Project-1
IPO Study

In This study, you will examine stock IPOs over the period 2012-2023. The data is created using yahoo finance, iterating over tickers listed in the Nasdaq and the S&P 500 and a list of (Special Purpose Acquisition Corporations. (SPACs) stocks. In addition, you will use data files on list of SPACs IPOs, on the Russell1000 and the S&P500. In the project folder you will find the following files:

'stock_ipos_20231001.csv''
' list_of_all_spacs.xlsx'
'sp500_202308.xlsx'
'russ_1000_202308.xlsx'

You will use these data to examine (i) Whether an IPO return over a window is predictive of following-windows-returns; (ii) Whether SPACS IPOs returns differ from other IPOs returns; And (iii) the relationship between IPOs returns and its inclusion in the S&P500 or the Russell1000.

Place the above files in your working directory and complete the following activities

1. (i) Run the following code:

```
import os
import pandas as pd
cur_dir = os.getcwd()
files = os.listdir(cur_dir)
stock_ipos = pd.read_csv('stock_ipos_20231004.csv')
stock_ipos.info()
```

(ii) what is the data type of the column 'ipo_date' ? Run this code and explain what is doing.

```
stock_ipos=stock_ipos.dropna()
stock_ipos['ipo_date'] = pd.to_datetime(stock_ipos['ipo_date'])
stock_ipos['year']  = stock_ipos['ipo_date'].dt.year
stock_ipos['month'] = stock_ipos['ipo_date'].dt.month
stock_ipos.info()
```

(iii) Run the following code and explain what is doing.

```
Import numpy as np
stock_spacs = pd.read_excel('list_of_all_spacs.xlsx')
print (stock_spacs.shape)
spacs_tkrs = list( stock_spacs['symbol'] )
stock_ipos['spac'] = np.where(stock_ipos['symbol'].isin(spacs_tkrs) , 'yes', 'no')
stock_ipos['spac'].value_counts()
```

How many stocks in 'list_of_all_spacs.xlsx' ? Explain what does the 4th line in the code achieve? How many SPACs we have in the stock_ipos data frame?

(iv) Replicate the code in (iii) reading the following two excel file:
'sp500_202308.xlsx'
'russ_1000_202308.xlsx'
Create two columns in the data frame stock_ipos:
(i) 'sp' with two values 'yes and 'no' to flag stocks in S&P.
(ii) 'russell' with two values 'yes and 'no' to flag stocks in Russel1000.
How many stocks in the data frame stock_ipos are in the S&P500? How many are in Russell1000

2. (i) Run the following and explain what is doing.

```
ipos_spacs_count =stock_ipos.groupby([ 'year', 'spac'])['symbol'].count().reset_index()
import seaborn as sns
sns.set(rc={"figure.figsize":(10, 6)})
sns.barplot(x= 'year', y= 'symbol',  hue = 'spac' , data= ipos_spacs_count )
```

Comment on the share of SPACs from all IPOs over time. Do your own web research to find why SPACs appeal has changed recently and how it is reflected in these data.

(ii) Use the groupby method in (i) to determine how many of the SPACS stocks are in the S&P, in Russell1000. Use the S&P and Russell dataframes and an analogous code as in item 1.3, line 4.

3. Use the describe method to examine the mean and the standard-deviation return difference between stock ipos and Russell 1000 return for each IPO return. Specifically, run this code:

```
stock_ipos[['ipo_date', 'sym_day0_OTC', 'iwv_day0_OTC',
                    'sym_1day_ret' , 'iwv_1day_ret',
                    'sym_5day_ret' , 'iwv_5day_ret',
                    'sym_22day_ret' , 'iwv_22day_ret',
                    'sym_91day_ret' , 'iwv_91day_ret',
                    'sym_252day_ret' , 'iwv_252day_ret'
                    ]].describe()
```

(i) Why it makes sense to organize the variables the way it is done here?
(ii) Compare the mean of stock IPOs returns with the Russell 1000 returns for different return window. What can you conclude?
(iii) Compare the median of stock IPOs returns with the Russell 1000 returns for different return window. What can you conclude? Contrast your conclusion here with your conclusion in (ii)
(iv) Compare the std of stock IPOs returns with the Russell 1000 returns for different return window. What can you conclude? Contrast your conclusion here with your conclusions in (ii) and (iii)

