web_scraping_post

January 19, 2021

[1]:

```
%reset -f
import requests
import urllib.request
import numpy as np
from bs4 import BeautifulSoup
import pandas as pd
import math
from re import search
from re import sub


######################################################################
################### Attributions ####################################
# Kevin Markham's Data School https://www.youtube.com/watch?v=zXif_9RVadI
# Theodore Petrou's Pandas Cookbook
# Wes McKinney's Python for Data Analysis
# Alvin Zuyin Zheng's "Python and Web Data Extraction: Introduction"
# Beautiful Soup Documentation
# Scraping EDGAR with Python by Rasha Ashraf
# Sigma Coding Scraping SEC XBRL Documents Parts 1-4
###################################################################
################################################################################
```

[2]:

```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

# Set the url to EDGAR's company.idx file for the first quarter of 2020. The␣
 ↪company.idx file will be read in
# as a fixed-width text file, the header will be removed, and then the program␣
 ↪can iterate through a table
# which will contain the company name, the CIK, and the url with a link to the .
 ↪txt file of the company filing.

url = 'https://www.sec.gov/Archives/edgar/full-index/2020/QTR1/company.idx'
urllib.request.urlretrieve(url, 'c:/web_scraping/company.idx')
```

[2]:

```
('c:/web_scraping/company.idx', <http.client.HTTPMessage at 0x1d84fea4370>)
```

[3]:

```
# Read in the fixed-width company.idx
df = pd.read_fwf('company.idx', colspecs=[
                 (0, 61), (62, 73), (74, 85), (86, 97), (98, 142)])


# Rows 0 through 6 are not needed since they are a header, row 7 is variable␣
 ↪names, and row 8 is dashes that
# should be gotten rid of.
df.head(15)
```

[3]:

```
   Description:           Master Index of EDGAR Dissemination Fe  d by Compan  \
0               Last Data Received:    March 31, 2020                     NaN
1            Comments:              webmaster@sec.gov                     NaN
2     Anonymous FTP:         ftp://ftp.sec.gov/edgar/                     NaN
3                                                 NaN                     NaN
4                                                 NaN                     NaN
5                                                 NaN                     NaN
6                                                 NaN                     NaN
7                                        Company Name               Form Type
8   ----------------------------------------------…             -----------
9                           &VEST Domestic Fund II LP                       D
10                        &VEST Offshore Fund II L.P.                       D
11                  &vest Domestic Fund II KPIV, L.P.                       D
12                                   024 Pharma, Inc.                   8-K/A
13                              1 800 FLOWERS COM INC                    10-Q
14                              1 800 FLOWERS COM INC                       4

           Name   Unnamed: 3                                   Unnamed: 4
0           NaN          NaN                                          NaN
1           NaN          NaN                                          NaN
2           NaN          NaN                                          NaN
3           NaN          NaN                                          NaN
4           NaN          NaN                                          NaN
5           NaN          NaN                                          NaN
6           NaN          NaN                                          NaN
7           CIK   Date Filed                                    File Name
8   -----------  -----------  -------------------------------------------
9       1800903   2020-01-27  edgar/data/1800903/0001800903-20-000001.txt
10      1800902   2020-01-27  edgar/data/1800902/0001800902-20-000001.txt
11      1802417   2020-02-06  edgar/data/1802417/0001802417-20-000001.txt
12      1307969   2020-02-20  edgar/data/1307969/0001683168-20-000541.txt
13      1084869   2020-02-07  edgar/data/1084869/0001437749-20-002005.txt
14      1084869   2020-02-28  edgar/data/1084869/0001437749-20-003844.txt
```

