Assignment3_Ji_Qi

April 10, 2022

1 *Student Name: Ji Qi , Session B1*

2 Calculating Risk Exposures at the Start of the Year 2022 for 100 Stocks

2.0.1 Import libraries and packages

[65]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

2.0.2 Upload WRDS CRSP Stock Return Data

- CSV file contains monthly stock returns for 100 stocks

[66]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[67]:
```
stockret = pd.read_csv('/content/drive/MyDrive/BA_870/HW/3/100-Stocks-Returns.
↪csv')
```

Examine varaibles in dataframe

[68]:
```
stockret.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   PERMNO  6000 non-null   int64
 1   date    6000 non-null   int64
 2   TICKER  6000 non-null   object
 3   RET     6000 non-null   float64
dtypes: float64(1), int64(2), object(1)
memory usage: 187.6+ KB
```

Print out header of dataframe

- Note that BWXT data appears first (60 monthly observations)

[69]:

```
stockret.head()
```

[69]:

```
   PERMNO      date TICKER       RET
0   10220  20170131   BWXT  0.045088
1   10220  20170228   BWXT  0.119306
2   10220  20170331   BWXT  0.026916
3   10220  20170428   BWXT  0.032983
4   10220  20170531   BWXT -0.009355
```

Print out ”tail” (the last 5 obs) of dataframe

- Note that BLDR data is at the end (60 monthly observations)

[70]:

```
stockret.tail()
```

[70]:

```
      PERMNO      date TICKER       RET
5995   90720  20210831   BLDR  0.197528
5996   90720  20210930   BLDR -0.029086
5997   90720  20211029   BLDR  0.126208
5998   90720  20211130   BLDR  0.191694
5999   90720  20211231   BLDR  0.234303
```

2.0.3 Create 100 new dataframes for each stock: monthly data

[71]:

```
# A 100-stocks list
ticker = stockret.TICKER.unique().tolist()
```

[72]:

```
# Create 100 new stock dataframes and store into a dictionary
stockret_dict = {}
for i in ticker:
  stockret_dict[i] = stockret[stockret['TICKER'] == i]
```

