = c$

$$\pi' \mathbf{1} = 1$$

$$\pi \geq 0$$

Sharpe란?

- 초과 수익률/표준편차
- 위험을 고려한 경우 최고의 수익률로 해석 가능

1) 가중치 생성

	삼성전자	현대차	POSCO	현대모비스	SK하이닉스	...	한화손해보험	JB금융지주	부광약품	부산가스	크라운제과
week											
19Y_week1	4.65	0	0	0	0	...	0	0	0.47	0	0
19Y_week2	4.56	0	0	0	0	...	0	0	0	3.5	0
19Y_week3	4.62	0	0	0	0	...	0	0	0	0	0
19Y_week4	4.7	0	0	0	0	...	0	0	0	0	0
19Y_week5	4.76	0	0.01	0	0.01	...	0	0	0	0.01	0
...
19Y_week48	4.93	0	0	0	5.44	...	0	0	0	0	0
19Y_week49	4.73	0	0	0	5.27	...	0	0	0	0	0
19Y_week50	4.74	0	0	0	6.54	...	0	0	0	0	0
19Y_week51	4.83	0	0	0	7.32	...	0	0	0	0	0
19Y_week52	4.75	0	0	0	6.88	...	0	0	0	0	0

2) 포트폴리오 구성

	EW	VW	MV_sha	MV_var
week				
15Y_week1	-0.007394	-0.012904	0.029309	0.003129
15Y_week2	0.001896	0.008866	0.019637	0.008332
15Y_week3	-0.003627	-0.013704	0.011890	0.017004
15Y_week4	0.011634	0.019657	0.021816	0.012279
15Y_week5	0.024299	0.016394	0.013613	0.010659
...
19Y_week48	-0.008233	-0.011578	-0.017323	0.008585
19Y_week49	-0.014826	-0.002344	0.023013	0.007494
19Y_week50	0.026661	0.032153	0.034755	0.012831
19Y_week51	0.010754	0.016586	0.025721	0.015851
19Y_week52	-0.003587	-0.010228	-0.008010	0.020156

- Thompson Sampling (EW(VW) + MV(min variance))

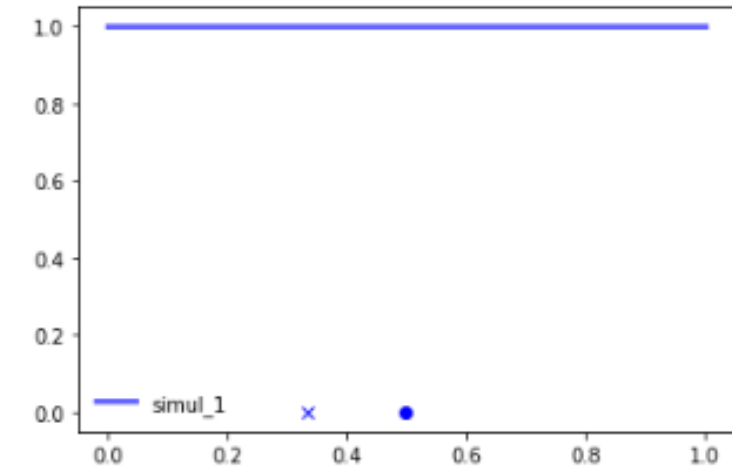
12 8
0.60,0.69 & -0.00,-0.00
False,True
12 9
0.57,0.53 & -0.00,-0.00
True,False
12 10
0.55,0.54 & 0.01,0.01
True,True
13 10
0.57,0.40 & -0.01,-0.02
True,True
14 10
0.58,0.69 & 0.00,0.01
False,False
15 10
0.60,0.52 & 0.00,0.00
True,False
15 11
0.58,0.48 & -0.02,-0.02
True,False

...

