SEC Financial Statement Data III: Sort and Group the Data

In [1]:
```
import pandas as pd
import requests, zipfile, io
import os
from pathlib import Path
```

Get latest file from SEC (https://www.sec.gov/dera/data/financial-statement-and-notes-data-set.html):

In [2]:
```
def download_file(period):
    url = 'https://www.sec.gov/files/dera/data/financial-statement-and-notes-dat

    unzip_folder_name = 'data/sec/downloads/' + period

    r = requests.get(url)
    if r.ok:
        print('Downloaded:', url, 'to:', unzip_folder_name)
        Path(unzip_folder_name).mkdir(parents=True, exist_ok=True)
        z = zipfile.ZipFile(io.BytesIO(r.content))
        z.extractall(members=['sub.tsv','num.tsv'], path=unzip_folder_name)
    else:
        print('File not found')


def merge_sec_files(folder):

    keep_these_columns = ['cik','sic','countryinc','tag','filed','ddate','qtrs',

    filings = pd.read_table('data/sec/downloads/'+folder+'/sub.tsv')
    numbers = pd.read_table('data/sec/downloads/'+folder+'/num.tsv', encoding='I

    filings = filings[filings.form.isin(['10-Q','10-K']) & filings.cik.notnull()
    numbers = numbers[(numbers.dimh=='0x00000000')]

    merged = numbers.merge(filings, on='adsh', how='inner')[keep_these_columns]

    merged['filed'] = pd.to_datetime(merged.filed, format='%Y%m%d', errors='coer
    merged['ddate'] = pd.to_datetime(merged.ddate, format='%Y%m%d', errors='coer

    merged = merged[merged.filed.notnull() & merged.ddate.notnull()].drop_duplic

    merged.to_csv('data/sec/merged/'+folder+'.csv', index=False)

    return merged
```

In [3]:
```
download_file('2021_01')
```

```
Downloaded: https://www.sec.gov/files/dera/data/financial-statement-and-notes-da
ta-sets/2021_01_notes.zip to: data/sec/downloads/2021_01
```

In [4]:
```
merge_sec_files('2021_01')
```

Out[4]:

	cik	sic	countryinc	tag	filed	ddate	qtr

	cik	sic	countryinc	tag	filed	ddate	qtr
0	1517389	7371.0	US	AccountsPayableAndAccruedLiabilitiesCurrent	2021-01-06	2020-11-30	
1	1517389	7371.0	US	AccountsPayableAndAccruedLiabilitiesCurrent	2021-01-06	2020-02-29	
2	1517389	7371.0	US	AccountsReceivableNetCurrent	2021-01-06	2020-11-30	
3	1517389	7371.0	US	AccountsReceivableNetCurrent	2021-01-06	2020-02-29	
4	1517389	7371.0	US	AdditionalPaidInCapital	2021-01-06	2020-11-30	
...	...	...	...	...	...	...	...
118256	1593812	6221.0	US	RedemptionsCostBasis	2021-01-29	2019-10-31	
118257	1593812	6221.0	US	RedemptionsCostBasis	2021-01-29	2020-10-31	
118258	1593812	6221.0	US	WeightedAverageNumberOfGoldReceipts	2021-01-29	2018-10-31	
118259	1593812	6221.0	US	WeightedAverageNumberOfGoldReceipts	2021-01-29	2019-10-31	
118260	1593812	6221.0	US	WeightedAverageNumberOfGoldReceipts	2021-01-29	2020-10-31	

106707 rows × 8 columns

Read example file:

In [5]:
```
directory = 'data/sec/merged/'
filename  = '2010q2.csv'
data      = pd.read_csv(directory+filename, parse_dates=['filed','ddate'])
data
```

Out[5]:

	cik	sic	countryinc	tag	filed	ddate
0	4904	4911	US	EarningsPerShareBasic	2010-04-30	2009-03-31
1	4904	4911	US	EarningsPerShareBasic	2010-04-30	2010-03-31
2	4904	4911	US	EarningsPerShareDiluted	2010-04-30	2009-03-31
3	4904	4911	US	EarningsPerShareDiluted	2010-04-30	2010-03-31
4	4904	4911	US	ElectricProductionExpense	2010-04-30	2009-03-31

	cik	sic	countryinc	tag	filed	ddate
...	...	...	...	...	...	...
96107	796343	7372	US	WeightedAverageNumberOfDilutedSharesOutstanding	2010-04-09	2009-02-28
96108	796343	7372	US	WeightedAverageNumberOfDilutedSharesOutstanding	2010-04-09	2010-02-28
96109	796343	7372	US	WeightedAverageNumberOfSharesOutstandingBasic	2010-04-09	2009-02-28
96110	796343	7372	US	WeightedAverageNumberOfSharesOutstandingBasic	2010-04-09	2010-02-28
96111	796343	7372	US	WeightedAverageRecognitionPeriodRelatedToNonVe...	2010-04-09	2010-02-28