[4]:

```
df = df.iloc[7:]

df.reset_index(inplace=True, drop=True)
df.head()
```

```
[4]:  Description:            Master Index of EDGAR Dissemination Fe  d by Compan  \
    0                                        Company Name                Form Type
    1  ----------------------------------------------…             -----------
    2                          &VEST Domestic Fund II LP                        D
    3                        &VEST Offshore Fund II L.P.                        D
    4                  &vest Domestic Fund II KPIV, L.P.                        D

             Name  Unnamed: 3                                 Unnamed: 4
    0         CIK  Date Filed                                  File Name
    1 ----------- -----------  -------------------------------------------
    2     1800903  2020-01-27  edgar/data/1800903/0001800903-20-000001.txt
    3     1800902  2020-01-27  edgar/data/1800902/0001800902-20-000001.txt
    4     1802417  2020-02-06  edgar/data/1802417/0001802417-20-000001.txt
```

```
[5]: df = df.drop([1])

     # Now the DataFrame is in the format that is needed.
     df.head()
```

```
[5]:  Description:            Master Index of EDGAR Dissemination Fe d by Compan  \
    0                                        Company Name               Form Type
    2                          &VEST Domestic Fund II LP                       D
    3                        &VEST Offshore Fund II L.P.                       D
    4                  &vest Domestic Fund II KPIV, L.P.                       D
    5                                024 Pharma, Inc.                    8-K/A

         Name  Unnamed: 3                                 Unnamed: 4
    0     CIK  Date Filed                                  File Name
    2 1800903  2020-01-27  edgar/data/1800903/0001800903-20-000001.txt
    3 1800902  2020-01-27  edgar/data/1800902/0001800902-20-000001.txt
    4 1802417  2020-02-06  edgar/data/1802417/0001802417-20-000001.txt
    5 1307969  2020-02-20  edgar/data/1307969/0001683168-20-000541.txt
```

```
[6]: # Reset the indices
     df.reset_index(inplace=True, drop=True)
     df.head(15)
```

```
[6]:   Description:            Master Index of EDGAR Dissemination Fe d by Compan  \
    0                                        Company Name               Form Type
    1                          &VEST Domestic Fund II LP                       D
    2                        &VEST Offshore Fund II L.P.                       D
    3                  &vest Domestic Fund II KPIV, L.P.                       D
    4                                024 Pharma, Inc.                    8-K/A
    5                               1 800 FLOWERS COM INC                    10-Q
    6                               1 800 FLOWERS COM INC                       4
```

```
7                               1 800 FLOWERS COM INC                    8-K
8                               1 800 FLOWERS COM INC                    8-K
9                               1 800 FLOWERS COM INC                 SC 13G
10                              1 800 FLOWERS COM INC               SC 13G/A
11                              1 800 FLOWERS COM INC               SC 13G/A
12                        1 NORTH WEALTH SERVICES LLC                 13F-HR
13      10 Federal Self Storage Acquisition Co 2, LLC                    D/A
14                             10-15 ASSOCIATES, INC.                 13F-HR

       Name  Unnamed: 3                                   Unnamed: 4
0       CIK  Date Filed                                    File Name
1   1800903  2020-01-27  edgar/data/1800903/0001800903-20-000001.txt
2   1800902  2020-01-27  edgar/data/1800902/0001800902-20-000001.txt
3   1802417  2020-02-06  edgar/data/1802417/0001802417-20-000001.txt
4   1307969  2020-02-20  edgar/data/1307969/0001683168-20-000541.txt
5   1084869  2020-02-07  edgar/data/1084869/0001437749-20-002005.txt
6   1084869  2020-02-28  edgar/data/1084869/0001437749-20-003844.txt
7   1084869  2020-01-30  edgar/data/1084869/0001157523-20-000125.txt
8   1084869  2020-02-18  edgar/data/1084869/0001157523-20-000215.txt
9   1084869  2020-02-14  edgar/data/1084869/0001398344-20-003415.txt
10  1084869  2020-02-05  edgar/data/1084869/0000834237-20-004766.txt
11  1084869  2020-02-12  edgar/data/1084869/0001258897-20-000739.txt
12  1641761  2020-02-14  edgar/data/1641761/0001641761-20-000001.txt
13  1783074  2020-01-09  edgar/data/1783074/0001578563-20-000005.txt
14  1511144  2020-01-24  edgar/data/1511144/0001511144-20-000001.txt
```

```
[7]: # Make the variable names equal to the contents of row 0, then remove row 0.
     df.columns = df.iloc[0]
     df = df.drop([0])
     df.head()
```

```
[7]: 0                       Company Name Form Type      CIK  Date Filed  \
     1          &VEST Domestic Fund II LP         D  1800903  2020-01-27
     2        &VEST Offshore Fund II L.P.         D  1800902  2020-01-27
     3  &vest Domestic Fund II KPIV, L.P.         D  1802417  2020-02-06
     4                   024 Pharma, Inc.     8-K/A  1307969  2020-02-20
     5              1 800 FLOWERS COM INC      10-Q  1084869  2020-02-07

     0                                    File Name
     1  edgar/data/1800903/0001800903-20-000001.txt
     2  edgar/data/1800902/0001800902-20-000001.txt
     3  edgar/data/1802417/0001802417-20-000001.txt
     4  edgar/data/1307969/0001683168-20-000541.txt
     5  edgar/data/1084869/0001437749-20-002005.txt
```

```
[8]: # Create a boolean series to identify 10-K's, which are the annual reports.
     bool = df['Form Type'] == '10-K'
```

[9]:

```
df[bool].head()
```

[9]:

```
0                              Company Name Form Type      CIK  Date Filed  \
33                       10x Genomics, Inc.      10-K  1770787  2020-02-27
143  1347 Property Insurance Holdings, Inc.      10-K  1591890  2020-03-30
250                       1847 Holdings LLC      10-K  1599407  2020-03-30
260         1895 Bancorp of Wisconsin, Inc.      10-K  1751692  2020-03-30
290                    1Life Healthcare Inc      10-K  1404123  2020-03-27

0                                      File Name
33   edgar/data/1770787/0001193125-20-052640.txt
143  edgar/data/1591890/0001493152-20-005206.txt
250  edgar/data/1599407/0001213900-20-007912.txt
260  edgar/data/1751692/0001564590-20-014188.txt
290  edgar/data/1404123/0001564590-20-013666.txt
```

[10]:

```
# Retain only 10-K's in the dataset.
df= df[bool]
df.reset_index(inplace=True, drop=True)
df.head(50)
```

