Quantum Returns: Gambler’s Ruin in the Stock Market

Ethan Bostick

Department of Mathematics, University of Mary Washington

STAT 381: Probability and Statistical Inference I

Dr. Debra L. Hydorn

December 11th, 2025

Table of Contents

Abstract

This study examines the applicability of the "Gambler’s Ruin" problem within the context of highly volatile stock market assets, focusing specifically on Rigetti Computing Inc (RGTI). Due to its dramatic daily price swings driven by speculative interest in quantum computing, RGTI serves as a prime example of an asset targeted by short-term traders seeking large returns through high risk. We simulated 4,000 trading scenarios over 30-day timelines using a Monte Carlo-like method based on five years of historical returns data. Four distinct trading strategies, including variations of stop-loss and profit-taking rules, were tested against the volatile price movements. The results unequivocally demonstrate that despite the implementation of risk-mitigating strategies and the potential for outlier returns exceeding 150%, the mean return across all strategies remained negative. This simulation confirms that for assets characterized by high volatility and a historical tendency toward negative drift, the inherent mathematical disadvantage—the "house edge"—ensures that repeated, continuous betting, irrespective of strategy, leads to a net loss over time, perfectly illustrating the principles of the Gambler’s Ruin in modern finance.

Introduction

This report explores the application of the "Gambler's Ruin" problem in the stock market. Specifically, this problem will be applied to the public shares of Rigetti Computing Inc (RGTI).

RGTI has recently received attention due to renewed interest in quantum computing, as well as the dramatic and lucrative price swings the stock experiences on a daily basis. This volatile movement has made it a prime subject for seeking "fast and easy money," where traders take on high risk for potential large daily returns. That being said, the share price of RGTI represents a purely speculative value where investors look toward the potential of the stock in the future. Such stocks with speculative prices tend to undergo dramatic swings based on technological advancements or government contracts offered to the company.

The nature of an experimental tech stock like RGTI has led it to yield more negative daily returns than positive ones over the past five years, even accounting for "black swan" events (rare and unpredictable occurrences) such as the recent renewed interest. This report will demonstrate that, regardless of the strategy, betting on these stocks continually will result in losses more often than not.

Methods

The experiment was simulated using a Monte Carlo-like method written in Python, utilizing data from the past five years of RGTI's daily returns. To generate random daily returns for the simulation, the mean and standard deviation of the log daily returns (to create additive data) were calculated using Python's NumPy library and its random normal value function. Using this method assumes the underlying distribution of RGTI's returns is normal; this is a safe assumption given the Law of Large Numbers applied to five years of data and a roughly normal-shaped distribution (Fig 1).

A total of 4,000 simulations were conducted. Technically, this consisted of 1,000 simulated 30-day timelines. For each timeline, four different trading strategies were applied. Each simulation assumed an initial investment of $10,000 over a period of 30 days (a typical timeframe for a short-term trade). The strategies were defined as follows:

- Strategy 1: Do nothing and hold the stock for 30 days.
- Strategy 2: Hold until at least a 7% return is achieved, then sell if the stock falls more than 7% (provided it does not cut into the 7% total return accrued). This is effectively a "sell while high" strategy.
- Strategy 3: Only sell if the total return drops to or below -7% (minimize losses).
- Strategy 4: A mix of the previous two: sell if the total return is at or below -7%, or sell while high.

With four different strategies applied to 1,000 different timelines of returns, the total return on the initial investment for each was recorded and graphed.

Results

Each simulated timeline for each strategy is displayed via a line graph showing each portfolio's value across the 30 days (Figs 2, 4, 6, and 9). Interestingly, there were some simulated runs in which the portfolio "blew up" to over a 150% return on the initial investment. However, for every single simulation that achieved this, there were many—if not most—that resulted in negative returns.

Table 1 below summarizes the percent change from the initial investment to the end value for each simulation.

Table 1

Summary of each strategy

Strategy	Mean	Standard Deviation	Min	25%	50%	75%	Max
1	-7.797	36.254	-79.196	-34.759	-13.525	12.300	161.620
2	-5.584	33.540	-79.196	-32.756	-3.570	15.793	161.620
3	-3.331	21.409	-23.965	-11.866	-9.223	-7.556	128.151
4	-2.144	18.553	-23.965	-11.515	-8.694	-7.031	128.151

Note: All values are percentages, not decimals. Each strategy uses data from the same 1,000 simulated daily return timelines with no insight into future values.

The summary information shows how the expected return shifts based on the method used. As expected, strategies implementing a stop-loss (minimizing loss) resulted in mean returns closer to zero. Conversely, the strategies without stop-losses maintained higher maximum values, even when compared to the "sell while high" approach (Strategy 2).