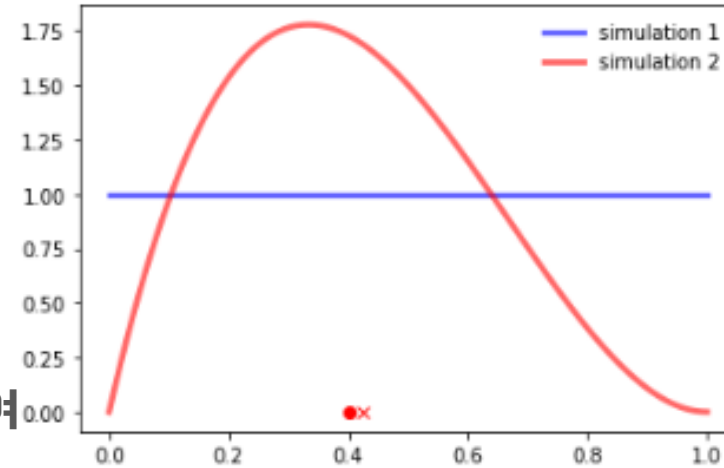
False,False
98 97
0.50,0.49 & 0.00,0.00
True,False
98 98
0.50,0.42 & 0.00,0.00
True,True
99 98
0.50,0.53 & -0.01,-0.01
False,False
100 98
0.51,0.51 & -0.01,-0.01
False,False
101 98
0.51,0.51 & -0.01,-0.01
False,False
102 98
0.51,0.55 & 0.02,0.02
False,True
102 99
0.51,0.60 & 0.01,0.01
False,False
103 99

```
# TS loop
for j in range(window_len) :
    # mean delta & weight & net retrun
    mean_delta = beta.mean(beta_a, beta_b)
    mean_weight_EM = mean_delta * portpolio1_weight.iloc[(j+i+8),] + #
        (1-mean_delta) * portpolio2_weight.iloc[(j+i+8),]
    mean_netreturn_EM = return_full.iloc[j+i+8,].T.dot(mean_weight_EM)
    # random delta & weight & net retrun
    rand_delta = beta.rvs(beta_a, beta_b)
    rand_weight_EM = rand_delta * portpolio1_weight.iloc[(j+i+8),] + #
        (1-rand_delta) * portpolio2_weight.iloc[(j+i+8),]
    rand_netreturn_EM = return_full.iloc[j+i+8,].T.dot(rand_weight_EM)
    # update
    if (mean_netreturn_EM > rand_netreturn_EM) == (mean_delta > rand_delta):
        # success
        beta_a += 1
    else :
        # failure
        beta_b += 1
    print(beta_a, beta_b)
```

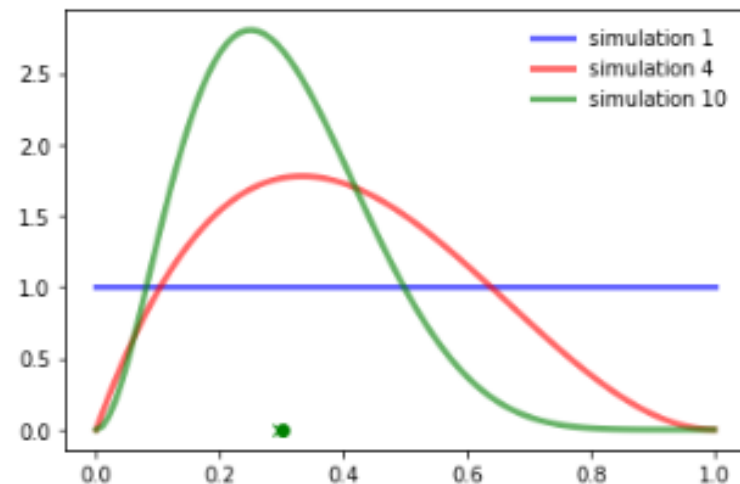
- Thompson Sampling (EW(VW) + MV(min variance))



