Project 3

FINA 6333 – Spring 2024

Richard Herron

# Assignment

## 5 Portfolios

* What are the pros and cons of each portfolio?
* Which one is best portfolio for the average risk averse investor who wants the mean-variance efficient portfolio?

# The 5 Portfolios

1. All-equity portfolio
2. Traditional 60/40 portfolio, which became common in 1926 and consists of the following:
   1. 60% equity
   2. 40% intermediate Treasury bonds
3. Harry Browne’s *permanent portfolio*, which he proposed in about 1980 and consists of the following:
   1. 25% equity
   2. 25% long-term Treasury bonds
   3. 25% Treasury bills (i.e., cash)
   4. 25% gold
4. Ray Dalio’s *all seasons portfolio*, which he proposed in about 2014 and consists of the following:
   1. 30% equity
   2. 40% long-term Treasury bonds
   3. 15% intermediate Treasury bonds
   4. 7.5% commodities
   5. 7.5% gold
5. Another portfolio that you build from the asset class returns in histretSP.csv

# Our Calculations

To determine the best portfolio, we…

1. Found the assoicated risk-free rate from the T-Bill
2. Established our portfolio weights with the corresponding portfolios
3. Calculated the average retruns, standard deviation, sharpe ratio, and utillity for the firdt 4 portfolios
4. Maximized the sharpe ratio to optimize our original portfolio

import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
import pandas\_datareader as pdr  
import scipy.optimize as sco  
import yfinance as yf  
import matplotlib.ticker as ticker  
import seaborn as sns

%precision 4  
pd.options.display.float\_format = '{:.4f}'.format  
%config InlineBackend.figure\_format = 'retina'

# Establishing Dataset for Asset Returns  
data = pd.read\_csv('histretSP.csv').set\_index('Year')  
data;

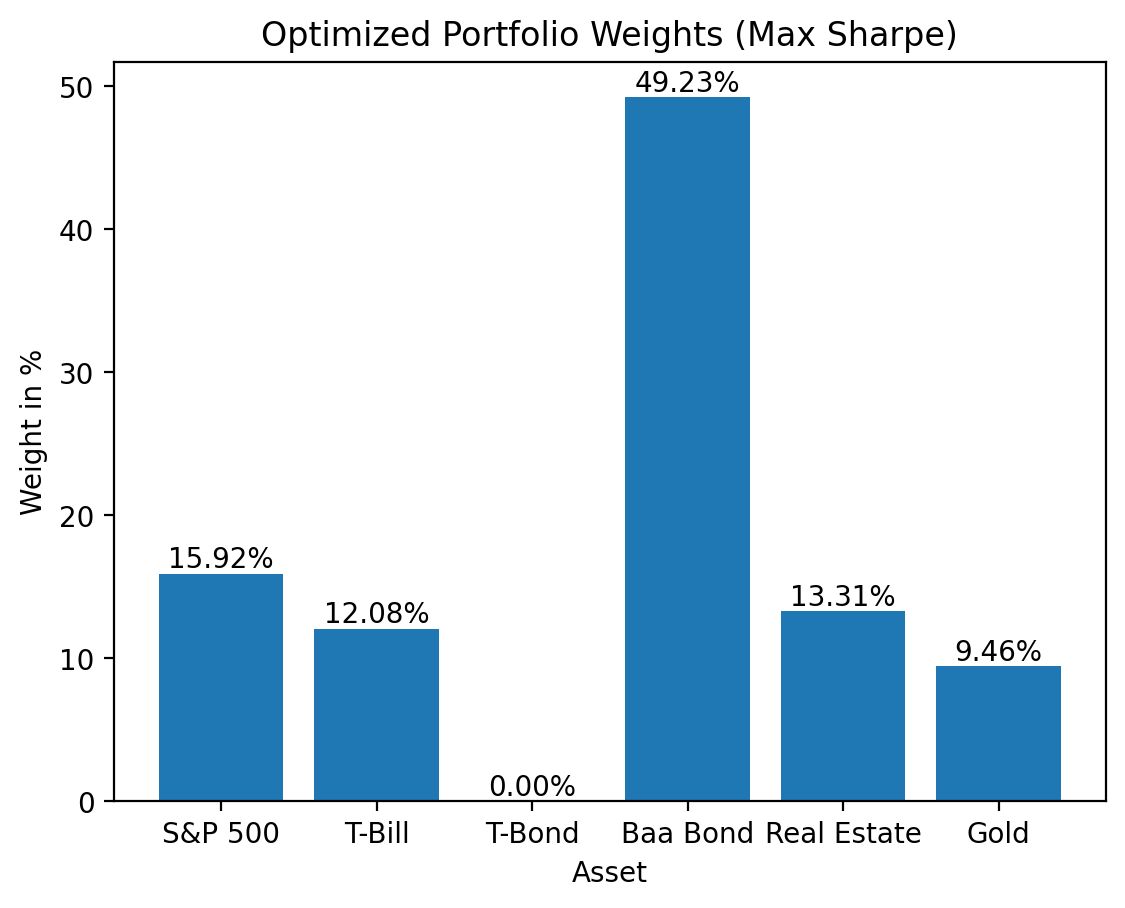
### Establishing Portfolio Weights ###  
# weight value order: [S&P 500, T-Bill, T-Bond, Baa Bond, Real Estate, Gold]  
  
# Portfolio 1: All-Equity Portfolio   
 # 100% equity  
portfolios = pd.DataFrame(  
 [[1] + list(np.zeros(data.shape[1]-1))],   
 columns = list(data.columns.values), index=['allequity']).T  
  
# Portfolio 2: Traditional 60/40 portfolio, became commonplace in 1926  
 # 60% equity  
 # 40% intermediate Treasury bonds  
portfolios['traditional'] = [0.6, 0, 0.4] + list(np.zeros(data.shape[1]-3))  
  
# Portfolio 3: Harry Browne’s permanent portfolio, proposed in ~1980  
 # 25% equity  
 # 25% long-term Treasury bonds  
 # 25% Treasury bills (i.e., cash)  
 # 25% gold  
portfolios['permanent'] = [0.25]\*3 + [0, 0, 0.25]  
  
# Portfolio 4: Ray Dalio’s all seasons portfolio, proposed in ~2014   
 # 30% equity  
 # 40% long-term Treasury bonds  
 # 15% intermediate Treasury bonds  
 # 7.5% commodities  
 # 7.5% gold  
portfolios['allseasons'] = [0.3, 0, 0.55, 0, 0, 0.075\*2]  
  
# Portfolio 5: Our Original Optimized Portfolio   
portfolios['original'] = [1/data.shape[1]]\*data.shape[1]  
portfolios = portfolios.T  
initial = portfolios.copy()  
portfolios

|  | S&P 500 | T-Bill | T-Bond | Baa Bond | Real Estate | Gold |
| --- | --- | --- | --- | --- | --- | --- |
| allequity | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| traditional | 0.6000 | 0.0000 | 0.4000 | 0.0000 | 0.0000 | 0.0000 |
| permanent | 0.2500 | 0.2500 | 0.2500 | 0.0000 | 0.0000 | 0.2500 |
| allseasons | 0.3000 | 0.0000 | 0.5500 | 0.0000 | 0.0000 | 0.1500 |
| original | 0.1667 | 0.1667 | 0.1667 | 0.1667 | 0.1667 | 0.1667 |

def calc\_excessret(x, r, ref, ppy):  
 pfrets = r.dot(x)   
 excessrets = pfrets.sub(ref.loc[str(pfrets.index[0]):pfrets.index[-1]].values)   
 return excessrets  
  
def calc\_sharpe(x, r, ref, ppy):  
 pfrets = r.dot(x).dropna().sub(ref, axis=0)  
 return pfrets.mean() / pfrets.std()  
  
def calcneg\_sharpe(x, r, ref, ppy):   
 return -1 \* calc\_sharpe(x, r, ref, ppy)  
  
def calc\_utility(x, r, ref, ppy, gamma): # annualized values  
 excessret = calc\_excessret(x, r, ref, ppy)  
 annreturn = excessret.mean() \* ppy  
 annstdev = excessret.std() \* np.sqrt(ppy)  
 annvar = annstdev \*\* 2  
 return np.array(annreturn - (0.5 \* annvar \* gamma))  
  