[10]:

```
0                             Company Name Form Type      CIK  Date Filed  \
0                       10x Genomics, Inc.      10-K  1770787  2020-02-27
1   1347 Property Insurance Holdings, Inc.      10-K  1591890  2020-03-30
2                        1847 Holdings LLC      10-K  1599407  2020-03-30
3          1895 Bancorp of Wisconsin, Inc.      10-K  1751692  2020-03-30
4                     1Life Healthcare Inc      10-K  1404123  2020-03-27
5                 1ST CONSTITUTION BANCORP      10-K  1141807  2020-03-16
6                          1ST SOURCE CORP      10-K    34782  2020-02-20
7              1st FRANKLIN FINANCIAL CORP      10-K    38723  2020-03-30
8                       20/20 Global, Inc.      10-K  1763329  2020-03-30
9                 22nd Century Group, Inc.      10-K  1347858  2020-03-11
10                                2U, Inc.      10-K  1459417  2020-02-28
11                         3D SYSTEMS CORP      10-K   910638  2020-02-26
12                                   3M CO      10-K    66740  2020-02-06
13                             89bio, Inc.      10-K  1785173  2020-03-18
14                      A. M. Castle & Co.      10-K    18172  2020-02-27
15                      A10 Networks, Inc.      10-K  1580808  2020-03-10
16                              AAON, INC.      10-K   824142  2020-02-27
17                             AARON'S INC      10-K   706688  2020-02-20
18        AB Private Credit Investors Corp      10-K  1634452  2020-03-30
19                     ABBOTT LABORATORIES      10-K     1800  2020-02-21
20                ABEONA THERAPEUTICS INC.      10-K   318306  2020-03-16
21             ABERCROMBIE & FITCH CO /DE/      10-K  1018840  2020-03-31
22                    ACACIA RESEARCH CORP      10-K   934549  2020-03-16
23              ACADIA PHARMACEUTICALS INC      10-K  1070494  2020-02-27
24                     ACADIA REALTY TRUST      10-K   899629  2020-02-21
```

```
25                    ACCELERON PHARMA INC      10-K  1280600  2020-02-27
26                        ACCESS-POWER INC      10-K  1041588  2020-01-02
27                        ACCO BRANDS Corp      10-K   712034  2020-02-27
28              ACELRX PHARMACEUTICALS INC      10-K  1427925  2020-03-16
29             ACHIEVE LIFE SCIENCES, INC.      10-K   949858  2020-03-13
30                     ACI WORLDWIDE, INC.      10-K   935036  2020-02-27
31                      ACM Research, Inc.      10-K  1680062  2020-03-24
32                        ACME UNITED CORP      10-K     2098  2020-03-13
33                               ACNB CORP      10-K   715579  2020-03-06
34                 ACORDA THERAPEUTICS INC      10-K  1008848  2020-02-28
35                      ACORN ENERGY, INC.      10-K   880984  2020-03-25
36                     ACQUIRED SALES CORP      10-K  1391135  2020-03-30
37               ACRO BIOMEDICAL CO., LTD.      10-K  1622996  2020-01-07
38              ACURA PHARMACEUTICALS, INC      10-K   786947  2020-03-31
39          ADAMS RESOURCES & ENERGY, INC.      10-K     2178  2020-03-06
40                ADESTO TECHNOLOGIES Corp      10-K  1395848  2020-03-16
41             ADIAL PHARMACEUTICALS, INC.      10-K  1513525  2020-03-20
42                    ADMA BIOLOGICS, INC.      10-K  1368514  2020-03-13
43                              ADOBE INC.      10-K   796343  2020-01-21
44                           ADOMANI, INC.      10-K  1563568  2020-03-10
45                                ADT Inc.      10-K  1703056  2020-03-10
46                              ADTRAN INC      10-K   926282  2020-02-25
47                     ADURO BIOTECH, INC.      10-K  1435049  2020-03-09
48                  ADVANCE AUTO PARTS INC      10-K  1158449  2020-02-18
49          ADVANCED ENERGY INDUSTRIES INC      10-K   927003  2020-03-02

0                                     File Name
0   edgar/data/1770787/0001193125-20-052640.txt
1   edgar/data/1591890/0001493152-20-005206.txt
2   edgar/data/1599407/0001213900-20-007912.txt
3   edgar/data/1751692/0001564590-20-014188.txt
4   edgar/data/1404123/0001564590-20-013666.txt
5   edgar/data/1141807/0001141807-20-000005.txt
6     edgar/data/34782/0000034782-20-000035.txt
7     edgar/data/38723/0001376474-20-000072.txt
8   edgar/data/1763329/0001445866-20-000291.txt
9   edgar/data/1347858/0001104659-20-031934.txt
10  edgar/data/1459417/0001459417-20-000003.txt
11   edgar/data/910638/0000910638-20-000010.txt
12    edgar/data/66740/0001558370-20-000581.txt
13  edgar/data/1785173/0001564590-20-011636.txt
14    edgar/data/18172/0000018172-20-000014.txt
15  edgar/data/1580808/0001580808-20-000014.txt
16   edgar/data/824142/0000824142-20-000025.txt
17   edgar/data/706688/0000706688-20-000012.txt
18  edgar/data/1634452/0001193125-20-090592.txt
19     edgar/data/1800/0001104659-20-023904.txt
```

```
20   edgar/data/318306/0001493152-20-004015.txt
21  edgar/data/1018840/0001018840-20-000021.txt
22   edgar/data/934549/0001683168-20-000836.txt
23  edgar/data/1070494/0001564590-20-006889.txt
24   edgar/data/899629/0001564590-20-005641.txt
25  edgar/data/1280600/0001280600-20-000012.txt
26  edgar/data/1041588/0001041588-20-000001.txt
27   edgar/data/712034/0000712034-20-000013.txt
28  edgar/data/1427925/0001437749-20-005304.txt
29   edgar/data/949858/0001564590-20-010642.txt
30   edgar/data/935036/0000935036-20-000009.txt
31  edgar/data/1680062/0001140361-20-006743.txt
32     edgar/data/2098/0001564590-20-010850.txt
33   edgar/data/715579/0001047469-20-001268.txt
34  edgar/data/1008848/0001564590-20-007789.txt
35   edgar/data/880984/0001493152-20-004708.txt
36  edgar/data/1391135/0001445866-20-000276.txt
37  edgar/data/1622996/0001640334-20-000007.txt
38   edgar/data/786947/0001104659-20-040632.txt
39     edgar/data/2178/0000002178-20-000013.txt
40  edgar/data/1395848/0001558370-20-002795.txt
41  edgar/data/1513525/0001213900-20-007020.txt
42  edgar/data/1368514/0001193805-20-000348.txt
43   edgar/data/796343/0000796343-20-000013.txt
44  edgar/data/1563568/0001564590-20-009553.txt
45  edgar/data/1703056/0001703056-20-000013.txt
46   edgar/data/926282/0001564590-20-006425.txt
47  edgar/data/1435049/0001564590-20-009364.txt
48  edgar/data/1158449/0001158449-20-000035.txt
49   edgar/data/927003/0001558370-20-001892.txt
```

[11]:
```
# web address for FCCY https://www.sec.gov/Archives/edgar/data/1141807/
 ↪0001141807-20-000005.txt, so you can see
# the CIK without leading zeros goes before the url with the CIK and leading␣
 ↪zeros.

# Set a string equal to the start of all url's.
begin = 'https://www.sec.gov/Archives/'

# Create an empty dataframe.
pl0 = pd.DataFrame()

# Initialize an empty list to track exceptions from the loop.
exceptions=[]
```

