The Efficient Frontier

Fama-French 3-Factor and

Markowitz Efficient Frontier in Python

Team 4

Roger C. Hahn

Lucas Gowland

Brody Wacker

Nicole Goldin

Evinski Benjamin

Feb. 22, 2020

CU-NYC-Fintech

MOTIVATION:
To Go Where No Banker has Gone Before

UNLESS YOU KNOW PYTHON
AND HAVE TAKEN THIS COURSE

- CAPM measures *MARKET* risk but nothing else
- Fama French 3-Model (FF3) improves on CAPM by adding *SIZE* risk and *VALUE* risk
- FF3 delivers more accurate expected returns than CAPM
- Markowitz efficient frontier theory can allocate weights to stocks in a portfolio more accurately

Questions

- Can we use FF3 to obtain more accurate expected values?
- Can we use those more accurate expected values to allocate weights to a stock portfolio using a Markowitz efficient frontier?

Data

P/E, EPS, Beta found on Yahoo finance

FFM3 coefficients found on Ken French's website

Answers

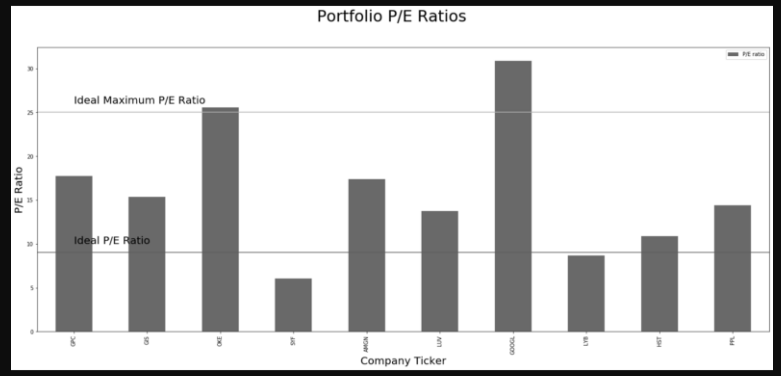
Yes, expected returns probably more accurate using FF3

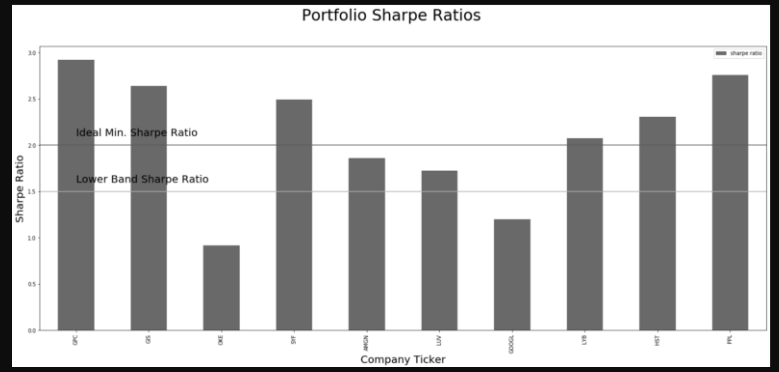
Yes, we could allocate weights to stocks using Modern Portfolio Theory

Portfolio Selection

- Investment Universe: S&P 500
  - Data Constraints – 100 randomly selected stocks
- Data Sources
  - Price history from **Yahoo Finance** for Sharpe Ratios
  - Earnings from **Google Finance** for P/E Ratios
- Selection Approach
  - Sector Diversification
  - Low P/E Ratio (undervalued)
  - High Sharpe Ratio (low risk to return)
- Libraries Used
  - PathLib for importing / exporting CSV files
  - NumPy for random selection
  - Pandas Datareader for API interactions
  - Pandas for data organization
  - Matplotlib for visualization