def performance(x, r, ref, ppy, gamma):  
   
 pflabel = list(x.index.values)  
 pf = x.copy()  
   
 pf['avgreturns'] = np.nan  
 pf['stdev'] = np.nan  
 pf['utility'] = np.nan  
 pf['sharpe'] = np.nan  
 returns = pd.DataFrame()  
   
 for i in range(len(pflabel)):  
 pf.loc[pflabel[i], 'avgreturns'] = calc\_excessret(x=x.loc[str(pflabel[i])].values, r=r, ref=ref, ppy=ppy).mean()  
 pf.loc[pflabel[i], 'stdev'] = calc\_excessret(x=x.loc[str(pflabel[i])].values, r=r, ref=ref, ppy=ppy).std()  
 pf.loc[pflabel[i], 'utility'] = calc\_utility(x=x.loc[str(pflabel[i])].values, r=r, ref=ref, ppy=ppy, gamma=gamma)  
 pf.loc[pflabel[i], 'sharpe'] = calc\_sharpe(x=x.loc[str(pflabel[i])].values, r=r, ref=ref, ppy=ppy)  
   
 returns[pflabel[i]] = calc\_excessret(x=x.loc[str(pflabel[i])].values, r=r, ref=ref, ppy=ppy)  
  
 returns = returns.set\_index(r.reset\_index()['Year'])   
   
 return pf, returns  
  
# maximizing sharpe:  
def max\_sharpe(x, r, ref, ppy):  
 label = 'maxsharpe'  
 pf = pd.DataFrame([], index=[label], columns=list(x.columns.values))  
 xval = np.array(x.loc['original'].values)  
   
 optim = sco.minimize(fun=calcneg\_sharpe,   
 x0=xval,   
 args=(r, ref, ppy),  
 bounds=((0,1) for c in r.columns),   
 constraints = ({'type':'eq', 'fun': lambda x: x.sum() - 1})  
 )  
   
 pf.loc[label] = optim['x']  
   
 return pf

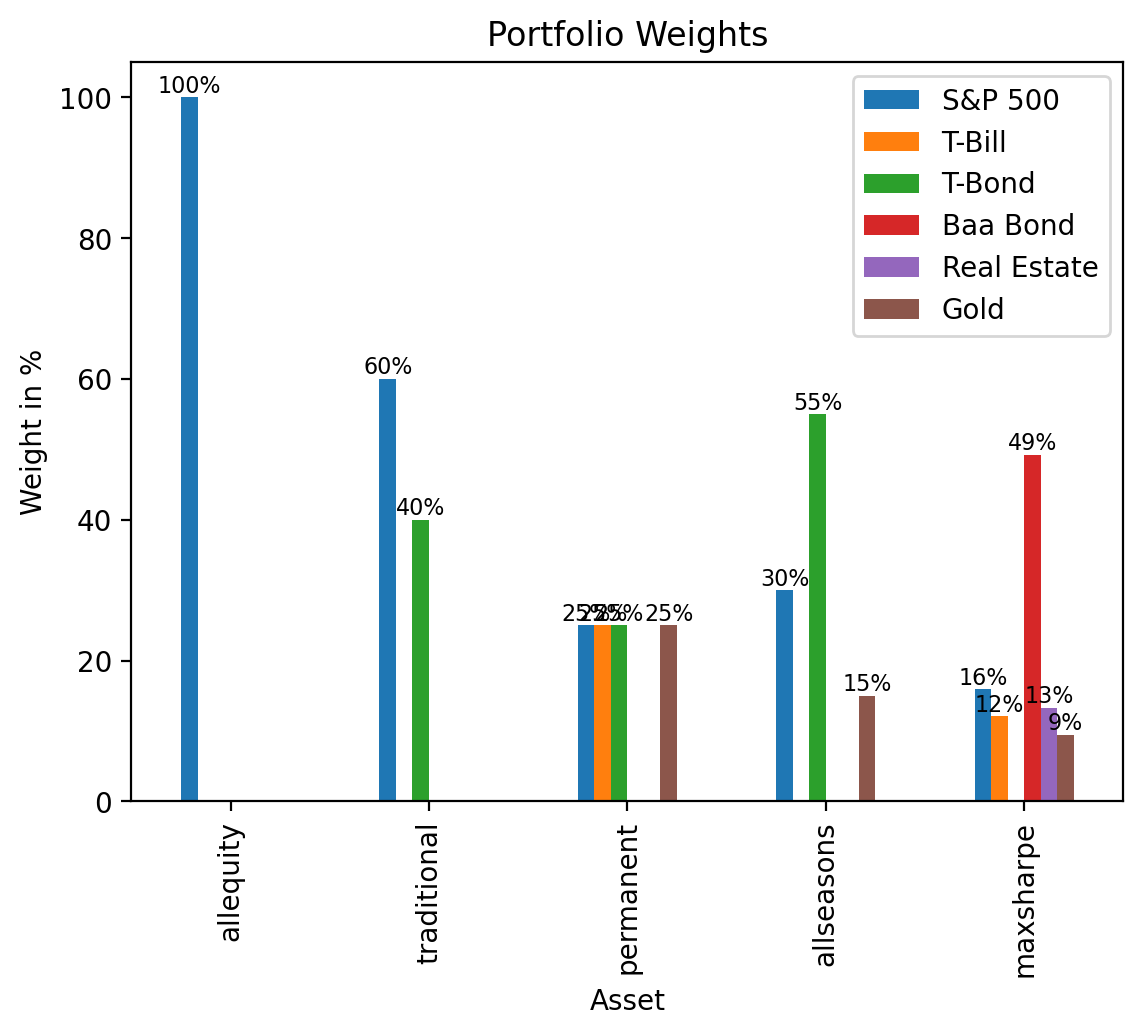
# gives initial portfolio performance based on initial weights  
initialpfs, initialexr = performance(x=portfolios, r=data, ref=data['T-Bill'], ppy=1, gamma=2)  
initialpfs;

# Maximizing Sharpe Ratio for Entire Period  
result = max\_sharpe(x=portfolios,  
 r=data,   
 ref=data['T-Bill'],  
 ppy=1  
 )  
  
optimized = portfolios.loc[portfolios.index.tolist()[:-1]].copy()  
optimized = pd.concat([optimized, result])  
optimized;  
  
  
# Data Visualization for Weights   
def plot\_portfolioweights(df):  
 visualwts = df.replace(0, np.nan) \* 100  
 visualwts.plot.bar()  
 [plt.gca().annotate(f'{bar.get\_height():.0f}%', (bar.get\_x() + bar.get\_width() / 2., bar.get\_height()), ha='center', va='bottom', fontsize=8) for bar in plt.gca().patches if bar.get\_height() > 0]  
   
 plt.xlabel('Asset')  
 plt.ylabel('Weight in %')  
 plt.title('Portfolio Weights')  
 plt.show()  
   
visualwts = optimized.loc['maxsharpe':,:] \* 100  
plt.bar(visualwts.columns, visualwts.iloc[0].tolist())  
[plt.gca().annotate(f'{bar.get\_height():.2f}%', (bar.get\_x() + bar.get\_width() / 2., bar.get\_height()), ha='center', va='bottom', fontsize=10) for bar in plt.gca().patches]  
plt.xlabel('Asset')  
plt.ylabel('Weight in %')  
plt.title('Optimized Portfolio Weights (Max Sharpe)')  
plt.show()  
  
