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_workstation_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.

```
maohuaxie@maohuaxie-virtual-machine:~$ ls
Anaconda2-4.2.0-Linux-x86_64.sh  cruise_ship_info.csv  Downloads      examples.desktop  Music         Pictures  python-scraping            spark-2.1.0-bin-hadoop2.7.tgz  Templates
anaconda3                        Desktop               edgar_idx.db   GSU               mysample.csv  Public    Spark                      Spark.ipynb                    Untitled.ipynb
Anaconda3-4.2.0-Linux-x86_64.sh  Documents             edgar_idx.dta  GSUGRA.ipynb      pandas        python    spark-2.1.0-bin-hadoop2.7  spark-warehouse                Videos
maohuaxie@maohuaxie-virtual-machine:~$ sqlite3
SQLite version 3.13.0 2016-05-18 10:57:30
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open edgar_idx.db
sqlite> .databases
seq  name             file
---  ---------------  ----------------------------------------------------------
0    main             /home/maohuaxie/edgar_idx.db
sqlite> .tables
idx
```

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

```
maohuaxie@maohuaxie-virtual-machine:~$ pwd
/home/maohuaxie
maohuaxie@maohuaxie-virtual-machine:~$ jupyter notebook
[I 16:48:48.389 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
[I 16:48:48.394 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook_cookie_secret
[I 16:48:49.619 NotebookApp] [nb_anacondacloud] enabled
[I 16:48:49.627 NotebookApp] [nb_conda] enabled
[I 16:48:49.775 NotebookApp] ✓ nbpresent HTML export ENABLED
[W 16:48:49.775 NotebookApp] ✗ nbpresent PDF export DISABLED: No module named 'nbbrowserpdf'
[I 16:48:49.790 NotebookApp] Serving notebooks from local directory: /home/maohuaxie
[I 16:48:49.790 NotebookApp] 0 active kernels
[I 16:48:49.790 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=ea3ccf148aa5dd98b9b48d2701861c4643d7133ab6cc0e22
[I 16:48:49.790 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:48:49.793 NotebookApp]

    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 16:48:52.693 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).

jupyter GSUGRA Last Checkpoint: Last Friday at 1:42 PM (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

In [2]:
```python
# Generate the list of index files archived in EDGAR since start_year (earliest: 2015) until the most recent quarter
import datetime

current_year = datetime.date.today().year
current_quarter = (datetime.date.today().month - 1) // 3 + 1
start_year = 2015
years = list(range(start_year, current_year))
quarters = ['QTR1', 'QTR2', 'QTR3', 'QTR4']
history = [(y, q) for y in years for q in quarters]
for i in range(1, current_quarter + 1):
    history.append((current_year, 'QTR%d' % i))
urls = ['https://www.sec.gov/Archives/edgar/full-index/%d/%s/master.idx' % (x[0], x[1]) for x in history]
urls.sort()

# Download index files and write content into SQLite
import sqlite3
import requests

con = sqlite3.connect('edgar_idx.db')
cur = con.cursor()
cur.execute('DROP TABLE IF EXISTS idx')
cur.execute('CREATE TABLE idx (cik TEXT, conm TEXT, type TEXT, date TEXT, path TEXT)')

for url in urls:
    lines = requests.get(url).text.splitlines()
    records = [tuple(line.split('|')) for line in lines[11:]]
    cur.executemany('INSERT INTO idx VALUES (?, ?, ?, ?, ?)', records)
    print(url, 'downloaded and wrote to SQLite')

con.commit()
con.close()
```

```
https://www.sec.gov/Archives/edgar/full-index/2015/QTR1/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2015/QTR2/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2015/QTR3/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2015/QTR4/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2016/QTR1/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2016/QTR2/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2016/QTR3/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2016/QTR4/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2017/QTR1/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2017/QTR2/master.idx downloaded and wrote to SQLite
https://www.sec.gov/Archives/edgar/full-index/2017/QTR3/master.idx downloaded and wrote to SQLite
```

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:~$
```

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python [default]

Code

In [1]:
```
import pandas as pd
df=pd.read_csv("/home/maohuaxie/mysample.csv")
df
```