	ticker	name	sector	sharpe ratio	FFM returns	P/E ratio	EPS	beta	mkt cap	shares
0	GPC	Genuine Parts	Consumer Discretionary	2.924464	0.074228	17.71	5.45	0.89	14016415887	145293000
1	GIS	General Mills	Consumer Staples	2.641281	0.063147	15.34	3.48	0.73	32242795100	604817000
2	OKE	ONEOK	Energy	0.917326	0.230148	25.56	3.01	1.11	31799253894	413085000
3	SYF	Synchrony Financial	Financials	2.493336	0.091836	6.07	5.56	1.21	20704848750	613477000
4	AMGN	Amgen Inc	Health Care	1.861649	0.151749	17.36	12.88	1.12	131810065840	589807000
5	LUV	Southwest Airlines	Industrials	1.726576	0.107833	13.77	4.21	1.47	29987649751	517296000
6	GOOGL	Alphabet Inc Class A	Information Technology	1.198026	0.214640	30.89	49.16	1.02	1044236513876	299895000
7	LYB	LyondellBasell	Materials	2.072393	0.103512	8.65	9.55	1.45	27505799491	333000000
8	HST	Host Hotels & Resorts	Real Estate	2.308137	0.052833	10.89	1.55	1.17	12127479870	717178000
9	PPL	PPL Corp.	Utilities	2.755488	0.098239	14.38	2.46	0.51	25573676437	723033000





Capital Asset Pricing Model (CAPM)

$$(Ri\text{-}Rf) = Rf + \beta i(ERm - Rf)$$

Fama French Three Factor Model

$$(Ri\text{-}Rf) = \alpha it + \beta 1(RMt - Rft) + \beta 2(SMBt) + \beta 3(HMLt) + \epsilon it$$

β1(RMt – Rft)	Regression correlation to S&P
β2(SMBt)	Regression correlation to Small minus Big
β3(HMLt)	Regression correlation to High minus Low

Coding Fama French Model

- Pull in the SMB and HML constants
- Create loop using StatsModel Regressions for betas

```
M↓
import statsmodels.api as sm
from statsmodels.regression import linear_model as lm
```

```
M↓
results_dict = {}
params = {}

for ticker in monthly_df_daily_returns.columns:
    results_dict[ticker] = lm.OLS(endog=monthly_df_daily_returns.loc[:,ticker],
                  exog=sm.add_constant(ffm_data[['Mkt-RF','SMB','HML']])).fit()

    params[ticker] = results_dict[ticker].params
```

- Create dataframe to hold regressions

Efficient Frontier

Modern Portfolio Theory

What is Efficient Frontier?