[12]:
```
# Run the loop for the first 10 firms
for i in range(0, 10):
    try:
        test = begin + df['File Name'].loc[i]
        first_word_name= df['Company Name'].loc[i].split(" ")[0]
        first_word_name= first_word_name.lower()
        print(first_word_name)
        print(i, test)

        u= requests.get(test)


        # The html5lib seemed like the only parser that would pull in the␣
↪entire xbrl data.
        # The html and lxml seemed like they ignored half of the text file, but␣
↪I can't explain this.
        soup = BeautifulSoup(u.text, 'html5lib')

        results = soup.find_all(['table'])

        # Below looks for the tables that could be the income statement. First,␣
↪the titles are converted to lowercase
        # and then single spaces, double spaces, and new line characteres are␣
↪removed. Note that due to the substantial
        # amount of variation in income statement titles in the 10-K's, I␣
↪created a series of "if" statements to try
        # an pick up the various titles I saw from manually inspecting annual␣
↪reports.

        table_list = []
        for i, j in enumerate(results):
            j = str(j)
            j= j.lower()
            j= j.replace(" ", "")
            j= j.replace("  ", "")
            j= j.replace("/n", "")
            if search("consolidatedstatementsofincome",  j):
                table_list.append(i)
            if search("consolidatedstatementsofoperations",  j):
                table_list.append(i)
            if search("consolidatedstatementsofcomprehensive",  j):
                table_list.append(i)
            if search("statementofincome",  j):
                table_list.append(i)
            if search("statementsofincome",  j):
                table_list.append(i)
            if search("statementofoperations",  j):
                table_list.append(i)
            if search("statementsofoperations",  j):
                table_list.append(i)
            if search("conslidatedstatementofincome",  j):
                table_list.append(i)
            if search("consolidatedstatementofoperations",  j):
                table_list.append(i)
            if search("consolidatedstatementofcomprehensive",  j):
                table_list.append(i)
            if search("consolidatedstatementofearnings", j):
                table_list.append(i)
            if search("consolidatedstatementsofearnings", j):
                table_list.append(i)
            if search("consolidatedstatementsofloss", j):
                table_list.append(i)
            if search("consolidatedstatementofloss", j):
                table_list.append(i)

        # The loop below contains scaffolding with print statements that show␣
↪the results the loop is returning. Even though
        # this makes the notebook larger, I thought it was helpful to include␣
↪this for clarity. Note that some firms use
        # NetIncomeLoss as the Net Income tag whereas others use ProfitLoss.

        tds = []
        pulled = []
        sub_list = []
        for i, j in enumerate(table_list):
            print(j)
            temp_table = results[j]
            temp_results = temp_table.find_all(['tr'])
            for x, y in enumerate(temp_results):
                y_str = str(y)
                # Modification if one were searching for earnings per share (i.
↪e., EPS) ->
                # if search("'defref_us-gaap_EarningsPerShareBasic',",  y_str):
                if search("'defref_us-gaap_NetIncomeLoss',",  y_str):
                    print('condition met', i, j, x, y, len(y))
                    sub_list.append(x)
                    pulled.append(y)
                    td = y.find_all('td')
                    tds.append(td)
                # Modifcation if one were searching for earnings per share (i.e.
↪, EPS) ->
                # if␣
↪search("'defref_us-gaap_EarningsPerShareBasicAndDiluted',",  y_str):
                if search("'defref_us-gaap_ProfitLoss',",  y_str):
                    print('condition met', i, j, x, y, len(y))
                    sub_list.append(x)
                    pulled.append(y)
                    td = y.find_all('td')
                    tds.append(td)

        # For firms that are reporting three years for their income statement,␣
↪there will be a lenfth of 4 based on the above.
        # The first hit will be for the XBRL tag searched for, here␣
↪NetIncomeLoss or ProfitLoss, and the next three for each
        # of the three reported years. However, if the firm was only in␣
↪operation for two years, the length would be 3, etc.
        # The "if" statements below are trying to appropriately deal with the␣
↪length of time the firm has been in operations
        # with a maximum of 3 years reported and a minimum of a single year␣
↪being reported.

        td_keeper = None
        for i, j in enumerate(tds):
            print(i, len(j) == 4)
            if len(j) == 4:
                td_keeper = i
            if (td_keeper == None) & (len(j) == 3):
                td_keeper = i
            if (td_keeper == None) & (len(j) == 2):
                td_keeper = i
            if (td_keeper == None) & (len(j) == 1):
                td_keeper = i

        balances= tds[td_keeper]


        url= test
        start = 'https://www.sec.gov/Archives/edgar/data/'
        # remove above then keep everything until / and that will be the CIK.
        url2 = url.replace(start, '')
        url2
        sep = '/'
        # Split on seperator one time, and keep the fist element.

        cik = url2.split(sep)[0]
        cik

        if len(balances)==4:
            df_temp= pd.DataFrame(data=[[cik, 'defref_us-gaap_NetIncomeLoss',␣
↪balances[-3], balances[-2], balances[-1]]], columns=['cik', 'xbrl_tag',␣
↪'cy', 'l_cy', 'l2_cy'])
```

```
            pl0= pl0.append(df_temp)
            del df_temp
        if len(balances)==3:
            df_temp= pd.DataFrame(data=[[cik, 'defref_us-gaap_NetIncomeLoss',␣
↪balances[-2], balances[-1], 'null']], columns=['cik', 'xbrl_tag', 'cy',␣
↪'l_cy', 'l2_cy'])
            pl0= pl0.append(df_temp)
            del df_temp
        if len(balances)==2:
            df_temp= pd.DataFrame(data=[[cik, 'defref_us-gaap_NetIncomeLoss',␣
↪balances[-1], 'null', 'null']], columns=['cik', 'xbrl_tag', 'cy', 'l_cy',␣
↪'l2_cy'])
            pl0= pl0.append(df_temp)
            del df_temp
        if (len(balances)==0) | (len(balances) >4):
            df_temp= pd.DataFrame(data=[[cik, 'defref_us-gaap_NetIncomeLoss',␣
↪'null', 'null', 'null']], columns=['cik', 'xbrl_tag', 'cy', 'l_cy', 'l2_cy'])
            pl0= pl0.append(df_temp)
            del df_temp

    except Exception:
        continue
```

```
10x
0 https://www.sec.gov/Archives/edgar/data/1770787/0001193125-20-052640.txt
151
151
183
183
541
541
1077
condition met 6 1077 19 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net loss</a></td>
<td class="num">(31,251)<span></span>
</td>
<td class="num">(112,485)<span></span>
</td>
<td class="num">(18,762)<span></span>
</td>
</tr> 9
1077
condition met 7 1077 19 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net loss</a></td>
<td class="num">(31,251)<span></span>
</td>
<td class="num">(112,485)<span></span>
</td>
<td class="num">(18,762)<span></span>
</td>
</tr> 9
1124
1124
1124
1124
1203
1203
1235
1235
1235
1235
1465
1465
1555
1555
1644
1644
1644
1644
0 True
1 True
1347
1 https://www.sec.gov/Archives/edgar/data/1591890/0001493152-20-005206.txt
46
46
99
326
326
457
457
457
813
813
822
822
1047
1047
1047
1130
1130
1153
```

```
condition met 17 1153 18 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">311<span></span>
</td>
<td class="nump">804<span></span>
</td>
</tr> 7
condition met 17 1153 28 <tr class="re">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">$ 311<span></span>
</td>
<td class="nump">$ 804<span></span>
</td>
</tr> 7
1153
condition met 18 1153 18 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">311<span></span>
</td>
<td class="nump">804<span></span>
</td>
</tr> 7
condition met 18 1153 28 <tr class="re">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">$ 311<span></span>
</td>
<td class="nump">$ 804<span></span>
</td>
</tr> 7
1153
condition met 19 1153 18 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">311<span></span>
</td>
<td class="nump">804<span></span>
</td>
</tr> 7
condition met 19 1153 28 <tr class="re">
```

```
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">$ 311<span></span>
</td>
<td class="nump">$ 804<span></span>
</td>
</tr> 7
1153
condition met 20 1153 18 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">311<span></span>
</td>
<td class="nump">804<span></span>
</td>
</tr> 7
condition met 20 1153 28 <tr class="re">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="nump">$ 311<span></span>
</td>
<td class="nump">$ 804<span></span>
</td>
</tr> 7
1204
1205
1281
1281
0 False
1 False
2 False
3 False
4 False
5 False
6 False
7 False
1847
2 https://www.sec.gov/Archives/edgar/data/1599407/0001213900-20-007912.txt
265
265
582
958
condition met 3 958 25 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">NET LOSS BEFORE NON-CONTROLLING
INTERESTS</a></td>
<td class="num">(3,381,423)<span></span>
</td>
<td class="num">(1,541,873)<span></span>
</td>
</tr> 7
958
condition met 4 958 25 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">NET LOSS BEFORE NON-CONTROLLING
INTERESTS</a></td>
<td class="num">(3,381,423)<span></span>
</td>
<td class="num">(1,541,873)<span></span>
</td>
</tr> 7
1058
1058
1106
1106
0 False
1 False
1895
3 https://www.sec.gov/Archives/edgar/data/1751692/0001564590-20-014188.txt
102
102
102
1270
condition met 3 1270 34 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income (loss)</a></td>
<td class="nump">$ 449<span></span>
</td>
<td class="num">$ (19)<span></span>
</td>
</tr> 7
1270
condition met 4 1270 34 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income (loss)</a></td>
<td class="nump">$ 449<span></span>
</td>
<td class="num">$ (19)<span></span>
</td>
```

```
</tr> 7
1633
condition met 5 1633 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income (loss)</a></td>
