Use Python to download SEC filings on EDGAR

Hardware and Software Requirements

As a guideline, I run the code in this report on a virtual machine with Ubuntu 16.04 Linux on my windows platform Laptop

Please check out the following link and follow the YouTube guide to set up your Ubuntu 16.04 Linux platform on your windows Laptop.

https://www.youtube.com/watch?v=wHxvu t-wAc

The software can be found and downloaded:

https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstatio n_player/12_0

To bulk download the files what you are interested, it would be more convenient and stable by using Linux platform, So I highly recommend you bulk download the 10K,10Q as well as 8K files by using python jupyter notebook on Linux platform.

One step I want to point out is that it is much better for the Linux beginner to partition the disk as below.

For simplicity, we partition 3 disks:

One is boot, for boot, if you have enough disk space, please allocate more than 100 GB for boot. One is swap, basically, it is allocated twice size with your RAM, if you RAM is 16 GB, please allocate 32 GB for swap

One is data, all the left disk space are allocated to data disk.

After finish installation of your Ubuntu platform, you may need to set up static IP address:

First thing you need to do is to enable SSH in Ubuntu 16.04.

sudo apt-get install openssh-server sudo nano /etc/ssh/sshd_config

Change Permit RootLogin to yes

Then go to /etc/network/interfaces folder to set up the static IP address by using the following code:

sudo nano /etc/network/interfaces

And then replace everything with the following content:

auto eth0

iface eth0 inet static

address 192.168.107.133

gateway 192.168.107.2 // you can get the infor from the properties of your network

netmask 255.255.255.0

dns-nameservers 8.8.8.8

And then, go to /etc/NetworkManager/NetworkManager.conf folder

sudo nano /etc/NetworkManager/NetworkManager.conf

[if updown] managed = false // change false to true

Now, Ubuntu is ready to install software for web scraping.

Here, I am going to install anaconda3.

The software can be found and downloaded:

https://www.continuum.io/downloads

For simplicity, please download anaconda3.sh file to your home directory:

In my case, the home directory is /home/maohuaxie.

And then run the code as below:

bash anaconda3.sh

And then you will get notice to export the path export PATH=/home/maohuaxie/anaconda3/bin:\$PATH >> ~/.bashrc And then run:

source ~/.bashrc

I write the following Python program to pull out the data sets containing Cik, Sticker as well as the file path information. This program borrows from Kai Chen's blog. Please look at his blog page.

http://kaikaichen.com/?p=59

Please note: my program stores all paths in a SQLite database. I personally like the lightweight database product very much.

The following is my code to install SQLite. Please note that the directory where the SQLite is installed is home directory. You can change to any directory if you want.

sudo apt-get install sqlite3 libsqlite3-dev

After installation check installation, sqlite terminal will give you a prompt and version.

```
Manbuaxtegnachuaxtevrtrual-machine:-5 ls
Anaconda2-4,2.0-linux-x86-6.4.sh rouse ship_info.csv bownloads examples.desktop Music Pictures python-scraping spark-2.1.0-bin-hadoop2.7.tgz Templates anaconda3-4,2.0-linux-x86-64.sh Documents edgar_idx.db GSU mysample.csv Public spark Spark.lpynb Unititled.lpynb machuaxtegnachuaxtegnachuaxtevrtrual-machine:-5 sqlite3 Sqlite vs. 13.13.0 2016-05-18 103.73.00 Enter ".help" for usage hints.
Connected to a transient in-menory database.
Use ".open FILENARE" to reopen on a persistent database. Sqlite> - squire> - open edgar_idx.db sqlite> - databases seq name file

O nain /hone/machuaxie/edgar_idx.db sqlite> - tables idx

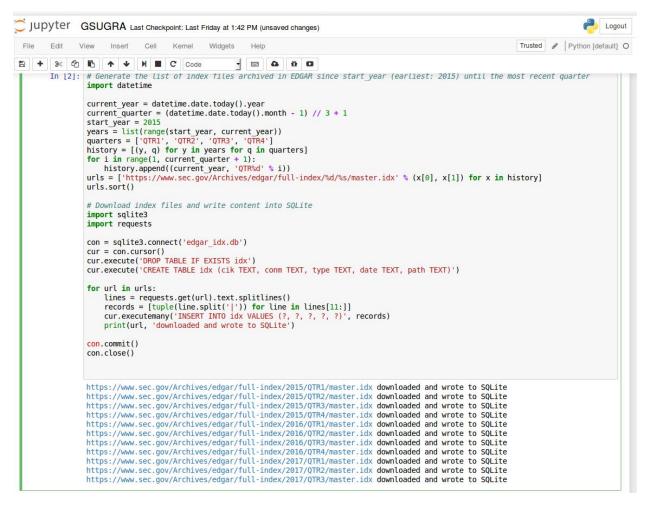
| Mone/machuaxie/edgar_idx.db | sqlite> - tables idx
| Mone/machuaxie/edgar_idx.db | sqlite> - tables idx
| Mone/machuaxie/edgar_idx.db | sqlite> - tables idx
```

