Assignment5_Ji_Qi

April 24, 2022

1 *Student Name: Ji Qi , Session B1*

2 Import packages

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import statsmodels.api as sm
from statsmodels.sandbox.regression.predstd import wls_prediction_std
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

3 Upload CSV file with data

[3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

3.1 tempGICS.csv

- Download Compustat GGROUP for "Data Date" 2021-01 through 2021-12 from WRDS Compustat and store into tempGICS.csv
- tempGICS CSV file contains 1886 data
- the `ggroup variable` represents `Industry Group GICS code`

[4]:
```
gics = pd.read_csv('/content/drive/MyDrive/BA_870/HW/5/tempGICS.csv')
```

- No missing value for tempGICS.csv

[5]:
```
gics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1886 entries, 0 to 1885
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   gvkey     1886 non-null   int64
 1   datadate  1886 non-null   int64
 2   fyear     1886 non-null   int64
 3   indfmt    1886 non-null   object
 4   consol    1886 non-null   object
 5   popsrc    1886 non-null   object
 6   datafmt   1886 non-null   object
 7   tic       1886 non-null   object
 8   curcd     1886 non-null   object
 9   costat    1886 non-null   object
 10  ggroup    1886 non-null   int64
dtypes: int64(4), object(7)
memory usage: 162.2+ KB
```

- first 5 rows of tempGICS.csv

[6]:
```
gics.head()
```

[6]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	curcd	costat	ggroup
0	1004	20210531	2020	INDL	C	D	STD	AIR	USD	A	2010
1	1045	20211231	2021	INDL	C	D	STD	AAL	USD	A	2030
2	1075	20211231	2021	INDL	C	D	STD	PNW	USD	A	5510
3	1078	20211231	2021	INDL	C	D	STD	ABT	USD	A	3510
4	1161	20211231	2021	INDL	C	D	STD	AMD	USD	A	4530

- In total, 24 unique Industry Group GICS code for tempGICS.csv file

[7]:
```
gics.ggroup.nunique()
```

[7]: 24

3.2 ProjectTickers.csv

- Import ProjectTickers.csv file

- 3 columns: **Ticker** (the stock's ticker symbol), **Name** (the name of each company), and **RetTYD** (the year-to-date stock return of each company from January 1, 2022 to April 14, 2022).

[8]:
```
ticker = pd.read_csv('/content/drive/MyDrive/BA_870/HW/5/ProjectTickers.csv')
ticker.head()
```

[8]:

	Ticker	Name	RetYTD
0	A	Agilent Technologies	-0.2080
1	AA	Alcoa Corp	0.4731
2	AAL	American Airlines Gp	0.0579
3	AAN	Aarons Holdings Company	-0.1327
4	AAON	Aaon Inc	-0.3456

- No missing value for ProjectTickers.csv file

[9]:
```
ticker.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1886 entries, 0 to 1885
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Ticker  1886 non-null   object
 1   Name    1886 non-null   object
 2   RetYTD  1886 non-null   float64
dtypes: float64(1), object(2)
memory usage: 44.3+ KB
```

- first 5 rows of ProjectTickers.csv file

[10]:
```
ticker.head()
```

[10]:

	Ticker	Name	RetYTD
0	A	Agilent Technologies	-0.2080
1	AA	Alcoa Corp	0.4731
2	AAL	American Airlines Gp	0.0579
3	AAN	Aarons Holdings Company	-0.1327
4	AAON	Aaon Inc	-0.3456

4 Data Merging (tempGICS.csv & ProjectTickers.csv)

[11]:
```
df = pd.merge(gics, ticker, how = 'outer', left_on= 'tic', right_on= 'Ticker',␣
 ↪indicator= True)
df._merge.value_counts()
```

[11]:
```
both          1886
left_only        0
right_only       0
Name: _merge, dtype: int64
```