2535830	9892	BARD C R INC /NJ/	4	2017-07-05	edgar/data/9892/0001225208-17-012201.txt
2535831	9892	BARD C R INC /NJ/	4	2017-07-05	edgar/data/9892/0001225208-17-012202.txt
2535832	99188	FPA CAPITAL FUND INC	497	2017-07-06	edgar/data/99188/0001104659-17-043841.txt
2535833	99203	FPA NEW INCOME INC	497	2017-07-06	edgar/data/99203/0001104659-17-043842.txt
2535834	99771	TRINITY CAPITAL CORP	8-K	2017-07-05	edgar/data/99771/0000099771-17-000129.txt
2535835	99771	TRINITY CAPITAL CORP	8-K	2017-07-07	edgar/data/99771/0000099771-17-000131.txt
2535836	99780	TRINITY INDUSTRIES INC	4	2017-07-05	edgar/data/99780/0000099780-17-000087.txt
2535837	99780	TRINITY INDUSTRIES INC	4	2017-07-05	edgar/data/99780/0000099780-17-000088.txt
2535838	99780	TRINITY INDUSTRIES INC	4	2017-07-05	edgar/data/99780/0000099780-17-000089.txt
2535839	99780	TRINITY INDUSTRIES INC	4	2017-07-05	edgar/data/99780/0000099780-17-000090.txt
2535840	99780	TRINITY INDUSTRIES INC	4	2017-07-05	edgar/data/99780/0000099780-17-000091.txt
2535841	99780	TRINITY INDUSTRIES INC	4	2017-07-05	edgar/data/99780/0000099780-17-000092.txt

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.

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python [default]

Code

2535846 rows × 5 columns

In [2]:
```
df10q=df[df.type=="10-Q"]
df10q
```

Out[2]:

	cik	conm	type	date	path
1	1000045	NICHOLAS FINANCIAL INC	10-Q	2015-02-09	edgar/data/1000045/0001193125-15-038970.txt
222	1000230	OPTICAL CABLE CORP	10-Q	2015-03-10	edgar/data/1000230/0001437749-15-004601.txt
745	1000955	Southcorp Capital, Inc.	10-Q	2015-03-13	edgar/data/1000955/0001477932-15-001704.txt
758	1001039	WALT DISNEY CO/	10-Q	2015-02-03	edgar/data/1001039/0001001039-15-000060.txt
854	1001115	GEOSPACE TECHNOLOGIES CORP	10-Q	2015-02-05	edgar/data/1001115/0001564590-15-000498.txt
921	1001250	ESTEE LAUDER COMPANIES INC	10-Q	2015-02-05	edgar/data/1001250/0001104659-15-006983.txt
967	1001279	INTERNET AMERICA INC	10-Q	2015-02-11	edgar/data/1001279/0001144204-15-007720.txt
1088	1001426	PERICOM SEMICONDUCTOR CORP	10-Q	2015-02-02	edgar/data/1001426/0001145443-15-000124.txt
1206	1001907	ASTROTECH Corp \WA\	10-Q	2015-02-17	edgar/data/1001907/0001571049-15-001205.txt
1214	1002037	LEARNING TREE INTERNATIONAL, INC.	10-Q	2015-02-10	edgar/data/1002037/0001437749-15-002262.txt

10Q file was filtered out here.

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Code

In [3]:
```
df1510q=df10q[(df['date'] > '2015-01-01') & (df['date'] <= '2015-12-31')]
df1510q
len(df1510q)
df1510q.to_csv('/home/maohuaxie/GSU/df1510q.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

In [4]:
```
df1610q=df10q[(df['date'] > '2016-01-01') & (df['date'] <= '2016-12-31')]
df1610q
len(df1610q)
df1610q.to_csv('/home/maohuaxie/GSU/df1610q.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

In [5]:
```
df1710q=df10q[(df['date'] > '2017-01-01') & (df['date'] <= '2017-12-31')]
df1710q
len(df1710q)
df1710q.to_csv('/home/maohuaxie/GSU/df1710q.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

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

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Code

In [7]:
```
df10k=df[df.type=="10-K"]
df10k
```

2526101	1634117	Barnes & Noble Education, Inc.	10-K	2017-07-12	edgar/data/1634117/0001634117-17-000075.txt
2526348	1642363	SEGUIN NATURAL HAIR PRODUCTS INC.	10-K	2017-07-13	edgar/data/1642363/0001437749-17-012588.txt
2526651	1650101	ADDENTAX GROUP CORP.	10-K	2017-07-03	edgar/data/1650101/0001493152-17-007430.txt
2527250	1666114	Unleashed Inc.	10-K	2017-07-10	edgar/data/1666114/0001213900-17-007324.txt
2527620	1676852	Glolex, Inc.	10-K	2017-07-06	edgar/data/1676852/0001676852-17-000011.txt
2529882	1750	AAR CORP	10-K	2017-07-12	edgar/data/1750/0001047469-17-004528.txt
2531519	69891	NATIONAL BEVERAGE CORP	10-K	2017-07-13	edgar/data/69891/0001437749-17-012545.txt
2533687	857501	JACOBS FINANCIAL GROUP, INC.	10-K	2017-07-03	edgar/data/857501/0001065949-17-000085.txt
2533688	857501	JACOBS FINANCIAL GROUP, INC.	10-K	2017-07-03	edgar/data/857501/0001065949-17-000086.txt
2533689	857501	JACOBS FINANCIAL GROUP, INC.	10-K	2017-07-03	edgar/data/857501/0001065949-17-000087.txt
2533755	862651	Investview, Inc.	10-K	2017-07-13	edgar/data/862651/0001144204-17-036649.txt

21826 rows × 5 columns

10K file was filtered out here.

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Code

In [9]:
```
df1510k=df10k[(df['date'] > '2015-01-01') & (df['date'] <= '2015-12-31')]
df1510k
len(df1510k)
df1510k.to_csv('/home/maohuaxie/GSU/df1510k.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