Please google documentations of SQLite, Pandas. If you have any installation problems.

```
maohuaxte@maohuaxie
maohuaxie@maohuaxie-vtrtual-machine:~$ jupyter notebook
[I 10:48:48:389 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
[I 10:48:48:389 NotebookApp] [mb_conda_kernels] enabled, 2 kernels found
[I 10:48:48:389 NotebookApp] [mb_conda_kernels] enabled
[I 10:48:48:389 NotebookApp] [nb_conda] enabled
[I 10:48:49:367 NotebookApp] [nb_conda] enabled
[I 10:48:49:627 NotebookApp] [nb_conda] enabled
[I 10:48:49:627 NotebookApp] / nbpresent HTML export ENABLED
[W 10:48:49:775 NotebookApp] / nbpresent HTML export ENABLED: No module named 'nbbrowserpdf'
[I 10:48:49:795 NotebookApp] Serving notebooks from local directory: /home/maohuaxie
[I 10:48:49:795 NotebookApp] 0 active kernels
[I 10:48:49:795 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=ea3ccf148aa5dd98b9b48d2701861c4643d7133ab6cc0e22
[I 10:48:49:795 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

Copy/paste this URL into your browser when you connect for the first time, to login with a token:
    http://localhost:8888/?token=ea3ccf148aa5dd98b9b48d2701861c4643d7133ab6cc0e22
[I 10:48:49:795 NotebookApp] Accepting one-time-token-authenticated connection from 127.0.0.1
```

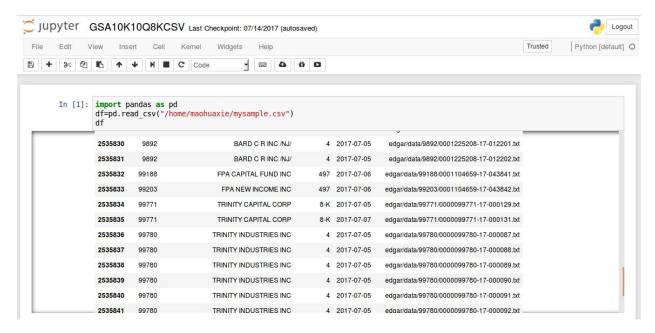
Please note: edgar_idx.db will be created right after running the following code (GSUGRA). The index database includes all types of filings (e.g., 10-K, 10-Q and 8-K).



After we got the edgar_idx.db, we select all data and export them into mysample.csv file with the following code:

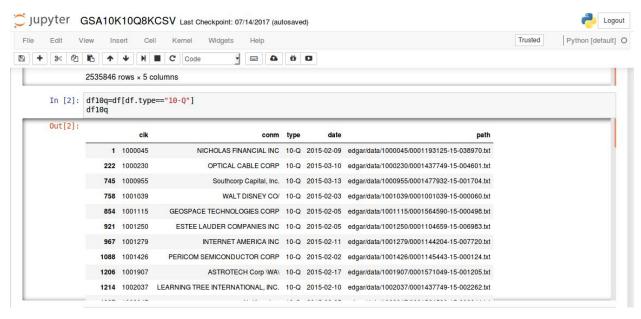
sqlite3 -header -csv edgar_idx.db "select * from idx;" > mysample.csv

```
maohuaxie@maohuaxie-virtual-machine:-$ sqlite3 -header -csv edgar_idx.db "select * from idx;" > sample.csv
maohuaxie@maohuaxie-virtual-machine:-$ ls
Anaconda2-4.2.0-Linux-x86_64.sh cruise_ship_info.csv Downloads examples.desktop Music Pictures python-scraping
anaconda3 Desktop edgar_idx.db GSU mysample.csv Public sample.csv
Anaconda3-4.2.0-Linux-x86_64.sh Documents edgar_idx.dta GSUGRA.ipynb pandas python Spark
maohuaxie@maohuaxie-virtual-machine:-$
```