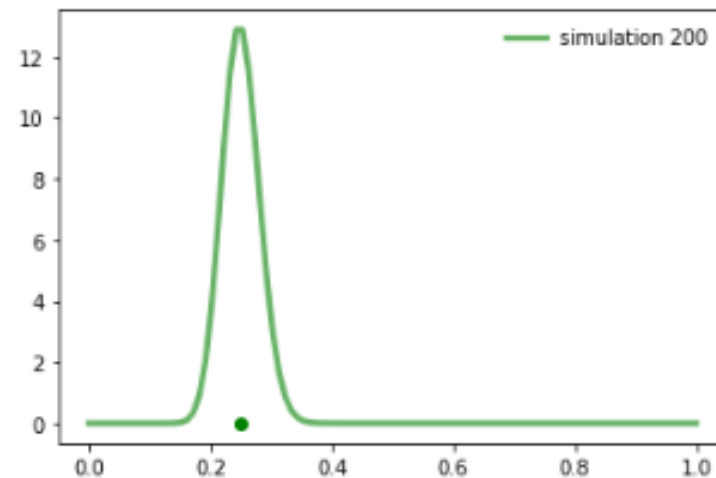
1. 0회 시행
-Uniform, $\alpha=1$, $b=1$ 부여



2. 3회 시행
-Exploit한 것의 분포 변화

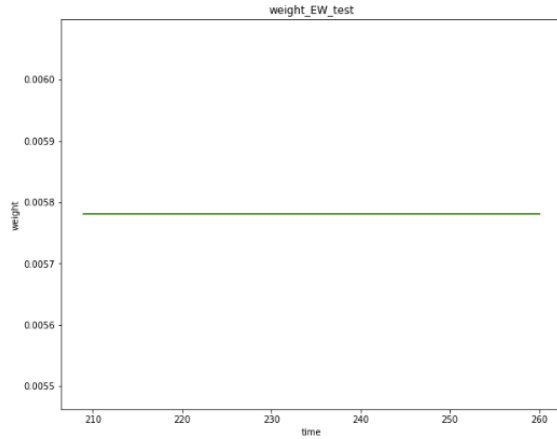


3. 10회 시행
-분포가 점점 밀도 있게
Mean으로 좁혀짐

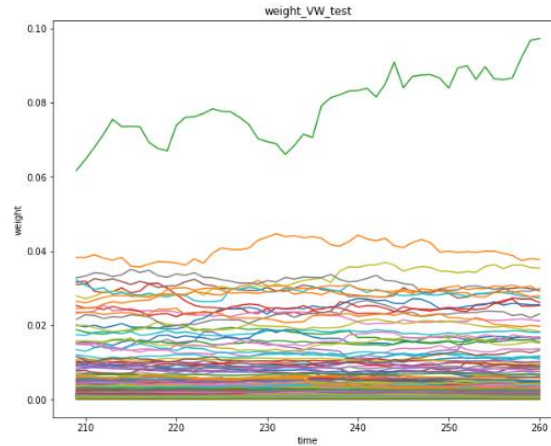


4. 200회 시행
-random x와 mean이
거의 겹칠 정도로 분포가