# Storing DataFrame "newpfs" for Portfolio Weights & Performance Measures and "expfreturns" for Excess Portfolio Returns  
newpfs, expfreturns = performance(x=optimized, r=data, ref=data['T-Bill'], ppy=1, gamma=2)  
expfreturns = expfreturns.set\_index(pd.DatetimeIndex(pd.to\_datetime(expfreturns.index, format='%Y')))  
newpfs



|  | S&P 500 | T-Bill | T-Bond | Baa Bond | Real Estate | Gold | avgreturns | stdev | utility | sharpe |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| allequity | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0832 | 0.1987 | 0.0437 | 0.4187 |
| traditional | 0.6000 | 0.0000 | 0.4000 | 0.0000 | 0.0000 | 0.0000 | 0.0560 | 0.1247 | 0.0405 | 0.4492 |
| permanent | 0.2500 | 0.2500 | 0.2500 | 0.0000 | 0.0000 | 0.2500 | 0.0326 | 0.0717 | 0.0275 | 0.4554 |
| allseasons | 0.3000 | 0.0000 | 0.5500 | 0.0000 | 0.0000 | 0.1500 | 0.0381 | 0.0786 | 0.0320 | 0.4855 |
| maxsharpe | 0.1592 | 0.1208 | 0.0000 | 0.4923 | 0.1331 | 0.0946 | 0.0355 | 0.0649 | 0.0313 | 0.5471 |

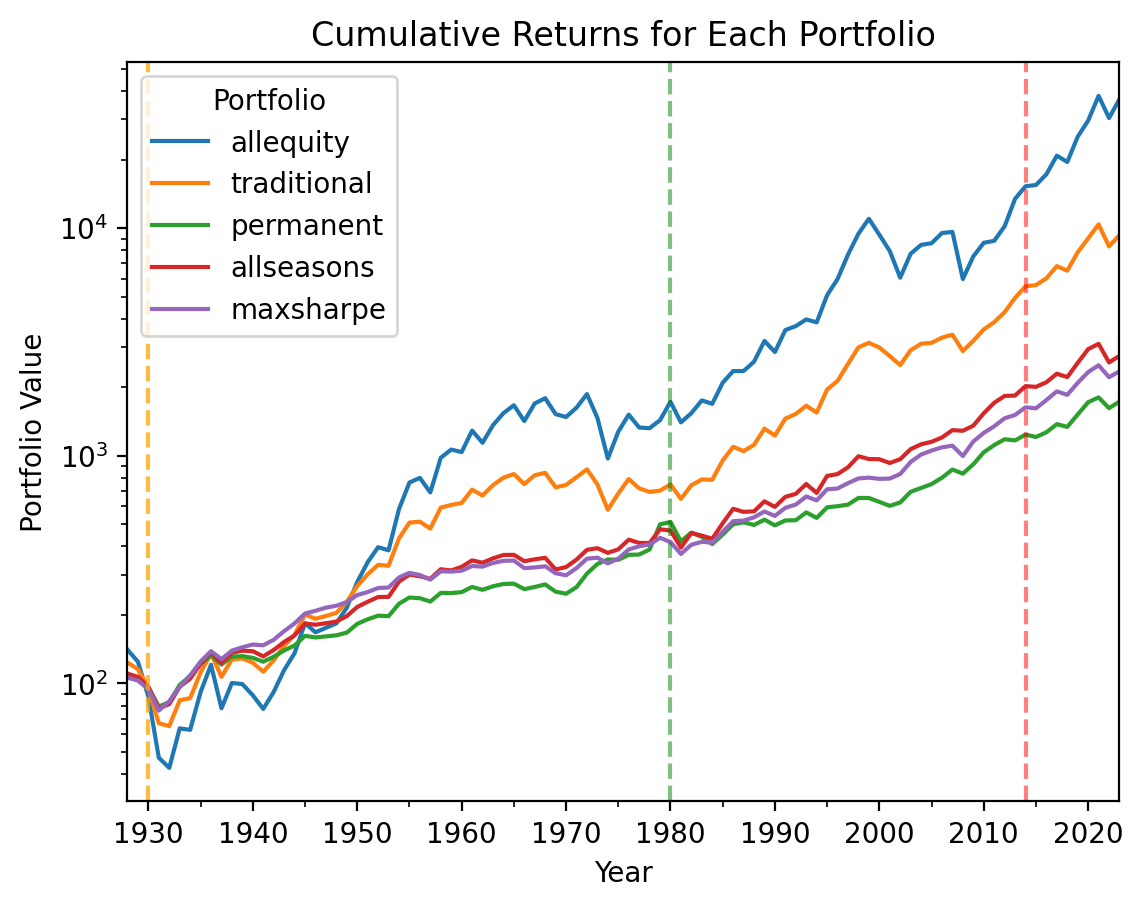
visualwts = optimized.replace(0, np.nan) \* 100  
visualwts.plot.bar()  
[plt.gca().annotate(f'{bar.get\_height():.0f}%', (bar.get\_x() + bar.get\_width() / 2., bar.get\_height()), ha='center', va='bottom', fontsize=8) for bar in plt.gca().patches if bar.get\_height() > 0]  
   
plt.xlabel('Asset')  
plt.ylabel('Weight in %')  
plt.title('Portfolio Weights')  
plt.show()



# Comparing Portfolios:

## Cumulative Returns

# Cumulative Returns  
def plot\_cumrets(returndata):  
 returns = returndata.dropna().rename\_axis(columns='Portfolio')  
 returns\_cum = returns.add(1).cumprod()  
 returns\_cum.mul(100).plot()  
 plt.gca().get\_yaxis().set\_major\_formatter(ticker.StrMethodFormatter('${x:,.0f}'))  
 plt.yscale('log')   
 plt.axvline(x='1930', linestyle='--', color = 'orange', alpha=0.75)  
 plt.axvline(x='1980', linestyle='--', color = 'green', alpha=0.5)  
 plt.axvline(x='2014', linestyle='--', color = 'red', alpha=0.5)  
 plt.ylabel('Portfolio Value')  
 plt.title('Cumulative Returns for Each Portfolio')  
 plt.show()  
plot\_cumrets(expfreturns)



This graph shows us the cumulative product returns for the entire period from 1928 to 2023. - Each portfolio is labeled respectively by their individual color. - The veritical lines also represents the when the particular portfolio was established to examine how the portfolio did following the time it was proposed.

Is this a good measure to evaluate the portfolios? - Cumulative product returns allow us to visualize how the total returns of each portfolio does in a given time. However, returns are unpredictable and should not be used as the primary deciding factor as historical analysis does not show future returns.

## Beta and Drawdown

# Beta Calculation  
def calc\_betas(returns, asset):  
 excess = returns.sub(asset['T-Bill'].values, axis=0)  
 excess['market'] = asset['S&P 500'].sub(asset['T-Bill'].values, axis=0).values  
 mktvar = excess.cov().iloc[-1,-1]  
 covassets = excess.cov().iloc[-1,:-1]  
 betas = pd.DataFrame(covassets.div(mktvar)).T  
 return betas  
  
Browne\_before = calc\_betas(returns= expfreturns.loc['1970':'1980'], asset= data.loc['1970':'1980'])  
Browne\_after = calc\_betas(returns= expfreturns.loc['1980':'1990'], asset= data.loc['1980':'1990'])  
Dalio\_before = calc\_betas(returns= expfreturns.loc['2004':'2014'], asset= data.loc['2004':'2014'])  
Dalio\_after = calc\_betas(returns= expfreturns.loc['2014':'2024'], asset= data.loc['2014':'2024'])  
print('\n Beta for Harry Browne’s permanent portfolio, the decade before and the decade after the portfolio was proposed:')  
display(Browne\_before, Browne\_after)  
  
print('\n Beta for Ray Dalio’s all seasons portfolio, the decade before and the decade after the portfolio was proposed:')  
display(Dalio\_before, Dalio\_after)

Beta for Harry Browne’s permanent portfolio, the decade before and the decade after the portfolio was proposed:  
  
 Beta for Ray Dalio’s all seasons portfolio, the decade before and the decade after the portfolio was proposed:

|  | allequity | traditional | permanent | allseasons | maxsharpe |
| --- | --- | --- | --- | --- | --- |
| market | 1.0076 | 0.6247 | 0.0228 | 0.1838 | 0.1563 |

|  | allequity | traditional | permanent | allseasons | maxsharpe |
| --- | --- | --- | --- | --- | --- |
| market | 1.0591 | 0.7812 | 0.5515 | 0.6266 | 0.4317 |

|  | allequity | traditional | permanent | allseasons | maxsharpe |
| --- | --- | --- | --- | --- | --- |
| market | 1.0285 | 0.4714 | 0.1478 | 0.0930 | 0.3115 |