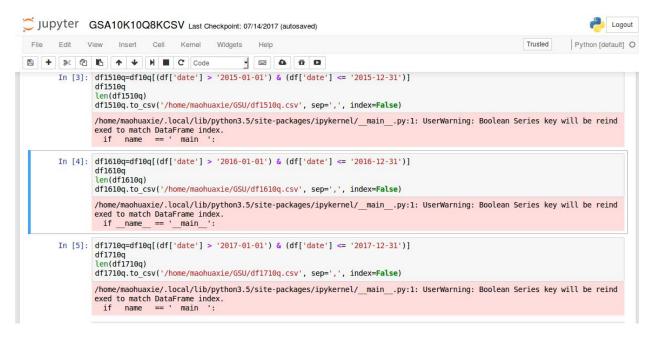


When we have mysample.csv file. We can use read_csv to read it into a DataFrame:

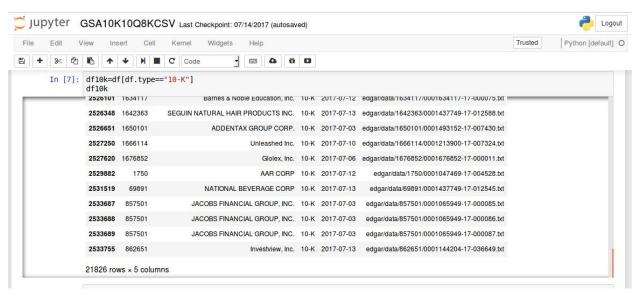
As we can see here, mysample.csv file contains Cik, Company name and file path information.



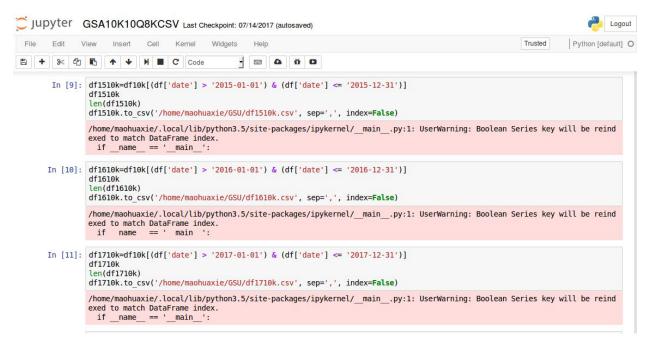
10Q file was filtered out here.



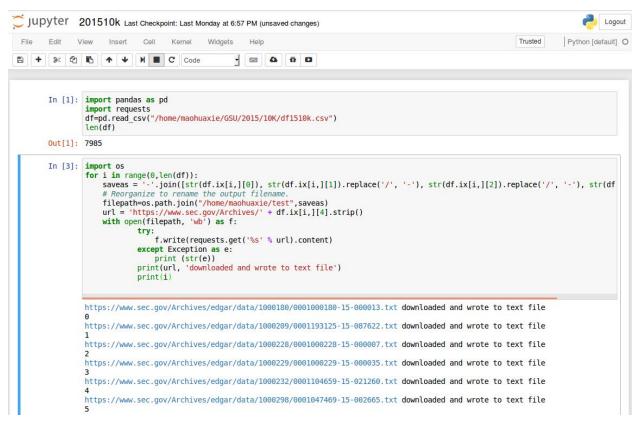
df1510q.csv, df1610q.csv and df1710q.csv files were generated by filtering over year and writing them to a csv file respectively.



10K file was filtered out here.



df1510k.csv, df1610k.csv and df1710k.csv files were generated by filtering over year and writing them to a csv file respectively.

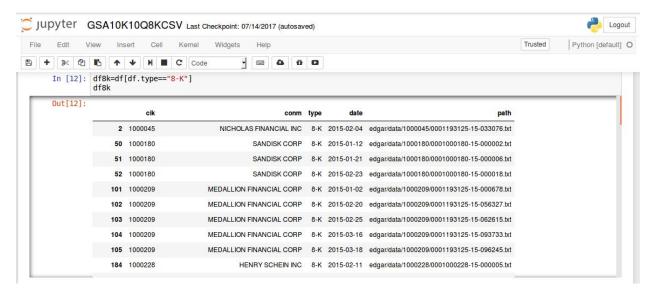


After run 201510K.ipynb, we will get the txt files in (/home/maohuaxie/test) folder as the following shown. Process other files as do for df1510k.csv.