The Collaboration of all previous Models

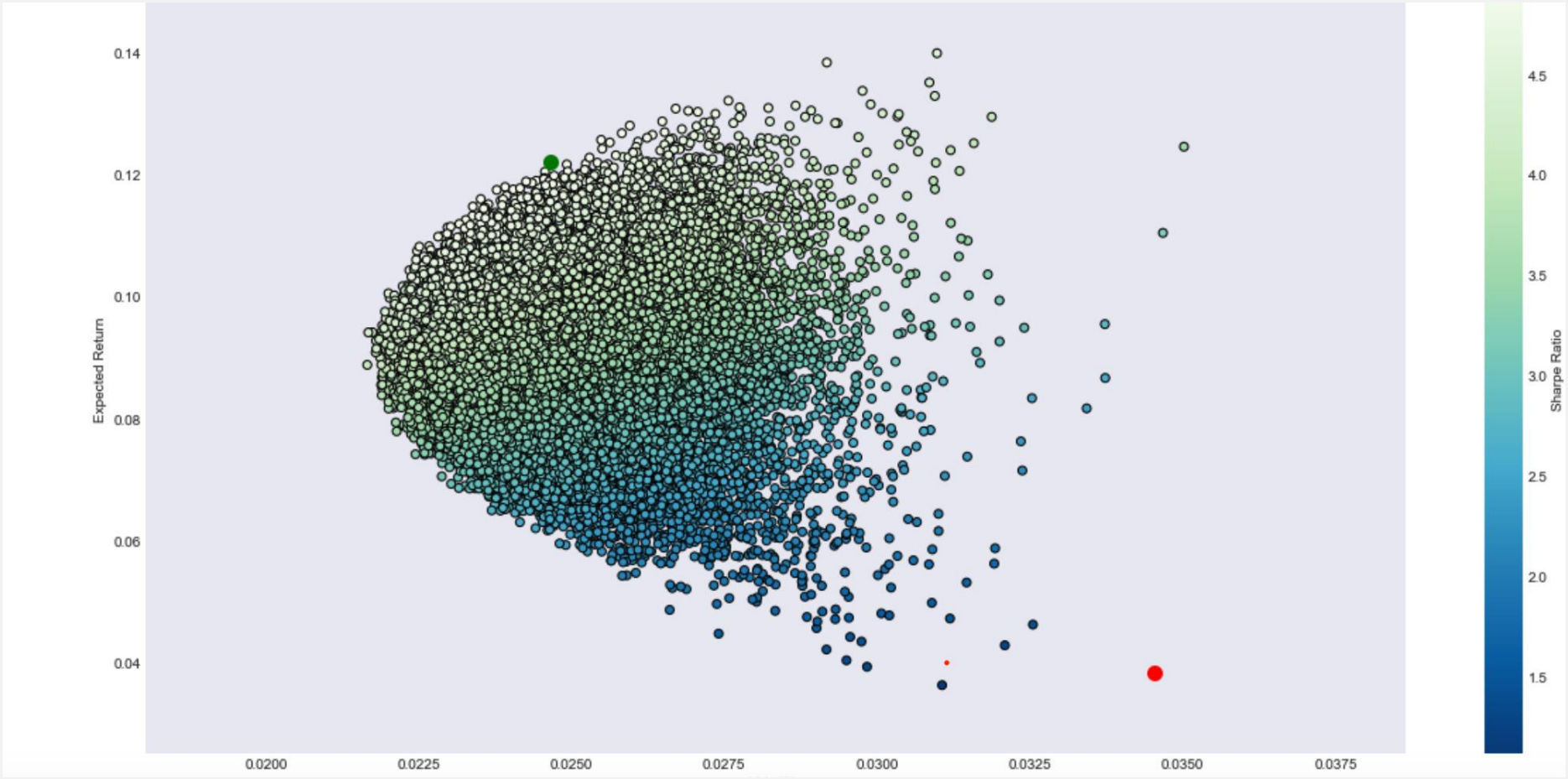
Portfolios that satisfy above requirements are called efficient portfolios.

Coding Efficient Frontier

1. Load data set correlating Sharpe ratio and FFM.
2. Back out volatility by using Sharp ratios and returns
3. Determine the amount of portfolios you would like to test
4. Create the empty lists with each portfolios returns, risk and weight
5. Create a loop which will simulate several different combinations of the ten stocks and save their weight.

```
    for x in range(num_portfolios):
        # produce random values for each stock
        weights = np.array(np.random.random(len(Efficient_df.index)))

        # get weights for each stock by normalizing sum to 1
        weights = weights/np.sum(weights)

        # store weights for portfolio
        all_weights[x,:] = weights

        # store returns for portfolio
        ret_arr[x] = np.sum(Efficient_df['FFM returns'] * weights)

        # get vol
        vol_arr[x] = np.sqrt(np.dot(weights.T, np.dot(Efficient_df['FFM returns'].cov(
            Efficient_df['FFM returns']), weights)))
```

Efficient Frontier
Portfolio Result



Final Portfolio Weightings

Genuine Parts – 1.3%

General Mills – 20.9%

ONEOK – 13.1%

Synchrony Financial – 15.6%

Amgen – 11.7%

Southwest Airlines – 18.6%

Alphabet – 0.07%

LyondellBasell – 6%

Host Hotels – 10.5%

PPL Corp. - 1.7%

Challenges
Next Steps

- Git What?
- Sensitivity Analysis
- Monte Carlo

Questions?