<td class="nump">$ 449<span></span>
</td>
<td class="num">$ (19)<span></span>
</td>
</tr> 7
1633
condition met 6 1633 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income (loss)</a></td>
<td class="nump">$ 449<span></span>
</td>
<td class="num">$ (19)<span></span>
</td>
</tr> 7
1650
1651
1652
1689
0 False
1 False
2 False
3 False
1life
4 https://www.sec.gov/Archives/edgar/data/1404123/0001564590-20-013666.txt
161
161
303
303
310
310
310
835
835
835
835
1395
1395
1411
1411
1842
condition met 15 1842 3 <tr class="ro">
```

```
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net loss</a></td>
<td class="num">$ (53,695)<span></span>
</td>
<td class="num">$ (45,501)<span></span>
</td>
<td class="num">$ (31,686)<span></span>
</td>
</tr> 9
1842
condition met 16 1842 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net loss</a></td>
<td class="num">$ (53,695)<span></span>
</td>
<td class="num">$ (45,501)<span></span>
</td>
<td class="num">$ (31,686)<span></span>
</td>
</tr> 9
1855
1856
1879
2025
2025
2097
2097
2097
2097
2220
condition met 26 2220 18 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net loss</a></td>
<td class="num">(53,695)<span></span>
</td>
<td class="num">(45,501)<span></span>
</td>
<td class="num">(31,686)<span></span>
</td>
</tr> 9
condition met 26 2220 20 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net loss attributable to 1Life Healthcare, Inc.
stockholders</a></td>
```

```
<td class="num">$ (52,554)<span></span>
</td>
<td class="num">$ (44,415)<span></span>
</td>
<td class="num">$ (30,797)<span></span>
</td>
</tr> 9
2220
condition met 27 2220 18 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net loss</a></td>
<td class="num">(53,695)<span></span>
</td>
<td class="num">(45,501)<span></span>
</td>
<td class="num">(31,686)<span></span>
</td>
</tr> 9
condition met 27 2220 20 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net loss attributable to 1Life Healthcare, Inc.
stockholders</a></td>
<td class="num">$ (52,554)<span></span>
</td>
<td class="num">$ (44,415)<span></span>
</td>
<td class="num">$ (30,797)<span></span>
</td>
</tr> 9
2263
2263
2269
2269
2289
2289
0 True
1 True
2 True
3 True
4 True
5 True
1st
5 https://www.sec.gov/Archives/edgar/data/1141807/0001141807-20-000005.txt
107
107
108
```

```
108
110
110
492
492
635
condition met 8 635 9 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income</a></td>
<td class="nump">$ 3,244<span></span>
</td>
<td class="nump">$ 3,623<span></span>
</td>
<td class="nump">$ 3,370<span></span>
</td>
<td class="nump">$ 3,397<span></span>
</td>
<td class="nump">$ 3,313<span></span>
</td>
<td class="nump">$ 4,011<span></span>
</td>
<td class="nump">$ 1,871<span></span>
</td>
<td class="nump">$ 2,853<span></span>
</td>
<td class="nump">$ 574<span></span>
</td>
<td class="nump">$ 2,486<span></span>
</td>
<td class="nump">$ 1,919<span></span>
</td>
<td class="nump">$ 1,949<span></span>
</td>
<td class="nump">13,634<span></span>
</td>
<td class="nump">12,048<span></span>
</td>
<td class="nump">6,928<span></span>
</td>
</tr> 33
condition met 8 635 23 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income</a></td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
```

```
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="nump">13,634<span></span>
</td>
<td class="nump">12,048<span></span>
</td>
<td class="nump">6,928<span></span>
</td>
</tr> 33
635
condition met 9 635 9 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income</a></td>
<td class="nump">$ 3,244<span></span>
</td>
<td class="nump">$ 3,623<span></span>
</td>
<td class="nump">$ 3,370<span></span>
</td>
<td class="nump">$ 3,397<span></span>
</td>
<td class="nump">$ 3,313<span></span>
</td>
<td class="nump">$ 4,011<span></span>
</td>
<td class="nump">$ 1,871<span></span>
</td>
<td class="nump">$ 2,853<span></span>
```

```
</td>
<td class="nump">$ 574<span></span>
</td>
<td class="nump">$ 2,486<span></span>
</td>
<td class="nump">$ 1,919<span></span>
</td>
<td class="nump">$ 1,949<span></span>
</td>
<td class="nump">13,634<span></span>
</td>
<td class="nump">12,048<span></span>
</td>
<td class="nump">6,928<span></span>
</td>
</tr> 33
condition met 9 635 23 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income</a></td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="text"> <span></span>
</td>
<td class="nump">13,634<span></span>
</td>
<td class="nump">12,048<span></span>
</td>
```

```
<td class="nump">6,928<span></span>
</td>
</tr> 33
1222
1222
1521
1521
1556
condition met 14 1556 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="th" style="border-bottom: 0px;"><sup></sup></td>
<td class="nump">$ 13,634<span></span>
</td>
<td class="nump">$ 12,048<span></span>
</td>
<td class="nump">$ 6,928<span></span>
</td>
</tr> 11
1556
condition met 15 1556 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="th" style="border-bottom: 0px;"><sup></sup></td>
<td class="nump">$ 13,634<span></span>
</td>
<td class="nump">$ 12,048<span></span>
</td>
<td class="nump">$ 6,928<span></span>
</td>
</tr> 11
1556
condition met 16 1556 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="th" style="border-bottom: 0px;"><sup></sup></td>
<td class="nump">$ 13,634<span></span>
</td>
<td class="nump">$ 12,048<span></span>
</td>
<td class="nump">$ 6,928<span></span>
</td>
</tr> 11
1556
condition met 17 1556 3 <tr class="ro">
```

```
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net income</a></td>
<td class="th" style="border-bottom: 0px;"><sup></sup></td>
<td class="nump">$ 13,634<span></span>
</td>
<td class="nump">$ 12,048<span></span>
</td>
<td class="nump">$ 6,928<span></span>
</td>
</tr> 11
1557
1557
1604
1605
2249
2249
2526
2526
2607
2607
2712
condition met 28 2712 36 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income</a></td>
<td class="nump">$ 13,634<span></span>
</td>
<td class="nump">$ 12,048<span></span>
</td>
<td class="nump">$ 6,928<span></span>
</td>
</tr> 9
2712
condition met 29 2712 36 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income</a></td>
<td class="nump">$ 13,634<span></span>
</td>
<td class="nump">$ 12,048<span></span>
</td>
<td class="nump">$ 6,928<span></span>
</td>
</tr> 9
2795
2795
0 False
```

```
1 False
2 False
3 False
4 False
5 False
6 False
7 False
8 True
9 True
1st
6 https://www.sec.gov/Archives/edgar/data/34782/0000034782-20-000035.txt
6
6
6
100
107
112
139
139
139
429
456
456
465
465
689
689
701
701
1011
1167
1252
1252
1326
1326
1332
condition met 24 1332 38 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net income</a></td>
<td class="nump">92,015<span></span>
</td>
<td class="nump">82,414<span></span>
</td>
<td class="nump">68,051<span></span>
</td>
</tr> 9
1332
```

```
condition met 25 1332 38 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net income</a></td>
<td class="nump">92,015<span></span>
</td>
<td class="nump">82,414<span></span>
</td>
<td class="nump">68,051<span></span>
</td>
</tr> 9
1548
1549
1554
1556
1564
1564
1564
1566
1579
1580
2329
2329
2350
2356
2356
2361
condition met 41 2361 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net income</a></td>
<td class="nump">$ 92,015<span></span>
</td>
<td class="nump">$ 82,414<span></span>
</td>
<td class="nump">$ 68,051<span></span>
</td>
</tr> 9
2361
condition met 42 2361 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net income</a></td>
<td class="nump">$ 92,015<span></span>
</td>
<td class="nump">$ 82,414<span></span>
</td>
<td class="nump">$ 68,051<span></span>
```

```
</td>
</tr> 9
2380
2381
2451
2451
2456
2456
2456
2458
2465
2467
0 True
1 True
2 True
3 True
1st
7 https://www.sec.gov/Archives/edgar/data/38723/0001376474-20-000072.txt
17
17
17
26
39
64
64
132
condition met 7 132 3 <tr class="ro">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income (Loss)</a></td>
<td class="nump">$ 13,348,373<span></span>
</td>
<td class="nump">$ 17,340,931<span></span>
</td>
<td class="nump">$ 14,905,754<span></span>
</td>
</tr> 9
629
860
903
903
909
909
1021
1052
condition met 15 1052 25 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income (Loss)</a></td>
<td class="th" style="border-bottom: 0px;"><sup></sup></td>
<td class="nump">$ 13,348,373<span></span>
</td>
<td class="nump">$ 17,340,931<span></span>
</td>
<td class="nump">$ 14,905,754<span></span>
</td>
</tr> 11
1052
condition met 16 1052 25 <tr class="rou">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net Income (Loss)</a></td>
<td class="th" style="border-bottom: 0px;"><sup></sup></td>
<td class="nump">$ 13,348,373<span></span>
</td>
<td class="nump">$ 17,340,931<span></span>
</td>
<td class="nump">$ 14,905,754<span></span>
</td>
</tr> 11
1281
1286
0 True
1 False
2 False
20/20
8 https://www.sec.gov/Archives/edgar/data/1763329/0001445866-20-000291.txt
7
7
310
310
335
condition met 4 335 22 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net income</a></td>
<td class="nump">$ 4,803<span></span>
</td>
<td class="nump">$ 61,202<span></span>
</td>
</tr> 7
335
condition met 5 335 22 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_ProfitLoss', window );">Net income</a></td>
```

```
<td class="nump">$ 4,803<span></span>
</td>
<td class="nump">$ 61,202<span></span>
</td>
</tr> 7
0 False
1 False
22nd
9 https://www.sec.gov/Archives/edgar/data/1347858/0001104659-20-031934.txt
100
100
107
107
155
155
409
condition met 6 409 28 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net loss</a></td>
<td class="num">(26,558,544)<span></span>
</td>
<td class="num">(7,966,911)<span></span>
</td>
<td class="num">(13,029,117)<span></span>
</td>
</tr> 9
409
condition met 7 409 28 <tr class="reu">
<td class="pl" style="border-bottom: 0px;" valign="top"><a class="a"
href="javascript:void(0);" onclick="top.Show.showAR( this, 'defref_us-
gaap_NetIncomeLoss', window );">Net loss</a></td>
<td class="num">(26,558,544)<span></span>
</td>
<td class="num">(7,966,911)<span></span>
</td>
<td class="num">(13,029,117)<span></span>
</td>
</tr> 9
517
517
517
517
529
529
705
705
1157
```

```
1157
1157
1157
1217
1456
1456
0 True
1 True
```