- No missing value for this merged dataset

[12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1886 entries, 0 to 1885
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   gvkey     1886 non-null   int64
 1   datadate  1886 non-null   int64
 2   fyear     1886 non-null   int64
 3   indfmt    1886 non-null   object
 4   consol    1886 non-null   object
 5   popsrc    1886 non-null   object
 6   datafmt   1886 non-null   object
 7   tic       1886 non-null   object
 8   curcd     1886 non-null   object
 9   costat    1886 non-null   object
 10  ggroup    1886 non-null   int64
 11  Ticker    1886 non-null   object
 12  Name      1886 non-null   object
 13  RetYTD    1886 non-null   float64
 14  _merge    1886 non-null   category
dtypes: category(1), float64(1), int64(4), object(9)
memory usage: 223.0+ KB
```

- First 5 rows of this merged dataset

[13]:
```
df.head()
```

[13]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	curcd	costat
0	1004	20210531	2020	INDL	C	D	STD	AIR	USD	A
1	1045	20211231	2021	INDL	C	D	STD	AAL	USD	A
2	1075	20211231	2021	INDL	C	D	STD	PNW	USD	A
3	1078	20211231	2021	INDL	C	D	STD	ABT	USD	A
4	1161	20211231	2021	INDL	C	D	STD	AMD	USD	A

	ggroup	Ticker	Name	RetYTD	_merge
0	2010	AIR	AAR Corp	0.2944	both
1	2030	AAL	American Airlines Gp	0.0579	both
2	5510	PNW	Pinnacle West Capital Corp	0.0985	both
3	3510	ABT	Abbott Laboratories	-0.1638	both

```
4    4530    AMD           Adv Micro Devices  -0.3533   both
```

4.1 Export assign5.csv file

[14]:
```
df.to_csv('assign5.csv', index = False)
```

5 Create Industry Indicator (category) variables for the 24 Industry categories

- Select useful columns for the downstream tasks

[15]:
```
df = df[['Ticker', 'ggroup', 'RetYTD']]
df.head()
```

[15]:
```
  Ticker  ggroup  RetYTD
0    AIR    2010  0.2944
1    AAL    2030  0.0579
2    PNW    5510  0.0985
3    ABT    3510 -0.1638
4    AMD    4530 -0.3533
```

[16]:
```
df_new = pd.get_dummies(df, columns=['ggroup'])
df_new
```

[16]:
```
      Ticker  RetYTD  ggroup_1010  ggroup_1510  ggroup_2010  ggroup_2020  \
0        AIR  0.2944            0            0            1            0
1        AAL  0.0579            0            0            0            0
2        PNW  0.0985            0            0            0            0
3        ABT -0.1638            0            0            0            0
4        AMD -0.3533            0            0            0            0
…      …      …           …            …            …            …
1881     KRG  0.0275            0            0            0            0
1882     LYB  0.1664            0            1            0            0
1883     FRO  0.3380            1            0            0            0
1884    ALLE -0.1888            0            0            1            0
1885     LPG  0.2427            1            0            0            0

      ggroup_2030  ggroup_2510  ggroup_2520  ggroup_2530  …  ggroup_4010  \
0               0            0            0            0  …            0
1               1            0            0            0  …            0
2               0            0            0            0  …            0
3               0            0            0            0  …            0
4               0            0            0            0  …            0
…             …          …            …            …  …          …
```

```
1881            0            0            0            0  …            0
1882            0            0            0            0  …            0
1883            0            0            0            0  …            0
1884            0            0            0            0  …            0
1885            0            0            0            0  …            0

      ggroup_4020  ggroup_4030  ggroup_4510  ggroup_4520  ggroup_4530  \
0               0            0            0            0            0
1               0            0            0            0            0
2               0            0            0            0            0
3               0            0            0            0            0
4               0            0            0            0            1
…             …            …            …            …            …
1881            0            0            0            0            0
1882            0            0            0            0            0
1883            0            0            0            0            0
1884            0            0            0            0            0
1885            0            0            0            0            0

      ggroup_5010  ggroup_5020  ggroup_5510  ggroup_6010
0               0            0            0            0
1               0            0            0            0
2               0            0            1            0
3               0            0            0            0
4               0            0            0            0
…             …            …            …            …
1881            0            0            0            1
1882            0            0            0            0
1883            0            0            0            0
1884            0            0            0            0
1885            0            0            0            0