[88]:

```
# All 100 stock tickers
stockret_dict.keys()
```

```
[88]: dict_keys(['BWXT', 'BCPC', 'CAL', 'BC', 'BAH', 'BKU', 'BCOV', 'BLMN', 'BERY',
      'BFAM', 'BCC', 'BLUE', 'BNFT', 'BRX', 'BURL', 'BRG', 'BLBD', 'BWFG', 'BCLI',
      'BGSF', 'BOOT', 'CALA', 'BOX', 'BPMC', 'CABO', 'BLD', 'BATRA', 'BATRK', 'BL',
      'BPOP', 'BOH', 'BCO', 'BRC', 'BMY', 'BA', 'CACI', 'CALM', 'CAMP', 'CAH', 'BAX',
      'BOOM', 'BEN', 'BDX', 'B', 'BK', 'BMI', 'CAG', 'BRT', 'BLL', 'BAC', 'BKH',
      'BXMT', 'BRO', 'BIG', 'BDN', 'C', 'BLFS', 'BHE', 'BIIB', 'BOKF', 'BKE', 'BSX',
      'BBBY', 'CACC', 'CAKE', 'BLX', 'BANF', 'BBSI', 'BWA', 'BFS', 'BDC', 'BYD',
      'BZH', 'BCRX', 'BANR', 'BJRI', 'BXP', 'BAM', 'BHB', 'CAC', 'BRKL', 'BBY',
      'BMTC', 'BELFB', 'BUSE', 'BCOR', 'BSRR', 'BMRN', 'BLK', 'BMRC', 'BGCP', 'BHLB',
      'BRKR', 'BG', 'BDSI', 'BANC', 'BLKB', 'BECN', 'BFIN', 'BLDR'])
```

```
[96]: # Display the first 180 rows (3 DataFrames:'BWXT', 'BCPC', 'CAL')
      list(stockret_dict.values())[:3]
```

```
[96]: [    PERMNO      date TICKER       RET
      0     10220  20170131   BWXT  0.045088
      1     10220  20170228   BWXT  0.119306
      2     10220  20170331   BWXT  0.026916
      3     10220  20170428   BWXT  0.032983
      4     10220  20170531   BWXT -0.009355
      5     10220  20170630   BWXT  0.003086
      6     10220  20170731   BWXT  0.080615
      7     10220  20170831   BWXT  0.040812
      8     10220  20170929   BWXT  0.023757
      9     10220  20171031   BWXT  0.069618
      10    10220  20171130   BWXT  0.044059
      11    10220  20171229   BWXT -0.031385
      12    10220  20180131   BWXT  0.048768
      13    10220  20180228   BWXT -0.007566
      14    10220  20180329   BWXT  0.011595
      15    10220  20180430   BWXT  0.067212
      16    10220  20180531   BWXT -0.013274
      17    10220  20180629   BWXT -0.066227
      18    10220  20180731   BWXT  0.055199
      19    10220  20180831   BWXT -0.065085
      20    10220  20180928   BWXT  0.019896
      21    10220  20181031   BWXT -0.065238
      22    10220  20181130   BWXT -0.223743
      23    10220  20181231   BWXT -0.154578
      24    10220  20190131   BWXT  0.214230
      25    10220  20190228   BWXT  0.127747
      26    10220  20190329   BWXT -0.049666
      27    10220  20190430   BWXT  0.030657
      28    10220  20190531   BWXT -0.085910
      29    10220  20190628   BWXT  0.119467
      30    10220  20190731   BWXT  0.034741
      31    10220  20190830   BWXT  0.101280
```

```
32   10220  20190930   BWXT -0.033615
33   10220  20191031   BWXT  0.015557
34   10220  20191129   BWXT  0.037866
35   10220  20191231   BWXT  0.032430
36   10220  20200131   BWXT  0.024323
37   10220  20200228   BWXT -0.137600
38   10220  20200331   BWXT -0.108315
39   10220  20200430   BWXT  0.089304
40   10220  20200529   BWXT  0.182812
41   10220  20200630   BWXT -0.094774
42   10220  20200731   BWXT -0.037429
43   10220  20200831   BWXT  0.023478
44   10220  20200930   BWXT  0.012588
45   10220  20201030   BWXT -0.023087
46   10220  20201130   BWXT  0.037448
47   10220  20201231   BWXT  0.059775
48   10220  20210129   BWXT -0.105508
49   10220  20210226   BWXT  0.075853
50   10220  20210331   BWXT  0.140321
51   10220  20210430   BWXT  0.014862
52   10220  20210528   BWXT -0.062313
53   10220  20210630   BWXT -0.070675
54   10220  20210730   BWXT -0.011872
55   10220  20210831   BWXT  0.003657
56   10220  20210930   BWXT -0.062163
57   10220  20211029   BWXT  0.053472
58   10220  20211130   BWXT -0.155622
59   10220  20211231   BWXT  0.003774,      PERMNO      date TICKER       RET
60    10318  20170131   BCPC  0.015729
61    10318  20170228   BCPC  0.022642
62    10318  20170331   BCPC -0.054491
63    10318  20170428   BCPC -0.015287
64    10318  20170531   BCPC -0.030064
65    10318  20170630   BCPC -0.012830
66    10318  20170731   BCPC -0.001416
67    10318  20170831   BCPC -0.034021
68    10318  20170929   BCPC  0.084445
69    10318  20171031   BCPC  0.036905
70    10318  20171130   BCPC  0.035354
71    10318  20171229   BCPC -0.071617
72    10318  20180131   BCPC -0.019851
73    10318  20180228   BCPC -0.047468
74    10318  20180329   BCPC  0.086379