[13]:

```
# Examine the dataframe created from the loops. Note that "cy" is the current␣
 ↪year's net income, "l_cy" is lagged net income,
# and "l2_cy" is the second lag of net income. For a firm that filed in 2020␣
 ↪and reported 2019, 2018, and 2017 on the
# income statement "cy" is the 2019 Net Income and "l2_cy" is the 2017 Net␣
 ↪Income.

pl0
```

[13]:

```
       cik                        xbrl_tag                      cy  \
0  1770787  defref_us-gaap_NetIncomeLoss      [(31,251), [], \n]
0  1591890  defref_us-gaap_NetIncomeLoss           [311, [], \n]
0  1599407  defref_us-gaap_NetIncomeLoss   [(3,381,423), [], \n]
0  1751692  defref_us-gaap_NetIncomeLoss         [$ 449, [], \n]
0  1404123  defref_us-gaap_NetIncomeLoss    [$ (52,554), [], \n]
0  1141807  defref_us-gaap_NetIncomeLoss      [$ 13,634, [], \n]
0    34782  defref_us-gaap_NetIncomeLoss      [$ 92,015, [], \n]
0    38723  defref_us-gaap_NetIncomeLoss  [$ 13,348,373, [], \n]
0  1763329  defref_us-gaap_NetIncomeLoss       [$ 4,803, [], \n]
0  1347858  defref_us-gaap_NetIncomeLoss  [(26,558,544), [], \n]

                    l_cy                  l2_cy
0     [(112,485), [], \n]      [(18,762), [], \n]
0           [804, [], \n]                    null
0   [(1,541,873), [], \n]                    null
0        [$ (19), [], \n]                    null
0    [$ (44,415), [], \n]    [$ (30,797), [], \n]
0      [$ 12,048, [], \n]       [$ 6,928, [], \n]
0      [$ 82,414, [], \n]      [$ 68,051, [], \n]
0  [$ 17,340,931, [], \n]  [$ 14,905,754, [], \n]
0      [$ 61,202, [], \n]                    null
0   [(7,966,911), [], \n]  [(13,029,117), [], \n]
```