Interestingly, the quartiles shift significantly between strategies. Those implementing stop-losses have much lower third-quartile values for returns, while the others have significantly higher third-quartile values. However, despite the high upper quartiles, Strategies 1 and 2 exhibit extremely low negative values in their lower quartiles. These deep losses demonstrate how having no limitations on such a volatile asset can lead to losing the majority of the initial investment in worst-case scenarios.

Looking at the distributions for each strategy, Figure 3 illustrates the true returns of each simulation over the 30-day period. The distribution is right-skewed and mostly unimodal. Interestingly, this shape mimics a log-normal distribution; however, this distribution is discrete, as each day represents a step.

When implementing Strategy 2, the distribution (Fig 5) remains right-skewed but becomes bimodal, forming a valley around the zero-return value. This is likely explained by the strategy aiming for at least a 7% return, resulting in a cluster of results near the ~10% return mark.

Strategy 3's distribution is notable in that a large number of simulations effectively ended within the first five days (Fig 7), a trend reflected in the histogram in Figure 8. The histogram is unimodal with a heavy right skew, representing the simulated portfolios that survived without encountering immediate misfortune.

Strategy 4's histogram appears exactly as expected, reflecting a mix of Strategies 2 and 3. It is bimodal and right skewed, featuring a large number of returns around the -10% mark and a smaller, yet visible, group around the 10% return mark. This confirms that blending the stop-loss and "sell while high" methods retains returns but creates distinct outcome clusters.

Ultimately, while each strategy reduced risk in some way (noting that higher deviation typically correlates with higher risk and potential return), every strategy ended with a mean return below 0%. Each simulation effectively functions as a different attempt to profit from RGTI; the mean results confirm that, on average, the bet will end with a negative return regardless of the strategy employed.

Conclusions

Overall, this study demonstrates that even with a defined strategy, repeated bets on a highly volatile asset such as RGTI—while potentially scoring big—will result in net losses over time, perfectly displaying the Gambler's Ruin.

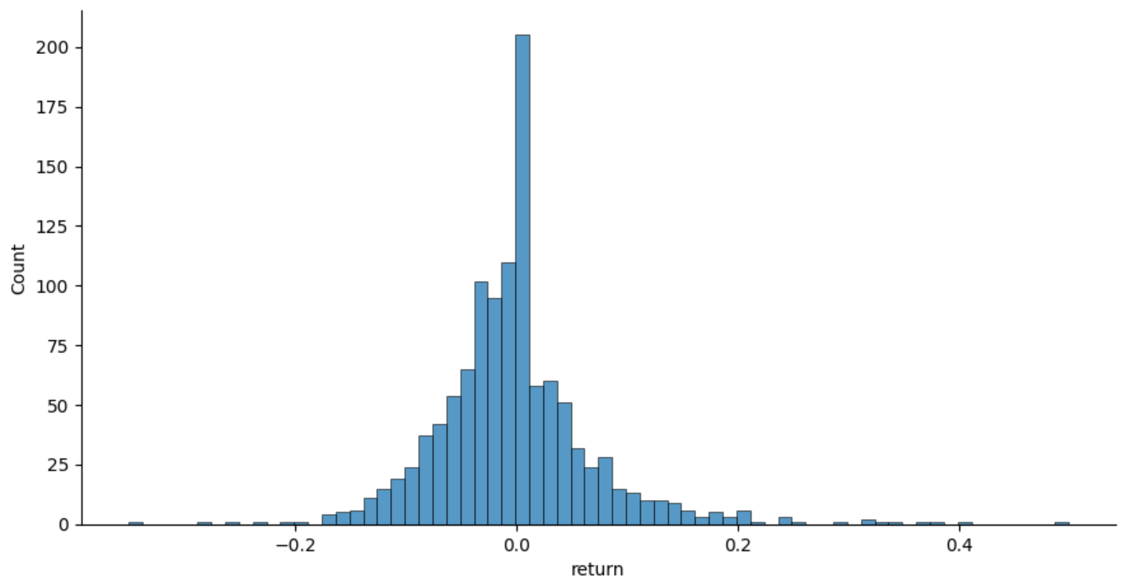
It is important to note that having a strategy can vastly limit the losses an individual faces in any single attempt. That being said, this study represents a naïve simulation of the stock market based on historic daily returns. The real stock market is currently impossible to simulate with total accuracy due to the inability to predict mass psychology and black swan events.

However, this study could be improved by expanding the strategies employed. Only four strategies were showcased here, one of which was inaction and another a hybrid. Future research could implement more complex algorithms to further test the resilience of the Gambler’s Ruin theory. Until then, the data suggests that for speculative assets like RGTI, the "house edge"—driven by negative drift and volatility—remains a formidable barrier that simple trading rules cannot easily overcome.

