Multi factor Model and Futures hedging

Fogli Alessandro
ID 231273
Project #2

Install packages

In [1]:

```
from scipy import stats
import pandas as pd
import numpy as np
import statsmodels.api as sm
import yfinance as yf
import pandas_datareader as pdr
from IPython.display import display, HTML
import datetime as dt
import getFamaFrenchFactors as gff
from fredapi import Fred
fred = Fred(api_key='8e31dd837475abc71d8beb8400da3bdf')
```

Data

Get data of difference in yields between BAA and AAA rated U.S. corporate bonds.

In [2]:

```
BAA = fred.get_series('BAA', observation_start="1992-02-01", observation_end="2022-02-0
1", frequency='m')
AAA = fred.get_series('AAA', observation_start="1992-02-01", observation_end="2022-02-0
1", frequency='m')
credit = BAA-AAA
credit = credit.tolist()
```

Get data of difference in yields between 10 year and 3 months U.S. Treasuries

In [3]:

```
term = fred.get_series('T10Y3M', observation_start="1992-02-01", observation_end= "2022
-02-01" ,frequency='m')
term = term.tolist()
```

Get data of S$P 500

In [4]:

```
sp500 = yf.download('^GSPC','1992-01-01','2022-03-01', interval ='1mo')
sp500_rtn = sp500.pct_change()
sp500_rtn = sp500_rtn['Adj Close']
sp500_rtn.fillna(0, inplace=True)
sp500_rtn.drop(index = sp500_rtn.index[0], axis=0, inplace=True)
sp500_rtn = sp500_rtn.apply(lambda x: x* 100) # get % return
sp500_rtn.index = sp500_rtn.index + pd.offsets.MonthEnd()
```

```
[*********************100%***********************]  1 of 1 completed
```

Get Fama French Data

In [5]:

```
fama_data = gff.famaFrench3Factor(frequency="m")
fama_data.rename(columns={"date_ff_factors": 'Date'}, inplace=True)
fama_data.set_index('Date', inplace=True)
fama_data = fama_data.loc[fama_data.index >= '1992-02-01']
fama_data = fama_data.loc[fama_data.index <= '2022-02-28']
fama_data.columns = ['mkt', 'smb', 'hml', 'rf']
fama_data = fama_data.drop('mkt', axis=1)
fama_data = fama_data.apply(lambda x: x* 100) # transform data in %
```

In [6]:

```
ff_data = fama_data
ff_data['credit'] = credit
ff_data['term'] = term
ff_data['mkt'] = sp500_rtn
ff_data.fillna(0, inplace=True)
ff_data = ff_data.replace([np.inf, -np.inf], 0)
factors = ff_data.drop(['rf'], axis=1)
```

Load data of S&P500 Futures - Jun 2022 expiring date

In [7]:

```
sp500_futures = pd.read_csv('S&P 500 Futures Historical Data.csv', index_col = 0)
sp500_futures.index = pd.to_datetime(sp500_futures.index, format= '%b %y') # date index
transforming
sp500_futures.index = sp500_futures.index + pd.offsets.MonthEnd()
sp500_futures_rtn = sp500_futures['Change %']
sp500_futures_rtn= sp500_futures_rtn.str.replace('%','') # remove str % from value
sp500_futures_rtn = pd.to_numeric(sp500_futures_rtn, errors='coerce')
sp500_futures_rtn.fillna(0, inplace=True)
sp500_futures_rtn = sp500_futures_rtn.iloc[::-1]
sp500_futures_rtn.drop(sp500_futures_rtn.tail(1).index,inplace=True) # adjusting size t
o match stocks month
```

Get data of The Walt Disney stock

In [8]:

```
disney = yf.download('DIS','1992-01-01','2022-03-1', interval ='1mo')
disney = disney.dropna()
disney_change = disney.pct_change()
disney_rtn = disney_change['Adj Close']
disney_rtn.fillna(0, inplace=True)
disney_rtn.drop(index = disney_rtn.index[0], axis=0, inplace=True)
disney_rtn = disney_rtn.apply(lambda x: x* 100) # get % return
disney_rtn.index = disney_rtn.index + pd.offsets.MonthEnd()
```

```
[*********************100%***********************]  1 of 1 completed
```

Get Stocks data for portfolio creation

In [9]:

```
tickers = ['DIS', 'CVX', 'WFC', 'BAC', 'IBM', 'PEP', 'JPM', 'GE', 'AXP', 'BRK-A']
start = dt.datetime(1992,1,1)
end = dt.datetime(2022,2,28)
portfolio = pdr.get_data_yahoo(tickers, start, end, interval='m')
portfolio.fillna(0, inplace=True)
portfolio.index = portfolio.index + pd.offsets.MonthEnd()
```