[1886 rows x 26 columns]
```

6 Question 1: What are the average stock returns for each industry classification (24 industry groups)?

- The dataframe shown below demonstrates the **average stock returns** for each industry classification. The range of average stock returns is from **-0.25 to 0.52** among 24 different industries.
- **The lowest average stock returns** is around **-0.25** and its corresponding industry is **Semiconductors & Semiconductor Equipment**. Despite the continuous chip shortage starting from the outbreak of COVID 19, The 2022 stock price for the semiconductor industry appears to go down. Several leading stocks like Nvidia, Taiwan Semiconductor and Intel are **all down over 20% year to date** due to **downgrades** and the **increased risk of con-**

sumer spending decreasing. For instance, On April 11th, Baird downgraded Nvidia to neutral and reduced its price target from **$360 to $225**, because there exists a slowdown in consumer spending, especially in China's consumer market, the smartphone market shifts and headwinds caused by the Russian embargo. Furthermore, South Korea, a hub for semiconductor production, **relies on Neon, xenon and krypton** used in the production of advanced chips and **imported from Russia and Ukraine**. However, **the recent tension between Russia and Ukraine and accompanying sanctions on Russia have cut off its supply**.

- **The highest average stock returns is around -0.52** and its corresponding industry is **Energy**. In general, the Energy sector includes energy equipment and services, and oil, gas and consumable fuels. With the gradual reopening of the global economy, the demand in oil has incredibly increased. In the other hand, **the total oil inventories have declined due to some cautious producers like OPEC and U.S. Especially, rising tensions amid Russia/Ukraine war exacerbated the oil shortage, which in turn highly increase the oil prices**. In addition, **Valuations in the Energy sector are attractive relative to the other sectors**. Although there are the strong gains in energy stock prices, they have not kept up with rapidly rising earnings expectations.

[17]:
```
# Create a dataframe for avg stock return using "groupby method"
avg_ret = pd.DataFrame(df.groupby('ggroup').RetYTD.mean()).reset_index()
# sort RetYTD from the lowest to highest
avg_ret.sort_values('RetYTD')
```

[17]:

	ggroup	RetYTD
19	4530	-0.252813
6	2520	-0.208598
5	2510	-0.207992
18	4520	-0.172065
13	3520	-0.163937
11	3030	-0.153771
8	2550	-0.151223
15	4020	-0.122515
17	4510	-0.108644
2	2010	-0.090548
14	4010	-0.087832
21	5020	-0.070871
4	2030	-0.067454
12	3510	-0.066432
7	2530	-0.062583
23	6010	-0.035826
3	2020	-0.035157
16	4030	0.003779
20	5010	0.011838
10	3020	0.038780
22	5510	0.040759
1	1510	0.057846
9	3010	0.078312

```
0      1010  0.515035
```

7 Question 2: Run fixed effect regressions

- Use constant and drop one industry indicator (in this case "ggroup_1010")
- Note the interpretation of coefficients on industry indicators (relative to CONSTANT which represents dropped industry (ggroup_1010)

[18]:
```
# add constant column to the original dataframe
df_new['constant'] = 1
df_new
```

[18]:

	Ticker	RetYTD	ggroup_1010	ggroup_1510	ggroup_2010	ggroup_2020	\
0	AIR	0.2944	0	0	1	0	
1	AAL	0.0579	0	0	0	0	
2	PNW	0.0985	0	0	0	0	
3	ABT	-0.1638	0	0	0	0	
4	AMD	-0.3533	0	0	0	0	
…	…	…	…	…	…	…	
1881	KRG	0.0275	0	0	0	0	
1882	LYB	0.1664	0	1	0	0	
1883	FRO	0.3380	1	0	0	0	
1884	ALLE	-0.1888	0	0	1	0	
1885	LPG	0.2427	1	0	0	0	

	ggroup_2030	ggroup_2510	ggroup_2520	ggroup_2530	…	ggroup_4020	\
0	0	0	0	0	…	0	
1	1	0	0	0	…	0	
2	0	0	0	0	…	0	
3	0	0	0	0	…	0	
4	0	0	0	0	…	0	
…	…	…	…	…	…	…	
1881	0	0	0	0	…	0	
1882	0	0	0	0	…	0	
1883	0	0	0	0	…	0	
1884	0	0	0	0	…	0	
1885	0	0	0	0	…	0	

	ggroup_4030	ggroup_4510	ggroup_4520	ggroup_4530	ggroup_5010	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	1	0	
…	…	…	…	…	…	

```
1881            0            0            0            0            0
1882            0            0            0            0            0
1883            0            0            0            0            0
1884            0            0            0            0            0
1885            0            0            0            0            0

      ggroup_5020  ggroup_5510  ggroup_6010  constant
0               0            0            0         1
1               0            0            0         1
2               0            1            0         1
3               0            0            0         1
4               0            0            0         1
…             …            …            …         …
1881            0            0            1         1
1882            0            0            0         1
1883            0            0            0         1
1884            0            0            0         1
1885            0            0            0         1

