WILLIAM MARSH RICE UNIVERSITY

CAPM Validation

Dr. John Dobelman, Brandon Fantine, Edward Wong, Justin Jiang, Saswat Pati

Table of Contents

Introduction

The purpose of this mini project was to replicate and extend Wojciechowski and Thompson's conclusions that showed that in almost all years RANDOM portfolios lie above the capital market line. To do this, we replicated the CAPM validation method for randomly generated portfolios based on data from 1970 onwards.

CAPM Validation

We were able to successfully generate CAPM models for all years post 1970 with the exception of 1973, 1976, 1978, 1983-1986, and 1992. For the remainder of our successfully generated models we were able to show that Wojciechowski and Thompson's conclusions were, more-often-than-not, correct. For the randomly generated portfolios created, in which real-life data for the corresponding year was used as a basis for the Fama French Risk Free Rate and the data itself (see *ii. CAPM Tabulations*), we see a majority of the graphs - $\frac{31}{44}$ or roughly 70% - have most of (if not all) their data points lie above the Capital Market Line (abbreviated on each graph as *cml*). All CAPM plots can be seen in section *iv. CAPM Graphs*, where the x-axis represents volatility and the y-axis average return.

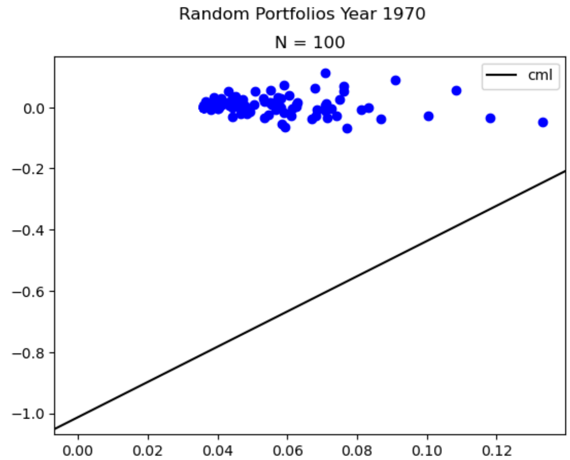
Tabulations used in CAPM Models and Finding

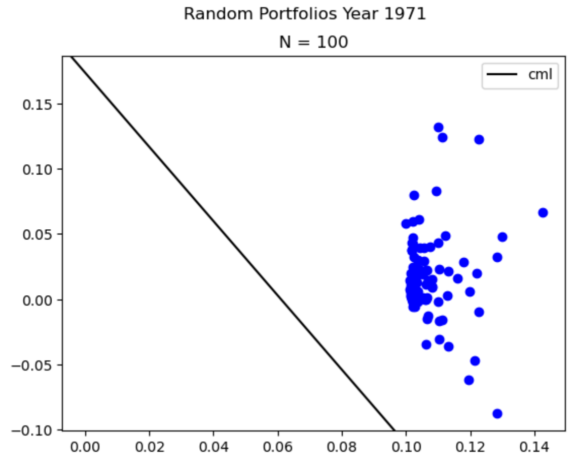
The following is a link to all tabulations used to construct the FF Risk Free Rate for all CAPM Models from 1970-2022: CAPM Tabulations. Due to size, these cannot be included in the document.

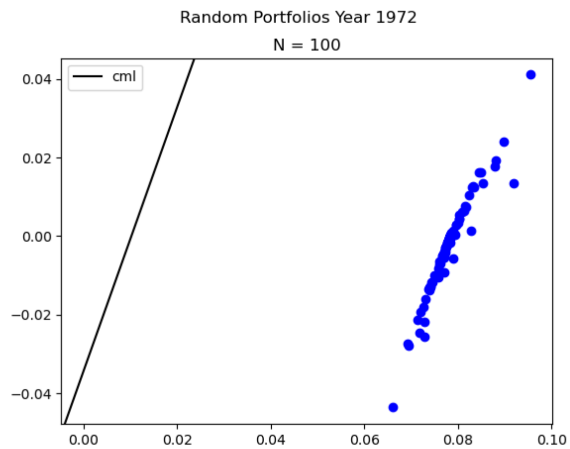
All graphs employed 100 random data points that were calculated using the equations for random weights and random returns postulated by Wojciechowski and Thompson in *Market Truths: Theory versus Empirical Situations* [1]. In addition to those values, 100 random stocks were also collected using the above data and fed into an algorithm combining that real-world information with the random weights and returns, creating an accurate model of what would be 100 theoretical returns for 100 theoretical stocks in each year's global market. As a result, this gave rise to many different distributions where data points were in various quadrants of the graph and could range anywhere from extremely scrunched up to very spread apart. Looking within the tabulations, something similar is modeled in the returns for all stocks each year in the sense that all stocks did either:

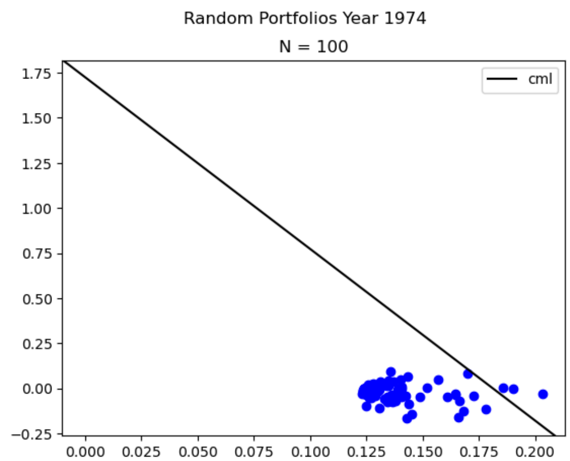
a) all relatively poor with either similar or different volatilities
b) all relatively well with either similar or different volatilities
c) some poor and some well with either similar or different volatilities