75    10318  20180430   BCPC  0.079388
76    10318  20180531   BCPC  0.092815
77    10318  20180629   BCPC  0.017733
78    10318  20180731   BCPC  0.021907
```

```
79     10318  20180831   BCPC  0.105693
80     10318  20180928   BCPC  0.010822
81     10318  20181031   BCPC -0.164511
82     10318  20181130   BCPC -0.074213
83     10318  20181231   BCPC -0.090888
84     10318  20190131   BCPC  0.059604
85     10318  20190228   BCPC  0.068779
86     10318  20190329   BCPC  0.045869
87     10318  20190430   BCPC  0.093858
88     10318  20190531   BCPC -0.106590
89     10318  20190628   BCPC  0.102327
90     10318  20190731   BCPC  0.026708
91     10318  20190830   BCPC -0.134938
92     10318  20190930   BCPC  0.117130
93     10318  20191031   BCPC  0.020365
94     10318  20191129   BCPC -0.015611
95     10318  20191231   BCPC  0.025294
96     10318  20200131   BCPC  0.062875
97     10318  20200228   BCPC -0.125532
98     10318  20200331   BCPC  0.045098
99     10318  20200430   BCPC -0.096029
100    10318  20200529   BCPC  0.127858
101    10318  20200630   BCPC -0.057526
102    10318  20200731   BCPC  0.056926
103    10318  20200831   BCPC -0.025534
104    10318  20200930   BCPC -0.000716
105    10318  20201030   BCPC  0.023763
106    10318  20201130   BCPC  0.037419
107    10318  20201231   BCPC  0.116790
108    10318  20210129   BCPC -0.071081
109    10318  20210226   BCPC  0.115201
110    10318  20210331   BCPC  0.050687
111    10318  20210430   BCPC  0.014193
112    10318  20210528   BCPC  0.029955
113    10318  20210630   BCPC  0.001985
114    10318  20210730   BCPC  0.027655
115    10318  20210831   BCPC  0.040996
116    10318  20210930   BCPC  0.033115
117    10318  20211029   BCPC  0.055284
118    10318  20211130   BCPC  0.032073
119    10318  20211231   BCPC  0.071139,      PERMNO      date TICKER       RET
120    10866  20170131    CAL -0.063071
121    10866  20170228    CAL -0.028618
122    10866  20170331    CAL -0.113157
123    10866  20170428    CAL  0.090840
124    10866  20170531    CAL -0.051353
125    10866  20170630    CAL  0.018654
```

```
126   10866  20170731    CAL -0.017999
127   10866  20170831    CAL -0.010997
128   10866  20170929    CAL  0.133803
129   10866  20171031    CAL -0.104522
130   10866  20171130    CAL  0.194292
131   10866  20171229    CAL  0.027880
132   10866  20180131    CAL -0.114695
133   10866  20180228    CAL -0.054993
134   10866  20180329    CAL  0.202071
135   10866  20180430    CAL -0.025893
136   10866  20180531    CAL  0.083410
137   10866  20180629    CAL -0.028201
138   10866  20180731    CAL -0.026170
139   10866  20180831    CAL  0.208719
140   10866  20180928    CAL -0.112401
141   10866  20181031    CAL -0.046291
142   10866  20181130    CAL -0.116082
143   10866  20181231    CAL -0.077076
144   10866  20190131    CAL  0.072224
145   10866  20190228    CAL  0.042225
146   10866  20190329    CAL -0.203859
147   10866  20190430    CAL  0.062373
148   10866  20190531    CAL -0.280976
149   10866  20190628    CAL  0.059915
150   10866  20190731    CAL -0.057229
151   10866  20190830    CAL  0.072950
152   10866  20190930    CAL  0.165261
153   10866  20191031    CAL -0.080735
154   10866  20191129    CAL  0.017193
155   10866  20191231    CAL  0.088168
156   10866  20200131    CAL -0.261053
157   10866  20200228    CAL -0.343020
158   10866  20200331    CAL -0.542931
159   10866  20200430    CAL  0.559615
160   10866  20200529    CAL -0.115906
161   10866  20200630    CAL  0.172943
162   10866  20200731    CAL -0.243405
163   10866  20200831    CAL  0.237718
164   10866  20200930    CAL  0.233035
165   10866  20201030    CAL -0.196653
166   10866  20201130    CAL  0.532552
167   10866  20201231    CAL  0.335599
168   10866  20210129    CAL -0.034505
169   10866  20210226    CAL  0.049636
170   10866  20210331    CAL  0.378941
171   10866  20210430    CAL  0.069266
172   10866  20210528    CAL  0.075933
```

```
173   10866  20210630    CAL  0.090909
174   10866  20210730    CAL -0.093441
175   10866  20210831    CAL -0.006063
176   10866  20210930    CAL -0.093534
177   10866  20211029    CAL  0.037804
178   10866  20211130    CAL  0.023851
179   10866  20211231    CAL -0.036425]
```