[1886 rows x 27 columns]
```

[30]:
```
# Define x as a subset of original dataframe
# only keep industy dummy variables and drop one industry indicator (let's␣
 ↪choose "gsector_10")
x = df_new.drop(columns=['Ticker', 'RetYTD', 'ggroup_1010'])
# Define y as a series
y = df_new['RetYTD']
# pass x as a dataframe, while pass y as a series
sm.OLS(y, x).fit().summary()
```

[30]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  RetYTD   R-squared:                       0.326
Model:                            OLS   Adj. R-squared:                  0.317
Method:                 Least Squares   F-statistic:                     39.07
Date:                Sun, 24 Apr 2022   Prob (F-statistic):          6.29e-141
Time:                        04:13:32   Log-Likelihood:                 357.63
No. Observations:                1886   AIC:                            -667.3
Df Residuals:                    1862   BIC:                            -534.2
Df Model:                          23
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
ggroup_1510    -0.4572      0.031    -14.692      0.000      -0.518      -0.396
```

```
ggroup_2010    -0.6056      0.028    -21.924      0.000      -0.660      -0.551
ggroup_2020    -0.5502      0.034    -15.980      0.000      -0.618      -0.483
ggroup_2030    -0.5825      0.040    -14.646      0.000      -0.660      -0.504
ggroup_2510    -0.7230      0.047    -15.540      0.000      -0.814      -0.632
ggroup_2520    -0.7236      0.034    -21.018      0.000      -0.791      -0.656
ggroup_2530    -0.5776      0.034    -16.849      0.000      -0.645      -0.510
ggroup_2550    -0.6663      0.032    -20.932      0.000      -0.729      -0.604
ggroup_3010    -0.4367      0.054     -8.070      0.000      -0.543      -0.331
ggroup_3020    -0.4763      0.038    -12.449      0.000      -0.551      -0.401
ggroup_3030    -0.6688      0.054    -12.359      0.000      -0.775      -0.563
ggroup_3510    -0.5815      0.030    -19.639      0.000      -0.640      -0.523
ggroup_3520    -0.6790      0.028    -24.154      0.000      -0.734      -0.624
ggroup_4010    -0.6029      0.028    -21.756      0.000      -0.657      -0.549
ggroup_4020    -0.6375      0.031    -20.342      0.000      -0.699      -0.576
ggroup_4030    -0.5113      0.035    -14.719      0.000      -0.579      -0.443
ggroup_4510    -0.6237      0.030    -20.963      0.000      -0.682      -0.565
ggroup_4520    -0.6871      0.032    -21.346      0.000      -0.750      -0.624
ggroup_4530    -0.7678      0.036    -21.470      0.000      -0.838      -0.698
ggroup_5010    -0.5032      0.061     -8.314      0.000      -0.622      -0.384
ggroup_5020    -0.5859      0.039    -15.090      0.000      -0.662      -0.510
ggroup_5510    -0.4743      0.035    -13.464      0.000      -0.543      -0.405
ggroup_6010    -0.5509      0.029    -18.884      0.000      -0.608      -0.494
constant        0.5150      0.023     22.140      0.000       0.469       0.561
==============================================================================
Omnibus:                      403.265   Durbin-Watson:                   2.008
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2314.656
Skew:                           0.877   Prob(JB):                         0.00
Kurtosis:                       8.136   Cond. No.                         26.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

7.1 What is the explanatory power of these regressions using different types of industry variables. Is this high or low? Explain.

Basically, After trying several linear regression models by dropping different industry varibales, I found that the highest R-squared is 0.326, and adjusted R-squared is 0.317. Those values are generated by including all 24 industry indicators in the regression model. If dropping one or more indicators, both R-squared and adj R-squared decrease. In terms of the explanatory power of these regressions, 32.6% of variation in Year to date Stock Return has been explained by the variation in 24 industry indicators. Although the model with 0.326 R-squared usually couldn't be considered as having a high explanatory power, this value in the stock market which is unstable and erratic could be deemed as a reasonbale value.

[ ]:
```
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
↪texlive-plain-generic
```

[ ]:
```
!jupyter nbconvert --to pdf '/content/drive/MyDrive/BA_870/HW/5/
↪Assignment5_Ji_Qi.ipynb'
```