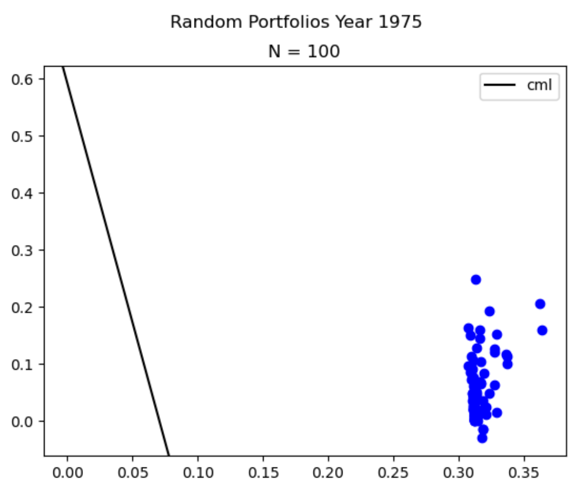
CAPM Graphs Using a Randomized Portfolio from 1976-2022

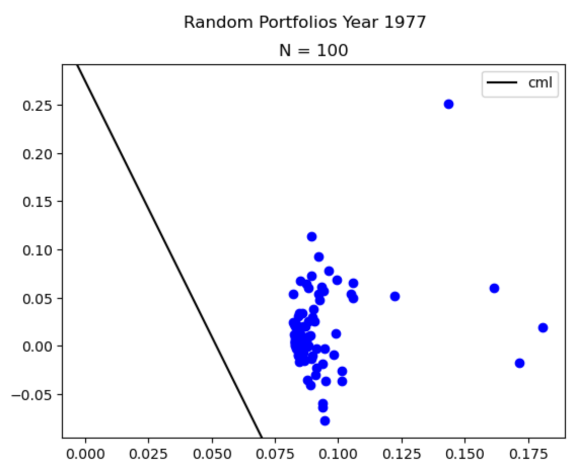


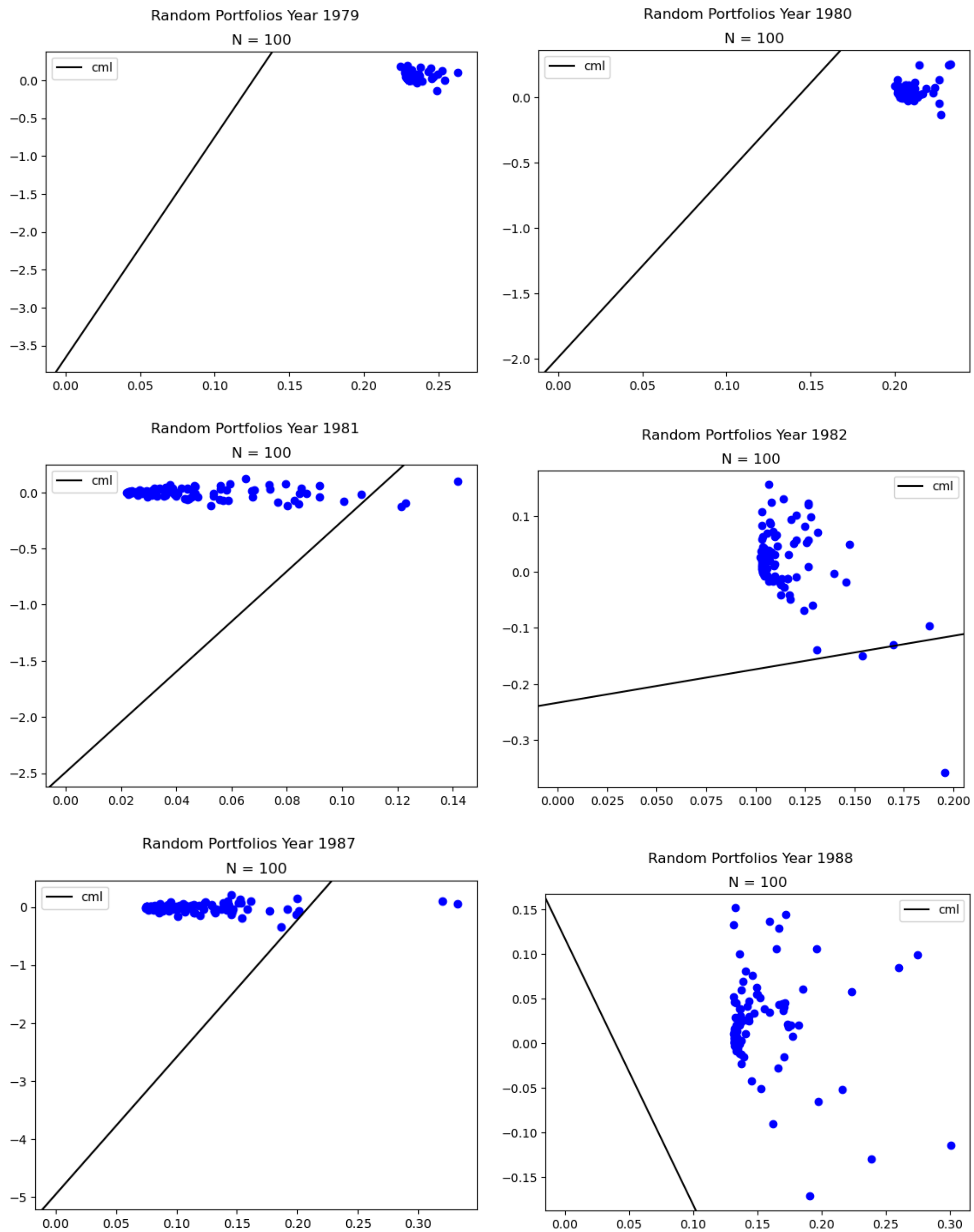


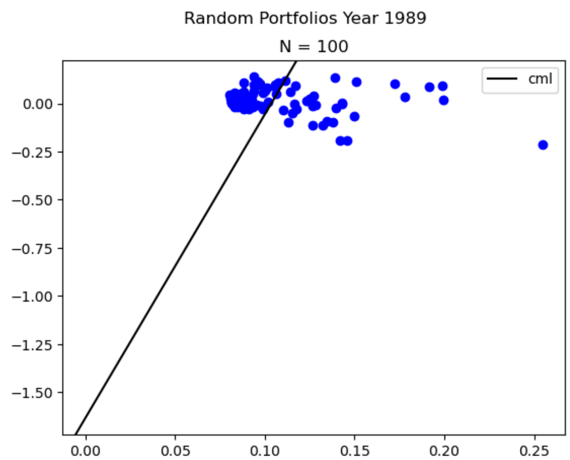


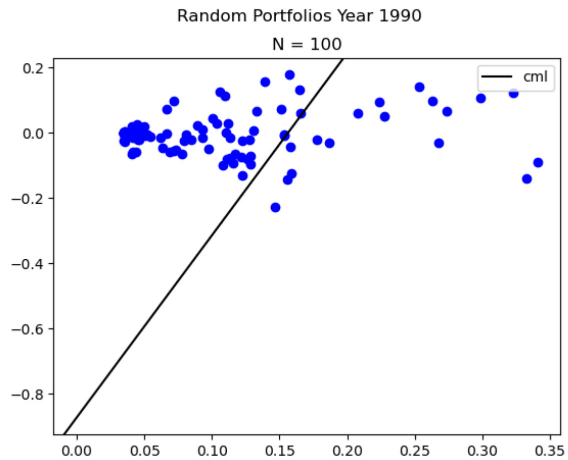


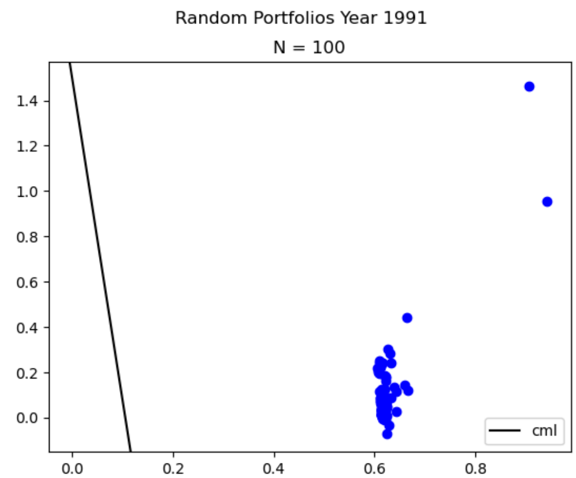


