q3

February 7, 2019

In [7]:
```
#Lisa He
#Obtain the time before we open the time:
import time
start_time = time.clock()
fs_file = open('/Users/Lisa/Documents/BigData/hw1/compustat_annual_updates_fundamentals_
variables_of_interest = ['datadate', 'aco', 'acominc', 'act', 'ao', 'aocipen', 'aodo','a
single_line =fs_file.readline()

variables = single_line.split(",")
index =0
vti = dict()
for variable in variables:
    variable = variable.strip()
    if variable in variables_of_interest:
        vti[variable]=index
        #print(variable+"\t"+str(index))
    index+=1

for var in variables_of_interest:
    print(var+"\t"+str(vti[var]))


#Create a list of lists:
ll = list()
#Create a list of dictionaries:

line_num=1
for line in fs_file:
    #we would like to skip the first line (the header):
    if line_num==1:
        line_num += 1
        continue
    vars = line.split(",")
    ll.append([vars[1],vars[39],vars[41],vars[54],vars[65],vars[68],vars[70],vars[72],va
    line_num+=1
fs_file.close()
end_time = time.clock()
```

```
        print ("Read line by line and storing into list of lists takes "+str(round(end_time - st
```

```
datadate        1
aco         39
acominc         41
act         54
ao         65
aocipen         68
aodo         70
aoloch         72
aox         73
ap         74
aqc         80
at         93
bkvlps         109
caps         115
capx         116
capxv         117
ceq         122
ceql         123
ceqt         124
ch         133
Read line by line and storing into list of lists takes 5.658268 seconds
```

```
In [9]: #Lisa He
        #Obtain the time before we open the time:
        start_time = time.clock()
        fs_file = open('/Users/Lisa/Documents/BigData/hw1/compustat_annual_updates_fundamentals_
        variables_of_interest = ['datadate', 'aco', 'acominc', 'act', 'ao', 'aocipen', 'aodo','a
        single_line =fs_file.readline()

        variables = single_line.split(",")
        index =0
        vti = dict()
        for variable in variables:
            variable = variable.strip()
            if variable in variables_of_interest:
                vti[variable]=index
                #print(variable+"\t"+str(index))
            index+=1

        for var in variables_of_interest:
            print(var+"\t"+str(vti[var]))



        #Create a list of dictionaries:
```

```
        ld = list()
        line_num=1
        for line in fs_file:
            #we would like to skip the first line (the header):
            if line_num==1:
                line_num += 1
                continue
            vars = line.split(",")

            td = dict()
            for var in variables_of_interest:
                        td[var]=vars[vti[var]]
            ld.append(td)
            #print(line, end='')
            line_num+=1
        fs_file.close()
        end_time = time.clock()
        print ("Read line by line and storing into list of dictionaries takes "+str(round(end_ti
```

```
datadate        1
aco        39
acominc        41
act        54
ao        65
aocipen        68
aodo        70
aoloch        72
aox        73
ap        74
aqc        80
at        93
bkvlps        109
caps        115
capx        116
capxv        117
ceq        122
ceql        123
ceqt        124
ch        133
Read line by line and storing into list of dictionaries takes 6.968535 seconds
```

```
In [10]: start_time = time.clock()
         import pandas as pd
         fs_pd = pd.read_csv('/Users/Lisa/Documents/BigData/hw1/compustat_annual_updates_fundame
         fsi_pd = fs_pd[variables_of_interest]
         #Obtain the time after we are done processing:
         end_time = time.clock()
```

```
        #Compute the running time
        print ("Pandas takes "+str(round(end_time - start_time,6))+" seconds")
```

```
/Users/Lisa/anaconda3/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2785: DtypeWa
  interactivity=interactivity, compiler=compiler, result=result)
```

```
Pandas takes 19.015816 seconds
```

In [ ]:
```
#as expected, Panda takes the longest time
#list of lists and list of dictionaries both take shorter time than panda
#but list of lists is faster than list of dictionaries
```