Project 1: Monte Crypto

Floriane Beyegue, Rawad Habib, Nigil Jeyashekar, Cassandra Johnson, Ken Lindgren, Antonio Pinkston

Motivation & Summary

Core Questions

- **What types of investments produce the best returns (cryptocurrency, agricultural commodities, metals, a stock portfolio, and the S&P 500)*?**
- **What's the ideal portfolio mix to maximize profit while minimizing risk?**
- **What impact did COVID have on the ideal investments?**

Sub-questions & Findings:

1. Which investment type has yielded the highest return?
    a. Finding: Cryptocurrency had the best historical returns. However, this investment was also very high risk and volatile.
2. Which investment type is likely to produce the highest returns going forward?
    a. Finding: Investing in a stock portfolio - either a mix of individual stocks or the S&P 500 - projected the greatest return over 5 years using a Monte Carlo simulation.

*Pricing data used: January 1, 2015 - October 31, 2020. Pre/Post COVID data cut-off date: February 1, 2020.

Motivation & Summary

3. Using the aforementioned investments, what is the optimal portfolio mix based on maximizing the Sharpe ratio?

4. How has COVID-19 impacted the optimal portfolio mix?

    a. Finding:

The ideal portfolio is a mix of all investments, and changed significantly pre- and post-COVID.

For example the largest 3 investments in the pre-COVID optimal portfolio are: Gold, AAPL, and AMZN. The 3 largest investments in the post-COVID portfolio are: ETH, Soybeans, and AMZN.

Questions & Data

- Datasets utilized:
  - Cryptocurrency dataset--Yahoo Finance
    - Bitcoin, Litecoin, Ethereum
  - Commodities dataset--Yahoo Finance
    - Soybean, Wheat, Corn
  - Metals dataset--Markets Insider
    - Gold, Silver, Platinum
  - Stock dataset--Yahoo Finance
    - Apple, Amazon, Southwest Airlines, Pfizer
  - S&P 500 ETF--Yahoo Finance
    - SPDR S&P 500 ETF Trust (SPY)
- Loaded data two ways:
  - Load csv files, wrote a function to clean the datasets and return data frames with date and closing price information to analyze daily and cumulative returns
  - Utilized Alpaca to load data for Monte Carlo simulations

Data Cleanup & Exploration

- *Step-1*: Data was sourced from Resources using Path
- *Step-2*: Function was coded to perform the following operations:
  - Reading csv to Dataframe and subsetting Date and Closing Prices.
    - Removed all duplicate or irrelevant observations from each dataset
  - Index was set to date and extraneous columns removed, with dates sorted in ascending order
- *Step-3*: Verify Null values were dropped
    - Filtered out unwanted outliers
    - Address any missing data issues
- By using Alpaca API for Monte Carlo simulations were able to validate the accuracy of the csv file

Step-1

Crypto

```
BTC_USD_path = Path("Resources1/BTC-USD.csv")
ETH_USD_path = Path("Resources1/ETH-USD.csv")
LTC_USD_path = Path("Resources1/LTC-USD.csv")
```

Commodities

```
Corn_path = Path("Resources1/Corn.csv")
Soybeans_path = Path("Resources1/Soybean.csv")
Wheat_path = Path("Resources1/Wheat.csv")
```

Benchmark

```
SPY_path = Path("Resources1/SPY.csv")
```

Stocks

```
AMZN_path = Path("Resources1/AMZN.csv")
LUV_path = Path("Resources1/LUV.csv")
PFE_path = Path("Resources1/PFE.csv")
AAPL_path = Path("Resources1/AAPL.csv")
```

Metals

```
Gold_path = Path("Resources1/Gold.csv")
Silver_path = Path("Resources1/Silver.csv")
Platinum_path = Path("Resources1/Platinum.csv")
```