2.0.4 Upload Fama-French monthly risk factor data

- Fama French Risk Factors for 2017-2021 from the file ”FF-Factors-2017-2021.csv”

```
[73]: ff_factors = pd.read_csv('/content/drive/MyDrive/BA_870/HW/3/
       ↪FF-Factors-2017-2021.csv')
```

List varaibles in FF dataframe

```
[74]: ff_factors.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   dateff  60 non-null     int64
 1   mktrf   60 non-null     float64
 2   smb     60 non-null     float64
 3   hml     60 non-null     float64
 4   rf      60 non-null     float64
dtypes: float64(4), int64(1)
memory usage: 2.5 KB
```

Look at head and tail of dataframe

```
[75]: ff_factors.head()
```

```
[75]:      dateff   mktrf     smb     hml      rf
      0  20170131  0.0194 -0.0113 -0.0274  0.0004
      1  20170228  0.0357 -0.0204 -0.0167  0.0004
      2  20170331  0.0017  0.0113 -0.0333  0.0003
      3  20170428  0.0109  0.0072 -0.0213  0.0005
      4  20170531  0.0106 -0.0252 -0.0375  0.0006
```

```
[76]: ff_factors.tail()
```

```
[76]:       dateff   mktrf     smb     hml      rf
      55  20210831  0.0290 -0.0048 -0.0013  0.0000
```

```
56  20210930 -0.0437  0.0080  0.0509  0.0000
57  20211029  0.0665 -0.0228 -0.0044  0.0000
58  20211130 -0.0155 -0.0135 -0.0053  0.0000
59  20211231  0.0310 -0.0157  0.0323  0.0001
```

Rename date column to ”date” to match WRDS data ”date” column for 100 stocks

```
[77]: ff_factors.rename(columns={'dateff':'date'}, inplace=True)
      ff_factors.head()
```

```
[77]:        date   mktrf     smb     hml      rf
      0  20170131  0.0194 -0.0113 -0.0274  0.0004
      1  20170228  0.0357 -0.0204 -0.0167  0.0004
      2  20170331  0.0017  0.0113 -0.0333  0.0003
      3  20170428  0.0109  0.0072 -0.0213  0.0005
      4  20170531  0.0106 -0.0252 -0.0375  0.0006
```