수렴함

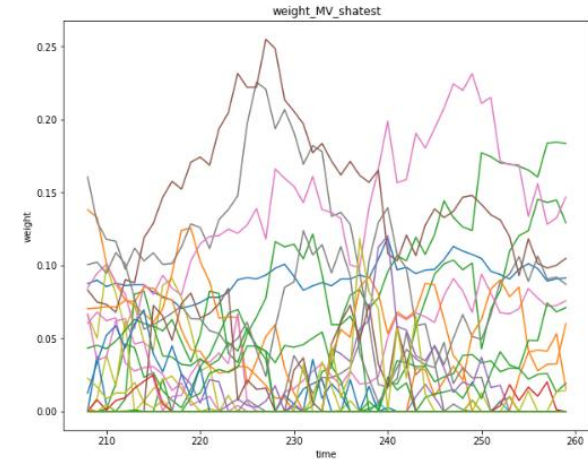
- 2019년 1주~50주에 가중치의 변화 PLOT



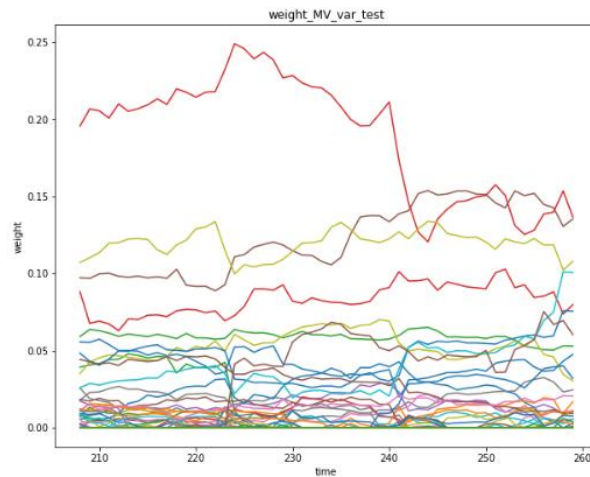
- EW Weight



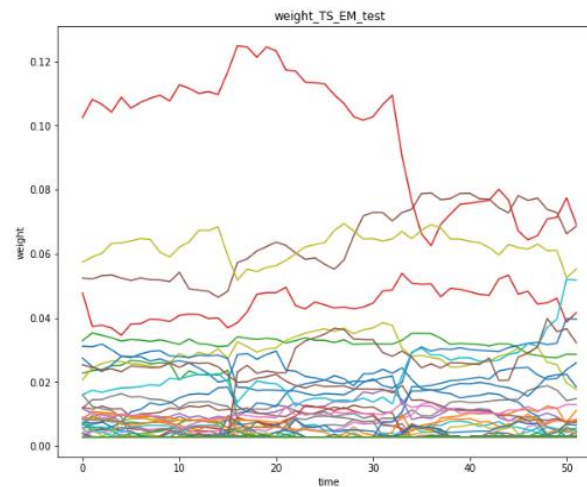
- VW Weight



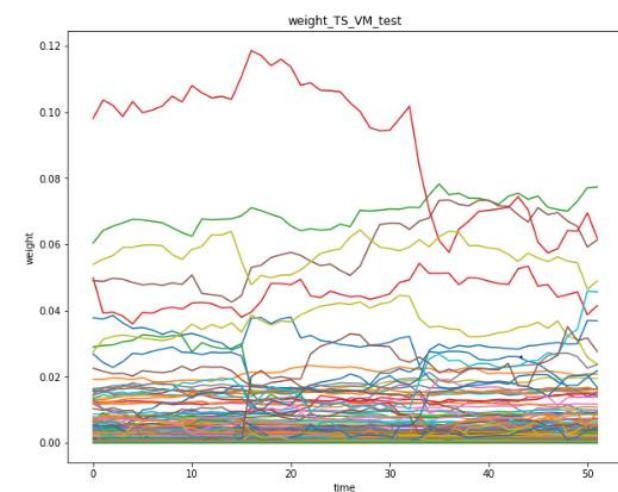
- MW_SHA Weight



- MW_VAR Weight



- TS_EM_MV Weight