Figures

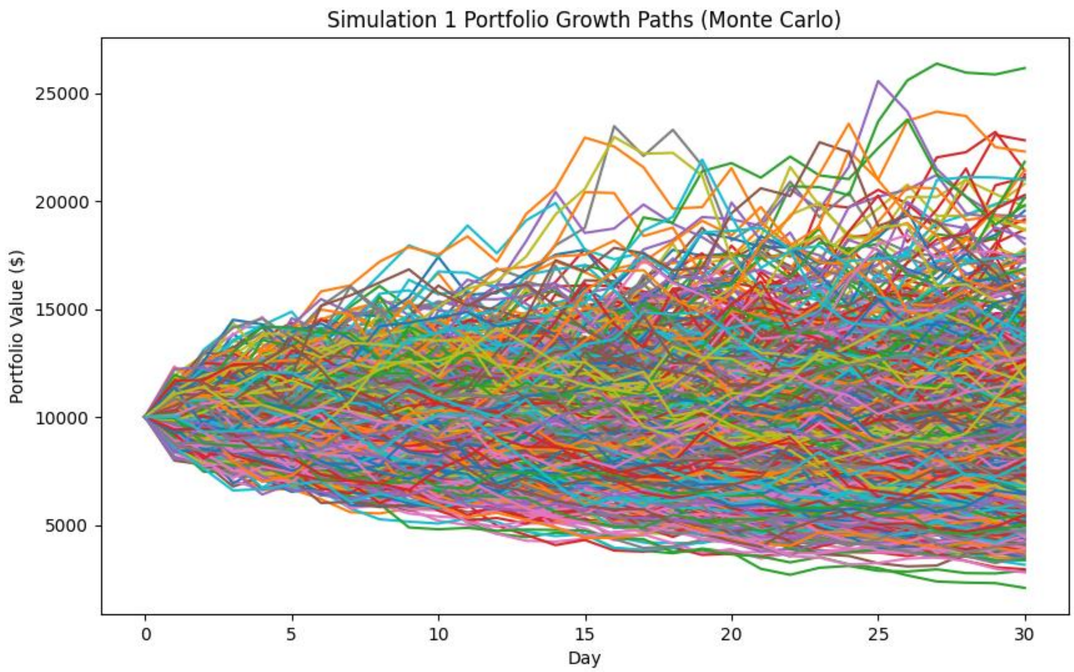
Figure 1:

Historic daily returns of RGTI (five years)



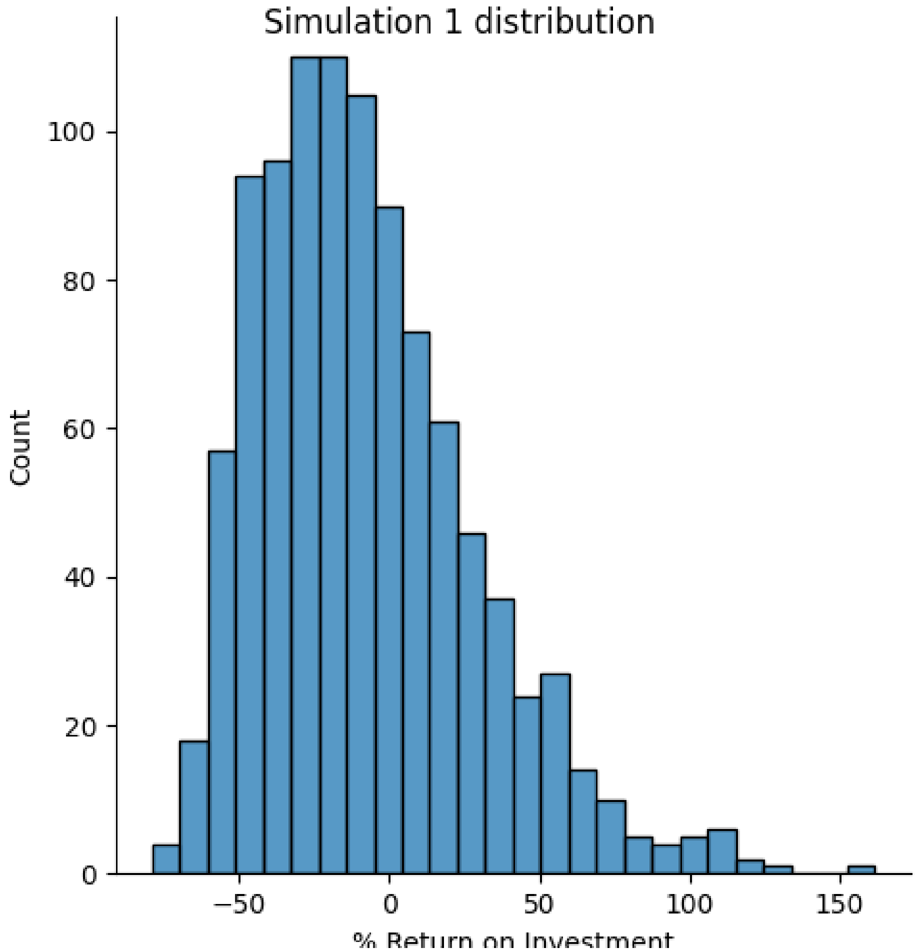
Note: historic data of the past five years of RGTI's daily return percentage