2.0.5 Merge the 100 stock return data and Fama-French market data based on ”date”

- Then list head and tail of dataframe for BLDR

```
[78]: stockret_ff = {}
      for i in stockret_dict.keys():
        stockret_ff[i] = pd.merge(stockret_dict[i], ff_factors, on = 'date', how =␣
       ↪'outer')
```

```
[79]: stockret_ff['BLDR'].head()
```

```
[79]:   PERMNO      date TICKER       RET   mktrf     smb     hml      rf
      0  90720  20170131   BLDR -0.019143  0.0194 -0.0113 -0.0274  0.0004
      1  90720  20170228   BLDR  0.202602  0.0357 -0.0204 -0.0167  0.0004
      2  90720  20170331   BLDR  0.151468  0.0017  0.0113 -0.0333  0.0003
      3  90720  20170428   BLDR  0.074497  0.0109  0.0072 -0.0213  0.0005
      4  90720  20170531   BLDR -0.146783  0.0106 -0.0252 -0.0375  0.0006
```

```
[80]: stockret_ff['BLDR'].tail()
```

```
[80]:    PERMNO      date TICKER       RET   mktrf     smb     hml      rf
      55  90720  20210831   BLDR  0.197528  0.0290 -0.0048 -0.0013  0.0000
      56  90720  20210930   BLDR -0.029086 -0.0437  0.0080  0.0509  0.0000
      57  90720  20211029   BLDR  0.126208  0.0665 -0.0228 -0.0044  0.0000
      58  90720  20211130   BLDR  0.191694 -0.0155 -0.0135 -0.0053  0.0000
      59  90720  20211231   BLDR  0.234303  0.0310 -0.0157  0.0323  0.0001
```

2.1 Run OLS regression for 100 stocks (60 months) using FF 3-factor model:

- [Ret(stock)-Rf] = alpha + B1(RetMkt-Rf) + b2(SMB) + b3(HML) + e

[81]:
```
# Create a empty output dataframe
output = pd.DataFrame(columns = ['TICKER', 'R-squared', 'Adj. R-squared',␣
↪'const', 'mktrf', 'smb', 'hml'])
output
```

[81]:
```
Empty DataFrame
Columns: [TICKER, R-squared, Adj. R-squared, const, mktrf, smb, hml]
Index: []
```

[82]:
```
# Define a Linear Regression function for FF model
def ffmodel(data,i):
  y = data[i]["RET"] - data[i]["rf"]
  X = data[i][['mktrf' , 'smb' , 'hml']]
  # Use statsmodels
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y, X).fit()

  #return regression output
  return (i, model.rsquared, model.rsquared_adj, model.params[0], model.
↪params[1],model.params[2],model.params[3])

for i in stockret_ff.keys():
  output.loc[len(output.index)] = ffmodel(stockret_ff, i)

# Display the output regression statistics
output
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:117:
FutureWarning: In a future version of pandas all arguments of concat except for
the argument 'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

[82]:
```
   TICKER  R-squared  Adj. R-squared     const     mktrf       smb       hml
0    BWXT   0.319518        0.283063 -0.008452  1.035051 -0.255460 -0.025344
1    BCPC   0.151024        0.105543  0.007626  0.477501  0.231980  0.122664
2     CAL   0.491911        0.464692 -0.008616  2.010956  1.691033  1.102032
3      BC   0.614279        0.593615 -0.001431  1.446841  0.968595  0.372144
4     BAH   0.333477        0.297770  0.002267  0.764677 -0.241293 -0.486009
..    …        …               …         …        …         …         …
95   BANC   0.646651        0.627721 -0.001240  1.230841  1.623205  0.772896
96   BLKB   0.367581        0.333701 -0.007162  0.931270  0.530980 -0.047068
97   BECN   0.635610        0.616089 -0.007835  1.774667  0.088601  0.971480
98   BFIN   0.329431        0.293508 -0.003914  0.436431  0.576216  0.514933
```

```
99   BLDR   0.552135        0.528142  0.016584  2.091193  0.249562  0.377770

[100 rows x 7 columns]
```

[83]:

```
# Store the output into a csv file
output_file = output.to_csv('Assign3-Output.csv', index = False)
```

[ ]:

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

[86]:

```
!jupyter nbconvert --to pdf '/content/drive/MyDrive/BA_870/HW/3/
 ↪Assignment3_Ji_Qi.ipynb'
```

```
[NbConvertApp] Converting notebook
/content/drive/MyDrive/BA_870/HW/3/Assignment3_Ji_Qi.ipynb to pdf
[NbConvertApp] Writing 69290 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 75892 bytes to
/content/drive/MyDrive/BA_870/HW/3/Assignment3_Ji_Qi.pdf
```