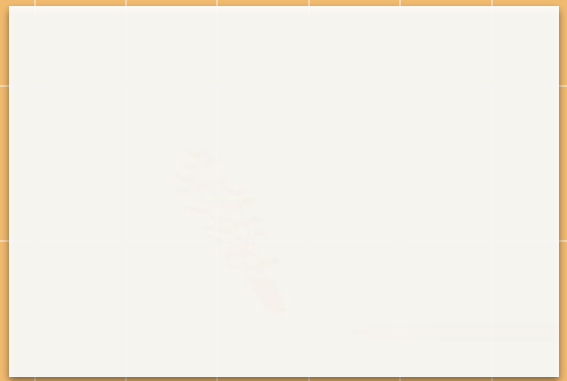
ZOOMERS VS BOOMERS:

CRYPTO VS COMMODITIES

The Ultimate Portfolio Showdown



Team 7

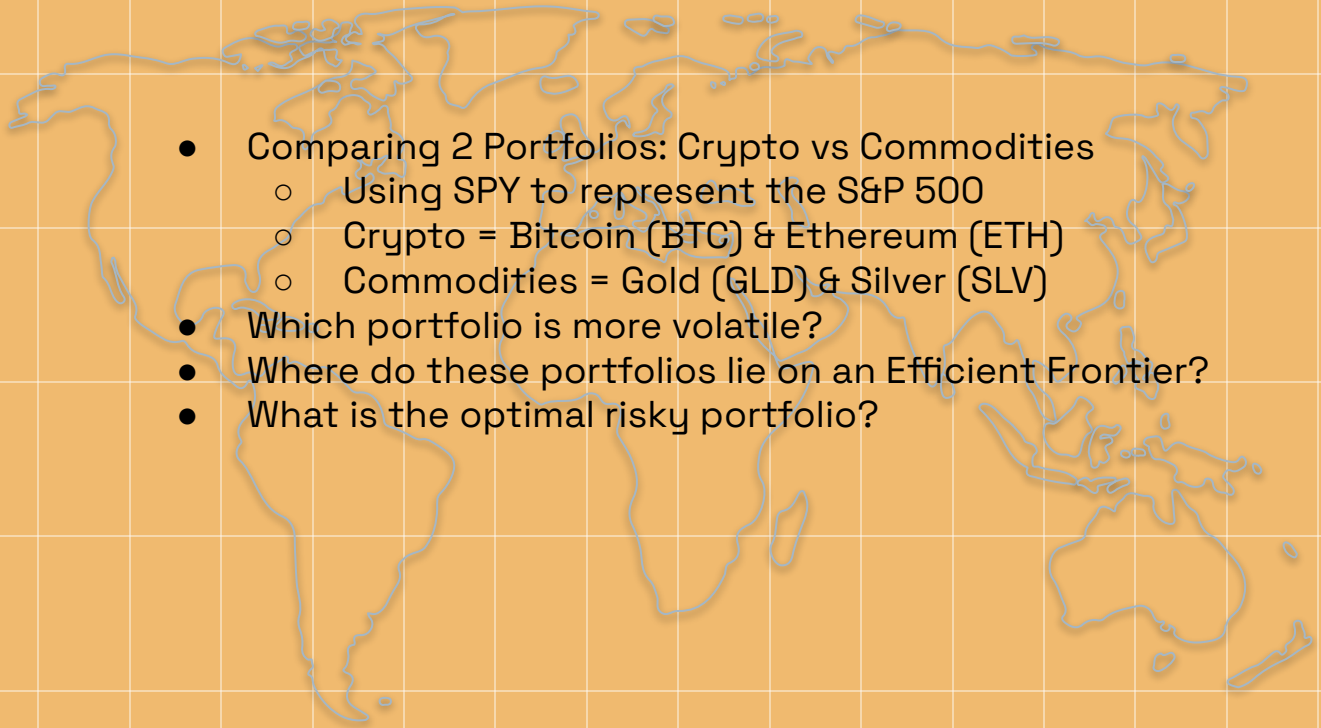
ABOUT THE TEAM

- Sean Patel
- Bailey Richterman
- Ed Foote
- Jose Sampedro

01

THE CONCEPT

The Concept



- Comparing 2 Portfolios: Crypto vs Commodities
  - Using SPY to represent the S&P 500
  - Crypto = Bitcoin (BTC) & Ethereum (ETH)
  - Commodities = Gold (GLD) & Silver (SLV)
- Which portfolio is more volatile?
- Where do these portfolios lie on an Efficient Frontier?
- What is the optimal risky portfolio?