- TS_VM_MV Weight

CONTENTS

프로젝트 개요

데이터셋 생성

포트폴리오 구성

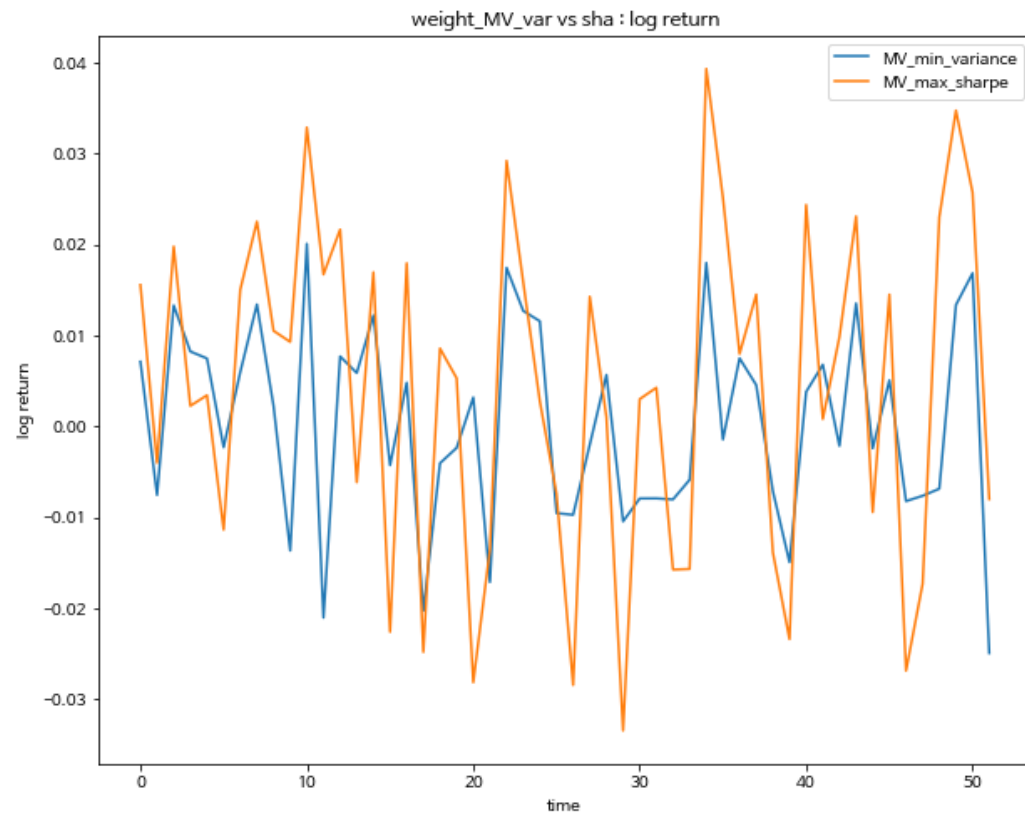
분석 결과

- 결과 비교

시사점 및 한계

- 결과 비교

Sharpe-ratio	EW	VW	MV_var	MV_Sha	TS EM	TS VM	TS EM_Sha	TS VM_Sha
Return-rate	-0.0508	0.0684	0.0326	0.2320	-0.0204	0.0586	0.1368	0.1824