[14]:

```
pl0['cy']
```

[14]:

```
0        [(31,251), [], \n]
0             [311, [], \n]
0     [(3,381,423), [], \n]
```

```
0            [$ 449, [], \n]
0       [$ (52,554), [], \n]
0         [$ 13,634, [], \n]
0         [$ 92,015, [], \n]
0     [$ 13,348,373, [], \n]
0          [$ 4,803, [], \n]
0     [(26,558,544), [], \n]
Name: cy, dtype: object
```

[15]:
```
# The output above looks like a list object, but let's take a closer look at␣
 ↪the first and second observations in the
# dataset.

pl0['cy'].iloc[0]
```

[15]:
```
<td class="num">(31,251)<span></span>
</td>
```

[16]:
```
pl0['cy'].iloc[1]
```

[16]:
```
<td class="nump">311<span></span>
</td>
```

[17]:
```
# You can see above that the class is sometimes "num" and sometimes "nump" in␣
 ↪the underlying 10-K. Below retains the Net Income
# and discards the rest.
pl0['cy']= pl0['cy'].apply(lambda x: str(x).replace('<td class="nump">', '').
 ↪split('<span>')[0])
pl0['cy']= pl0['cy'].apply(lambda x: str(x).replace('<td class="num">', '').
 ↪split('<span>')[0])
```

[18]:
```
pl0['l_cy']= pl0['l_cy'].apply(lambda x: str(x).replace('<td class="nump">',␣
 ↪'').split('<span>')[0])
pl0['l_cy']= pl0['l_cy'].apply(lambda x: str(x).replace('<td class="num">', '').
 ↪split('<span>')[0])
```

[19]:
```
pl0['l2_cy']= pl0['l2_cy'].apply(lambda x: str(x).replace('<td class="nump">',␣
 ↪'').split('<span>')[0])
pl0['l2_cy']= pl0['l2_cy'].apply(lambda x: str(x).replace('<td class="num">',␣
 ↪'').split('<span>')[0])
```

[20]:
```
pl0
```

[20]:
```
       cik                       xbrl_tag           cy          l_cy  \
0  1770787  defref_us-gaap_NetIncomeLoss     (31,251)     (112,485)
0  1591890  defref_us-gaap_NetIncomeLoss          311           804
0  1599407  defref_us-gaap_NetIncomeLoss  (3,381,423)   (1,541,873)
```

```
0  1751692  defref_us-gaap_NetIncomeLoss         $ 449        $ (19)
0  1404123  defref_us-gaap_NetIncomeLoss    $ (52,554)    $ (44,415)
0  1141807  defref_us-gaap_NetIncomeLoss      $ 13,634      $ 12,048
0    34782  defref_us-gaap_NetIncomeLoss      $ 92,015      $ 82,414
0    38723  defref_us-gaap_NetIncomeLoss  $ 13,348,373  $ 17,340,931
0  1763329  defref_us-gaap_NetIncomeLoss       $ 4,803      $ 61,202
0  1347858  defref_us-gaap_NetIncomeLoss  (26,558,544)   (7,966,911)

          l2_cy
0      (18,762)
0          null
0          null
0          null
0    $ (30,797)
0       $ 6,928
0      $ 68,051
0  $ 14,905,754
0          null
0  (13,029,117)
```