4. Examine how predictive is 'sym_day0_OTC' of future returns following the IPO date.
(i) Explain what the following code does:

```
sym_ret = ['sym_day0_OTC', 'sym_1day_ret', 'sym_5day_ret', 'sym_22day_ret',
'sym_91day_ret',   'sym_252day_ret']
stock_ipos [ sym_ret].corr()
```

How predictive is 'sym_day0_OTC' of the return on the IPO stock a week following the ipo_date?
How predictive is 'sym_day0_OTC' of the return on the IPO stock a month, a quarter, a year following the ipo_date?
What is the highest correlation in the correlation matrix? Why do you think it is that high?

(ii) The following imports statsmodels and runs a regression of the one-year return of an IPO against other returns symbol returns

```
import statsmodels.formula.api as smf
lm_22d = smf.ols('sym_252day_ret ~  sym_22day_ret'
      , data = stock_ipos).fit()
lm_22d.summary()
```

Use the model summary output to answer the comment on the ability to predict one year return based on one month return? Specifically, on R-squared, the F-stat and regression coefficients.

(iii) The results of the model are too good to be true. If they are true, you can make significant money by buying a stock after one month of its IPO, hold it for another 11 months and make money? Before Rushing into conclusions, create a scatter plot of the regression data.

(a) Verify that the following code achieves this plot.

```
import matplotlib.pyplot as plt
sns.set(rc={"figure.figsize":(8, 4)})
g= sns.scatterplot(x='sym_22day_ret',
               y='sym_252day_ret',
               data=stock_ipos)
x_label = 'sym_22day_ret'
y_label = 'sym_252day_ret'
#g.set_xlim(left=0, right=5)
#g.set_ylim(bottom=-1, top=10)
plt.title("One Year Return against First Day Return")
plt.show(g)
```

What conclusions you can draw from this graph? Can you explain the cause of the high correlation?

(b) Run the same code but un-comment the two lines highlighted in yellow. What does this picture show?

(c) Verify that the following creates a data frame stock_ipos_ that filters out outlier based on the graph in the uncommented lines.

```
stock_ipos_ = stock_ipos[ (stock_ipos['sym_22day_ret'] < 5 ) ]
```

(d) Run a regression in (ii) after filtering out outlier (Hint: the data is now the data frame defined in (c))
Comment on the F-stat and the regression coefficient. Can you argue that sym_22day_ret is predictive of sym_252day_ret based on the significance of the regression coefficients?

(e) Still there is something wrong in using the regression in (d) to predict its target variable. Google the concept 'Look-ahead Bias ', and try examining how it applies here.

(f) Create a column in the data frame and name it 'sym_22_252_ret'. That measures the return in the 11 months following the first month. Mathematical, 'sym_22_252_ret' = (1+ sym_252day_ret)/(1+ sym_22day_ret) – 1.

(g) Examine the ability predicting sym_22_252_ret using sym_22day_ret both graphically and using by examining the correlation matrix of ['sym_22day_ret', 'sym_252day_ret', 'sym_22_252_ret'] in the data frame stock_ipos_.

5. Let us examine whether the SPACs return differ from other IPO returns. The following starts with returns on the IPO date.
   (i) Verify that the following code flags abnormal return and summarized day0 return by SPACs

```
stock_ipos['day0_lvl'] = np.where ( stock_ipos['sym_day0_OTC'] < 1 , 'normal', 'abnormal' )
stock_ipos.groupby(['day0_lvl','spac'])['sym_day0_OTC'].agg(['mean', 'median','std', 'count','min' ,'max'])
```

   Comment on SPACs return with None-SPACs return on day 0.

   (ii) Apply the analysis above, to 'sym_5day_ret' , sym_22day_ret, sym_91day_ret, and sym_91day_ret.

6. Compare the performance IPO return between stock that made to be included in the S&P and those that were not included in the S&P. You may focus on one-year-returns
   (i) Answer the above question for the Russell1000.

Sources:
https://www.stockmarketmba.com/listofshellcompanies.php