**MV_min_variance와 MV_max_sharpe의
log(return)값 비교 PLOT**

CONTENTS

프로젝트 개요

데이터셋 생성

포트폴리오 구성

분석 결과

시사점 및 한계

- 시사점 및 한계

〈의도〉

포트폴리오 모델에 반영된 bias를
제거하기 위한 펀드 결합

〈시사점〉

툼슨 샘플링을 통해 목적에 부합하는
적정 cut-off 포인트를 생성할 수 있음

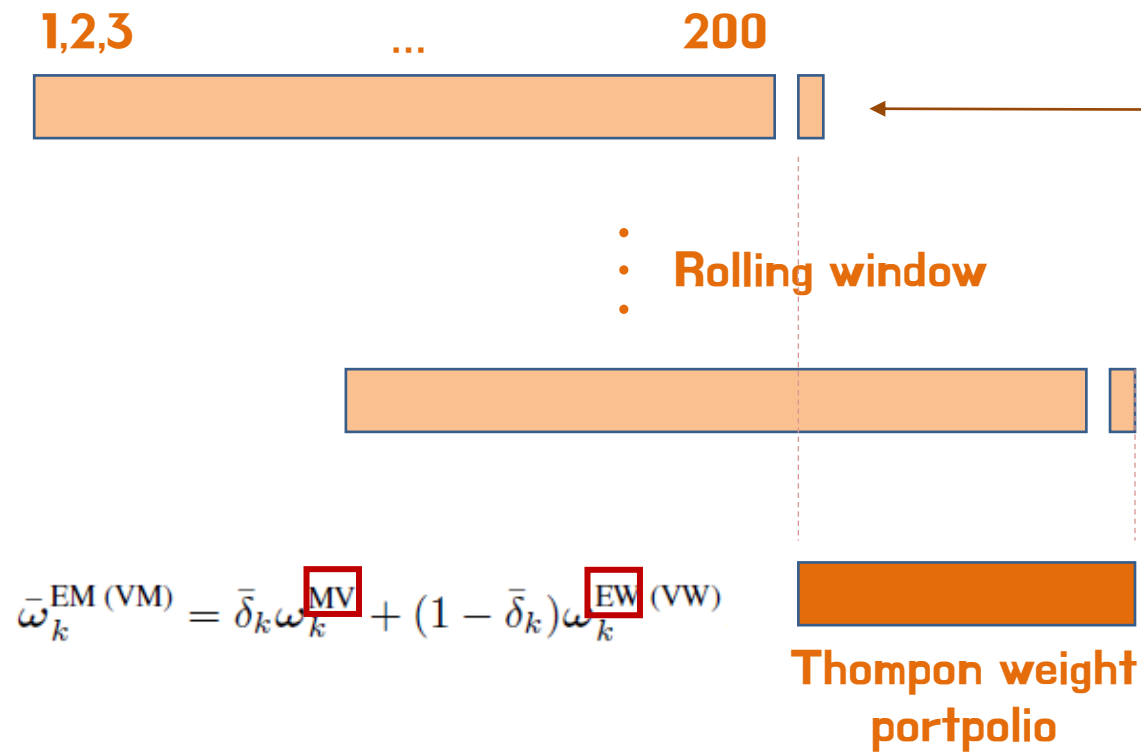
〈한계〉

1) MV를 잘못 구했을 가능성
(코딩, overfitting)

2) 데이터셋 문제

3) 톼슨 샘플링의 success/fail 기준에서
noise 가능성

- Thompson Sampling (Detail)

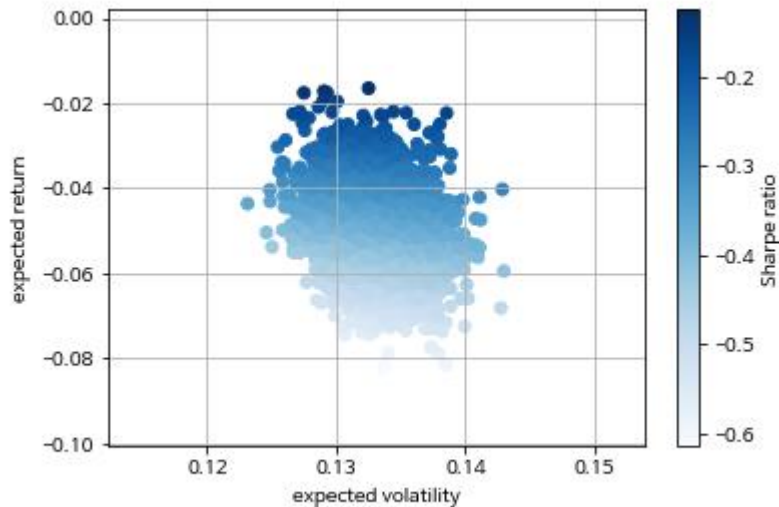


- ✓ 한 시점의 포트폴리오를 지난 200주간의 최적화된 blending 계수값(사후 베타분포의 평균)을 찾아냄

18	12 8	False,False
	0.60,0.69 & -0.00,-0.00	98 97
19	False,True	0.50,0.49 & 0.00,0.00
	12 9	True,False
	0.57,0.53 & -0.00,-0.00	98 98
20	True,False	0.50,0.42 & 0.00,0.00
	12 10	True,True
	0.55,0.54 & 0.01,0.01	99 98
	True,True	0.50,0.53 & -0.01,-0.01
	13 10	False,False
	0.57,0.40 & -0.01,-0.02	100 98
	True,True	0.51,0.51 & -0.01,-0.01
	14 10	False,False
	0.58,0.69 & 0.00,0.01	101 98
	False,False	0.51,0.51 & -0.01,-0.01
	15 10	False,False
	0.60,0.52 & 0.00,0.00	102 98
	True,False	0.51,0.55 & 0.02,0.02
	15 11	False,True
	0.58,0.48 & -0.02,-0.02	102 99
	True,False	0.51,0.60 & 0.01,0.01
	.	False,False
	.	103 99
	.	