96112 rows × 8 columns

Get all earnings:

In [6]:
```
tag  = 'NetIncomeLoss'
item = data[data.tag==tag]
item
```

Out[6]:

	cik	sic	countryinc	tag	filed	ddate	qtrs	value
66	4904	4911	US	NetIncomeLoss	2010-04-30	2009-03-31	1	3.610000e+08
67	4904	4911	US	NetIncomeLoss	2010-04-30	2010-03-31	1	3.450000e+08
393	7084	2070	US	NetIncomeLoss	2010-05-10	2009-03-31	1	3.000000e+06
394	7084	2070	US	NetIncomeLoss	2010-05-10	2009-03-31	3	1.626000e+09
395	7084	2070	US	NetIncomeLoss	2010-05-10	2010-03-31	1	4.210000e+08
...	...	...	...	...	...	...	...	...
95168	896159	6331	CH	NetIncomeLoss	2010-05-07	2009-03-31	1	5.670000e+08
95575	1109357	4931	US	NetIncomeLoss	2010-04-23	2010-03-31	1	7.490000e+08
95589	1109357	4931	US	NetIncomeLoss	2010-04-23	2009-03-31	1	7.120000e+08
95885	796343	7372	US	NetIncomeLoss	2010-04-09	2009-02-28	1	1.564350e+08

	cik	sic	countryinc	tag	filed	ddate	qtrs	value
95886	796343	7372	US	NetIncomeLoss	2010-04-09	2010-02-28	1	1.271540e+08

923 rows × 8 columns

Get SEC file with ticker symbols:

In [7]:
```
symbols = pd.read_json('https://www.sec.gov/files/company_tickers.json').transpo
symbols
```

Out[7]:

cik_str	ticker	title
320193	AAPL	Apple Inc.
789019	MSFT	MICROSOFT CORP
1018724	AMZN	AMAZON COM INC
1652044	GOOG	Alphabet Inc.
1293451	TCEHY	Tencent Holdings Ltd
...	...	...
1819516	ASPL-WT	Aspirational Consumer Lifestyle Corp.
1819574	STIC-UN	Northern Star Acquisition Corp.
1819574	STIC-WT	Northern Star Acquisition Corp.
1819584	SNPR-UN	Tortoise Acquisition Corp. II
1819584	SNPR-WT	Tortoise Acquisition Corp. II

10906 rows × 2 columns

Find CIKs for Apple and Amazon:

In [8]:
```
apple = symbols[symbols.ticker=='AAPL'].index[0]
apple
```

Out[8]:
```
320193
```

In [9]:
```
amazon = symbols[symbols.ticker=='AMZN'].index[0]
amazon
```

Out[9]:
```
1018724
```

Get all repoted earnings for these two firms:

In [10]:
```
t = item[item.cik.isin([apple,amazon])]
t
```

Out[10]:

	cik	sic	countryinc	tag	filed	ddate	qtrs	value
21114	320193	3571	US	NetIncomeLoss	2010-04-21	2009-03-31	2	3.875000e+09
21134	320193	3571	US	NetIncomeLoss	2010-04-21	2010-03-31	2	6.452000e+09
21154	320193	3571	US	NetIncomeLoss	2010-04-21	2009-03-31	1	1.620000e+09
21163	320193	3571	US	NetIncomeLoss	2010-04-21	2010-03-31	1	3.074000e+09
43593	1018724	5961	US	NetIncomeLoss	2010-04-23	2009-03-31	4	6.790000e+08
43608	1018724	5961	US	NetIncomeLoss	2010-04-23	2010-03-31	4	1.024000e+09
43624	1018724	5961	US	NetIncomeLoss	2010-04-23	2009-03-31	1	1.770000e+08
43651	1018724	5961	US	NetIncomeLoss	2010-04-23	2010-03-31	1	2.990000e+08

We want: for each company and each filing: most recent period and shortest quarters.

Step 1: sort:

In [11]:
```
shortest = t.sort_values(['cik','filed','ddate','qtrs'], ascending=[True,True,Tr
shortest
```

Out[11]:

	cik	sic	countryinc	tag	filed	ddate	qtrs	value
21114	320193	3571	US	NetIncomeLoss	2010-04-21	2009-03-31	2	3.875000e+09
21154	320193	3571	US	NetIncomeLoss	2010-04-21	2009-03-31	1	1.620000e+09
21134	320193	3571	US	NetIncomeLoss	2010-04-21	2010-03-31	2	6.452000e+09
21163	320193	3571	US	NetIncomeLoss	2010-04-21	2010-03-31	1	3.074000e+09
43593	1018724	5961	US	NetIncomeLoss	2010-04-23	2009-03-31	4	6.790000e+08
43624	1018724	5961	US	NetIncomeLoss	2010-04-23	2009-03-31	1	1.770000e+08
43608	1018724	5961	US	NetIncomeLoss	2010-04-23	2010-03-31	4	1.024000e+09
43651	1018724	5961	US	NetIncomeLoss	2010-04-23	2010-03-31	1	2.990000e+08