In [10]:
```
df1610k=df10k[(df['date'] > '2016-01-01') & (df['date'] <= '2016-12-31')]
df1610k
len(df1610k)
df1610k.to_csv('/home/maohuaxie/GSU/df1610k.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

In [11]:
```
df1710k=df10k[(df['date'] > '2017-01-01') & (df['date'] <= '2017-12-31')]
df1710k
len(df1710k)
df1710k.to_csv('/home/maohuaxie/GSU/df1710k.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

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

jupyter 201510k Last Checkpoint: Last Monday at 6:57 PM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Code

In [1]:
```
import pandas as pd
import requests
df=pd.read_csv("/home/maohuaxie/GSU/2015/10K/df1510k.csv")
len(df)
```

Out[1]: 7985

In [3]:
```
import os
for i in range(0,len(df)):
    saveas = '-'.join([str(df.ix[i,][0]), str(df.ix[i,][1]).replace('/', '-'), str(df.ix[i,][2]).replace('/', '-'), str(df
    # Reorganize to rename the output filename.
    filepath=os.path.join("/home/maohuaxie/test",saveas)
    url = 'https://www.sec.gov/Archives/' + df.ix[i,][4].strip()
    with open(filepath, 'wb') as f:
            try:
                f.write(requests.get('%s' % url).content)
            except Exception as e:
                print (str(e))
            print(url, 'downloaded and wrote to text file')
            print(i)
```

```
https://www.sec.gov/Archives/edgar/data/1000180/0001000180-15-000013.txt downloaded and wrote to text file
0
https://www.sec.gov/Archives/edgar/data/1000209/0001193125-15-087622.txt downloaded and wrote to text file
1
https://www.sec.gov/Archives/edgar/data/1000228/0001000228-15-000007.txt downloaded and wrote to text file
2
https://www.sec.gov/Archives/edgar/data/1000229/0001000229-15-000035.txt downloaded and wrote to text file
3
https://www.sec.gov/Archives/edgar/data/1000232/0001104659-15-021260.txt downloaded and wrote to text file
4
https://www.sec.gov/Archives/edgar/data/1000298/0001047469-15-002665.txt downloaded and wrote to text file
5
```