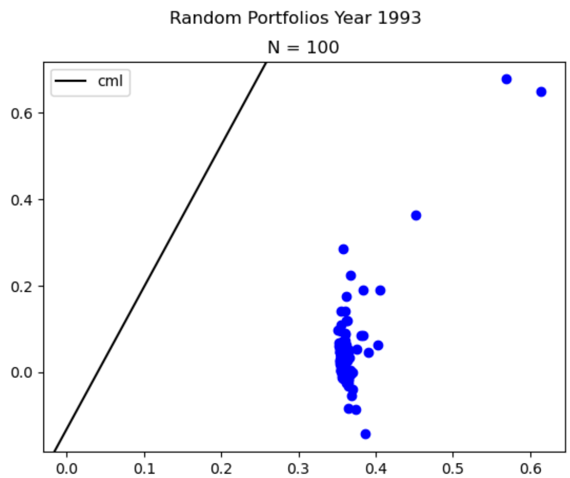


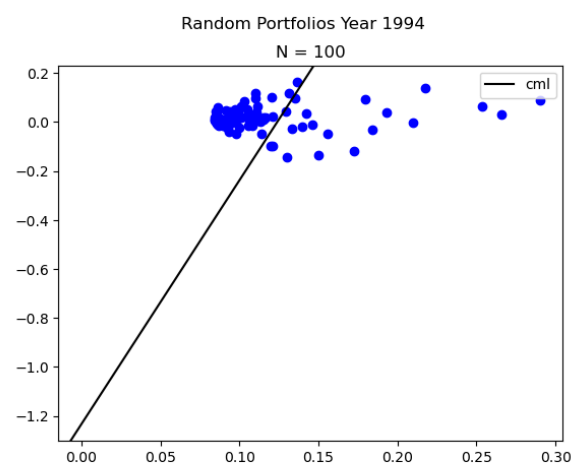

Random Portfolios Year 1979
N = 100
cml
Random Portfolios Year 1980
N = 100
cml
Random Portfolios Year 1981
N = 100
cml
Random Portfolios Year 1982
N = 100
cml
Random Portfolios Year 1987
N = 100
cml
Random Portfolios Year 1988
N = 100
cml