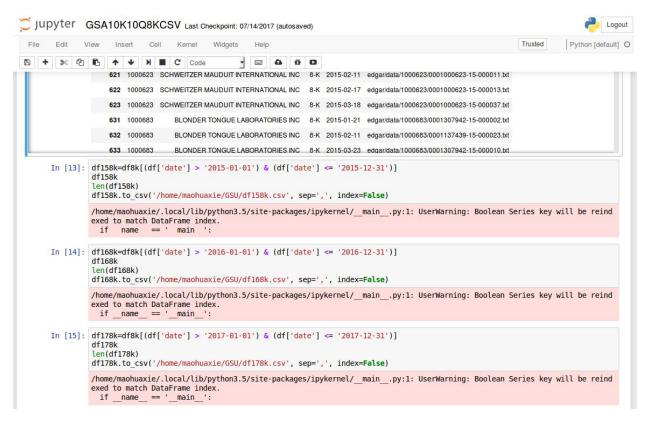
Please note: filepath=os.path.join("/home/maohuaxie/test") can be changed to as we need(e.g

"/home/maohuaxie/GSU/2015/10K")

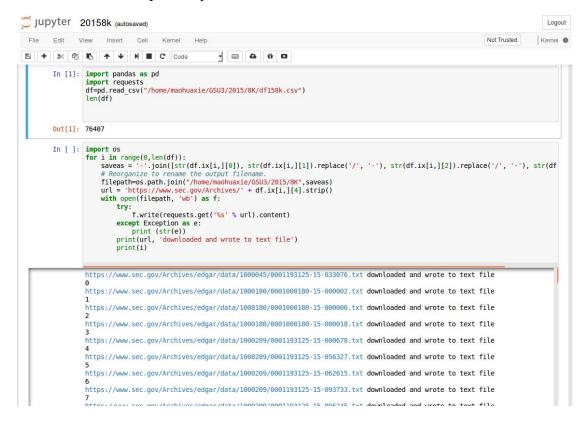
```
maohuaxie@maohuaxie-virtual-machine:~$ cd test
maohuaxie@maohuaxie-virtual-machine:~/test$ ls
1000180-SANDISK CORP-10-K-2015-02-10
1000209-MEDALLION FINANCIAL CORP-10-K-2015-03-11
1000228-HENRY SCHEIN INC-10-K-2015-02-11
1000228-HENRY SCHEIN INC-10-K-2015-02-17
10000232-KENTUCKY BANCSHARES INC -KY--10-K-2015-03-20
1000693-WATERS CORP -DE--10-K-2015-02-27
```



8K file was filtered out here.



df158k.csv, df168k.csv and df178k.csv files were generated by filtering over year and writing them to a csv file respectively.



For 8K files, I have added some code to catch the retrieving errors and save data to a specify directory.

After all the steps done, we need classify the data by quarter, we can perform this by using Linux shell.

mv *-01* /home/maohuaxie/GSU/2015/QTR1/8K //The files in current directory will go destination directory(/home/maohuaxie/GSU/2015/QTR1/8K)

To make directory, please use this code: mkdir -p /home/maohuaxie/GSU/2015/QTR1/8K

mv *-02* /home/maohuaxie/GSU/2015/QTR1/8K

mv *-03* /home/maohuaxie/GSU/2015/QTR1/8K

mv *-04* /home/maohuaxie/GSU/2015/QTR2/8K

mv *-05* /home/maohuaxie/GSU/2015/QTR2/8K

mv *-06* /home/maohuaxie/GSU/2015/QTR2/8K

mv *-07* /home/maohuaxie/GSU/2015/QTR3/8K

mv *-08* /home/maohuaxie/GSU/2015/QTR3/8K

mv *-09* /home/maohuaxie/GSU/2015/QTR3/8K

mv *-10* /home/maohuaxie/GSU/2015/QTR4/8K

mv *-11* /home/maohuaxie/GSU/2015/QTR4/8K

mv *-12* /home/maohuaxie/GSU/2015/QTR4/8K

There may have more easy way to do this, at the writing time, I only could work out this in a hard way.