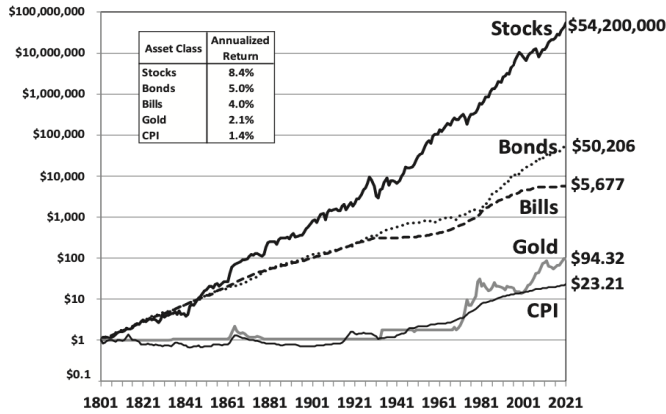


데이터 수집 프로젝트 제안서

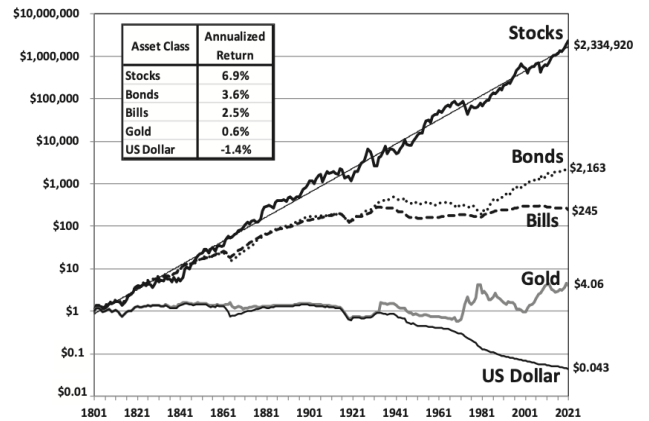
- 김태영, 류채 환, 신해술, 조은소리

주제 및 목적

<Total Nominal Return indexes, 1802-2021>¹



<Total Real Return Indexes, 1802-2021>²



Jeremy Siegel의 유명한 저서 Stocks for the Long Run에서 언급한 바, 자산으로서 주식은 다른 자산과 달리 실현수익률 측면에서 뛰어나며, 인플레이션에 대하여 헤지(Hedge)효과가 있는 자산임을 언급한 바 있음. 이에 우리나라의 주식과 채권도 이와 유사한 추이가 성립하는지 살펴보고자하며, 추가적으로 해외주식과 금, 부동산도 비교분석하고자함. 이들의 경우 안전자산과 대체투자자산 중 어느 성격이 더 강한지 살펴볼 예정임. 따라서 주식(국내, 해외), 채권, 금, 부동산의 수익률을 인플레이션율과 비교하여 각 자산이 인플레이션에 얼마나 효과적으로 헤지되는지 그래프로 표현하고자함.

방법론

각 자산별 원시데이터를 이용하여 비교가능한 지표인 수익률로 전환하고, 데이터 수집이 현실적으로 가능한 2000년 부터의 시계열을 상정함. 이를 바탕으로 초기시점(2000년 1월)에 1,000원을 투자했을때 현재시점(2023년 4월)까지 수익이 어떻게 변화하는지 살펴봄.

데이터 수집

<수집 데이터 설명>

자산종류	주식	채권	금	해외주식	부동산	(인플레이션)	(환율)
수집대상	KOSPI 지수	통안채(91 일) 국고채(1년)	국제 금 가격 (미국 기준)	S&P 500 index	매매가격지수 (종합주택유형)	소비자물가지수	원/달러
수집출처	KRX	한국은행 경제통계시스템	런던금시장협회	알파벳티지	한국부동산원 부동산통계 정보시스템	한국은행 경제통계시스템	한국은행 경제통계시스템
수집형식	API	csv 파일	API	API	API	csv 파일	csv 파일
수집기간	2000.01~	2000.01~	2000.01~	2000.01~	2003.11~	2000.01~	2000.01~
수집형식	고가 / 저가 / 종가 / 거래량 (1980.1.4=100)	시장금리(연%,월), 발행액(십억원,월)	1 트로이온스 당 미국 달러	고가 / 저가 / 종가 / 거래량 (1957.3.4=100)	원지수 (2021.06=100)	원지수 (2020=100)	달러당 원화
데이터 설명	유가증권시장의 보통주로 구성되어 산출하는 한국의 대표지수, 시가 총액 가중 평균 방식	T-bills와 유사한 기간인 한국은행에서 발행하는 통안채(91 일), 추가로 국고채(1년)	LBMA에서 결정되는 금 가격 중 15:00PM 가격정보	미국의 주요 산업을 대표하는 대형주 중에서 시가총액이 큰 500개 주식, 시가 총액 가중 평균 방식	주택가격동향조사를 반영한 주택가격지수를 대표. 부동산가격지수로 상정	통계청에서 조사하는 매월 조사하는 소비자물가지수를 대표 물가지수로 상정	-
기타	-	-	데이터 분석 시 환율 적용	데이터 분석 시 환율 적용	-	실질수익률 도출에 사용	달러표시 자산의 원화가치 변환

¹ Jeremy Siegel, Stocks for the Long Run, 2022, New York: McGraw Hill, p.23.