Step-2

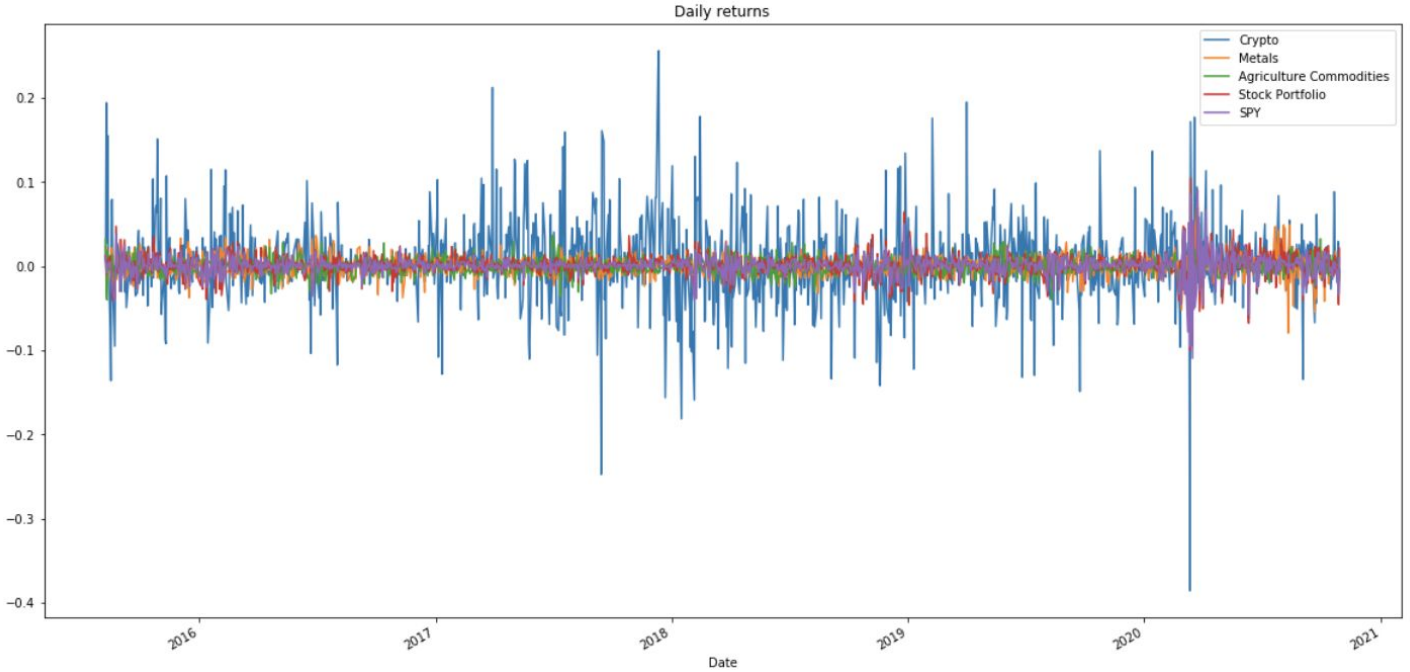
```
def returns(csvpath):
    df=pd.read_csv(csvpath)
    df=df[["Date","Close"]]
    df.set_index(pd.to_datetime(df['Date'],
                                infer_datetime_format=True),
                                inplace=True)
    df.drop(columns=['Date'], inplace=True)
    df_1=df.sort_values(by=['Date'], ascending=True)
    df_new=df_1.dropna()
    return df_new
```