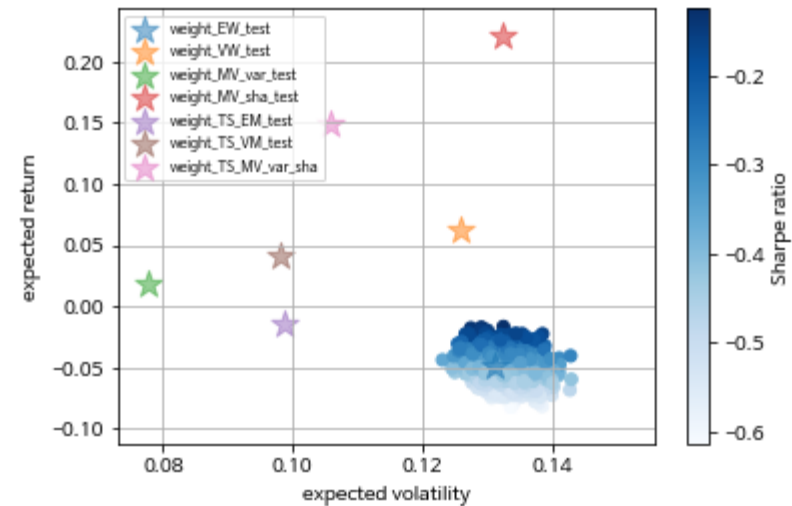
200

- 추가 업데이트 1. Risk-Return 그래프 / 각 포트폴리오 포지션



< 2019년 랜덤 포트폴리오 집합 >

- ✓ 랜덤한 weight를 준 포트폴리오의 기대 수익은 - 수치를 기록하고 분산 또한 큰 값을 가진다
- ✓ 각 포트폴리오들의 위치를 비교하여서 살펴보자

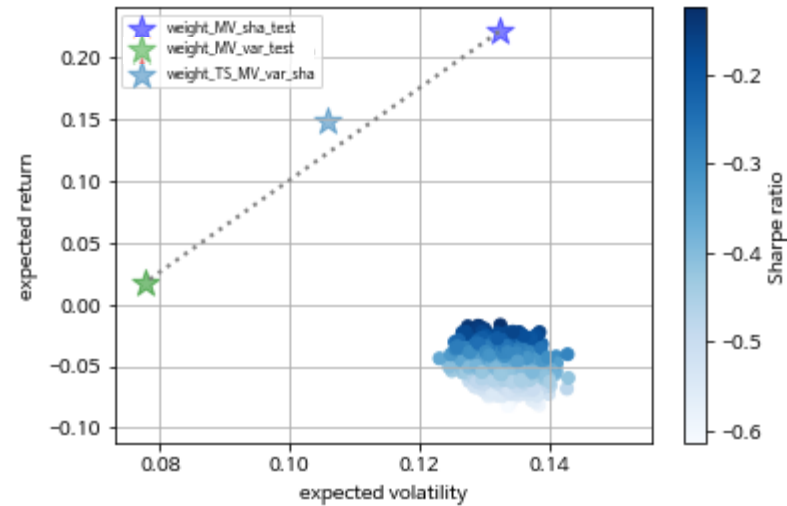


< 우리의 포트폴리오들의 포지션 >

- ✓ **EW포트폴리오**는 랜덤 포트폴리오 안에 속하여 크게 수익을 내지 못하는 모습을 보인다.
- ✓ **Min-variance max sharpe 포트폴리오**는 가장 높은 수익을 보이지만 분산이 크다
- ✓ **Min-variance max sharpe 포트폴리오**는 분산은 가장 작지만 수익을 내지 못한다

-> 이 두 포트폴리오를 blending한 결과를 보자!

