**Covid Capitalism**

**Background:** The Covid-19 Virus sent the world into a global frenzy like never seen before, and perhaps gave birth to the rise of the retail trader. Robinhood made it easy for the average Joe to full blown dive into securities trading, cold turkey. The combination of easy access from apps like Robinhood and the market conditions of the global Coronavirus pandemic created a massive shakedown of the stock market.

**Scenario:** It is March 09, 2020 and you are anxiously watching the impact of the Covid-19 pandemic unfold into society. The stock market crash of 2020 has just begun on Monday, with history's largest point plunge for the Dow Jones Industrial Average (DJIA) up to that date. It was followed by two more record-setting point drops on March 12 and March 16. The stock market crash included the three worst point drops in U.S. stock market history.

Amidst the global frenzy, your good buddy, seasoned trading veteran, Jon Raccah is convinced that there is no better time ever to take advantage of the huge drop in share prices and capitalize on this opportunity under the assumption that share prices will rebound sooner rather than later, leading to glorious gains.

After consulting with John and all things considered, you are convinced that it is indeed the ultimate opportunity to capitalize on chaos and decide to invest **$200,000** into the stock market and cryptocurrency on **March 17, 2020.** You decide to invest half of your portfolio in blue chips to mitigate long term risk, and the other half in crypto currency, volatile stocks, pandemic related airline stocks and pandemic related hospitality stocks. You are locked and loaded, eager to see how your investment plays out, and how soon the economy recovers from the initial shock of the Covid-19 scare.

**Your portfolio consists of the following distribution:**

**Bluechips: Total $100,000**

**APPL:** $25,000

**MSFT:** $25,000

**AMZN:** $25,000

**BRK.A:** $25,000

**Cryptocurrency: Total $25,000**

**BTC:** $12,500

**ETH:** $12,500

**Volatile stocks: Total $25,000**

**TSLA:** $12,500

**GME:** $12,500

**Pandemic Crash Airlines Stocks: Total $25,000**

**BA:** $12,500

**DAL:** $12,500

**Pandemic Crash Hospitality Stocks: Total $25,000**

**MGM:** $12,500

**WYNN:** $12,500

**STEP 1: Retrieve Data**

-Import Libraries And Dependencies:

- Load .env enviroment variables

- Set Alpaca API key and secret

### - Get all open, high, low and closing prices for entire portfolio from March 17, 2020 to current day of Data via API Call and Read in as DataFrame.

### -Data cleaning, Concat into single Dataframe for whole Portfolio to prepare for analysis and visual presentation.

### -view summary statistics df.describe()

### -Plot portfolio daily close Dataframe on line chart to visualize all data from March 17- today

### -Sort the DataFrame by Close to Get Records with Top Daily Returns, Slice out 5 records and plot top 5 performing days of entire portfolio and bottom 5

### --Group DataFrame by category (bluechip, cyrpto, volatile, airlines, hospitality) and calculate the average closing price

**STEP 2: Portfolio Analysis**

### -Calculate daily returns .pctchange

### - Plot daily returns

### -Visualize the distribution of daily returns across all stocks using a histogram plot

### Hint: To make the plot easier to read, set the alpha arguement to 0.5

### df\_daily\_returns.plot.hist(alpha=0.5)

### Visualize the distribution of daily returns across all stocks using a density plot

### df\_daily\_returns.plot.density()

### - Calculate volatility = daily\_returns.std() \* np.sqrt(252)

### volatility.sort\_values(inplace=True)

### -Calculate cumulative returns. cumulative\_returns = (1 + daily\_returns).cumprod()

### -plot cumulative returns

### -Calcualte standard deviation using std function. Sort standard deviation in desc order. Plot standard deviation. Calculate the annualized standard deviation. Plot annualized standard deviation

### -Plot standard deviation for the 5 different categories to determine which has the most risk

### -Caculate and plot sharpe ratios using risk free rate and annualized std deviation

### -Calculate and plot correlation between each investment in portfolio

### -Rolling 7-Day Mean & Standard Deviation of Closing Prices. 30 day. 180 day. Plot all on same figure. (ax=ax)

### -Calculate Covariance of Portfolio Returns vs. S&P 500 Returns

### -Calculate Variance of S&P 500 Returns

### -Calculate Beta Values of Portfolio

### -Calculate and plot 30-Day Rolling Betas of Portfolio Returns vs. S&P 500 Returns. 50 day. 180 day.

### -Calculate current value in dollars of stock portfolio. Create a DataFrame with the current value of shares. # Create a pie chart to show the proportion of stocks in the portfolio. # Create a bar plot to show the value of shares

**STEP 3: Forecasting**

### -Simulate five year portfolio growth using Monte Carlo simulation

Plot simulation outcomes

line\_plot = MC\_fiveyear.plot\_simulation()

Save the plot for future usage

line\_plot.get\_figure().savefig("MC\_fiveyear\_sim\_plot.png", bbox\_inches="tight")

-Plot probability distribution and confidence intervals

dist\_plot = MC\_fiveyear.plot\_distribution()

Save the plot for future usage

dist\_plot.get\_figure().savefig('MC\_fiveyear\_dist\_plot.png',bbox\_inches='tight')

-# Fetch summary statistics from the Monte Carlo simulation results

tbl = MC\_fiveyear.summarize\_cumulative\_return()

### -- Calculate and plot the Simulated Profits/Losses of each initial Investment  Over the Next 5 years

### cumulative\_pnl = initial\_investment \* df\_simulated\_returns

-Use the lower and upper `95%` confidence intervals to calculate the range of the possible outcomes of our investments.

ci\_lower = round(tbl[8]\*10000,2)

ci\_upper = round(tbl[9]\*10000,2)

### -Simulate five year portfolio growth with with different weights. Plot all outcomes

### -Simulate ten year growth with current weights. Plot all outcomes.

### - Simulate ten year portfolio growth with with different weights. Plot all outcomes.

### \*\*\*\* Implement plotly, HVplot, interactive plots etc when applicable and create dashboard for all visualizations.

![San Francisco Park Reading](Images/san-Francisco=park-reading.jpg)

\*[Title](picture path)