02

THE CODE

Data Collection

- **Data Sources Explored:**
  - Quandl
  - Alpaca
  - Yahoo Finance

- **Data Sources Used:**
  - Yahoo Finance

Data Cleaning

- Imported data into Jupyter Notebook
- Create individual dataframes (GLD, SLV, SPY, BTC, ETH)
- Concatenate dataframes into their respective portfolios
  - GLD + SLV
  - BTC + ETH

```
eth_daily_rts=eth_df_close.pct_change().dropna()
eth_daily_rts.loc[~eth_daily_rts.index.duplicated(keep='first')]

eth_daily_rts.head()
```

- Road block: Pulling data from both the Alpaca API and Yahoo Finance
  - Difficult to concatenate dataframes that were being pulled from 2 different sources
  - Settled on solely using Yahoo Finance
  - Alpaca API had a limit on dates for free content (could only 4 months of data)

Data Exploration

- Crypto Portfolio
- Covid-19
  - Both dipped slightly at the beginning, but came back up over time
- Had overall higher daily returns

- Commodities Portfolio
- Covid-19
  - Gold held steady
  - Silver dipped, but bounced back before the end of 2020

Code Examples

In [63]:
```
# COMMODITIES PORTFOLIO SOLVER

comm_p_ret = []
comm_p_vol = []
comm_p_weights = []

num_comm_assets = len(comm_assets.columns)
num_comm_portfolios = 10000
```

In [64]:
```
for portfolio in range(num_comm_portfolios):
    comm_weights = np.random.random(num_comm_assets)
    #print(comm_weights)
    comm_weights = comm_weights/np.sum(comm_weights)
    #print(comm_weights)
    comm_p_weights.append(comm_weights)
    #print(comm_p_weights)
    comm_returns = np.dot(comm_weights, comm_ind_er) # Returns are the product of individual expected returns of asset and its
   # print(comm_returns)                                    # weights
    comm_p_ret.append(comm_returns)
    #print(len(comm_p_ret))
    comm_var = comm_cov_matrix.mul(comm_weights, axis=0).mul(comm_weights, axis=1).sum().sum()# Portfolio Variance
    #print(comm_var)
    comm_sd = np.sqrt(comm_var) # Daily standard deviation
    #print(comm_sd)
    comm_ann_sd = comm_sd*np.sqrt(250) # Annual standard deviation = volatility

    #print(ann_sd)
    comm_p_vol.append(comm_ann_sd)
    #print(comm_p_vol)
```

In [65]:
```
data = {'Returns':comm_p_ret, 'Volatility':comm_p_vol}
for counter, symbol in enumerate(comm_assets_transposed.columns.tolist()):
    print(counter, symbol)
    data[symbol+' weight'] = [w[counter] for w in comm_p_weights]
```

Code Examples

In [66]:
```
portfolios  = pd.DataFrame(data)
portfolios.head() # Dataframe of the 10000 portfolios created
```

Out[66]:

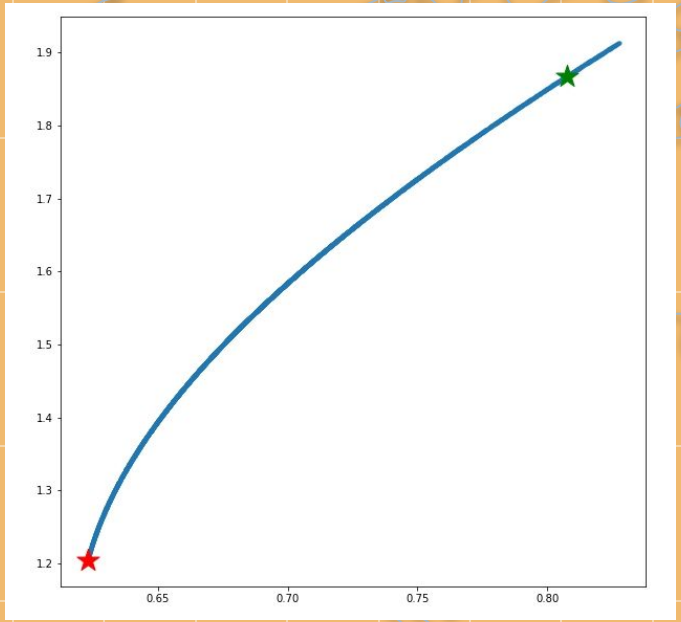
	Returns	Volatility	Gold Daily Returns weight	Silver Daily Returns weight
0	-0.002803	0.157895	0.921413	0.078587
1	0.000351	0.262666	0.254588	0.745412
2	0.000274	0.259660	0.270808	0.729192
3	-0.001740	0.187330	0.696612	0.303388
4	-0.002914	0.155394	0.944808	0.055192