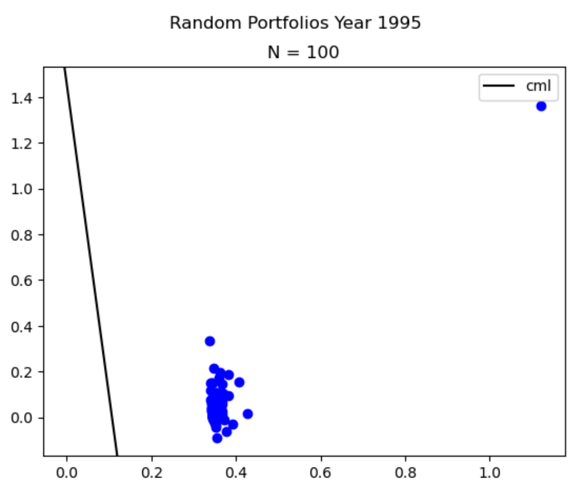

Random Portfolios Year 1989
N = 100
cml

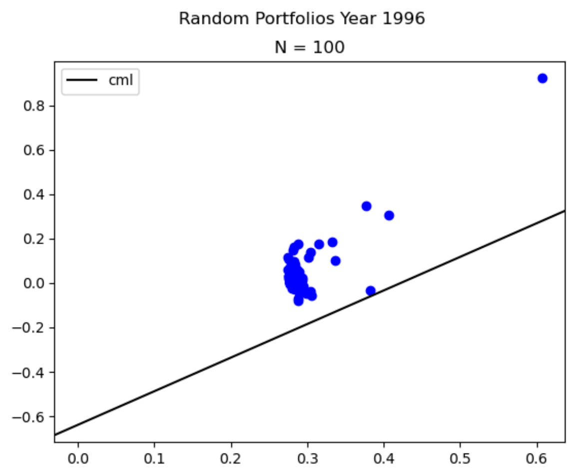

Random Portfolios Year 1990
N = 100
cml

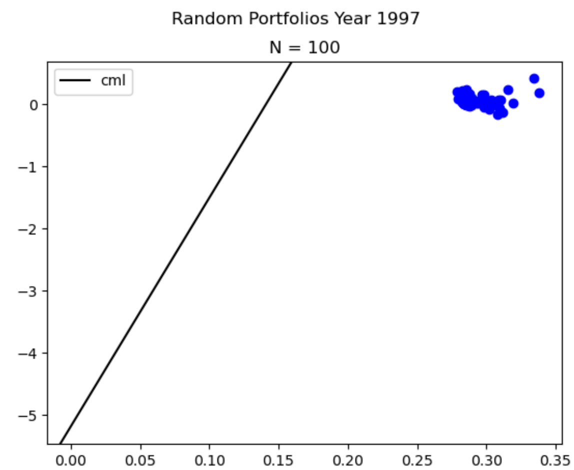

Random Portfolios Year 1991
N = 100
cml

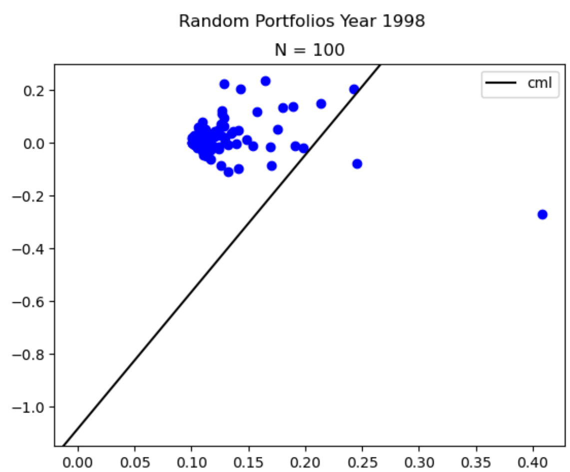

Random Portfolios Year 1993
N = 100
cml

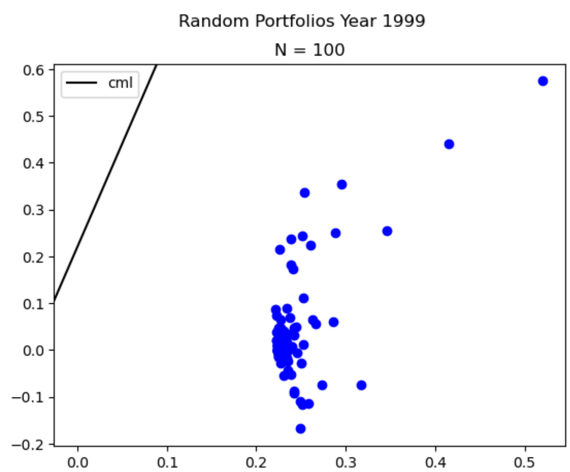

Random Portfolios Year 1994
N = 100
cml

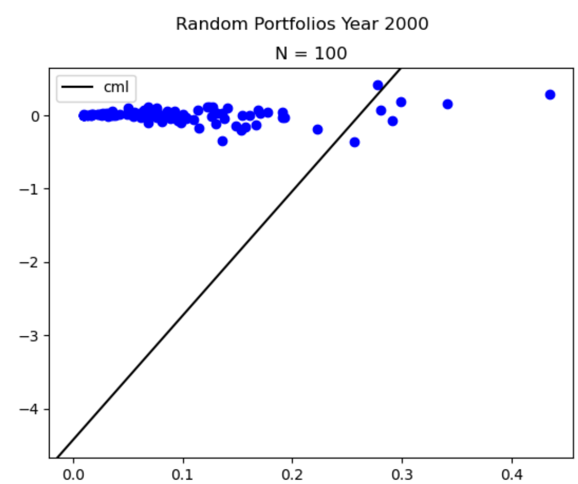

Random Portfolios Year 1995
N = 100
cml

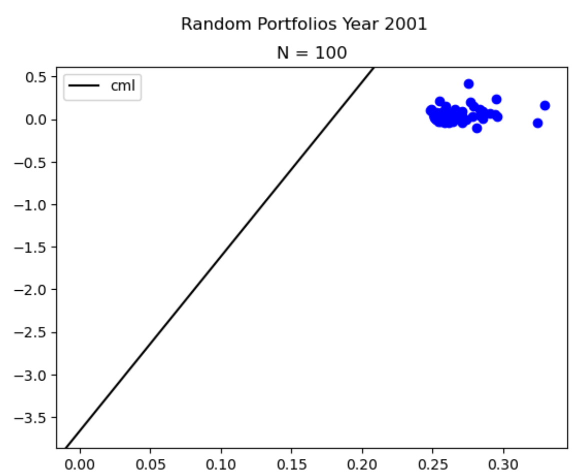

Random Portfolios Year 1996
N = 100
cml

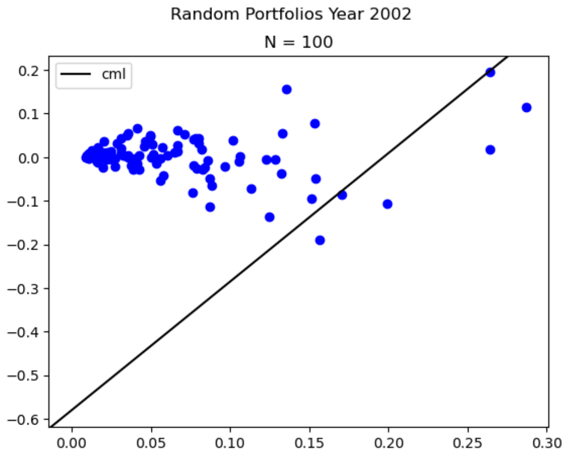

Random Portfolios Year 1997
N = 100
cml

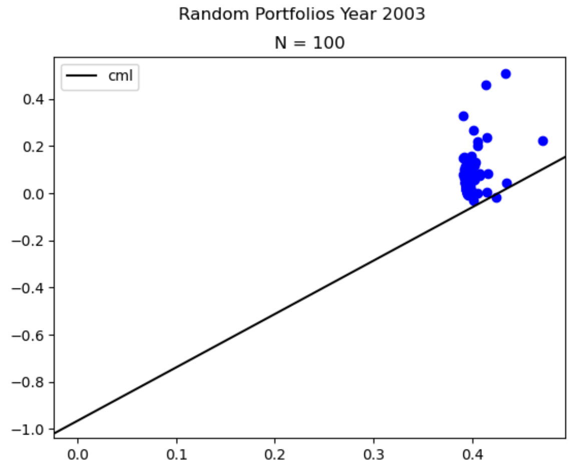

Random Portfolios Year 1998
N = 100
cml

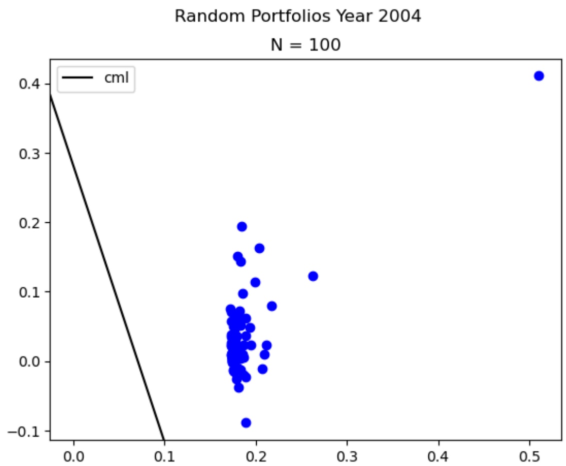

Random Portfolios Year 1999
N = 100
cml

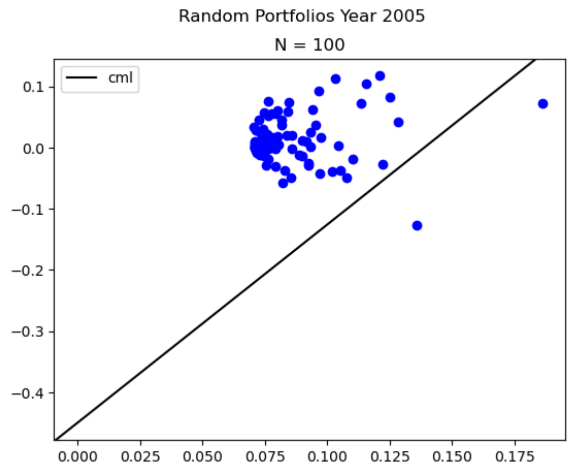

Random Portfolios Year 2000
N = 100
cml

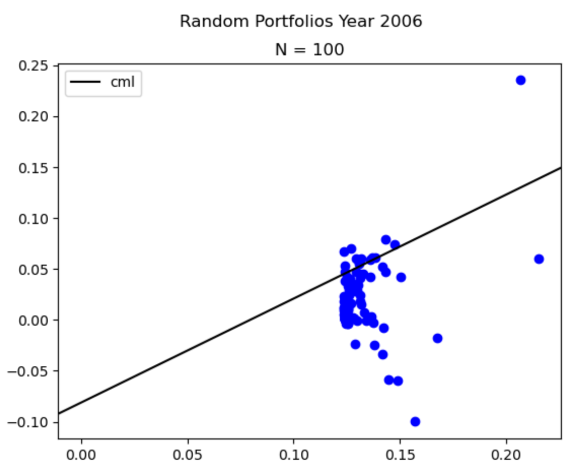

Random Portfolios Year 2001
N = 100
cml

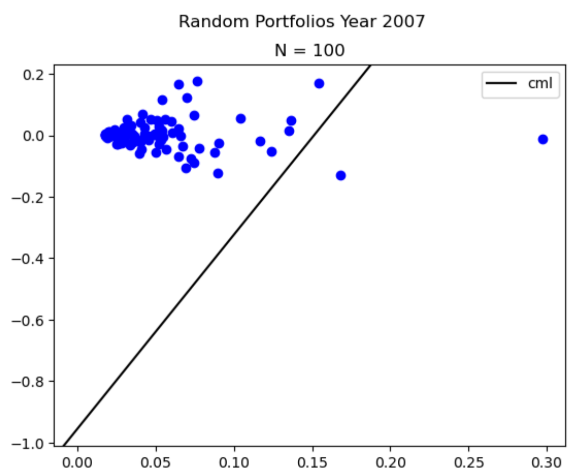

Random Portfolios Year 2002
N = 100
cml

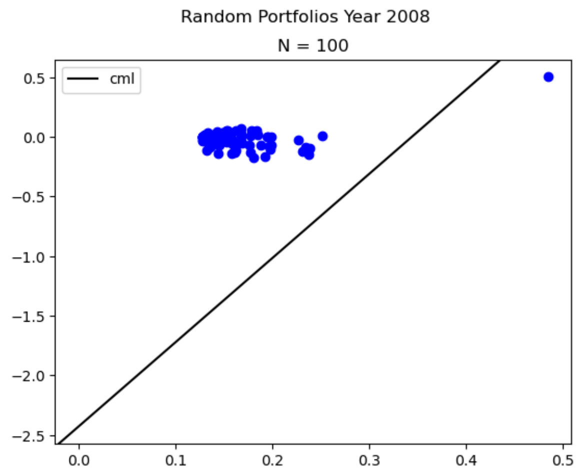

Random Portfolios Year 2003
N = 100
cml

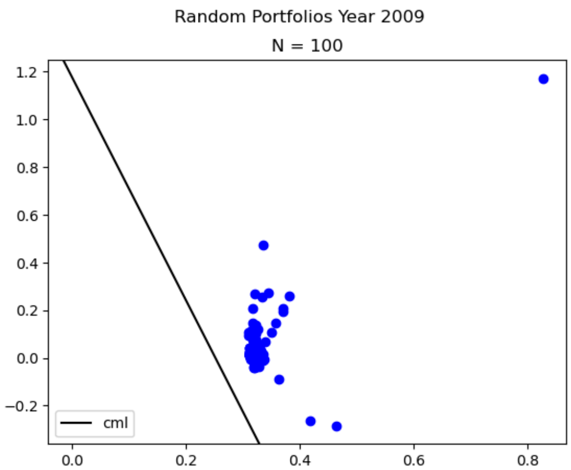