Figure 2:



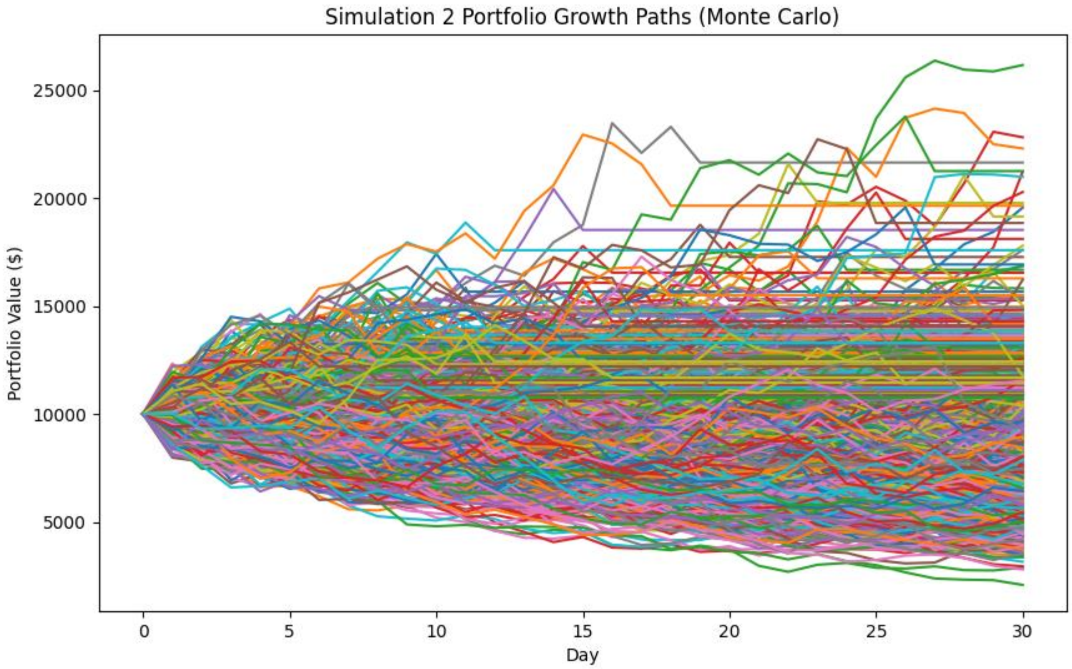
Note: simulation of 1000 random daily returns over 30 days functionally working as 1000 different timelines. This is also the result of strategy 1.

Figure 3:



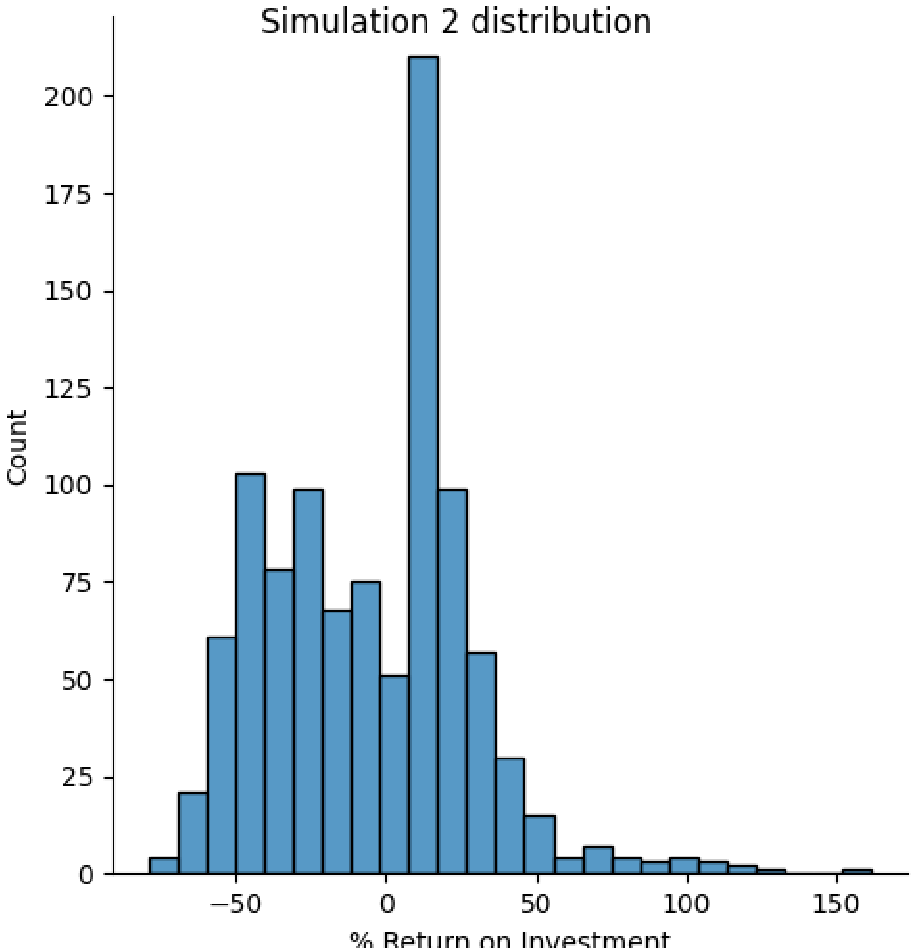
Note: strategy one's resulting histogram displaying the distribution of final returns on investment for the 1000 simulated timelines.

Figure 4:



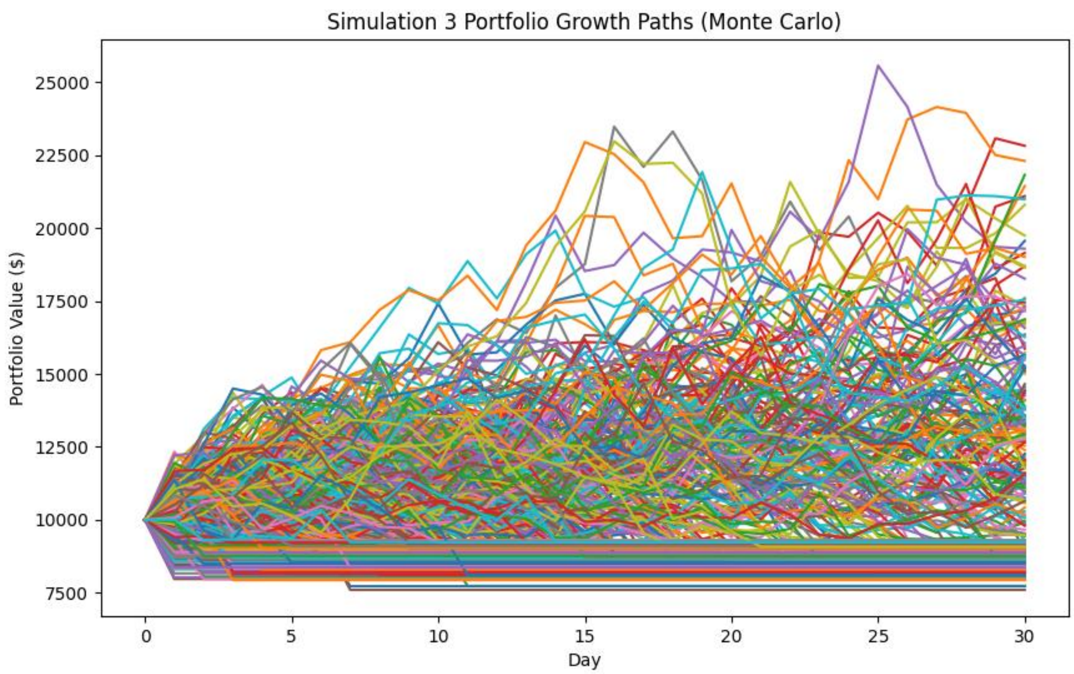
Note: simulation follows the same timelines as from the first 1000 but applies strategy two.