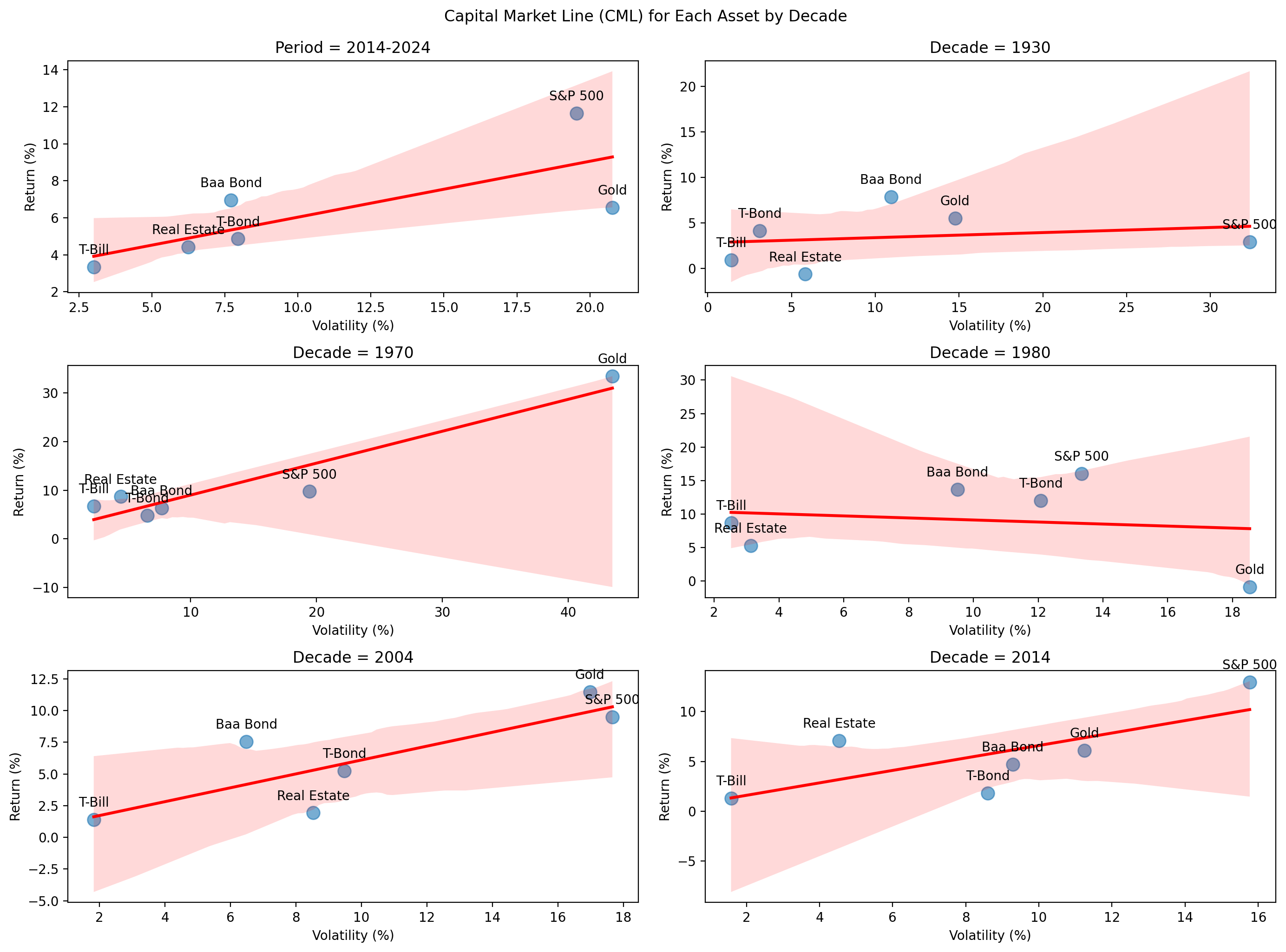
|  | allequity | traditional | permanent | allseasons | maxsharpe |
| --- | --- | --- | --- | --- | --- |
| market | 1.0023 | 0.7462 | 0.4298 | 0.5527 | 0.4499 |

# Calculate Drawdown  
def calculate\_drawdown(returns):  
 cumulative\_returns = (1 + returns).cumprod()  
 peak = cumulative\_returns.cummax()  
 drawdown = (cumulative\_returns - peak) / peak  
 return drawdown.min()  
  
drawdowns = {}  
for column in expfreturns.columns:  
 drawdowns[column] = round(calculate\_drawdown(expfreturns[column]), 4)  
  
# Display the results  
print("Drawdowns:", drawdowns)

Drawdowns: {'allequity': -0.6972, 'traditional': -0.4738, 'permanent': -0.2729, 'allseasons': -0.2963, 'maxsharpe': -0.2828}

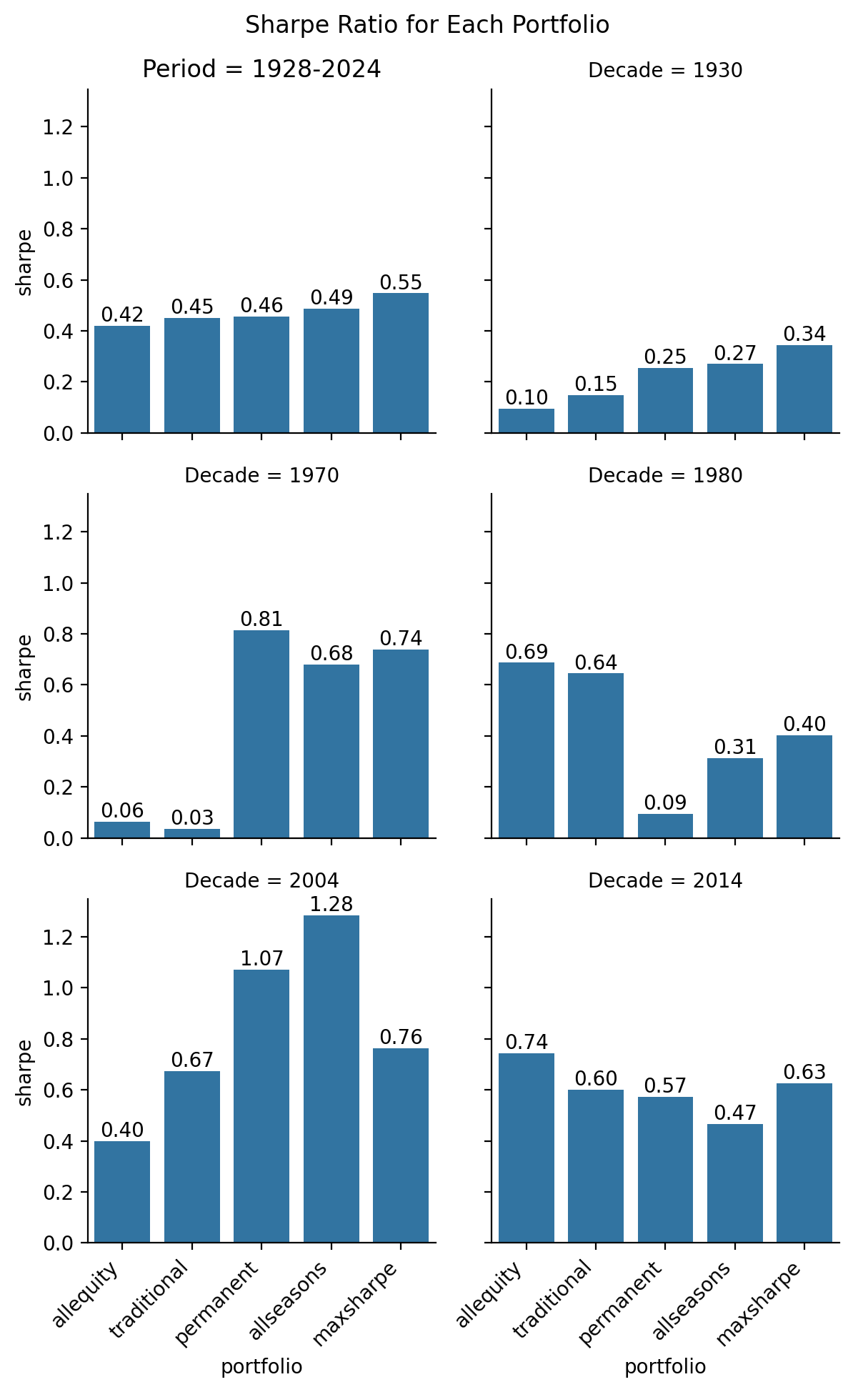
## Capital Market Line

def plot\_cml(periods, data):  
 # Define the time periods  
   
 # Create a figure and a 2x2 grid of subplots  
 fig, axs = plt.subplots(3, 2, figsize=(14,10))   
   
 axs = axs.flatten()   
   