03

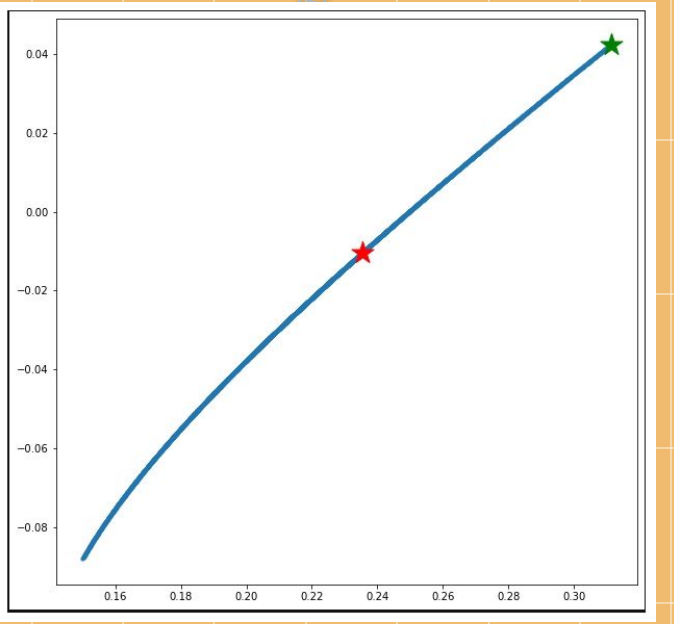
THE RESULTS

Visualizations & Analysis

Crypto Efficient Frontier

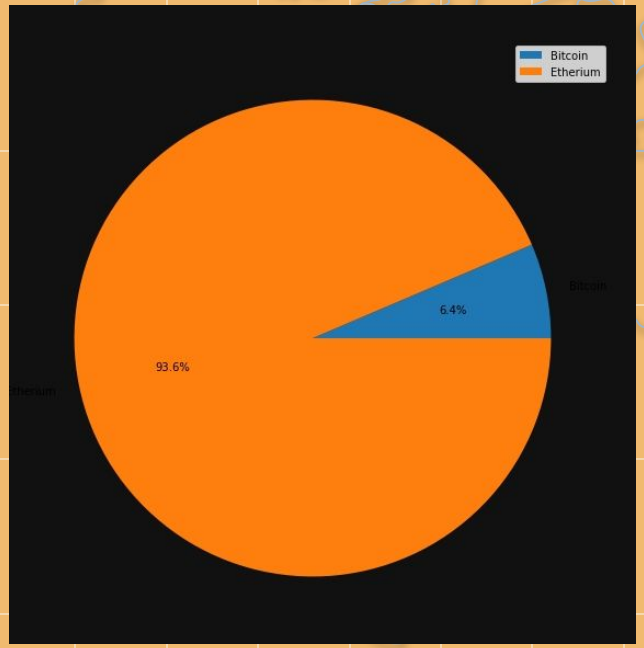


Commodities Efficient Frontier

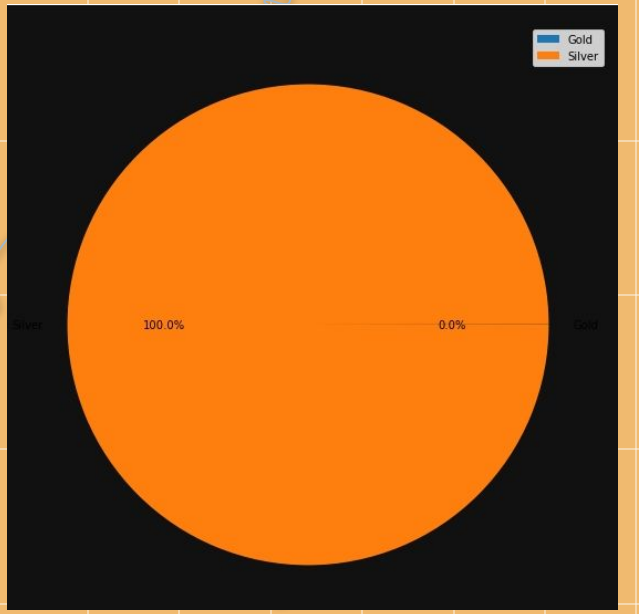


Visualizations & Analysis

Crypto Portfolio Weights



Commodity Portfolio Weights



04 THE CONCLUSION

The Conclusion

COMMODITIES

- **Silver preferred to Gold**
- Past year return of gold = -8.79%
- Past year return of silver = 4.23%

CRYPTOCURRENCY

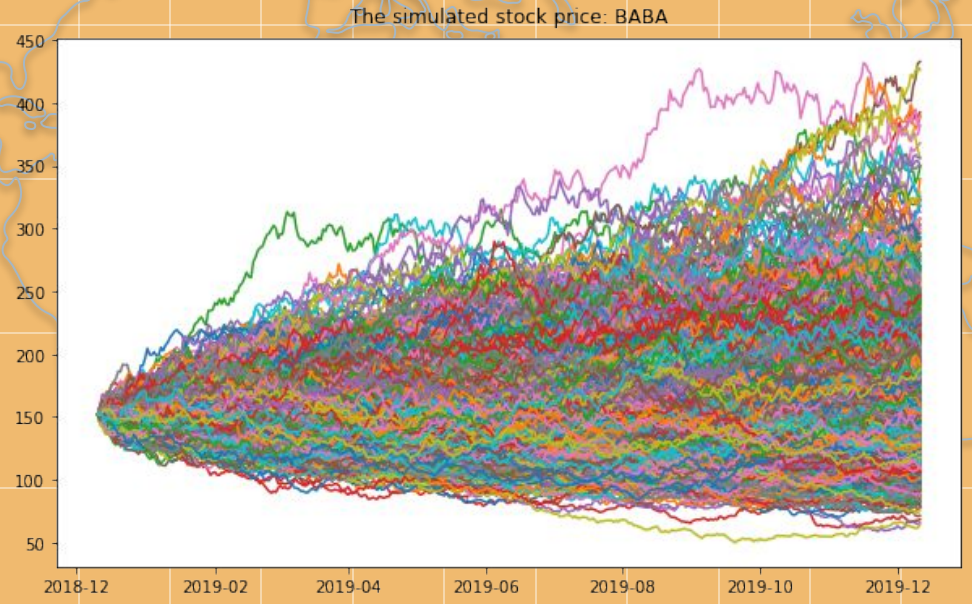
- **Ethereum preferred to Bitcoin**
- Past year return of bitcoin= 120%
- Past year return of ethereum= 191%

05

Next Steps

Next Steps

- Will standardizing the trading days make an impact on our conclusion?
  - Only include crypto data within the normal trading days for commodities
- Add in a Monte Carlo Simulation



RESOURCES

Yahoo Finance: https://finance.yahoo.com/

Team 7 GitHub: https://github.com/BRichterman/Team7-Project-1

Code Documentation:
https://www.machinelearningplus.com/machine-learning/portfolio-optimization-python-example/

Questions?