Figure 5:



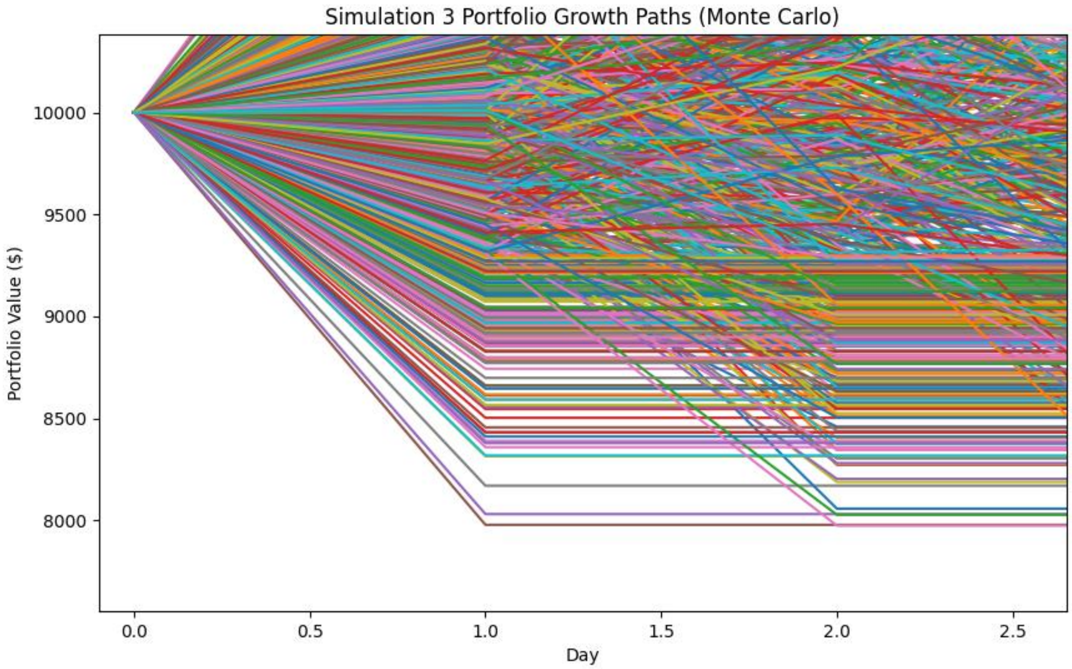
Note: strategy two's resulting histogram displaying the distribution of final returns on investment for the 1000 simulated timelines.

Figure 6:



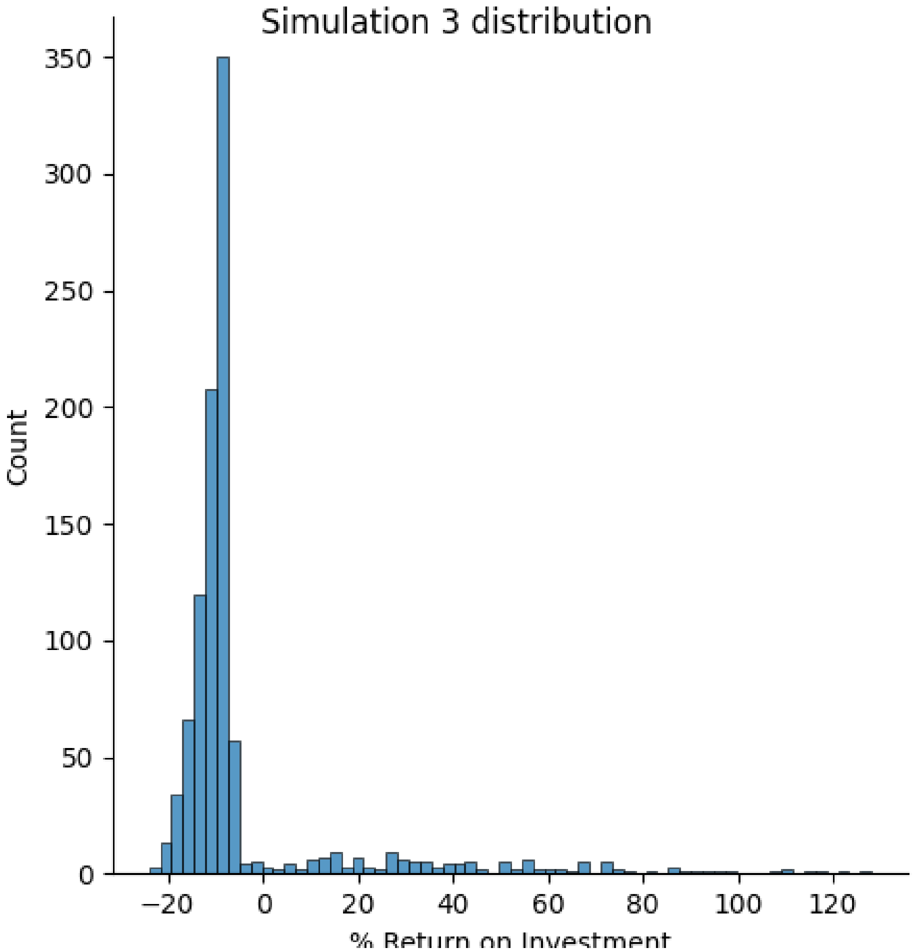
Note: simulation follows the same timelines as from the first 1000 but applies strategy three.