In [10]:

```
single_stocks_rtn = portfolio['Adj Close'].pct_change(1, fill_method='ffill')
single_stocks_rtn.fillna(0, inplace=True)
stocks_rtn = single_stocks_rtn.replace([np.inf, -np.inf], 0)
wts1 = [0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1] #set weights of the stocks in the port
folio
port_ret = (stocks_rtn* wts1).sum(axis = 1) # total montlhy return balanced by sotcks w
eights
port_ret.drop(index = port_ret.index[0], axis=0, inplace=True)
port_ret = port_ret.apply(lambda x: x* 100)
```

Multi-factor model

"A multi-factor model is a financial model that employs multiple factors in its calculations to explain market phenomena and/or equilibrium asset prices. A multi-factor model can be used to explain either an individual security or a portfolio of securities. It does so by comparing two or more factors to analyze relationships between variables and the resulting performance." Investopedia (https://www.investopedia.com/terms/m/multifactor-model.asp#:~:text=A%20multi%2Dfactor%20model%20is,or%20a%20portfolio%20of%20securities.)

In the previous project we considered the market return as the only factor affecting the return of any asset/portfolio with the following formula:

$$E_r - E_r : \alpha + \beta_1(R_m - R_f) + \epsilon$$

In this project we are also considering other factors deriving the following formula:

$$E_r - E_r : \alpha + \beta_1 Mkt + \beta_2 SMB + \beta_3 HML + \beta_4 Term + \beta_5 Credit + \epsilon$$

Where:

- E_r : expected return of stock/portfolio
- **α** : intercept
- β_i : slope coefficient for each explanatory variable
- **MKT** : the excess return of the market. It's the value-weighted return of all CRSP firms incorporated in the US and listed on the NYSE, AMEX, or NASDAQ minus the 1-month Treasury Bill rate.
- **SMB** : (Small Minus Big) measures the excess return of stocks with small market cap over those with larger market cap. It's a size discriminant factor also called (Short-Long portoflio), long on small companies stock and short on big companies stock. The use of this factor helps to include in the evaluation the size of the companies in the portfolio which is not considered with only the risk premium as factor.
- **HML** : (High Minus Low) measures the excess return of value stocks over growth stocks. Value stocks have high book to market ratio (B/P) than growth stocks. It is a disciminant value, usually small companies have high evaluation (book value) compared to market value. There are 2 values:
  - growth : young corporations have a market value > book value
  - mature/value stocks : corporations have market value < book value
- **term** : difference in yields between 10 year and 3 months U.S. Treasuries;
- **credit** : difference in yields between BAA and AAA rated U.S. corporate bonds
- ϵ : model error term (residual)

Multi factor with one stock

Defining a function to make a regression with explanatory variable (our factors) and dependent variable (stocks return)

In [11]:

```
def regression(explanatory, dependent):
    X = explanatory
    y = dependent
    X1 = sm.add_constant(X)
    # make regression model
    ff_model = sm.OLS(y, X1).fit()
    # fit model and print results
    print(ff_model.summary())
    global saved_values
    saved_values = ff_model.params
    saved_values = saved_values.tolist()
```

Regression between our factors and The Walt Disney Stock excess of return

In [19]:

```
regression(factors, (disney_rtn - ff_data['rf']))
```

```
                            OLS Regression Results
==========================================================================
====
Dep. Variable:                      y   R-squared:
0.413
Model:                            OLS   Adj. R-squared:
0.405
Method:                 Least Squares   F-statistic:                     4
9.99
Date:                Sun, 03 Apr 2022   Prob (F-statistic):           3.93
e-39
Time:                        12:03:42   Log-Likelihood:                -11
34.8
No. Observations:                 361   AIC:                             2
282.
Df Residuals:                     355   BIC:                             2
305.
Df Model:                           5
Covariance Type:            nonrobust
==========================================================================
====
                 coef    std err          t      P>|t|      [0.025      0.
975]
--------------------------------------------------------------------------
----
const         -1.2508      0.850     -1.472      0.142      -2.922
0.420
smb            0.1224      0.097      1.258      0.209      -0.069
0.314
hml            0.1949      0.095      2.047      0.041       0.008
0.382
credit         0.7471      0.799      0.935      0.350      -0.824
2.318
term           0.3193      0.273      1.169      0.243      -0.218
0.856
mkt            1.1218      0.072     15.481      0.000       0.979
1.264
==========================================================================
====
Omnibus:                       46.979   Durbin-Watson:
2.188
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               11
8.547
Skew:                           0.632   Prob(JB):                     1.81
e-26
Kurtosis:                       5.507   Cond. No.
15.8
==========================================================================
====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is cor
rectly specified.
```

Single stock analysis

By analyzing the results we have and R^2 of 0.405, it means that the 40% variation of returns of our stock can be explained by our selected factors : SMB, HML, CREDIT, TERM, MARKET.