[21]:
```
# Results above do not appear the same when I run this in VS Code or a local␣
 ↪notebook. On my machine, cy, l_cy, and l2_cy
# contain only the Net Income numbers and nothing else. But when uploading to␣
 ↪Github it appears
# that there are other entries next to the net income numbers. I can't explain␣
 ↪this and can't reproduce it locally. I have
# uploaded an export of the data to Excel (xbrl_PL_SEARCH_NI.xlsx) and a copy␣
 ↪of the local notebook in PDF format
# (web_scraping_post.PDF) so you can see what I see locally. This notebook␣
 ↪makes it look like there are errors only when
# uploaded to Github.
```

[22]:
```
pl0.to_excel('xbrl_PL_SEARCH_NI.xlsx', index=False)
```

[23]:
```
df= pd.read_excel('xbrl_PL_SEARCH_NI.xlsx')
```

[24]:
```
df
```

[24]:
```
       cik                      xbrl_tag            cy          l_cy  \
0  1770787  defref_us-gaap_NetIncomeLoss      (31,251)     (112,485)
1  1591890  defref_us-gaap_NetIncomeLoss           311           804
2  1599407  defref_us-gaap_NetIncomeLoss   (3,381,423)   (1,541,873)
3  1751692  defref_us-gaap_NetIncomeLoss         $ 449        $ (19)
4  1404123  defref_us-gaap_NetIncomeLoss    $ (52,554)    $ (44,415)
5  1141807  defref_us-gaap_NetIncomeLoss      $ 13,634      $ 12,048
6    34782  defref_us-gaap_NetIncomeLoss      $ 92,015      $ 82,414
7    38723  defref_us-gaap_NetIncomeLoss  $ 13,348,373  $ 17,340,931
```

```
8  1763329  defref_us-gaap_NetIncomeLoss       $ 4,803      $ 61,202
9  1347858  defref_us-gaap_NetIncomeLoss  (26,558,544)   (7,966,911)

          l2_cy
0      (18,762)
1           NaN
2           NaN
3           NaN
4    $ (30,797)
5       $ 6,928
6      $ 68,051
7  $ 14,905,754
8           NaN
9  (13,029,117)
```

[25]:

```
'''
############################################################
Discussion:
############################################################

1) No exceptions noted reconciling the local machine's output to the actual␣
 ↪10-K's (see web_scraping_post.PDF or
xbrl_PL_SEARCH_NI.xlsx).

2) When I ran this on the first 100 CIK's, I ended up getting data for around␣
 ↪88 of them. The above code was built from
looking at the exceptions in the exceptions list, identifying what tripped the␣
 ↪code, and modifying the loop for future runs.

3) It appears that there is variation in how the tag is applied to Net Income␣
 ↪Attributable to Stockholders vs. Non-Controlling
Interests. See CIK's 1599407 and 1404123 for examples.

4) Some firms have financial statements that contain lengths > 4 because they␣
 ↪also show monthly or quarterly breakdowns in
addition to the annual numbers. Examples include CIK's 1158449, 1420565,␣
 ↪1423689, and 824142. Refer to "td_keeper" in block [12]
to see that I am keeping 4 or less and the appliacable discussion.

5) Some firms appear to not have XBRL tagged documents. I can't explain this,␣
 ↪but it appears that CIK's 1366928, 1539816, and
1775098 are examples of this.

6) CIK 1514281 had five tags for the thee-year income statement and not 4. This␣
 ↪could be an isolated incident so I didn't
code around it, but it is something to be aware of.
```

```
'''
```

[25]:
```
'\n############################################################\nDiscussion:
\n############################################################\n\n1) No
exceptions noted reconciling the local machine\'s output to the actual 10-K\'s
(see web_scraping_post.PDF or \nxbrl_PL_SEARCH_NI.xlsx). \n\n2) When I ran this
on the first 100 CIK\'s, I ended up getting data for around 88 of them. The
above code was built from \nlooking at the exceptions in the exceptions list,
identifying what tripped the code, and modifying the loop for future runs.
\n\n3) It appears that there is variation in how the tag is applied to Net
Income Attributable to Stockholders vs. Non-Controlling\nInterests. See CIK\'s
1599407 and 1404123 for examples. \n\n4) Some firms have financial statements
that contain lengths > 4 because they also show monthly or quarterly breakdowns
in \naddition to the annual numbers. Examples include CIK\'s 1158449, 1420565,
1423689, and 824142. Refer to "td_keeper" in block [12]\nto see that I am
keeping 4 or less and the appliacable discussion. \n\n5) Some firms appear to
not have XBRL tagged documents. I can\'t explain this, but it appears that
CIK\'s 1366928, 1539816, and\n1775098 are examples of this. \n\n6) CIK 1514281
had five tags for the thee-year income statement and not 4. This could be an
isolated incident so I didn\'t \ncode around it, but it is something to be aware
of. \n'
```

[26]:
```
'''
#######################
Remaining Issues
#######################

1) Pulling the "units" of the income statement to make reported numbers␣
 ↪comparable. Some firms report in dollars, thousands
of dollars, millions, etc.

2) Capturing the SIC code for each firm. I wasn't able to think of a good way␣
 ↪to do this.

3) Efficiency. My loops are slow. It seems like there should be a way to find␣
 ↪the income statements faster, but this is
the best I could come up with. When I was scraping full income statements and␣
 ↪not using the XBRL tags, that approach took
over a full week for the loop to run.

4) Modifying the code to capture multiple financial statement variables on each␣
 ↪pass.

5) Conditioning on the year of the EDGAR fillings to perhaps create the␣
 ↪applicable years rather than the "cy", "l_cy" convention
currently used.
```

```
'''
```

```
[26]: '\n#######################\nRemaining Issues \n#######################\n\n1)
      Pulling the "units" of the income statement to make reported numbers comparable.
      Some firms report in dollars, thousands\nof dollars, millions, etc. \n\n2)
      Capturing the SIC code for each firm. I wasn\'t able to think of a good way to
      do this. \n\n3) Efficiency. My loops are slow. It seems like there should be a
      way to find the income statements faster, but this is\nthe best I could come up
      with. When I was scraping full income statements and not using the XBRL tags,
      that approach took\nover a full week for the loop to run. \n\n4) Modifying the
      code to capture multiple financial statement variables on each pass. \n\n5)
      Conditioning on the year of the EDGAR fillings to perhaps create the applicable
      years rather than the "cy", "l_cy" convention\ncurrently used. \n\n'
```