

```
1 !pip install yfinance
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: yfinance in /usr/local/lib/python3.8/dist-packages (0.2.12)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.8/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.8/dist-packages (from yfinance) (2022.7.1)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.22.4)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.8/dist-packages (from yfinance) (2.3.5)
Requirement already satisfied: appdirs>=1.4.4 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.4.4)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.8/dist-packages (from yfinance) (4.11.2)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.8/dist-packages (from yfinance) (4.9.2)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.3.5)
Requirement already satisfied: cryptography>=3.3.2 in /usr/local/lib/python3.8/dist-packages (from yfinance) (39.0.1)
Requirement already satisfied: requests>=2.26 in /usr/local/lib/python3.8/dist-packages (from yfinance) (2.28.2)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.8/dist-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.8/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.4)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.8/dist-packages (from cryptography>=3.3.2->yfinance) (1.15.1)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.8/dist-packages (from html5lib>=1.1->yfinance) (1.15.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.8/dist-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (3.0.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (1.24.3)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26->yfinance) (2.10)
Requirement already satisfied: pycparser in /usr/local/lib/python3.8/dist-packages (from cffi>=1.12->cryptography>=3.3.2->yfinance) (2.21)
```

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import yfinance as yf
```

```
1 price_history = yf.Ticker('TSLA').history(period='2y', # valid periods: 1d,5d,1mo,3mo,6mo,1y,2y,5y,10y,ytd,max
2                                     interval='1d', # valid intervals: 1m,2m,5m,15m,30m,60m,90m,1h,1d,5d,1wk,1mo,3mo
3                                     actions=False)
```

```
1 price_history
```

	Open	High	Low	Close	Volume
Date					
2021-03-01 00:00:00-05:00	230.036667	239.666672	228.350006	239.476669	81408600
2021-03-02 00:00:00-05:00	239.426666	240.369995	228.333328	228.813339	71196600
2021-03-03 00:00:00-05:00	229.330002	233.566666	217.236664	217.733337	90624000
2021-03-04 00:00:00-05:00	218.600006	222.816666	200.000000	207.146667	197758500
2021-03-05 00:00:00-05:00	208.686661	209.279999	179.830002	199.316666	268189500
...	...	...	...	...	...
2023-02-21 00:00:00-05:00	204.990005	209.710007	197.220001	197.369995	180018600
2023-02-22					

```
1 def find_volatility(ticker):
2     data = yf.Ticker(ticker).history(period='3mo', interval='1d', actions=False)
3     data['Log returns'] = np.log(data['Close']/data['Close'].shift())
4     volatility = data['Log returns'].std()*252**.5
5     return volatility
```

```
1 def find_beta_against_spy(ticker):
2     data = yf.Ticker(ticker).history(period='12mo', interval='1d', actions=False)
3     data['Log returns'] = np.log(data['Close']/data['Close'].shift())
4     data_spy = yf.Ticker('SPY').history(period='12mo', interval='1d', actions=False)
5     data_spy['Log returns'] = np.log(data_spy['Close']/data_spy['Close'].shift())
6     cov = data['Log returns'].cov(data_spy['Log returns'])
7     var = data_spy['Log returns'].var()
8     beta_spy = cov/var
9     return beta_spy
```

```
1 def find_beta_against_iwm(ticker):
2     data = yf.Ticker(ticker).history(period='12mo', interval='1d', actions=False)
3     data['Log returns'] = np.log(data['Close']/data['Close'].shift())
4     data_iwm = yf.Ticker('iwm').history(period='12mo', interval='1d', actions=False)
5     data_iwm['Log returns'] = np.log(data_iwm['Close']/data_iwm['Close'].shift())
6     cov = data['Log returns'].cov(data_iwm['Log returns'])
7     var = data_iwm['Log returns'].var()
8     beta_iwm = cov/var
9     return beta_iwm
```

```
1 def find_beta_against_dia(ticker):
2     data = yf.Ticker(ticker).history(period='12mo', interval='1d', actions=False)
3     data['Log returns'] = np.log(data['Close']/data['Close'].shift())
4     data_dia = yf.Ticker('dia').history(period='12mo', interval='1d', actions=False)
5     data_dia['Log returns'] = np.log(data_dia['Close']/data_dia['Close'].shift())
6     cov = data['Log returns'].cov(data_dia['Log returns'])
7     var = data_dia['Log returns'].var()
8     beta_dia = cov/var
9     return beta_dia
```

```

1 def find_average_weekly_drawdown(ticker):
2     data = yf.Ticker(ticker).history(period='12mo', interval='1d', actions=False)
3     avg_weekly_drawdown = (data['Close'].min() - data['Close'].max())/data['Close'].max()
4     return avg_weekly_drawdown

```