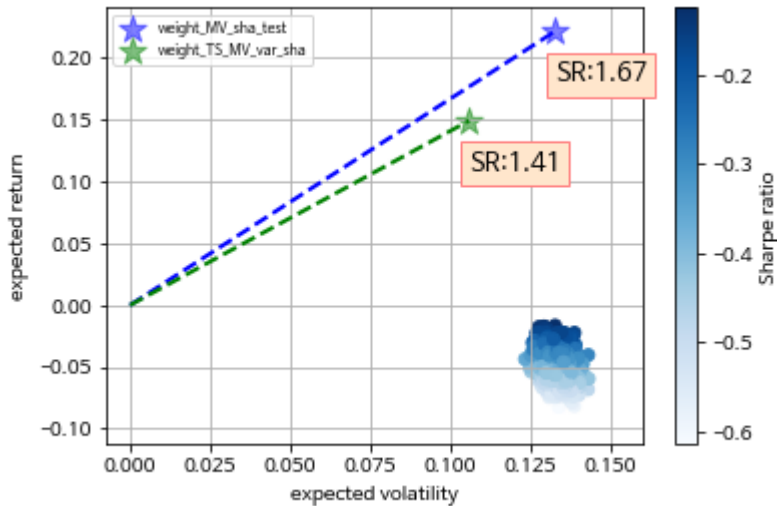
- 추가 업데이트 2. Risk-Return 그래프, TS Blending 효과



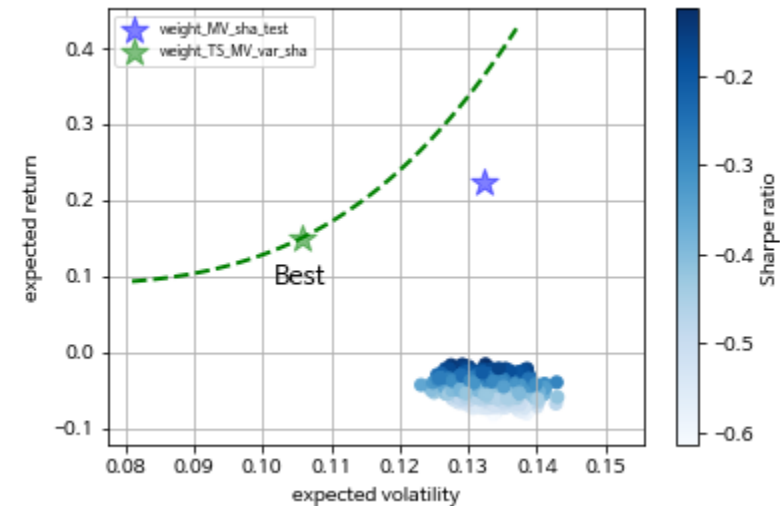
< Thomson sampling blending >

- ✓ 이 두 포트폴리오와 blending한 포트폴리오를 찍어보자.
- ✓ 각 시점에서 blending weight를 다르게 가진다.
- ✓ 따라서 단순 선형 결합이 아니라 더 높은 수익률을 기록하는 것을 알 수있다.

- 추가 업데이트 3. Risk-Return 그래프, SR과 무차별 곡선



< Metric : Sharpe Ratio >



< Metric : 무차별 곡선 >

- ✓ Min variance(max sharpe) 포트폴리오와 Thompson sampling [Min variance(max sharpe), Min variance(min variance)] 포트폴리오를 그래프 상에서 비교해보자
- ✓ 단순 Sharpe ratio는 Min variance(max sharpe) 포트폴리오가 높은 것으로 나타난다.
- ✓ 하지만 위험 성향을 가진 투자자가 가지는 무차별곡선에서의 효용값은 TS blending 포트폴리오 이 더 높을 것이다.
-> 따라서 톰슨 샘플링을 통한 블렌딩으로서 효율적인 포트폴리오를 만들어 낼다고 할 수 있을 것이다.



THANK YOU