 # Loop through each period and plot  
 for i, (start, end) in enumerate(periods):  
 # Filter data for the specific period  
 df\_period = data.loc[start:end]  
   
 # Calculate mean and standard deviation in percentage form  
 df\_stats = (df\_period \* 100).agg(['mean', 'std']).transpose()  
   
 # Use seaborn to plot the standard deviation and mean return relationship  
 sns.regplot(  
 ax=axs[i],  
 data=df\_stats,  
 x='std',  
 y='mean',  
 scatter\_kws={'s': 100, 'alpha': 0.6},   
 line\_kws={'color': 'red'}   
 )  
 # Add labels for each point  
 for t, (x, y) in df\_stats[['std', 'mean']].iterrows():  
 axs[i].annotate(text=t, xy=(x, y), textcoords="offset points", xytext=(0,10), ha='center')  
   
 # Set axis labels and chart title for each subplot  
 axs[i].set\_xlabel('Volatility (%)')  
 axs[i].set\_ylabel('Return (%)')  
 axs[i].set\_title(f'Decade = {start}')  
   
 axs[0].set\_title(f'Period = {start}-{end}')  
   
 # Adjust layout to prevent overlap  
 plt.tight\_layout()  
   
 # Display the figure with all subplots  
 plt.suptitle("Capital Market Line (CML) for Each Asset by Decade", y=1.02)  
 plt.show()  
  
periods = [(1928, 2024), (1930, 1940), (1970, 1980), (1980, 1990), (2004, 2014), (2014, 2024)]  
plot\_cml(periods, data)



## Sharpe Ratio

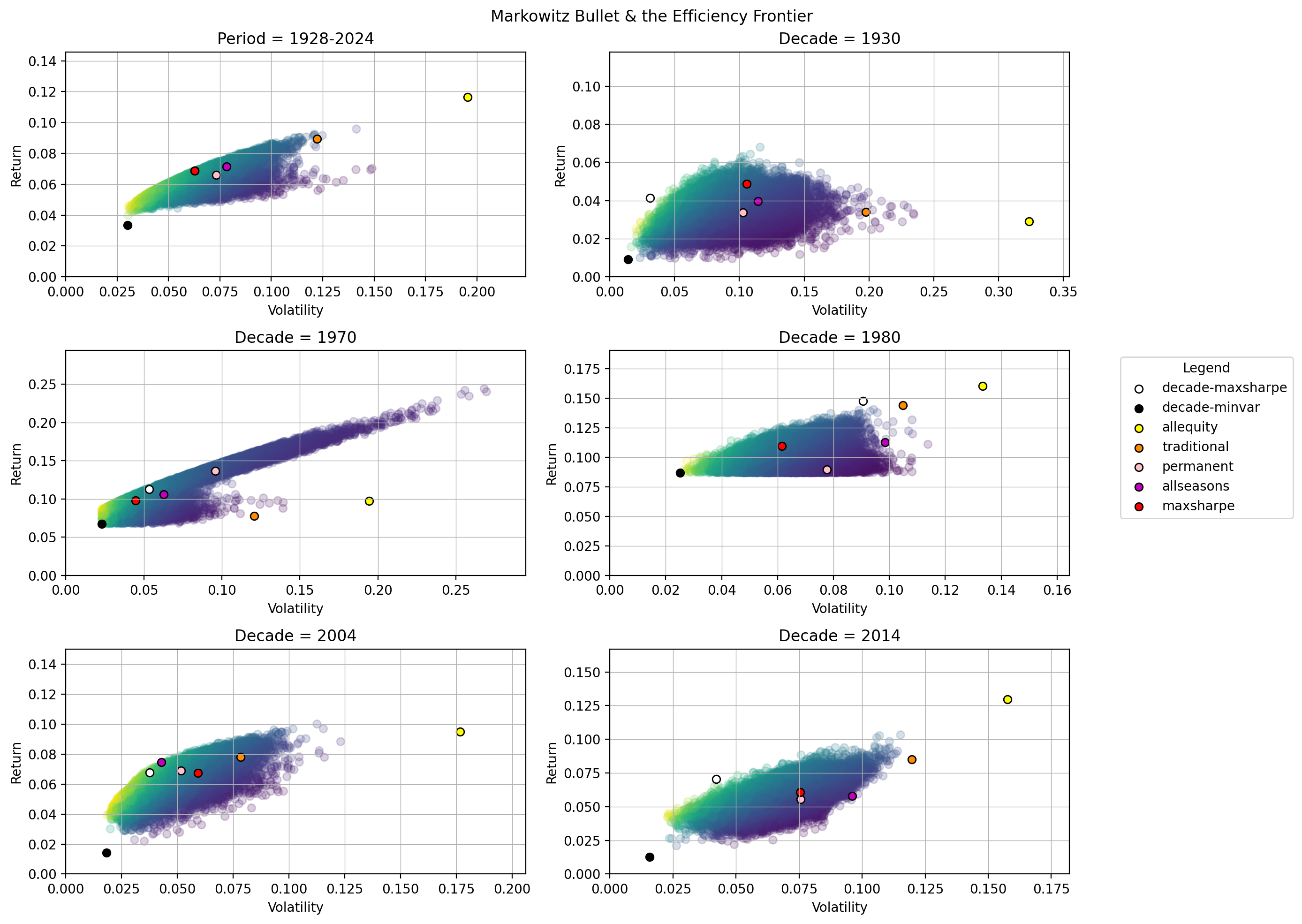
def get\_decade(r, d):  
 return r.loc[str(d): str(d+9)].copy()  
  
def get\_sharpe(r):  
 return r.mean() / r.std()  
  
def plot\_sharperatios(decades, returns, pf=True):  
 if pf == True:  
 title = 'Sharpe Ratio for Each Portfolio'  
 label = 'portfolio'  
 else:  
 title = 'Sharpe Ratio for Each Asset'  
 label= 'asset'  
 newdata = returns.copy()  
 seriesdata = pd.concat(  
 objs=[newdata.pipe(get\_decade, d=d).pipe(get\_sharpe) for d in decades],  
 keys=decades,  
 names=['Decade'])  
 visualdata = pd.DataFrame(seriesdata, columns=['sharpe']).reset\_index().rename(columns={'level\_1': label})  
 period = pd.DataFrame({'Decade': 1928, label: newdata.columns.tolist(), 'sharpe':(newdata.mean() / newdata.std()).values.tolist()})  
  
 visualdata = pd.concat([period, visualdata]).reset\_index(drop=True)  
   
 g = (  
 visualdata  
 .pipe(  
 sns.catplot,  
 y='sharpe',  
 x=label,  
 col='Decade',  
 col\_wrap=2,  
 kind='bar',  
 height=3  
 )  
 )  
 # Adding Weight Values to top of each bar   
 [  
 [ax.annotate(f'{bar.get\_height():.2f}', (bar.get\_x() + bar.get\_width() / 2., bar.get\_height()), ha='center', va='bottom') for bar in ax.patches]   
 for ax in g.axes.flatten()  
 ]  
   
 # Formatting  
 g.axes.flatten()[0].set\_title(f'Period = {data.index[0]}-{data.index[-1]+1}')  
 g.set\_xticklabels(newdata.columns.tolist(), rotation=45, ha='right')   
 plt.suptitle(title, y=1.02)  
 plt.show()  
decades = [1930, 1970, 1980, 2004, 2014]  
plot\_sharperatios(decades, expfreturns, True)  
# plot\_sharperatios(decades, data, False)