² 위의 글, p.6.

<수집 데이터 통합 DataFrame>

	USD_KRW	gold_close	kospi_close	kospi_volume	snp_close	snp_volume	kgb_1y_return	kgb_1y_volume	ktb_3m_return	ktb_3m_volume	cpi_index
date											
2022-11-30	1354.277273	1664.445238	2472.53	1.274876e+10	143.232857	4.458106e+06	3.838	10588.0	3.346	8680.0	109.10
2022-12-31	1293.515000	1726.447727	2236.40	8.436757e+09	144.581429	4.115535e+06	3.705	2700.0	3.444	7210.0	109.28
2023-01-31	1242.948636	1796.741667	2425.08	9.416084e+09	141.220500	5.278801e+06	3.584	12961.7	3.460	10770.0	110.10
2023-02-28	1278.717000	1898.628571	2412.85	9.658016e+09	134.177895	4.004246e+06	3.528	14188.2	3.464	11850.0	110.38
2023-03-31	1304.444348	1854.540000	2476.86	1.007835e+10	126.984783	6.004047e+06	3.452	18836.6	3.434	10820.0	110.56

예상 분석 절차³

데이터 수집

데이터 수집 방법

```
# csv 파일
# api
```

예시 : api를 활용한 데이터 추출

```
import requests
import json
from alpha_vantage.timeseries
import TimeSeries
```

```
# 데이터 요청 형식
api_key = 'demo'
r = requests.get(api_key)
data = r.json()
```

```
# TimeSeries 객체 생성
ts = TimeSeries(key=api_key,
output_format='pandas')
```

```
# 알파벳엔지 API를 사용하여 데이터 요청
symbol = 'SNP500' # SNP500의 심볼
data_SNP, meta_data =
ts.get_daily_adjusted(symbol=sym
bol,outputsize='full')
```

데이터 전처리

전처리 고려 사항

```
# 수익률 형식으로 전환
# Series를 월별 형식으로
# 자산간 상이한 기간 맞추기
```

예시 : mplfinance의 형식에 맞춰
snp500 전처리

```
snp = pd.DataFrame()
snp['open'] =
data_SNP_20['close'].resample(
'M').first()
snp['high'] =
data_SNP_20['close'].resample(
'M').max()
snp['low'] =
data_SNP_20['close'].resample(
'M').min()
snp['close'] =
data_SNP_20['close'].resample(
'M').last()
snp['volume'] =
data_SNP_20['volume'].resample(
'M').sum()
```

```
snp.index =
pd.to_datetime(snp.index)
```

데이터 표현

예시 : s&p500 그래프 그리기
macd = MACD(snp, 3, 12, 6)
stochastic = Stochastic(snp, 14, 3)

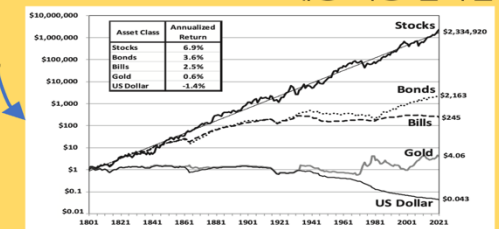
```
plots =
[ mpf.make_addplot((macd['macd'])),
color='#606060', panel=2,
ylabel='MACD (3,12,6)',
secondary_y=False),
mpf.make_addplot((macd['signal'])),
color='#1f77b4', panel=2,
secondary_y=False),
mpf.make_addplot((macd['bar_positiv
e'])), type='bar', color='#4dc790',
panel=2),
mpf.make_addplot((macd['bar_negativ
e'])), type='bar', color='#fd6b6c',
panel=2),
mpf.make_addplot((stochastic[['%D',
'%SD', 'UL', 'DL']]), ylim=[0, 100],
panel=3, ylabel='Stoch (14,3)') ]
mpf.plot(snp, type='candle',
style='yahoo', mav=(3,12),
volume=True, addplot=plots,
panel_ratios=(3,1,3,3),
figscale=1.5)
```

<예시 : S&P500>



같은 방식으로
겹쳐 그리기

예상 최종 결과물



³ 데이터 표현 참고 코드 출처 - <https://plainenglish.io/blog/plot-stock-chart-using-mplfinance-in-python-9286fc69689>