```

1 def find_maximum_weekly_drawdown(ticker):
2     data = yf.Ticker(ticker).history(period='12mo', interval='1d', actions=False)
3     avg_maximum_drawdown = (data['Close'].min() - data['Close'].max())/data['Close'].max()
4     return avg_maximum_drawdown

```

```

1 def find_return(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='3mo', actions=False)
3     ret = ((data['Close'].iloc[-1] - data['Close'].iloc[0])/data['Close'].iloc[0])
4     return ret

```

```

1 def find_annualized_return(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='3mo', actions=False)
3     ret = ((data['Close'].iloc[-1] - data['Close'].iloc[0])/data['Close'].iloc[0])
4     annualized_ret = ((1+ret)**(1/10)) - 1
5     return annualized_ret

```

```

1 tickers = ['TSLA', 'AAPL', 'MSFT', 'AMZN', 'UNH', 'GOOGL', 'NVDA']

```

```

1 table1 = pd.DataFrame(columns=['Ticker', 'Portfolio Weight (equally weighted)', 'Annualized Volatility (using trailing 3-months)', 'Beta against SPY (using trailing 12-m

```

```

1 table1['Ticker'] = tickers
2 table1['Portfolio Weight (equally weighted)'] = [1.0 for i in range(len(tickers))]
3 table1['Annualized Volatility (using trailing 3-months)'] = list(map(find_volatility, tickers))
4 table1['Beta against SPY (using trailing 12-months)'] = list(map(find_beta_against_spy, tickers))
5 table1['Beta against IWM (using trailing 12-months)'] = list(map(find_beta_against_iwm, tickers))
6 table1['Beta against DIA (using trailing 12-months)'] = list(map(find_beta_against_dia, tickers))
7 table1['Average Weekly Drawdown (52-week Low minus 52-week High) / 52-week High'] = list(map(find_average_weekly_drawdown, tickers))
8 table1['Maximum Weekly Drawdown (52-week Low minus 52-week High) / 52-week High'] = list(map(find_maximum_weekly_drawdown, tickers))
9 table1['Total Return (using trailing 10-years)'] = list(map(find_return, tickers))
10 table1['Annualized Total Return (using trailing 10-years)'] = list(map(find_annualized_return, tickers))

```

```

1 table1.head(len(tickers))

```

	Ticker	Portfolio Weight (equally weighted)	Annualized Volatility (using trailing 3-months)	Beta against SPY (using trailing 12-months)	Beta against IWM (using trailing 12-months)	Beta against DIA (using trailing 12-months)	Average Weekly Drawdown (52-week Low minus 52-week High) / 52-week High	Maximum Weekly Drawdown (52-week Low minus 52-week High) / 52-week High	Total Return (using trailing 10-years)	Annualized Total Return (using trailing 10-years)
0	TSLA	1.0	0.796812	1.739676	1.515709	1.706969	-0.716880	-0.716880	56.685688	0.500054
1	AAPL	1.0	0.294755	1.303787	1.037119	1.474275	-0.298248	-0.298248	9.893495	0.269746
2	MSFT	1.0	0.325976	1.295466	0.998773	1.466260	-0.317698	-0.317698	8.188563	0.248317
3	AMZN	1.0	0.417194	1.657138	1.361745	1.776833	-0.516759	-0.516759	6.388204	0.221389
4	UNH	1.0	0.228033	0.537093	0.371707	0.768203	-0.188245	-0.188245	8.405080	0.251227
5	GOOGL	1.0	0.400000	1.000000	1.000000	1.400000	-0.400000	-0.400000	0.000000	0.000000

```

1 etfs = ['EWD', 'SLX', 'CQQQ']

```

```

1 df = yf.Ticker(tickers[0]).history(period='10y', interval='1d', actions=False)['Close']
2 for t in tickers[1:]:
3     df = df + yf.Ticker(t).history(period='10y', interval='1d', actions=False)['Close']

```

```

1 df.head()

```

```

Date
2013-02-28 00:00:00-05:00    121.030101
2013-03-01 00:00:00-05:00    121.075366
2013-03-04 00:00:00-05:00    121.550338
2013-03-05 00:00:00-05:00    122.868421
2013-03-06 00:00:00-05:00    122.374424
Name: Close, dtype: float64

```

```

1 df.tail()

```

```

Date
2023-02-21 00:00:00-05:00    1482.749992
2023-02-22 00:00:00-05:00    1485.150009
2023-02-23 00:00:00-05:00    1521.280006
2023-02-24 00:00:00-05:00    1492.629997
2023-02-27 00:00:00-05:00    1507.670013
Name: Close, dtype: float64

```

```

1 df/=len(tickers)

```

```

1 df.tail()

```

```

Date
2023-02-21 00:00:00-05:00    211.821427
2023-02-22 00:00:00-05:00    212.164287
2023-02-23 00:00:00-05:00    217.325715
2023-02-24 00:00:00-05:00    213.232857
2023-02-27 00:00:00-05:00    215.381430
Name: Close, dtype: float64

```