## Efficient Frontier

# Functions to calculate portfolio returns, volatility, minimum variance portfolio, random & efficient frontier portfolios  
def portfolio\_performance(weights, returns, ref):  
 portfolio\_return = returns.dot(weights).mean()  
 portfolio\_volatility = np.sqrt(np.dot(weights.T, np.dot(returns.cov(), weights)))  
 return portfolio\_return, portfolio\_volatility  
  
def calc\_minvar(x, r, ref):  
 pfrets = r.dot(x).dropna().sub(ref, axis=0)  
 return pfrets.std()  
   
def optim\_minvar(x, r, ref):  
 label = 'minvar'  
 pf = pd.DataFrame([], index=[label], columns=list(x.columns.values))  
 xval = np.array(x.loc['original'].values)  
  
 optim = sco.minimize(fun=calc\_minvar,   
 x0=xval,   
 args=(r, ref),  
 bounds=((0,1) for c in r.columns),   
 constraints = ({'type':'eq', 'fun': lambda x: x.sum() - 1})  
 )  
   
 pf.loc[label] = optim['x']  
 return pf  
  
# Function to generate random portfolios  
def generate\_random\_portfolios(num\_portfolios, returns, ref):  
 num\_assets = returns.shape[1]  
 all\_weights = np.zeros((num\_portfolios, num\_assets))  
 all\_returns = np.zeros(num\_portfolios)  
 all\_volatilities = np.zeros(num\_portfolios)  
  
 for i in range(num\_portfolios):  
 weights = np.random.rand(num\_assets)  
 weights /= np.sum(weights)  
 portfolio\_return, portfolio\_volatility = portfolio\_performance(weights, returns, ref)  
 all\_weights[i, :] = weights  
 all\_returns[i] = portfolio\_return  
 all\_volatilities[i] = portfolio\_volatility  
  
 return all\_weights, all\_returns, all\_volatilities  
  
# Filtering random portfolios for Efficient Frontier Visualization  
def efficientfrontierpfs(num\_portfolios, returns, ref):  
 randwts, randrets, randvol = generate\_random\_portfolios(num\_portfolios, returns, ref)  
 randpfs = pd.DataFrame([randrets, randvol], index = ['randreturns', 'randvols']).T  
  
 initial = pd.DataFrame(np.ones(returns.shape[1]) / returns.shape[1], columns=['original'], index=returns.columns.values.tolist()).T  
 minvar = optim\_minvar(x=initial, r=returns, ref=ref)  
 minvstd = returns.dot(minvar.T).std().iloc[0]  
 minret = returns.dot(minvar.T).mean().iloc[0]  
  
 randefpf = randpfs[(randpfs['randvols'] >= minvstd) & (randpfs['randreturns'] >= minret)]  
 return randefpf.randreturns.values, randefpf.randvols.values  
  
# Plotting Efficient frontier  
def plot\_efficientfrontier(data, expfreturns, initial, periods, s):  
 fig, axs = plt.subplots(3, 2, figsize=(14,10))  
 axs = axs.flatten()  
 colors=['yellow', 'darkorange', 'pink', 'm', 'red']  
   
 for i, (start, end) in enumerate(periods):  
   
 # Slicing data for the specific decade  
 df\_period = data.loc[start:end]  
 rfrates = data['T-Bill'].loc[start:end].copy()  
 hist\_returns = df\_period.sub(rfrates, axis=0)  
 pf\_period = expfreturns.loc[str(start):str(end)].copy()  
   
 # Generate random portfolios  
 num\_portfolios = s  
 all\_returns, all\_volatilities = efficientfrontierpfs(num\_portfolios, df\_period, rfrates)  
   
 # Plot efficient frontier, portfolio points, period-specific max sharpe & min variance portfolios  
 axs[i].scatter(all\_volatilities, all\_returns, c=(all\_returns / all\_volatilities), marker='o', cmap='viridis', alpha=0.2)#, label='Random Portfolios')  
   
 maxsharpe = max\_sharpe(x=initial, r=df\_period, ref=rfrates, ppy=1 )  
 maxstd, maxsret = df\_period.dot(maxsharpe.T).std(), df\_period.dot(maxsharpe.T).mean()  
 axs[i].scatter(maxstd, maxsret, marker='o', edgecolors='black', color='white',label=f'decade-maxsharpe')  
   
 minvar = optim\_minvar(x=initial, r=df\_period, ref=rfrates)  
 minvstd, minvret = df\_period.dot(minvar.T).std(), df\_period.dot(minvar.T).mean()  
 axs[i].scatter(minvstd, minvret, marker='o', edgecolors='black', color='black', label=f'decade-minvar')  
   
 [axs[i].scatter(pf\_period.add(rfrates.values, axis=0).std().iloc[x], pf\_period.add(rfrates.values, axis=0).mean().iloc[x], marker='o', edgecolors='black', color=colors[x], label=pf\_period.columns.tolist()[x]) for x in range(pf\_period.shape[1])]  
  
 # Customize each subplot  
 axs[i].set\_title(f'Decade = {start}')  
 axs[i].set\_xlabel('Volatility')  
 axs[i].set\_ylabel('Return')  
 axs[i].set\_xlim(0, max(all\_volatilities.max(), pf\_period.std().max())+0.025)  
 axs[i].set\_ylim(0, max(all\_returns.max(), pf\_period.mean().max())+ 0.05)  
 axs[i].grid(True, linewidth='0.5')  
   
 axs[0].set\_title(f'Period = {data.index[0]}-{data.index[-1]+1}')  
 axs[3].legend(loc='upper right', bbox\_to\_anchor=(1.5, 1), title='Legend')   
 fig.suptitle("Markowitz Bullet & the Efficiency Frontier")  
 plt.tight\_layout()  
 plt.show()

periods = [(1928, 2024), (1930, 1940), (1970, 1980), (1980, 1990), (2004, 2014), (2014, 2024)]  
s = 30000 # number of random portfolios  
  
# Plotting Markowitz Bullet (omits inefficient portfolios whose risk-return tradeoff are dominated by min-variance)  
plot\_efficientfrontier(data, expfreturns, initialpfs.loc[:,:'Gold'].copy(), periods, s)



# periods = [(1928, 2024), (1930, 1940), (1970, 1980), (1980, 1990), (2004, 2014), (2014, 2024)]  
# decades=[1930, 1970, 1980, 2004, 2014]  
# plot\_portfolioweights(optimized)  
# plot\_cumrets(expfreturns)  
# plot\_cml(periods, data)  
# plot\_sharperatios(decades, expfreturns, True)  
# plot\_sharperatios(decades, data, False)  
# plot\_efficientfrontier(data, expfreturns, initialpfs.loc[:,:'Gold'].copy(), periods, s=50000)

# Portfolio Risk-Return Tradeoffs

## Portfolio 1: All Equity

Allocation: 100% S&P 500

Sharpe Ratio: 0.4187

#### PROS:

* **Potential for High Returns**: The all-equity portfolio can deliver substantial returns over the long term. Additionally, it is evident that this strategy dominated the others in achieving the highest return over the entire period, 1928 to 2024, see graph *Cumulative Returns for Each Portfolio*. The magnitude of these returns over the entire period is likely what caused utility to be the highest relative value in our investment universe, despite having the lowest sharpe.
* **Investor Preferences**: Despite its very low Sharpe ratio, this portfolio would potentially be attractive to risk-lover or risk-neutral investors.
* **Asset Class**: Managing an all-equity portfolio is straightforward since it involves only one asset class. This simplicity can be appealing to novice investors or those who prefer a hands-off approach to investing. Since this market-cap weighted index reflects the movements of all stocks in the market, this index is highly diversified in equity across multiple industries.

#### CONS:

* **Diversification**: An all-equity portfolio lacks diversification since it is solely invested in stocks. Without exposure to other asset classes like bonds or alternative investments, the portfolio’s performance is heavily reliant on the performance of the stock market.
* **Asset Class**: Indicated by the standard deviation at 19.87%. Equities are prone to significant price fluctuations, leading to high portfolio volatility. During market downturns, this volatility can result in substantial losses, since the S&P is highly exposed to systematic risk.