Step-3

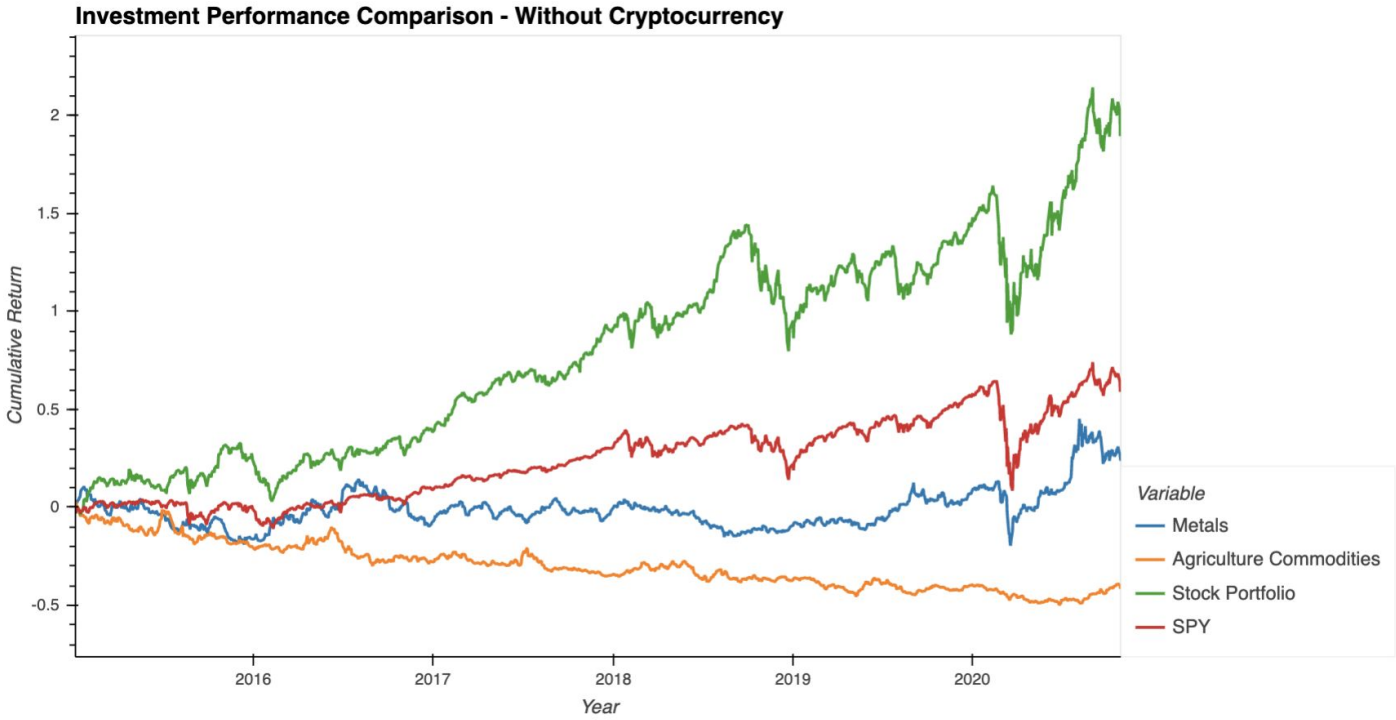
```
SPY_df=returns(SPY_path)
SPY_df.columns=["SPY"]
SPY_df.isnull().sum()
SPY_df=SPY_df.pct_change().dropna()
SPY_df.head()
```

	SPY
Date	
2015-01-05	-0.018060
2015-01-06	-0.009419
2015-01-07	0.012461
2015-01-08	0.017745
2015-01-09	-0.008014