Figure 7:



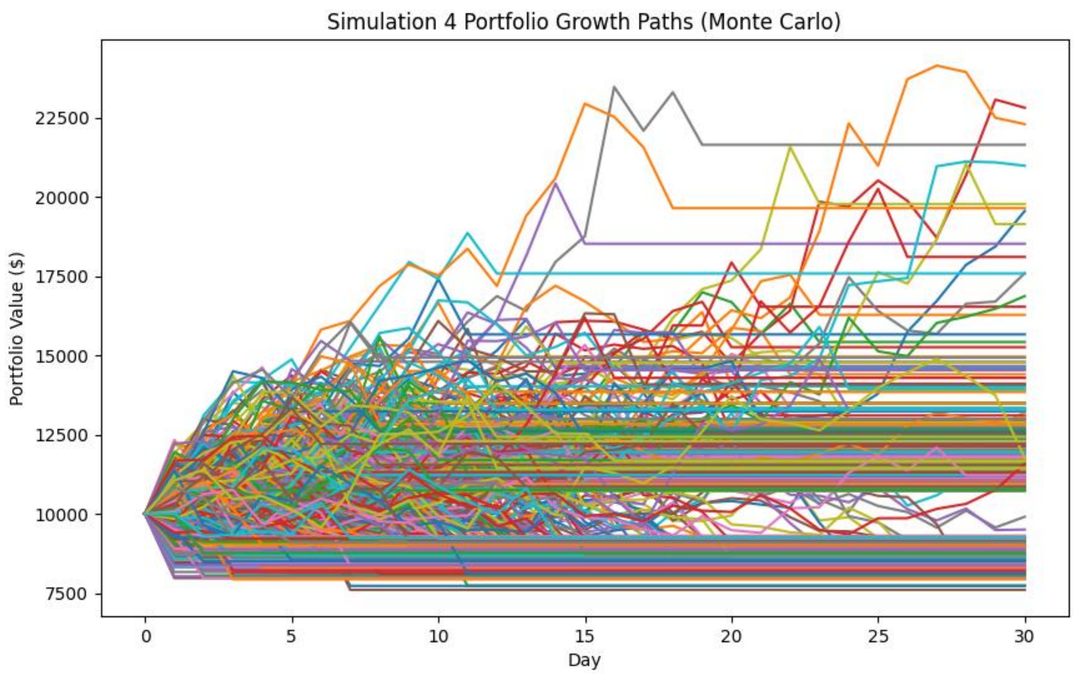
Note: zoom in on figure 6 displaying the investment strategy coming into play after one or two days.

Figure 8:



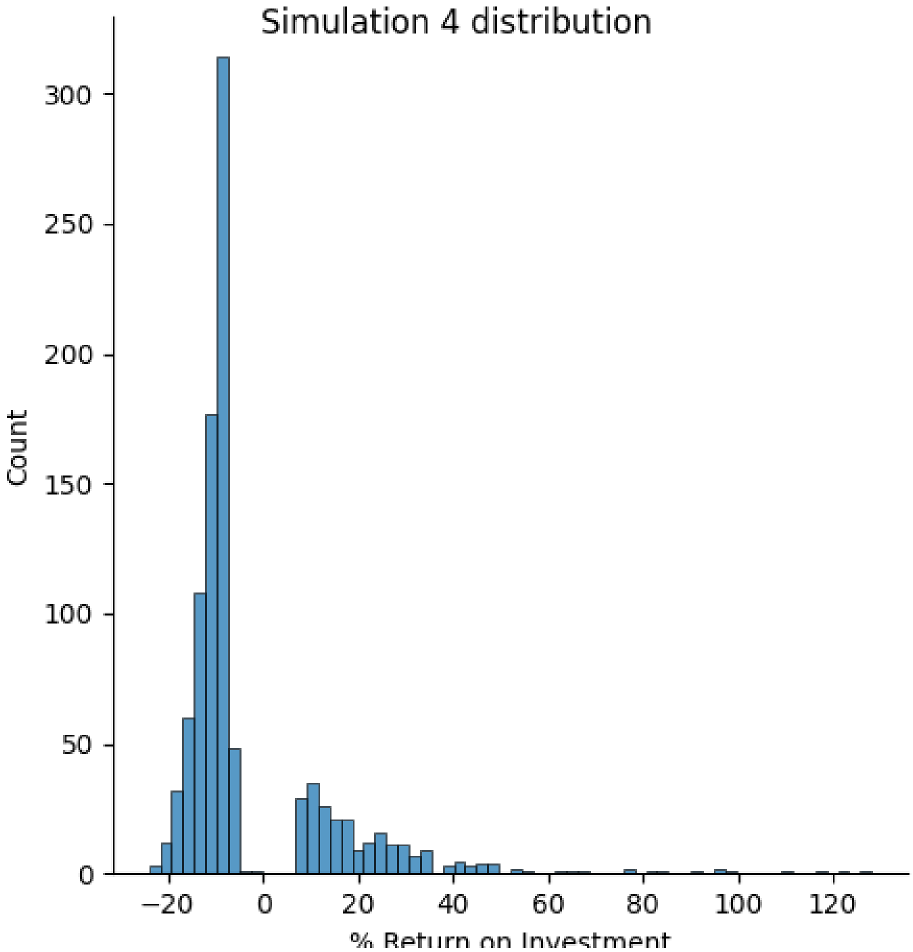
Note: strategy three's resulting histogram displaying the distribution of final returns on investment for the 1000 simulated timelines.

