How to download and clean Fama French 3 factor model data in Python

6/16/2019 — Written by DD

In this post we will download and clean the Fama/French 3 factors model data. First we will download the data like we did in the previous post.

To keep it brief we will execute the entire code at once.

```
import urllib.request
import zipfile
ff_url =
"https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/F-
F_Research_Data_Factors_CSV.zip"
# Download the file and save it
# We will name it fama_french.zip file
urllib.request.urlretrieve(ff_url,'fama_french.zip')
zip_file = zipfile.ZipFile('fama_french.zip', 'r')
# Next we extact the file data
# We will call it ff_factors.csv
zip_file.extractall()
# Make sure you close the file after extraction
zip_file.close()
import pandas as pd
ff_factors = pd.read_csv('F-F_Research_Data_Factors.csv',
skiprows = 3)
print(ff_factors.head())
```

```
##   Unnamed: 0    Mkt-RF       SMB       HML        RF
## 0     192607      2.96     -2.30     -2.87      0.22
## 1     192608      2.64     -1.40      4.19      0.25
## 2     192609      0.36     -1.32      0.01      0.23
```

```
## 3     192610     -3.24      0.04      0.51      0.32
## 4     192611      2.53     -0.20     -0.35      0.31
```

Now that the data has been downloaded lets look at the tail.

```
print(ff_factors.tail())
```

So the unwanted data is still in our dataframe. Lets confirm the rows are indeed 1114 on wards.

```
print(ff_factors.iloc[1112:1120],)
```

```
##                                Unnamed: 0    Mkt-RF    ...
HML        RF
## 1112                               201903      1.10    ...
-4.07      0.19
## 1113                               201904      3.96    ...
1.99      0.21
## 1114   Annual Factors: January-December        NaN    ...
NaN       NaN
## 1115                                 NaN    Mkt-RF    ...
HML        RF
## 1116                                1927     29.47    ...
-3.75      3.12
## 1117                                1928     35.39    ...
-6.15      3.56
## 1118                                1929    -19.54    ...
11.81      4.75
## 1119                                1930    -31.23    ...
-12.28      2.41
##
## [8 rows x 5 columns]
```

So we want to select only the first 1114 rows. Pandas as a built in function to do that. Its called `nrows()` . This lets us select the number of rows. So lets load the data again.

```
ff_factors = pd.read_csv('F-F_Research_Data_Factors.csv',
skiprows = 3,
nrows = 1114)
print(ff_factors.tail())
```

```
##       Unnamed: 0  Mkt-RF   SMB   HML    RF
## 1109      201812   -9.55 -2.58 -1.51  0.19
## 1110      201901    8.41  3.02 -0.60  0.21
## 1111      201902    3.40  2.02 -2.84  0.18
## 1112      201903    1.10 -3.15 -4.07  0.19
## 1113      201904    3.96 -1.69  1.99  0.21
```

We can see that the unwanted data is gone and we have April 2019 as our last data point. Next we want to use the first column as our index. So we will specify that.

```
ff_factors = pd.read_csv('F-F_Research_Data_Factors.csv',
skiprows = 3,
nrows = 1114, index_col = 0)
print(ff_factors.tail())
```

```
##         Mkt-RF   SMB   HML    RF
## 201812   -9.55 -2.58 -1.51  0.19
## 201901    8.41  3.02 -0.60  0.21
## 201902    3.40  2.02 -2.84  0.18
## 201903    1.10 -3.15 -4.07  0.19
## 201904    3.96 -1.69  1.99  0.21
```

Next we will convert our index into a date object.

```
ff_factors.index = pd.to_datetime(ff_factors.index, format=
'%Y%m')
print(ff_factors.tail())
```

```
##             Mkt-RF   SMB   HML    RF
## 2018-12-01   -9.55 -2.58 -1.51  0.19

## 2019-01-01    8.41  3.02 -0.60  0.21
## 2019-02-01    3.40  2.02 -2.84  0.18
## 2019-03-01    1.10 -3.15 -4.07  0.19
## 2019-04-01    3.96 -1.69  1.99  0.21
```

We have the same issue as before. The data starts at the first of the month. We can change it to the last of the month using the pd.offset method.

```
ff_factors.index = ff_factors.index + pd.offsets.MonthEnd()
print(ff_factors.tail())
```

```
##             Mkt-RF   SMB   HML    RF
## 2018-12-31   -9.55 -2.58 -1.51  0.19
## 2019-01-31    8.41  3.02 -0.60  0.21
## 2019-02-28    3.40  2.02 -2.84  0.18
## 2019-03-31    1.10 -3.15 -4.07  0.19
## 2019-04-30    3.96 -1.69  1.99  0.21
```

Now the data look much better. As our last steps lets convert the numbers into decimals. We will use a simple lambda function to that.

```
ff_factors = ff_factors.apply(lambda x: x/ 100)
ff_factors.tail()
```

We now have the data in the format that is useful to use. Below we will post all the steps needed to clean this data. You can write a script using the code below, which will automatically do this process for you.

```
import urllib.request
import zipfile
ff_url =
"https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/F-
F_Research_Data_Factors_CSV.zip"
# Download the file and save it
```

```
# We will name it fama_french.zip file
urllib.request.urlretrieve(ff_url,'fama_french.zip')

zip_file = zipfile.ZipFile('fama_french.zip', 'r')
# Next we extact the file data
# We will call it ff_factors.csv
zip_file.extractall()
# Make sure you close the file after extraction
zip_file.close()
import pandas as pd
ff_factors = pd.read_csv('F-F_Research_Data_Factors.csv',
skiprows = 3, nrows = 1114, index_col = 0)
ff_factors.index = pd.to_datetime(ff_factors.index, format=
'%Y%m')
ff_factors.index = ff_factors.index + pd.offsets.MonthEnd()
ff_factors = ff_factors.apply(lambda x: x/ 100)
```

READ OTHER POSTS

← Factor Based Analysis

How to download and clean Fama French 3 factor model data in R →

› coding finance