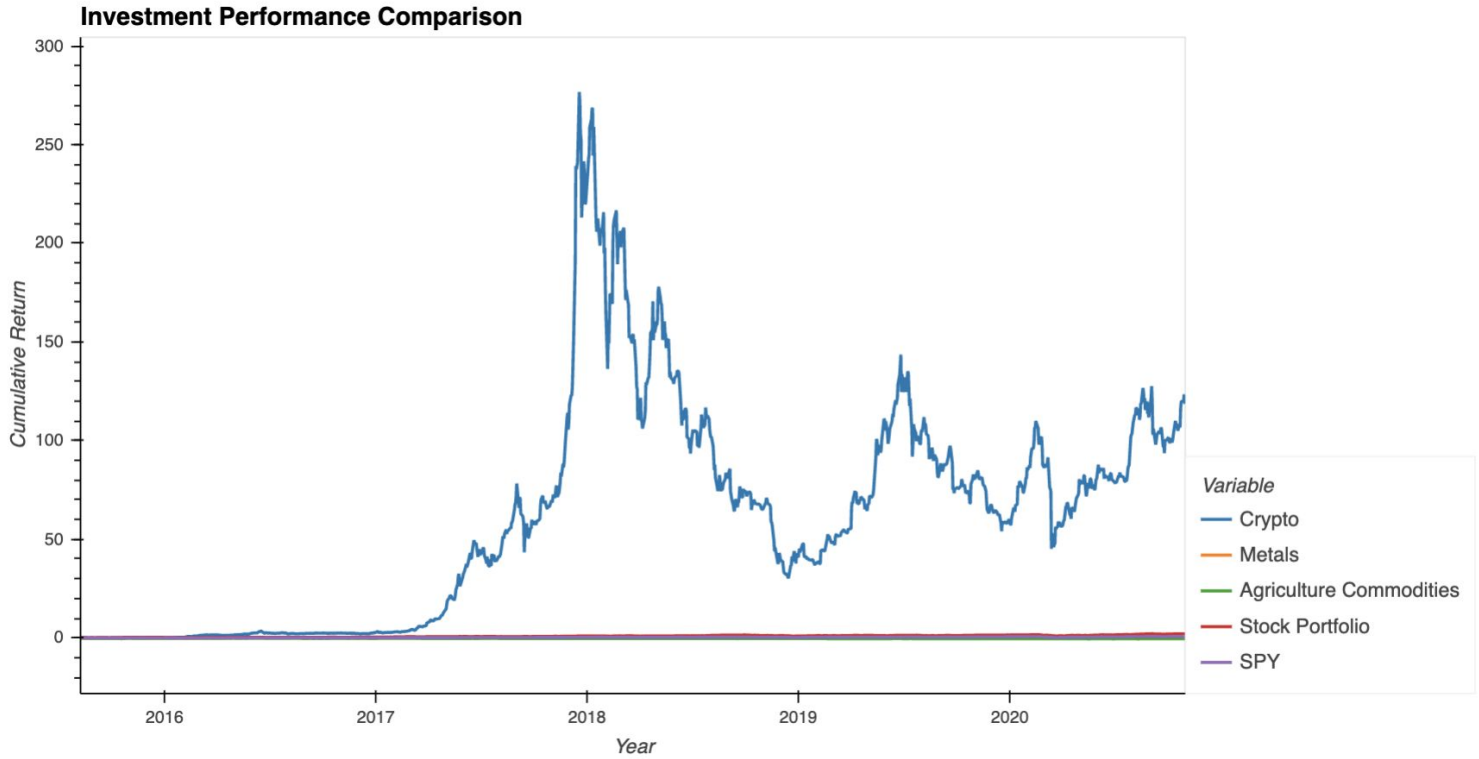
Exploration--Daily Returns



Exploration--Cumulative Returns w/o Crypto

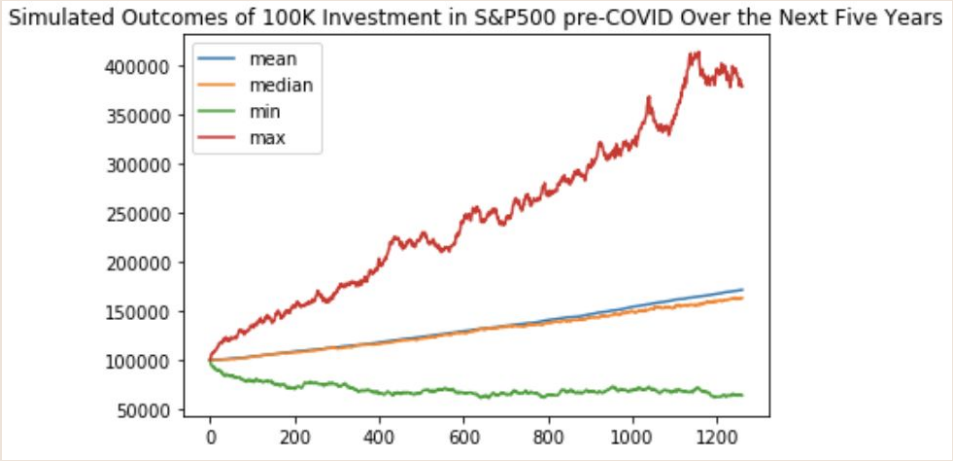


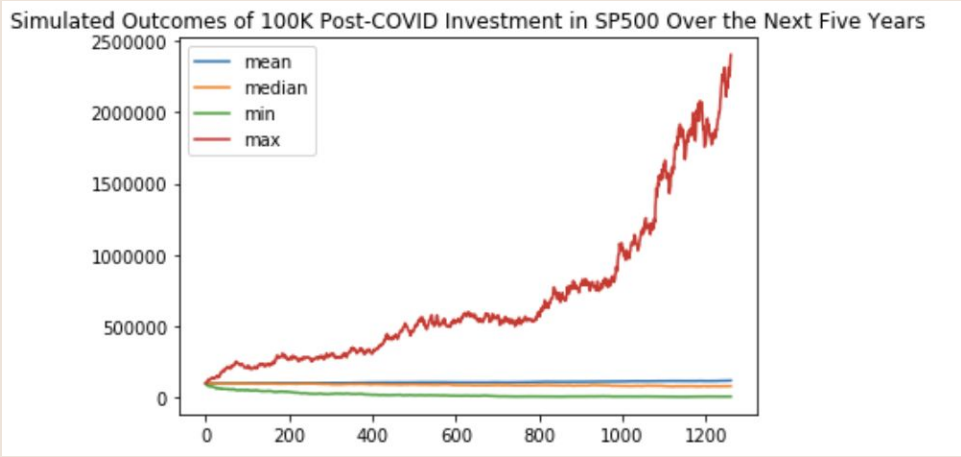
Exploration--Cumulative Returns with Crypto



Monte Carlo: S&P 500

- Used Monte Carlo simulations to obtain a distribution (array) of results of potential future investment behavior based on historical data of traditional portfolios (Commodities, Metals, Stocks, SP500)