Step 2: group (we want 1 observation per filing):

In [12]:
```
shortest.groupby(['cik','filed']).last()
```

Out[12]:

cik	filed	sic	countryinc	tag	ddate	qtrs	value
320193	**2010-04-21**	3571	US	NetIncomeLoss	2010-03-31	1	3.074000e+09
1018724	**2010-04-23**	5961	US	NetIncomeLoss	2010-03-31	1	2.990000e+08

Same for longest quarters:

In [13]:
```
longest = t.sort_values(['cik','filed','ddate','qtrs'], ascending=[True,True,Tru

longest.groupby(['cik','filed']).last()
```

Out[13]:

cik	filed	sic	countryinc	tag	ddate	qtrs	value
320193	**2010-04-21**	3571	US	NetIncomeLoss	2010-03-31	2	6.452000e+09
1018724	**2010-04-23**	5961	US	NetIncomeLoss	2010-03-31	4	1.024000e+09

Now do this for all firms:

In [14]:
```
shortest = item.sort_values(['cik','filed','ddate','qtrs'], ascending=[True,True
longest  = item.sort_values(['cik','filed','ddate','qtrs'], ascending=[True,True

shortest = shortest.groupby(['cik','filed']).last()     # Most recent, shortests
longest  = longest .groupby(['cik','filed']).last()     # Most recent, longest

shortest
```

Out[14]:

cik	filed	sic	countryinc	tag	ddate	qtrs	value
2969	**2010-04-26**	2810	US	NetIncomeLoss	2010-03-31	1	252000000.0
3673	**2010-05-07**	4911	US	NetIncomeLoss	2010-03-31	1	88200000.0
4127	**2010-05-11**	3674	US	NetIncomeLoss	2010-03-31	1	27744000.0
4281	**2010-04-22**	3350	US	NetIncomeLoss	2010-03-31	1	-201000000.0
4447	**2010-05-07**	2911	US	NetIncomeLoss	2010-03-31	1	538000000.0
...	**...**	...	...	...	...	...	...
1451505	**2010-05-05**	1381	CH	NetIncomeLoss	2010-03-31	1	677000000.0
1453090	**2010-05-03**	1381	CH	NetIncomeLoss	2010-03-31	1	-40009000.0
1465112	**2010-05-07**	4899	US	NetIncomeLoss	2010-03-31	1	558000000.0
1466258	**2010-05-07**	3822	IE	NetIncomeLoss	2010-03-31	1	1400000.0
1467373	**2010-06-25**	7389	IE	NetIncomeLoss	2010-05-31	1	490597000.0

403 rows × 6 columns

Repeat this for multiple quarters:

In [29]:
```
tag = 'NetIncomeLoss'

directory = 'data/sec/merged/'
filenames = ['2020q1.csv','2020q2.csv','2020q3.csv','2020_10.csv','2020_11.csv',

values_short   = pd.DataFrame()        # Values measured over shortest duration
values_long    = pd.DataFrame()        # Values measured over longest  duration

for filename in filenames:                                  # Loop over all files.
    print(filename)
    data = pd.read_csv(directory+filename, parse_dates=['filed','ddate'])  # Rea

    item  = data[data.tag==tag]                           # Select all data for th

    short = item.sort_values(['cik','filed','ddate','qtrs'], ascending=[True,Tru
    long  = item.sort_values(['cik','filed','ddate','qtrs'], ascending=[True,Tru
    short = short.groupby(['cik','filed']).last()     # Most recent, shortests f
    long  = long .groupby(['cik','filed']).last()     # Most recent, longest   f

    values_short   = values_short  .append( short[['value','qtrs']])
    values_long    = values_long   .append( long [['value','qtrs']])
```

```
2020q1.csv
2020q2.csv
2020q3.csv
2020_10.csv
2020_11.csv
2020_12.csv
2021_01.csv
```

Now check results for Apple:

In [30]:
```
# Apple CIK from above:
apple
```

Out[30]:
```
320193
```

In [35]:
```
values_short.loc[apple]
```

Out[35]:

filed	value	qtrs
2020-01-29	2.223600e+10	1
2020-05-01	1.124900e+10	1
2020-07-31	1.125300e+10	1
2020-10-30	1.267300e+10	1
2021-01-28	2.875500e+10	1

In [36]:
```
values_long.loc[apple]
```

Out[36]:

	value	qtrs

filed	value	qtrs

filed		
2020-01-29	2.223600e+10	1
2020-05-01	3.348500e+10	2
2020-07-31	4.473800e+10	3
2020-10-30	5.741100e+10	4
2021-01-28	2.875500e+10	1