Random Portfolios Year 2004
N = 100
cml

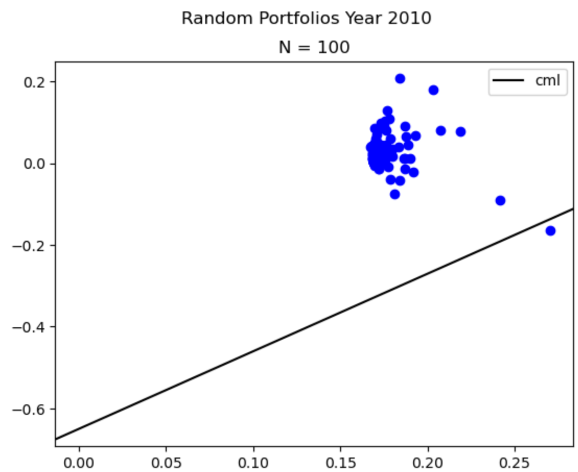

Random Portfolios Year 2005
N = 100
cml

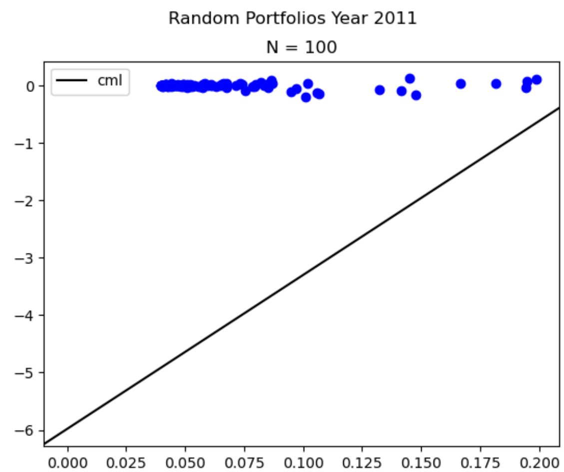

Random Portfolios Year 2006
N = 100
cml

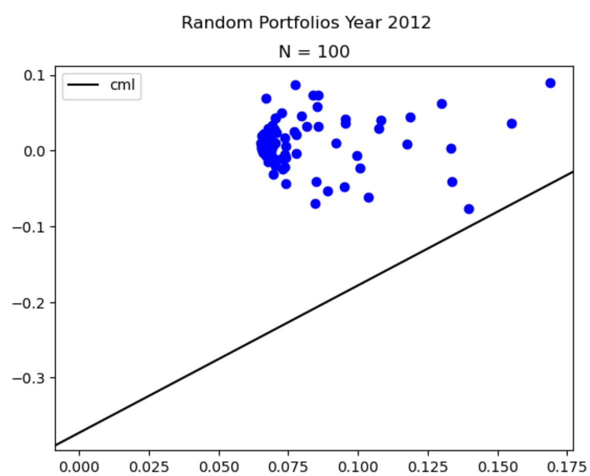

Random Portfolios Year 2007
N = 100
cml

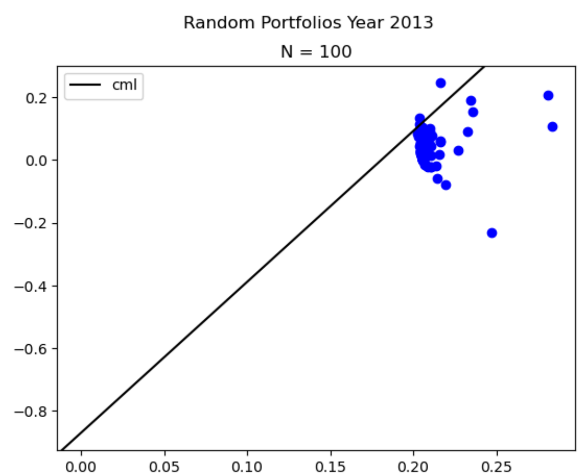

Random Portfolios Year 2008
N = 100
cml

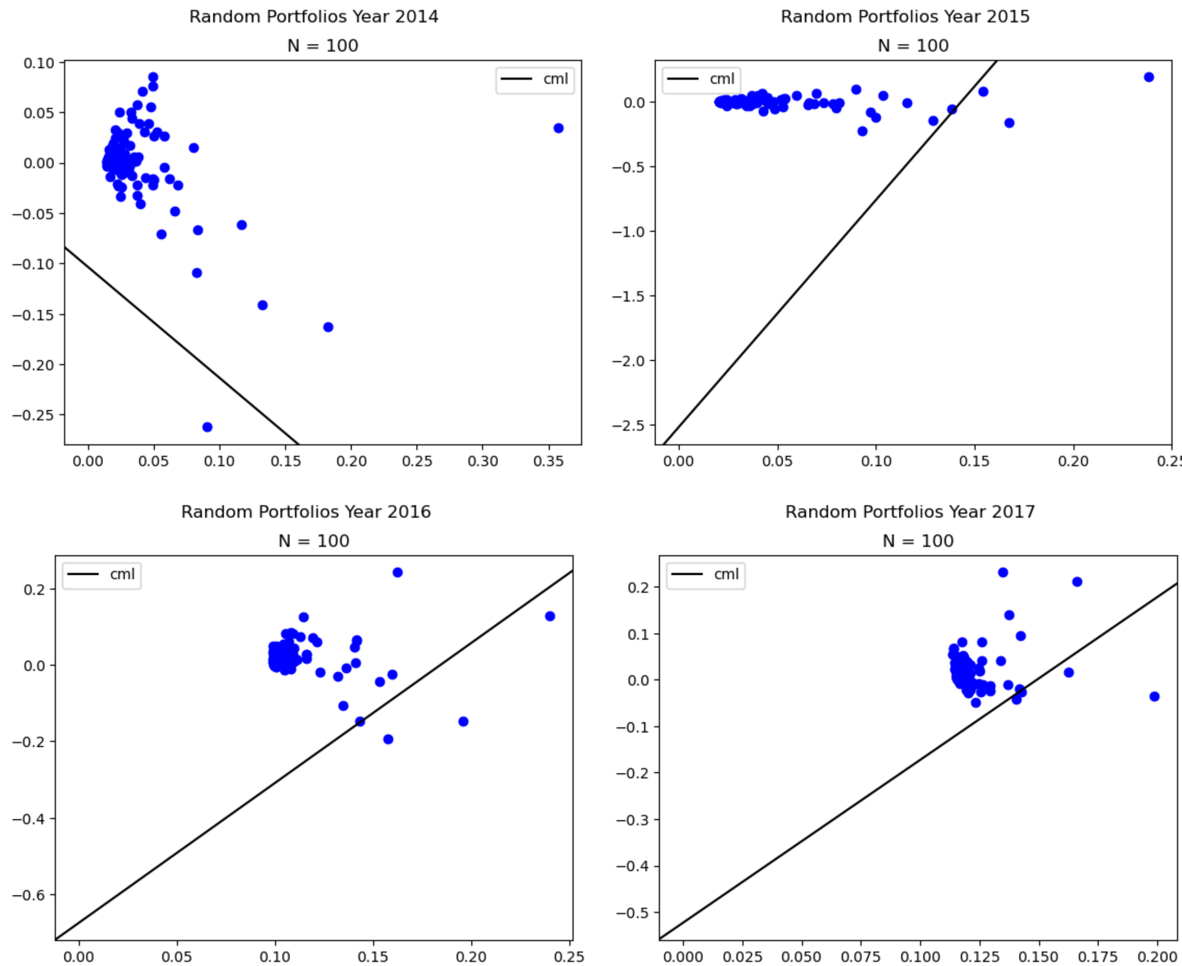

Random Portfolios Year 2009
N = 100
cml

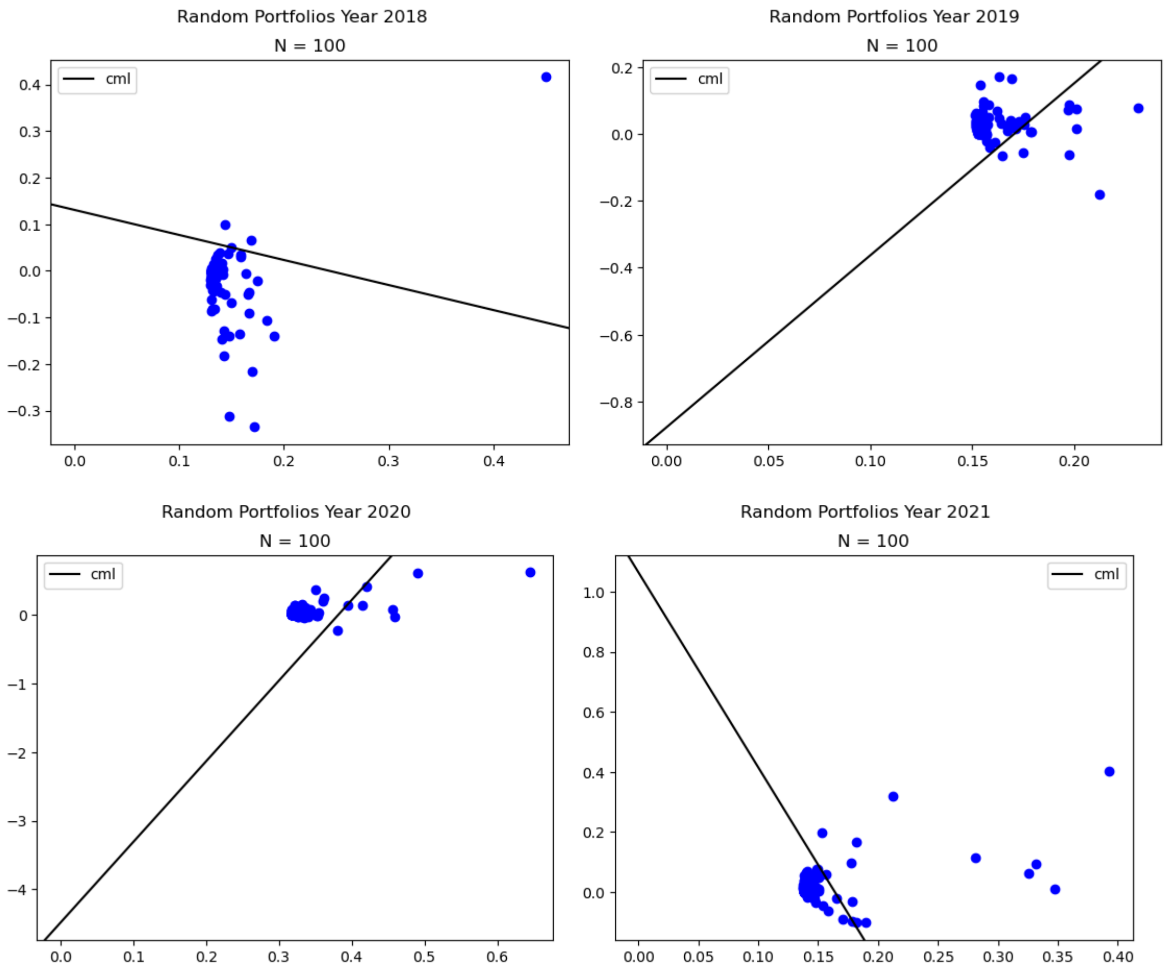

Random Portfolios Year 2010
N = 100
cml


Random Portfolios Year 2011
N = 100
cml


Random Portfolios Year 2012
N = 100
cml


Random Portfolios Year 2013
N = 100
cml


Random Portfolios Year 2014
N = 100
cml
Random Portfolios Year 2015
N = 100
cml
Random Portfolios Year 2016
N = 100
cml
Random Portfolios Year 2017
N = 100
cml


Random Portfolios Year 2018
N = 100
cml
Random Portfolios Year 2019
N = 100
cml
Random Portfolios Year 2020
N = 100
cml
Random Portfolios Year 2021
N = 100
cml

Appendix A: Python Code

```
# %%
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.lines as mlines
import matplotlib.transforms as mtransforms
%matplotlib inline