- Initial simulations were five year forecasts based on total dataset.
- Supplemental simulations generate forecast for pre-COVID timeframe and post-COVID period.





Sharpe Ratio

- Used Sharpe ratio to compare the risk-adjusted return of various investments such as stocks, ETFs, and commodities.

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

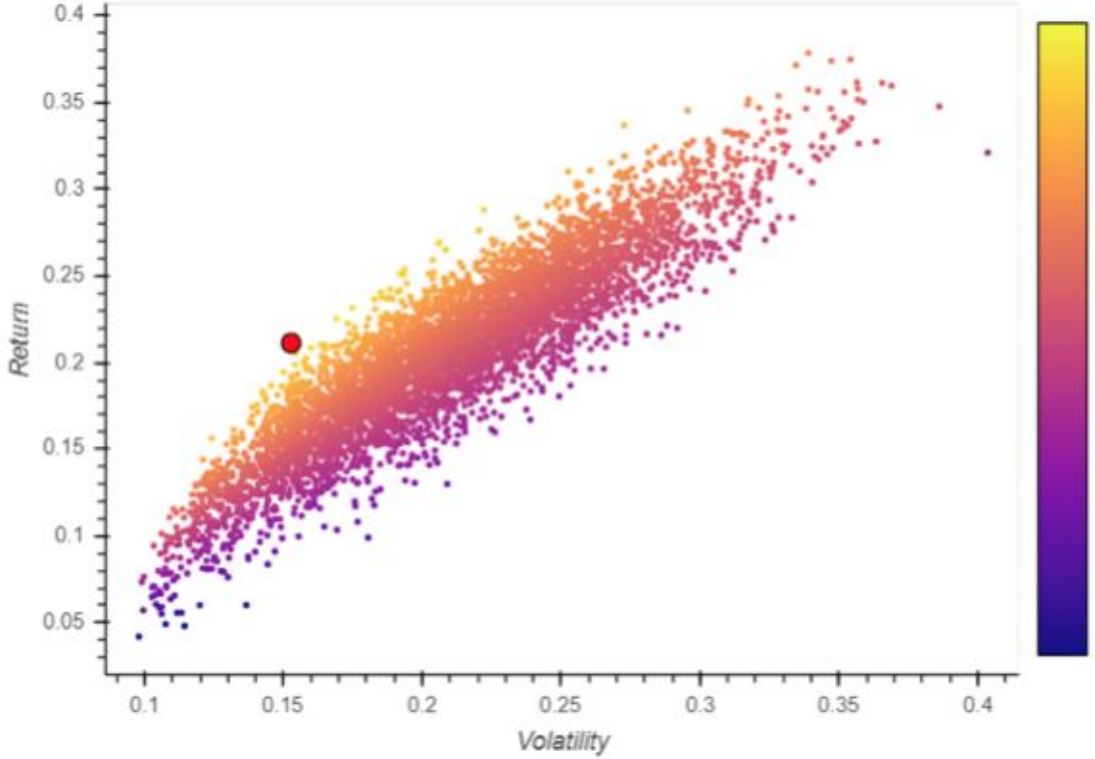
where:

R_p = return of portfolio

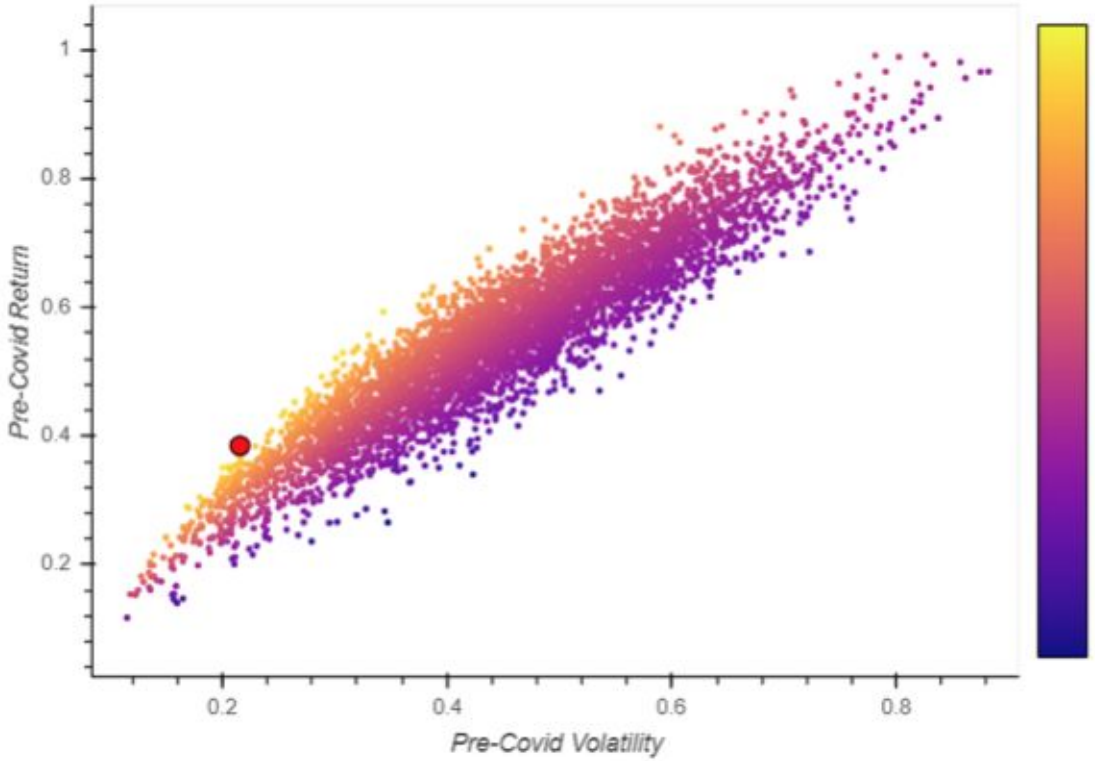
R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