SMB The coefficient of SMB is positive, it means that when small caps outperform large caps, the Small Cap Index will have higher returns, anyway the p_vaue tells us that our coefficient is not statistically significant.

HML The beta value shows a positive relation to the HML factor for value stock analysis, which explains that the portfolio's returns are attributable to the value premium. Even in that case the p_value is not statistically significant.

For Credit and Term we can also notice a negative p_value.

Market then we have a 1.12 β value which is statistically significant and it tells that for a 1% return by the market factor, we can expect our stock to return 1.12 * 1% in excess of the risk-free rate.

Const also known as α has a negative value, it explain everything that couldn't be explained by our factors, it means an underperforming respect to other, for a positive alpha portfolio there should be another person in the market with negative alpha. The wightes sum of all market alphas must be 0, that's because the weighted sum of the returns of all investors is equal to a market portfolio. For the market efficiency theory alpha should be almost 0. Here the alpha is not statistically significant.

Multi factor portfolio

Run regression between our factors and the stocks portfolio excess of return

In [18]:

```
regression(factors, (port_ret - ff_data['rf']))
```

```
                            OLS Regression Results
==============================================================================
====
Dep. Variable:                      y   R-squared:
0.827
Model:                            OLS   Adj. R-squared:
0.824
Method:                 Least Squares   F-statistic:                     3
38.4
Date:                Sun, 03 Apr 2022   Prob (F-statistic):          1.18e
-132
Time:                        12:03:34   Log-Likelihood:                -79
4.85
No. Observations:                 361   AIC:                             1
602.
Df Residuals:                     355   BIC:                             1
625.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
====
                 coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const         -0.7742      0.331     -2.337      0.020      -1.426      -
0.123
smb           -0.0423      0.038     -1.114      0.266      -0.117
0.032
hml            0.5242      0.037     14.117      0.000       0.451
0.597
credit         0.6604      0.312      2.120      0.035       0.048
1.273
term           0.1108      0.106      1.041      0.299      -0.099
0.320
mkt            1.0993      0.028     38.905      0.000       1.044
1.155
==============================================================================
====
Omnibus:                       49.657   Durbin-Watson:
2.079
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               16
4.617
Skew:                           0.575   Prob(JB):                     1.79
e-36
Kurtosis:                       6.102   Cond. No.
15.8
==============================================================================
====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is cor
rectly specified.
```

Portfolio analysis

By analyzing the results we have and R^2 of 0.827, it means that the 82% variation of returns of our portofolio can be explained by our selected factors : SMB, HML, CREDIT, TERM, MARKET.

SMB The coefficient of SMB is negative, it means that our portfolio is more oriented to companies with large capitalization, anyway the p_vaue tells us that our coefficient is not statistically significant.

HML The coefficient value is positive and it tells us that the portfolio behaves more as value stock. The p_value is statistically significant so the null hypotesis (no correlation between the portfolio and HML) is rejected.

For Credit and Term we can notice a negative p_value.

Market then we have a 1.08 β value which is statistically significant and it tells that for a 1% return by the market factor, we can expect our stock to return 1.08 * 1% in excess of the risk-free rate.

Const also known as α has a negative value, it means we have been rewarded less for the risk taken, it is statistically significant, so we can reject the null hypothesis.

Futures

Function to get beta coefficient

In [14]:

```
def beta(valx, valy):
    X = valx
    y = valy
    slope, intercept, r_value, p_value, std_err = stats.linregress(X, y)
    return round(slope,4)
```

Using the portofolio return as dependent variable and the S&P500 Futures return as explanatory variable we get the coefficient β also known as *optimal hedge ratio*

That coefficient can be used to derive the number of S&P500 Futures to hedge the portfolio, it can be helpful when we forecast an economic downturn and we want to avoid it.

We can compute the number of futures needed with the following formula:

$$N_{futures} : \beta(\frac{portfolio_{value}}{futures_{price}})$$

In [15]:

```
def futures_hedging_number():
    slope = beta(sp500_futures_rtn, port_ret.tail(len(sp500_futures_rtn))) #data len sh
ould match, we have less futures historical data
    sp500_futures['Price']= sp500_futures['Price'].str.replace(',','')
    futures_price = pd.to_numeric(sp500_futures['Price'])
    futures_price = futures_price.iloc[0]
    futures_price = pd.to_numeric(futures_price)
    futures_number = slope * (1000000/futures_price)
    return futures_number
futures_hedging_number()
```

Out[15]:

```
226.61448140900197
```

For the portfolio composed by the selected stocks we need #227 S&P500 Futures to hedge against a market downturn, and avoid the systematic risk.