Figure 9:



Note: simulation follows the same timelines as from the first 1000 but applies strategy four.

Figure 10:



Note: strategy four's resulting histogram displaying the distribution of final returns on investment for the 1000 simulated timelines.

Appendix

#Python Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

df = pd.read_csv('~/desktop/381Project/rgtiHistData.csv') #file path
for col in ['Open', 'Close/Last']:
  df[col] = df[col].astype(str).str.replace('$', '', regex=False).astype(float)

# transform to logs to allow arithmatic means!!! Importante
df['return'] = np.log(df['Close/Last'] / df['Open'])
m, std = df['return'].mean(), df['return'].std()

# Simulation parameters
z = 10000
i = 30
num_sims = 1000

#trade strategies
#0) pray and hold
#1) hold until at least 7% total return, pull out on bad day(if the days return is less than -7&) as long as it isn't cutting into 7% overall
#2) sell on -7% total return at any given point in time, otherwise hold
#3) strat 2 but with 1's sell while high strat
strat0 = np.zeros((i, num_sims))
strat1 = np.zeros((i, num_sims))
strat2 = np.zeros((i, num_sims))
strat3 = np.zeros((i, num_sims))

delta = 0.07

for sim in range(num_sims):
  per_returns = np.random.normal(m,std,i)
  sim_returns = z * np.exp(np.cumsum(per_returns))
  temp = np.copy(sim_returns)
  #0
  strat0[:, sim] = sim_returns

  #1
  out = False
  prev = 0
  for r in range(len(sim_returns)):
    if out:
      sim_returns[r:] = sim_returns[r-1]
```

```
            break
        tChange = (sim_returns[r] - z)/z
        if (per_returns[r] <= -delta) and (tChange >= 0.07):
            out = True
            continue
        prev = tChange
    strat1[:,sim] = sim_returns
    sim_returns = np.copy(temp)

    #2
    out = False
    for r in range(len(sim_returns)):
        if out:
            sim_returns[r:] = sim_returns[r-1]
            break
        tChange = (sim_returns[r] - z)/z
        if tChange <= -0.07:
            out = True
            continue
    strat2[:,sim] = sim_returns
    sim_returns = np.copy(temp)

    #3
    out = False
    prev =0
    for r in range(len(sim_returns)):
        if out:
            sim_returns[r:] = sim_returns[r-1]
            break
        tChange = (sim_returns[r] -z)/z
        if (per_returns[r] <= -delta) and (tChange >= 0.07):
            out = True
            continue
        if tChange <= -0.07:
            out = True
            continue
        prev = tChange
    strat3[:,sim] = sim_returns
    sim_returns = np.copy(temp)

#post sim
first = np.full((1,num_sims),z)
strat0 = np.insert(strat0, 0,first,axis=0) #insert the inital deposit before first trade day
strat1 = np.insert(strat1, 0,first,axis=0)
strat2 = np.insert(strat2, 0,first,axis=0)
strat3 = np.insert(strat3, 0,first,axis=0)
sims = [strat0,strat1,strat2,strat3]
```

```
k = 1
for s in sims:
   change = ((s[-1] - s[0])/s[0])*100
   x = pd.DataFrame(change)
   print('---------------------')
   print(f'sim {k} summary:')
   print(x.describe())

#plot
   fig = plt.figure(figsize=(10,6))
   plt.plot(s)
   plt.title(f"Simulation {k} Portfolio Growth Paths (Monte Carlo)")
   plt.xlabel("Day")
   plt.ylabel("Portfolio Value ($)")
   plt.show()
   plt.close(fig)

#dist
   g = sb.displot(change)
   g.fig.suptitle(f"Simulation {k} distribution")
   plt.xlabel("% Return on Investment")
   plt.show()
   plt.close(g.fig)

   k += 1
```