from pandas_datareader import data as pdr
from datetime import datetime,date
import yfinance as yfin
import requests
import scipy.optimize as sco
import scipy.interpolate as sci
import random

# %%
data = pd.read_csv('XXXX.csv', index_col=0, parse_dates =[2]).dropna()

# %%
data

# %%
def year_trans(data):
    data =
data.groupby([data.DlyCalDt.dt.year,'Ticker'])['IssuerNm'].count().unstack
(level = 'DlyCalDt')
    return(data)

def mode_trans(data):
    for i in data.columns:
        data[i] = data[i].where(data[i] == data[i].mode()[0],float('NaN'))
    return(data)

def weight_gen(port_num):
    p_weights = []
    weights = np.random.random(size = port_num)
    weights = weights / np.sum(weights)
    p_weights.append(weights)
```

```
    return p_weights[0].tolist()

def tday_filter (dataog,datafilter,year):
    mask1 = dataog['DlyCalDt'].dt.year == year
    datanew = dataog[mask1]
    mask2 = datanew.index.isin(datafilter[year].dropna().index)
    datanew2 = datanew[mask2]
    return datanew2

def random_select(universe,portsize):
    random_port = random.sample(list(universe.index.unique()),portsize)
    return random_port

#returns portfolio yearly return
def return_gen(data,assets,weights):
    m = data.index.isin(assets)
    data = data[m]
    ret = 0
    for i,j in zip(data.index.unique(),weights):
        ret = ret + data.loc[i,'DlyRet'].sum()*j

    return ret

#returns individual daily returns for items in portfolio
def return_gen2(data,assets,weights):
    empty = pd.DataFrame()
    m = data.index.isin(assets)
    data = data[m]
    ret = 0
    data2 = data.set_index(data['DlyCalDt'].dt.date,append = True)
    for i,j in
zip(data2.index.get_level_values('Ticker').unique(),weights):
        test = data2.loc[i,'DlyRet'] * j
        empty[i]=test
    return empty

def daily_return(data):
    row_sum = data.sum(axis=1)
    return row_sum
```

```
def vol(daily_returns,annual_return):
    sd = ((((daily_returns -
(annual_return/len(daily_returns)))**2).sum())*(len(daily_returns)/(len(da
ily_returns)-1)))**0.5
    return sd

# %%
data1=year_trans(data)
data1t = mode_trans(data1)
data1

# %%
N = 100
yr = XXXX
x = tday_filter(data,data1t,yr)
weight = weight_gen(N)
z = random_select(x,N)
ret = return_gen(x,z,weight)
ret

# %%
ret2 = return_gen2(x,z,weight)
ret2

# %%
dret = daily_return(ret2)
dret

# %%
sd = vol(dret,ret)
sd