#### MEAN-VARIANCE EFFICIENCY:

* This portfolio was not as nearly mean-variance efficient as any of the others listed below, evidenced by the location of this portfolio on the *Markowitz Bullet & Efficiency Frontier* graph in the 30’s and by the *Sharpe Ratios for Each Portfolio* graph for both the entire period and in the 30’s.

## Portfolio 2: Traditional

Allocation: 60% S&P 500, 40% T-Bond

Sharpe Ratio: 0.4492

#### PROS:

* **Historical Performance**: The 60/40 portfolio has a long track record of delivering competitive returns with lower volatility compared to an all-equity portfolio. Historical data and the added benefits of diversification supports its effectiveness as a balanced investment strategy.
* This strategy is an improvement from the one before. By including in T-Bond into the portfolio, a less risky asset than S&P, we see a higher Sharpe ratio with lower return and risk than S&P 500. This strategy has the second highest cumulative returns on the graph *Cumulative Returns for Each Portfolio*, mimicking the S&P over time. This would be attractive to investors who are not risk-neutral and who are not as risk-loving as those invested in the All Equity portfolio.
* **Asset Class**: Provides diversification with bonds, reducing overall portfolio volatility which is reduced to 12.47 percent. Bonds tend to perform differently than stocks, providing a hedge against equity market downturns. This also provides a more secured level of return with lower risk, and reduces the risk in the Traditional portfolio, relative to All Equity.

#### CONS:

* **Exposure to Systematic Risk**: May underperform in certain market conditions compared to more aggressive portfolios. In our case, the average return has dropped from 8.32 percent to 5.60 percent yearly.
* **Asset Class**: While the 60/40 portfolio is less volatile than an all-equity portfolio, it still experiences moderate volatility due to its equity exposure. During severe market downturns, the portfolio can still incur significant losses due to high interest rates & systematic risk. It also fails to capture additional benefits of diversification due to limited allocation to alternative assets like gold or commodities.

#### MEAN-VARIANCE EFFICIENCY:

* This portfolio was not as nearly mean-variance efficient as any of the others listed below, evidenced by the location of this portfolio on the *Markowitz Bullet & Efficiency Frontier* graph in the 30’s and by the *Sharpe Ratios for Each Portfolio* graph for both the entire period and in the 30’s.

## Portfolio 3: Permanent

Allocation: 25% S&P 500, 25% T-Bill, 25% T-Bond, 25% Gold

Sharpe Ratio: 0.4554

#### PROS:

* **Benefits from Diversification**: Harry Browne’s Permanent Portfolio is more diversified than the previously portfolios with equal asset allocation spanning across: equities, treasury bills, long-term bonds, and gold. This diversification helps mitigate risks associated with economic cycles.This portfolio aims to mitigate losses during drawdown (recessionary periods) by allocating investment to different assets that respond differently to economic events, thereby providing more stability and more resilience during market turbulence in order to achieve less but constant returns over the time frame of investment. The key idea behind the Permanent Portfolio is that regardless of the economic condition, at least one portion of the portfolio will perform well, thereby reducing the overall volatility and providing a more stable return over the long term. In this case, all the asset classes have equal weights which is why the standard deviation is 7.17 percent and it is lesser than the All Equity portfolio and Traditional 60/40 Portfolio.
* **Investor Preferences**: This continue to improve the Sharpe ratio, relative to Portfolios 1 & 2. This portfolio likely outranks the other two, from the viewpoint of a Risk-Averse investor.
* **Asset Class**: Gold and T-Bills outperformed the CML in the 1970s, seen by the graph *Capital Market Line (CML) for Each Asset*. The in-sample portfolio likely viewed the high returns in gold as a worthwhile investment despite its accompanying high level of risk. T-Bills were expected to provide some cushion to the portfolio returns and both S&P and T-Bonds likely provided a decent risk-return tradeoff that Browne viewed as a viable addition to the portfolio. T-Bonds and T-Bills would hopefully mitigate potential losses should some economic factors cause large increases in interest rates.

#### CONS:

* **Low Cumulative Returns**: This portfolio provides the lowest cumulative returns over the entire period relative to all portfolios, seen in the graph above, “Cumulative Returns for Each Portfolio Type.” Risk-loving and Risk-neutral investors would not be interested in investing this type of investment portfolio.
* **Asset Class**: With 25% of the weight in gold, while the in-sample portfolio may have performed well in the 1970s, in the 1980s the portfolio performed poorly out-of-sample, seeing how gold lost a significant level of return but maintained the same level of risk and that T-Bills fell below the CML (see graph *Capital Market Line (CML) for Each Asset*. While the risk-return tradeoff for investments in the S&P, T-Bonds, and T-Bills may have helped mitigate some losses, at least 25% of the portfolios value suffered due to the heavy allocation in gold.

#### MEAN-VARIANCE EFFICIENCY:

* The portfolio’s performance was harmed by the poor risk-return tradeoffs of gold & t-bills, evidenced by the CML. This is also witnessed in the cumulative returns graph where we see a steeper, upward slope for their returns in the 1970s which continuously grew before the year of 1980. However the slope is not as steep after the 1980s and more likely falls somewhat between 1980-1990.
* While this portfolio had the highest Sharpe ratio in the *Sharpe Ratios for Each Portfolio* graph in the 1970s, it also had the lowest in the 1980s. This portfolio also appears to much closer to the efficiency frontier in the 1970s, with significant drop from the frontier in the 1980s. Due to the oil crisis in the 1970s, there was the highest increase in the fed funds rate in the 1980s as an attempt to combat inflation. This unforeseen economic scenario likely caused this portfolio to lack mean-variance efficiency in the 1980s, as opposed to the 1970s.
* This portfolio would have been mean-variance efficient when it was proposed in 1980. However, it would not be today, given its distance from the effiency frontier seen on the graph for the entire time period (*Markowitz Bullet & Efficiency Frontier*) and its Sharpe ratio of 0.455 being the median of our five portfolios.

## Portfolio 4: All Seasons

Allocation: 30% S&P 500, 55% T-Bonds, 15% Gold

Sharpe Ratio: 0.4855

#### PROS:

* **Designed for Economic Resilience**: Ray Dalio’s All Seasons Portfolio is designed to perform well in various economic conditions by including assets with low correlation to each other. The average return is more 3.81 percent which is more than Harry Browne’s Permanent Portfolio which is capped at 3.26 percent. The added benefits of diversification helps mitigate losses and maintain portfolio stability. Ray Dalio’s portfolio prioritizes asset allocation in a majority of lesser risky stocks, in addition to S&P 500. His idea for proposing the strategy was based on the past recessionary drawdowns during dot com burst and 2008 recession.
* **Asset Classes**: The All Seasons Portfolio applies a majority of the weight allocation across safer asset classes, reducing overall portfolio risk and volatility compared to other asset classes. T-Bonds risk-return tradeoff appear to be very close to the CML, and has a risk-return tradeoff close to the fair estimate of the model and comprises of a majority of the asset allocation. For 2004-2014, in the *Capital Market Line (CML) for Each Asset* graph, we can see that Gold outperforms the CML. Here, gold and S&P have the highest returns, accompanied by the highest volatility. Seeing that none of the assets are significantly below the CML, they provided a decent risk-return tradeoff in the ten years following 2004. In the ten years following 2014, the portfolio benefited from the 30% allocation in S&P with returns rising above 10%, relative to the previous ten years closer to 7.5%.

#### CONS:

* **Complexity in Management**: Managing the All Seasons Portfolio can be relatively complex due to the inclusion of multiple asset classes and their specific weightings. Investors may require sophisticated strategies for portfolio rebalancing. Example: Allocating assets according to specific weightings, such as 7.5% to commodities and 7.5% to gold, requires meticulous attention to detail and periodic adjustments.
* **Potential for Lower Returns**: During bull markets, the All Seasons Portfolio may have lower returns compared to all-equity portfolios, as it prioritizes reduced volatility over maximum growth potential.
* **Investor Preferences**: While we see an increase in Sharpe ratio, this portfolio may be less attractive to the risk-averse investor in comparison to Portfolio 3.
* **Asset Classes**: For the out-of-sample portfolio performance, this portfolio suffered due to the underperformance of their 55% allocation in T-Bonds with returns falling close to 0, and 15% allocation in Gold with returns falling close to half its value from the previous ten years.