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                1000298-IMPAC MORTGAGE HOLDINGS INC-10-K-2015-03-25
1000209-MEDALLION FINANCIAL CORP-10-K-2015-03-11    1000623-SCHWEITZER MAUDUIT INTERNATIONAL INC-10-K-2015-02-27
1000228-HENRY SCHEIN INC-10-K-2015-02-11            1000683-BLONDER TONGUE LABORATORIES INC-10-K-2015-03-31
1000229-CORE LABORATORIES N V-10-K-2015-02-17       1000694-NOVAVAX INC-10-K-2015-02-27
1000232-KENTUCKY BANCSHARES INC -KY--10-K-2015-03-20 1000697-WATERS CORP -DE--10-K-2015-02-27
```

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

In [12]:
```
df8k=df[df.type=="8-K"]
df8k
```

Out[12]:

	cik	conm	type	date	path
2	1000045	NICHOLAS FINANCIAL INC	8-K	2015-02-04	edgar/data/1000045/0001193125-15-033076.txt
50	1000180	SANDISK CORP	8-K	2015-01-12	edgar/data/1000180/0001000180-15-000002.txt
51	1000180	SANDISK CORP	8-K	2015-01-21	edgar/data/1000180/0001000180-15-000006.txt
52	1000180	SANDISK CORP	8-K	2015-02-23	edgar/data/1000180/0001000180-15-000018.txt
101	1000209	MEDALLION FINANCIAL CORP	8-K	2015-01-02	edgar/data/1000209/0001193125-15-000678.txt
102	1000209	MEDALLION FINANCIAL CORP	8-K	2015-02-20	edgar/data/1000209/0001193125-15-056327.txt
103	1000209	MEDALLION FINANCIAL CORP	8-K	2015-02-25	edgar/data/1000209/0001193125-15-062615.txt
104	1000209	MEDALLION FINANCIAL CORP	8-K	2015-03-16	edgar/data/1000209/0001193125-15-093733.txt
105	1000209	MEDALLION FINANCIAL CORP	8-K	2015-03-18	edgar/data/1000209/0001193125-15-096245.txt
184	1000228	HENRY SCHEIN INC	8-K	2015-02-11	edgar/data/1000228/0001000228-15-000005.txt

8K file was filtered out here.

jupyter GSA10K10Q8KCSV Last Checkpoint: 07/14/2017 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Code

621	1000623	SCHWEITZER MAUDUIT INTERNATIONAL INC	8-K	2015-02-11	edgar/data/1000623/0001000623-15-000011.txt
622	1000623	SCHWEITZER MAUDUIT INTERNATIONAL INC	8-K	2015-02-17	edgar/data/1000623/0001000623-15-000013.txt
623	1000623	SCHWEITZER MAUDUIT INTERNATIONAL INC	8-K	2015-03-18	edgar/data/1000623/0001000623-15-000037.txt
631	1000683	BLONDER TONGUE LABORATORIES INC	8-K	2015-01-21	edgar/data/1000683/0001307942-15-000002.txt
632	1000683	BLONDER TONGUE LABORATORIES INC	8-K	2015-02-11	edgar/data/1000683/0001137439-15-000023.txt
633	1000683	BLONDER TONGUE LABORATORIES INC	8-K	2015-03-23	edgar/data/1000683/0001307942-15-000010.txt

In [13]:
```
df158k=df8k[(df['date'] > '2015-01-01') & (df['date'] <= '2015-12-31')]
df158k
len(df158k)
df158k.to_csv('/home/maohuaxie/GSU/df158k.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

In [14]:
```
df168k=df8k[(df['date'] > '2016-01-01') & (df['date'] <= '2016-12-31')]
df168k
len(df168k)
df168k.to_csv('/home/maohuaxie/GSU/df168k.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

In [15]:
```
df178k=df8k[(df['date'] > '2017-01-01') & (df['date'] <= '2017-12-31')]
df178k
len(df178k)
df178k.to_csv('/home/maohuaxie/GSU/df178k.csv', sep=',', index=False)
```

```
/home/maohuaxie/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: UserWarning: Boolean Series key will be reind
exed to match DataFrame index.
  if __name__ == '__main__':
```

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

jupyter 20158k (autosaved)

Logout

File Edit View Insert Cell Kernel Help Not Trusted Kernel

Code

In [1]:
```
import pandas as pd
import requests
df=pd.read_csv("/home/maohuaxie/GSU3/2015/8K/df158k.csv")
len(df)
```

Out[1]: 76407

In [ ]:
```
import os
for i in range(0,len(df)):
    saveas = '-'.join([str(df.ix[i,][0]), str(df.ix[i,][1]).replace('/', '-'), str(df.ix[i,][2]).replace('/', '-'), str(df
    # Reorganize to rename the output filename.
    filepath=os.path.join("/home/maohuaxie/GSU3/2015/8K",saveas)
    url = 'https://www.sec.gov/Archives/' + df.ix[i,][4].strip()
    with open(filepath, 'wb') as f:
        try:
            f.write(requests.get('%s' % url).content)
        except Exception as e:
            print (str(e))
        print(url, 'downloaded and wrote to text file')
        print(i)
```

```
https://www.sec.gov/Archives/edgar/data/1000045/0001193125-15-033076.txt downloaded and wrote to text file
0
https://www.sec.gov/Archives/edgar/data/1000180/0001000180-15-000002.txt downloaded and wrote to text file
1
https://www.sec.gov/Archives/edgar/data/1000180/0001000180-15-000006.txt downloaded and wrote to text file
2
https://www.sec.gov/Archives/edgar/data/1000180/0001000180-15-000018.txt downloaded and wrote to text file
3
https://www.sec.gov/Archives/edgar/data/1000209/0001193125-15-000678.txt downloaded and wrote to text file
4
https://www.sec.gov/Archives/edgar/data/1000209/0001193125-15-056327.txt downloaded and wrote to text file
5
https://www.sec.gov/Archives/edgar/data/1000209/0001193125-15-062615.txt downloaded and wrote to text file
6
https://www.sec.gov/Archives/edgar/data/1000209/0001193125-15-093733.txt downloaded and wrote to text file
7
```

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.