# %%
#calculate fama french risk free rate
def fama_french(portfolio):
    high_low = sorted(portfolio)
    num_trades = len(portfolio)
    r_f = (((high_low[0] + high_low[num_trades - 2])/2)*num_trades)
    return r_f
```

```
print(fama_french(dret))

#cml line assuming volatility is x axis and return is y
def cml(annual_return, interest, vol):
    cml_int = interest
    cml_slope = ((annual_return + interest)/vol)
    return cml_int, cml_slope

# %%
#generate the random distribution
x = []
y = []
ret2_df = pd.DataFrame(ret2)
for column in ret2_df:
    stock = ret2_df[column]
    y.append(sum(stock)*(N/10))
    x.append(vol(stock, ret)*(N/10))


# %%
#create the plot
cml_line = cml(ret, fama_french(dret), sd)
m = cml_line[1]
b = cml_line[0]
fig, ax = plt.subplots()
ax.scatter(x, y, c='blue')
ax.axline((0, b), slope=m, color='black', label='cml')
ax.legend()
plt.suptitle("Random Portfolios Year XXXX")
plt.title("N = 100")
plt.show()
```

Work Cited

[1] Wojciechowski, William C., and James R. Thompson. 2006. "Market Truths: Theory versus Empirical Simulations." *Journal of Statistical Computation and Simulation* 76 (5): 385–95. https://doi.org/10.1080/10629360500107709.