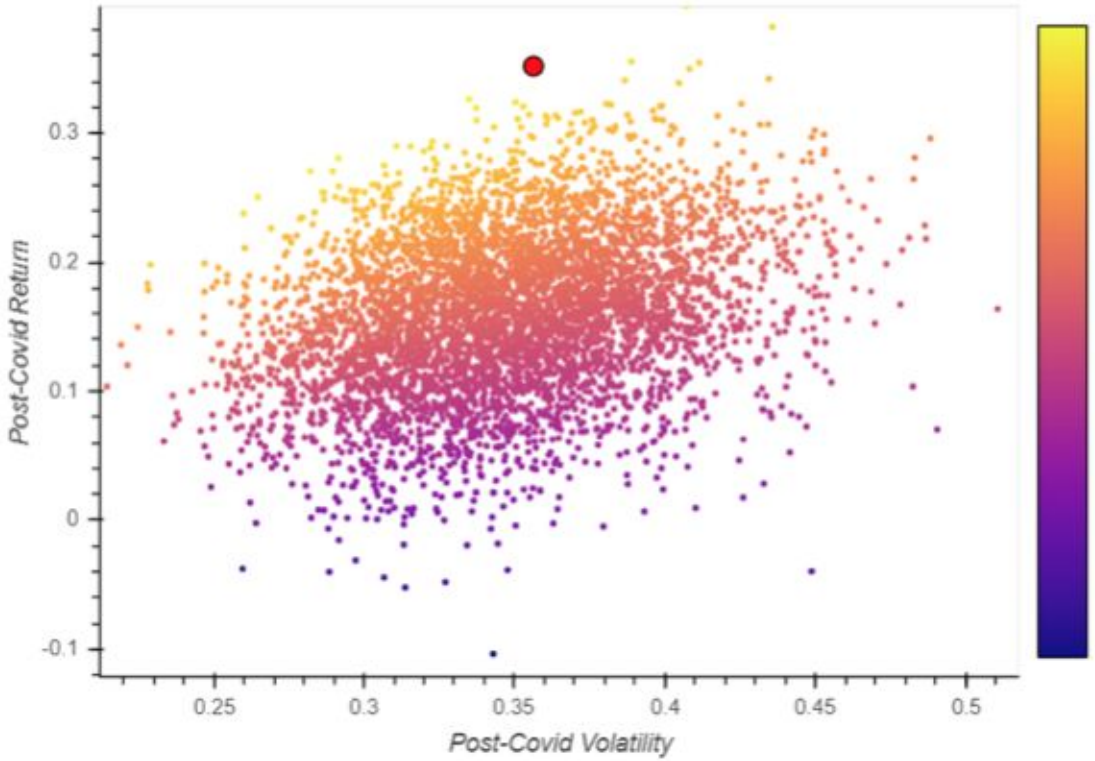
Sharpe Portfolio: Total Data



Sharpe Portfolio: Pre-Covid



Sharpe Portfolio: Post-Covid



Sharpe Portfolio Optimization Code Review

```
num_ports = 5000

all_weights = np.zeros((num_ports,len(Portfolio.columns)))
ret_arr = np.zeros(num_ports)
vol_arr = np.zeros(num_ports)
sharpe_arr = np.zeros(num_ports)

for ind in range(num_ports):

    # Create Random Weights
    weights = np.array(np.random.random(14))

    # Rebalance Weights
    weights = weights / np.sum(weights)

    # Save Weights
    all_weights[ind,:] = weights

    # Expected Return
    ret_arr[ind] = np.sum((log_ret.mean() * weights) *252)

    # Expected Variance
    vol_arr[ind] = np.sqrt(np.dot(weights.T, np.dot(log_ret.cov() * 252, weights)))

    # Sharpe Ratio
    sharpe_arr[ind] = ret_arr[ind]/vol_arr[ind]
```

Sharpe Portfolio - Summary

Asset Class Type	#	Asset Ticker	Optimum Portfolio %		
			Total	Pre-Covid	Post-Covid
S&P 500	1	SPY	8.32	2.62	3.41
Stocks	2	AMZN	13.60	12.40	12.01
	3	LUV	3.33	12.06	3.03
	4	PFE	9.77	9.62	6.89
	5	AAPL	13.99	13.12	11.24
Crypto	6	BTC	13.12	6.81	4.12
	7	ETH	2.16	1.80	15.89
	8	LTC	0.20	0.66	1.14
Commodities	9	Corn	6.10	6.93	5.38
	10	Soybeans	2.89	0.05	14.22
	11	Wheat	2.53	5.85	1.49
Metals	12	Gold	9.59	13.43	6.91
	13	Silver	12.84	11.06	11.58
	14	Platinum	1.56	3.59	2.67

Postmortem

Difficulties:

- Monte Carlo simulation for cryptocurrencies
  - After loading the datasets for BTC, LTC, and ETH, the pricing information for BTC was not populating
  - Utilized Quandl, Kraken, Coin Market Cap, but were unsuccessful in pulling in BTC data
- Dashboard
  - Data formatted as an array instead of a dataframe
  - Initially used plotly to visualize returns
  - Instead utilized hvplot and ensured data was in a dataframe format before running panel

Additional questions for future research:

- Conduct a Monte Carlo simulation to determine the possible outcomes for a $100,000 investment in the cryptocurrency portfolio--bitcoin, litecoin, and ethereum

Questions?