Assignment 9

Imports and Cleaning

In [9]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from tiingo import TiingoClient
import numpy as np
from datetime import date
import statsmodels.api as sm
from statsmodels import regression
import warnings
warnings.filterwarnings('ignore')
from dateutil.relativedelta import relativedelta
config = {}

config['session'] = True

config['api_key'] = "110ee73e29ec4269f49eb85cfb4b976ab8e73361"

client = TiingoClient(config)
```

In [5]:
```
def download_financial_data(ticker):
    fin_data = client.get_ticker_price(ticker,
                                       fmt='csv',
                                       startDate = date.today() - relativedelta(years=1),
                                       endDate = date.today(),
                                       frequency = 'daily')
    file_name = f"{ticker}.csv"
    with open(file_name,'w') as outfile:
        outfile.write(fin_data)
    print(f'{ticker}.csv created')
    return pd.read_csv(f"{ticker}.csv")
```

In [3]:
```
def compute_lin_reg(index,stock):
    x = index
    y = stock

    x = sm.add_constant(x)
    model = regression.linear_model.OLS(y,x).fit()

    x = x.drop(columns = 'const')
    return model.params[0],model.params[1]
```

In [6]:
```
spy_df = download_financial_data("SPY")
aapl_df = download_financial_data("AAPL")
wfc_df = download_financial_data("WFC")
ibm_df = download_financial_data("IBM")
ge_df = download_financial_data("GE")
tlsa_df = download_financial_data("TLSA")
```

```
SPY.csv created
AAPL.csv created
WFC.csv created
IBM.csv created
GE.csv created
TLSA.csv created
```

In [7]:
```
spy_df['date'] = pd.to_datetime(spy_df['date'])
aapl_df['date'] = pd.to_datetime(aapl_df['date'])
wfc_df['date'] = pd.to_datetime(wfc_df['date'])
ibm_df['date'] = pd.to_datetime(ibm_df['date'])
ge_df['date'] = pd.to_datetime(ge_df['date'])
tlsa_df['date'] = pd.to_datetime(tlsa_df['date'])
```

In [12]:
```
spy_df = spy_df[['date','adjClose']]
aapl_df = aapl_df[['date','adjClose']]
wfc_df = wfc_df[['date','adjClose']]
ibm_df = ibm_df[['date','adjClose']]
ge_df = ge_df[['date','adjClose']]
tlsa_df = tlsa_df[['date','adjClose']]
```

In [13]:
```
spy_df = spy_df.rename(columns = {'adjClose':'spy_adjClose'})
aapl_df = aapl_df.rename(columns = {'adjClose':'aapl_adjClose'})
wfc_df = wfc_df.rename(columns = {'adjClose':'wfc_adjClose'})
ibm_df = ibm_df.rename(columns = {'adjClose':'ibm_adjClose'})
ge_df = ge_df.rename(columns = {'adjClose':'ge_adjClose'})
tlsa_df = tlsa_df.rename(columns = {'adjClose':'tlsa_adjClose'})
```

In [14]:
```
df = spy_df.merge(aapl_df,on='date')
df = df.merge(wfc_df, on='date')
df = df.merge(ibm_df, on='date')
df = df.merge(ge_df,on='date')
df = df.merge(tlsa_df,on='date')
df = df.dropna()
df.head()
```

Out[14]:

	date	spy_adjClose	aapl_adjClose	wfc_adjClose	ibm_adjClose	ge_adjClose	tlsa_adjClose
0	2021-06-08	416.761382	126.020377	45.765644	135.641204	110.827522	2.30
1	2021-06-09	416.139615	126.408162	45.088144	137.097070	109.472077	2.31
2	2021-06-10	418.074001	125.393954	44.273179	136.978781	108.674757	2.41
3	2021-06-11	418.764853	126.626913	44.852491	137.652119	109.153149	2.49
4	2021-06-14	419.702438	129.739141	44.332092	136.514724	107.399045	2.61

In [16]:
```
df['spy_ret'] = df['spy_adjClose'].pct_change(1)
df['aapl_ret'] = df['aapl_adjClose'].pct_change(1)
df['wfc_ret'] = df['wfc_adjClose'].pct_change(1)
df['ibm_ret'] = df['ibm_adjClose'].pct_change(1)
df['ge_ret'] = df['ge_adjClose'].pct_change(1)
df['tlsa_ret'] = df['tlsa_adjClose'].pct_change(1)
df = df.dropna()
```

Alpha and Beta Calculations

In [17]:
```
aapl_a,aapl_b = compute_lin_reg(df['spy_ret'],df['aapl_ret'])
wfc_a,wfc_b = compute_lin_reg(df['spy_ret'],df['wfc_ret'])
ibm_a,ibm_b = compute_lin_reg(df['spy_ret'],df['ibm_ret'])
ge_a,ge_b = compute_lin_reg(df['spy_ret'],df['ge_ret'])
tlsa_a,tlsa_b = compute_lin_reg(df['spy_ret'],df['tlsa_ret'])
```

In [18]:
```
print(f"AAPL alpha: {aapl_a} beta: {aapl_b} \n"
      f"WFC alpha: {wfc_a} beta: {wfc_b} \n"
      f"IBM alpha: {ibm_a} beta: {ibm_b} \n"
      f"GE alpha: {ge_a} beta: {ge_b} \n"
      f"TLSA alpha: {tlsa_a} beta: {tlsa_b}")
```

```
AAPL alpha: 0.0007261221720641733 beta: 1.2798157567984034
WFC alpha: 0.00015941099031578502 beta: 1.0930878744237227
IBM alpha: 0.0003003383816437539 beta: 0.47573655124201997
GE alpha: -0.001259438904604089 beta: 1.0877589307030582
TLSA alpha: -0.0026456335675754433 beta: 0.42567236200472 55
```

In [19]:
```
df['high_risk_ret'] = (df['aapl_ret'] + df['wfc_ret'] + df['ge_ret']) /  3
df['low_risk_ret'] = (df['ibm_ret'] + df['tlsa_ret']) / 2
```

In [20]:
```
high_risk_a,high_risk_b = compute_lin_reg(df['spy_ret'],df['high_risk_ret'])
low_risk_a,low_risk_b = compute_lin_reg(df['spy_ret'],df['low_risk_ret'])
```

In [21]:
```
print(f"High Risk alpha: {high_risk_a} beta: {high_risk_b} \n"
      f"Low Risk alpha: {low_risk_a} beta: {low_risk_b}")
```

```
High Risk alpha: -0.00012463524740804368 beta: 1.1535541873083948
Low Risk alpha: -0.0011726475929658446 beta: 0.4507044566233728
```

In [24]:
```
df['middle_risk_ret'] = (.7 * df['aapl_ret']) + (.3 * df['ibm_ret'])
```

In [25]:
```
mid_risk_a,mid_risk_b = compute_lin_reg(df['spy_ret'],df['middle_risk_ret'])
```

In [32]:
```
print(f"Optimized Risk alpha: {mid_risk_a} beta: {mid_risk_b} \n"
      f"Contrast with High Risk Alpha: {round(((mid_risk_a-high_risk_a)/abs(high_risk_a))*100,2)}% beta: {round
      f"Contrast with Low Risk Alpha: {round(((mid_risk_a-low_risk_a)/abs(low_risk_a))*100,2)}% beta: {round((
```

```
Optimized Risk alpha: 0.0005983870349380477 beta: 1.0385919951314886
Contrast with High Risk Alpha: 580.11% beta: -9.97%
Contrast with Low Risk Alpha: 151.03% beta: 130.44%
```