```

1 def find_etf_correlation(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='1d', actions=False)['Close']
3     corr = data.corr(df)
4     return corr

```

```

1 def find_etf_covariance(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='1d', actions=False)['Close']
3     cov = data.cov(df)
4     return cov

```

```

1 def find_etf_tracking_errors(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='1d', actions=False)['Close']
3     tracking_error = np.std((df.values - data.values) / df.values)
4     return tracking_error

```

```

1 def find_etf_sharpe_ratio(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='1d', actions=False)['Close'].pct_change().dropna()
3     data = data.values
4     risk_free_Rate = 0.0
5     mean_daily_return = sum(data)/len(data)
6     std = np.std(data)
7     daily_sharpe_ratio = (mean_daily_return - risk_free_Rate) / std
8     sharpe_ratio = 252**(1/2) * daily_sharpe_ratio
9     return sharpe_ratio

```

```

1 def find_etf_annualized_volatility_spread(ticker):
2     data = yf.Ticker(ticker).history(period='10y', interval='1d', actions=False)
3     data['Log returns'] = np.log(data['Close']/data['Close'].shift())
4     etf_volatility = data['Log returns'].std()*252**.5
5     portfolio_log_returns = np.log(df[df.shift()])
6     portfolio_volatility = portfolio_log_returns.std()*252**.5
7     return portfolio_volatility - etf_volatility

```

```

1 table2 = pd.DataFrame(columns=['ETF Ticker', 'Correlation against ETF', 'Covariance of Portfolio against ETF', 'Tracking Errors (using trailing 10-years)', 'Sharpe Ratio

```

```

1 table2['ETF Ticker'] = etfs
2 table2['Correlation against ETF'] = list(map(find_etf_correlation, etfs))
3 table2['Covariance of Portfolio against ETF'] = list(map(find_etf_covariance, etfs))
4 table2['Tracking Errors (using trailing 10-years)'] = list(map(find_etf_tracking_errors, etfs))
5 table2['Sharpe Ratio (using current risk-free rate)'] = list(map(find_etf_sharpe_ratio, etfs))
6 table2['Annualized Volatility (252 days) Spread (Portfolio Volatility - ETF Volatility)'] = list(map(find_etf_annualized_volatility_spread, etfs))

```

```

1 table2.head()

```

	ETF Ticker	Correlation against ETF	Covariance of Portfolio against ETF	Tracking Errors (using trailing 10-years)	Sharpe Ratio (using current risk-free rate)	Annualized Volatility (252 days) Spread (Portfolio Volatility - ETF Volatility)
0	EWD	0.874130	415.070819	0.302942	0.320300	0.003271
1	SLX	0.794261	710.282297	0.385798	0.387549	-0.091469
2	CQQQ	0.667982	813.133906	0.340953	0.375248	-0.073745

..

```

1 table3 = pd.DataFrame()
2 table3['EquallyWeightedPortfolio'] = df
3 correlation_matrix_tickers = etfs + tickers
4 for t in correlation_matrix_tickers:
5     table3[t] = yf.Ticker(t).history(period='10y', interval='1d', actions=False)['Close']

```

```

1 table3.corr()

```

	EquallyWeightedPortfolio	EWD	SLX	CQQQ	TSLA	AAPL	MSFT
EquallyWeightedPortfolio	1.000000	0.874130	0.794261	0.667982	0.945736	0.991506	0.991801
EWD	0.874130	1.000000	0.790014	0.787748	0.858171	0.859056	0.869914
SLX	0.794261	0.790014	1.000000	0.458323	0.772197	0.785749	0.757554
CQQQ	0.667982	0.787748	0.458323	1.000000	0.576914	0.641645	0.684314
TSLA	0.945736	0.858171	0.772197	0.576914	1.000000	0.949035	0.917264
AAPL	0.991506	0.859056	0.785749	0.641645	0.949035	1.000000	0.983074
MSFT	0.991801	0.869914	0.757554	0.684314	0.917264	0.983074	1.000000
AMZN	0.917271	0.845088	0.620575	0.830828	0.807319	0.892662	0.936343
UNH	0.966159	0.773046	0.795778	0.590329	0.862377	0.956960	0.954339
GOOGL	0.983111	0.903593	0.786078	0.693499	0.924280	0.961525	0.979088
NVDA	0.975522	0.896157	0.803811	0.647499	0.938622	0.960729	0.962923

+ Code

+ Text

```

1 table3.head()

```

	EquallyWeightedPortfolio	EWD	SLX	CQQQ	TSLA	AAPL	MSFT	AMZN	
1									
2013-02-28 00:00:00-05:00		17.290014	22.332064	31.586666	21.968330	2.322000	13.615319	23.055151	13.2135 4%
2013-03-01 00:00:00-05:00		17.296481	22.532320	31.019159	21.903473	2.310000	13.278174	23.179544	13.2870 4%