#### MEAN-VARIANCE EFFICIENCY:

* All seasons portfolio neighbors the highest sharpe portfolio on the *Markowitz Bullet & Efficiency Frontier* graph in the ten years following 2004, seen above. This shows that this portfolio was mean-variance efficient for the years it was developed. From 2014-2024, the market suffered extreme loss in March 2023 and endured high levels of inflation due to the pandemic and the fed attempting to reduce the possibility of a recession. These unforeseen economic events caused a significant drop in returns for T-Bonds and commodities, in turn, the S&P 500 benefited from the initial drop in interest rates.
* The portfolio was efficient, seeing that they had the highest sharpe ratio in the *Sharpe Ratios for Each Portfolio* graph and a location close to the efficiency frontier from 2004-2014, despite the housing crisis in 2008. However, the out-of-sample performance in 2014-2024 shows that this portfolio fell significantly below the efficiency frontier and had the worst sharpe ratio out of all portfolios from 2014-2024.
* This portfolio was efficient when it was proposed in 2014. However, it would not be efficient if applied to today’s market.

## Portfolio 5: Optimized Maximum Sharpe

Allocation: 15.9% S&P 500, 12.1% T-Bill, 49.2% Baa Bond, 13.3% Real Estate, 9.5% Gold

Sharpe Ratio: 0.5471

#### PROS:

* **Maximum Risk-Return**: This portfolio is optimized for risk-adjusted returns, maximizing the Sharpe ratio for the entire time period. Here, this portfolio has the highest sharpe ratio among all the portfolios.
* **Diversified asset allocation**: By diversifying our assets moreso than the previous portfolios, this allows this portfolio to have less volatility, which can be beneficial in different market downturns and nullifying the downside risk of one asset underperforming the others due to unsystematic risk.
* **Asset Class**: With the two largest allocations in assets that lie significantly above the CML, Baa Bonds at 49% and S&P at 16%, we are able to capture returns above what is expected by the model for the entire time period. T-Bills also produce additional returns without any accompanying level of risk. The investment in Real Estate at 13% and Gold at 9%, provide additional diversifcation benefits and captures the value of these assets when their risk-return tradeoff rose above the CML.

#### CONS:

* **Potential for Lower Returns**: While the portfolio is optimized for risk-adjusted returns, due to the constraints in our optimization functions, its conservative asset allocation may result in lower average returns compared to more aggressive portfolios with security-specific equity allocations. Investors seeking higher growth potential may find the portfolio’s return profile suboptimal.
* **In Management**: The portfolio’s allocation across multiple asset classes with specific weightings may require ongoing monitoring and adjustment to maintain the desired risk-return profile. Managing and rebalancing the portfolio could be complex and time-consuming, particularly for individual investors without access to professional advisory services.
* **Historical Data**: Because average returns are less consistent than standard deviation year over year, these often fail to be replicated for the future period. Using these historical returns and volatility to calculate Sharpe will produce a portfolio that may or may not perform as well as expected.
* **Asset Class**: This portfolio has a high concentration of 49% in Baa Bonds, and any economic factors that could harm the performance of this asset will have a significant effect on the performance of our portfolio.

#### MEAN-VARIANCE EFFICIENCY:

* If the investor were to look at the performance of this portfolio for a select number of years, this portfolio would most likely fail to be mean-variance efficient. However, **for an investor who is invested in these assets for the entire timeframe of 1928-2024, we can clearly see, based on the Sharpe and Markowitz Bullet graphs, that this portfolio offers the best risk-return tradeoff due to its Sharpe ratio and its location on the efficiency frontier.**

In considerations with the whole project, the all equity portfolio gives the highest average excess returns of 8.32% among all other asset classes but the risk involved with the asset is also high having standard deviation at 19.87%, as can be seen in recession periods therefore all equity portfolios can create major drawdowns. The addition of Treasury bonds and other assets help in diversifying the risk as can be seen in Harry browne and Ray dalio portfolio which consists of various asset classes to support the portfolio for long run. These portfolios achieve excess returns of 3.26% and 3.81%, with lower standard deviations of 7.17% and 7.86%. Our fifth portfolio consists of maximizing the sharpe ratio to 0.5471, this helps us in achieving an average excess return of 3.55%. Reducing the standard deviation to 6.49, thereby cutting down on risk.

# Conclusion

Choosing the Portfolio 5, Optimized Maximum Sharpe, is currently the best strategic decision for both investors with an average risk aversion and investors who lean more risk-averse. However, highly risk-averse investors will most likely prefer investing in the minimum variance portfolio, which advised 100% investment in T-Bills, since this asset would provide returns with virtually no risk. Portfolio 5, Optimized Maximum Sharpe ratio had the highest Sharpe Ratio amongst the four other portfolios.

1. Value of a High Sharpe Ratio

Risk-Adjusted Returns: The Sharpe Ratio measures the excess return per unit of risk. A portfolio with a higher Sharpe Ratio implies receiving more return for every unit of risk. With a Sharpe Ratio of 0.5471, Portfolio 5 theoretically offers the best risk-adjusted returns among the options over the entire time period, making it particularly attractive for investors who are cautious about volatility with the same investment horizon.

1. Portfolio Optimization and Diversification

Diversified Investment: Portfolio 5’s allocation includes S&P 500, T-Bills, Baa Bonds, Real Estate, and Gold, which ultimately provides diversification benefits. Under Modern Portfolio Theory, diversification is key to reducing portfolio volatility without excessively sacrificing potential returns.

1. Optimized Asset Allocation

This portfolio strategically diversifies investments across asset classes, providing protection during market uncertainties and economic cycles. For instance, gold typically performs well during recessions and can offset declines in the stock market.

1. Consideration of Timeframes and Historical Events

Portfolio 5 takes into account a broad timeframe, including events like the Great Depression, the dot-com bubble, and the 2008 financial crisis. These events have a large impact on assessing and optimizing investment strategies. While past returns are no guarantee of future performance, Portfolio 5 aims to buffer against future uncertainties by optimizing for the Sharpe Ratio, considering historical performance. Evaluating more recent years in our dataset to optimize our Sharpe ratio, would like improve our portfolio performance using in-sample data.

## Limitations

* There is likely noise and outliers in our dataset, due to events such as the Great Depression, Oil Crisis, Dot Com bubble, 2008 financial crisis, Pandemic, etc, which has impacted the accuracy of the valuation of various securities. This has affected our historical analysis of assets and asset performance. Moreover, it has had a significant effect on how we derived our asset allocations and Sharpe ratio for our optimized portfolio.
* The optimization process of maximizing Sharpe under Markowitz mean-variance portfolio theory has led to overallocation of capital in assets. This would likely leave our portfolio overly exposed to risk in Baa Bonds. Setting new values as our ‘bounds’ to create a cutoff point for maximum asset allocation would help combat this. Alternatively, analyzing different optimization methods such as risk parity would likely help reduce this over-allocation.
* The idea of volatility = risk, when volatility represents the movement of stock returns both in a positive and negative direction will likely leave additional returns that were not achievable in our portfolio, since by seeking a high Sharpe ratio, we are also seeing lower volatility. Optimizing for a modified sharpe ratio where volatility is replaced with conditional value-at-risk might help us reduce the likelihood of receiving extremely negative returns as opposed avoiding both extremely positive and negative returns.
* Analyzing portfolios on a basis of one decade likely overstates or fails to see the actual performance of the asset or portfolio relative to what was actually occuring during the time that the proposed portfolio was in-sample verus out-of-sample.
* As seen through the analysis of the risk-return tradeoff for all portfolios, there is a consistent theme where historical data is used to analyze asset performance and this historical analysis is then assumed to also represent future performance. This assumption is detrimental to the performance of a portfolio once its strategy is tested in the market with new data. Either using most frequent rebalancing methods and/or using expected returns with implied volatility would likely provide better portfolio performance than the portfolios we have proposed here.