Project 1

October 16, 2021

[451]:

```
#Imports
from datetime import datetime
from datetime import timedelta
import requests, pandas as pd, numpy as np, time, datetime, matplotlib.pyplot␣
 ↪as plt, matplotlib.dates, math
from bs4 import BeautifulSoup
```

[ ]:

```

```

[452]:

```
#Part 1: Data Scraping and preparation
#Step 1: Scrape your competitor's data

r = requests.get('https://cmsc320.github.io/files/top-50-solar-flares.html')
root = BeautifulSoup(r.content)
root.prettify()
tables = pd.read_html('https://cmsc320.github.io/files/top-50-solar-flares.
 ↪html')
df_swl = tables[0]
df_swl = df_swl.rename(columns={"Unnamed: 0":"rank", "Unnamed: 1":
 ↪"x_classification",
                        "Unnamed: 2":"date", "Region":"region", "Start":
 ↪"start_time",
                       "Maximum":"maximum_time", "End":"end_time", "Unnamed: 7":
 ↪"movie"})
df_swl
```

[452]:

```
   rank x_classification        date  region start_time maximum_time  \
0     1             X28+  2003/11/04     486      19:29        19:53
1     2             X20+  2001/04/02    9393      21:32        21:51
2     3           X17.2+  2003/10/28     486      09:51        11:10
3     4             X17+  2005/09/07     808      17:17        17:40
4     5            X14.4  2001/04/15    9415      13:19        13:50
5     6              X10  2003/10/29     486      20:37        20:49
6     7             X9.4  1997/11/06    8100      11:49        11:55
7     8             X9.3  2017/09/06    2673      11:53        12:02
8     9               X9  2006/12/05     930      10:18        10:35
9    10             X8.3  2003/11/02     486      17:03        17:25
```

```
10    11             X8.2  2017/09/10    2673      15:35        16:06
11    12             X7.1  2005/01/20     720      06:36        07:01
12    13             X6.9  2011/08/09    1263      07:48        08:05
13    14             X6.5  2006/12/06     930      18:29        18:47
14    15             X6.2  2005/09/09     808      19:13        20:04
15    16             X6.2  2001/12/13    9733      14:20        14:30
16    17             X5.7  2000/07/14    9077      10:03        10:24
17    18             X5.6  2001/04/06    9415      19:10        19:21
18    19             X5.4  2012/03/07    1429      00:02        00:24
19    20             X5.4  2005/09/08     808      20:52        21:06
20    21             X5.4  2003/10/23     486      08:19        08:35
21    22             X5.3  2001/08/25    9591      16:23        16:45
22    23             X4.9  2014/02/25    1990      00:39        00:49
23    24             X4.9  1998/08/18    8307      22:10        22:19
24    25             X4.8  2002/07/23      39      00:18        00:35
25    26               X4  2000/11/26    9236      16:34        16:48
26    27             X3.9  2003/11/03     488      09:43        09:55
27    28             X3.9  1998/08/19    8307      21:35        21:45
28    29             X3.8  2005/01/17     720      06:59        09:52
29    30             X3.7  1998/11/22    8384      06:30        06:42
30    31             X3.6  2005/09/09     808      09:42        09:59
31    32             X3.6  2004/07/16     649      13:49        13:55
32    33             X3.6  2003/05/28     365      00:17        00:27
33    34             X3.4  2006/12/13     930      02:14        02:40
34    35             X3.4  2001/12/28    9767      20:02        20:45
35    36             X3.3  2013/11/05    1890      22:07        22:12
36    37             X3.3  2002/07/20      39      21:04        21:30
37    38             X3.3  1998/11/28    8395      04:54        05:52
38    39             X3.2  2013/05/14    1748      00:00        01:11
39    40             X3.1  2014/10/24    2192      21:07        21:41
40    41             X3.1  2002/08/24      69      00:49        01:12
41    42               X3  2002/07/15      30      19:59        20:08
42    43             X2.8  2013/05/13    1748      15:48        16:05
43    44             X2.8  2001/12/11    9733      07:58        08:08
44    45             X2.8  1998/08/18    8307      08:14        08:24
45    46             X2.7  2015/05/05    2339      22:05        22:11
46    47             X2.7  2003/11/03     488      01:09        01:30
47    48             X2.7  1998/05/06    8210      07:58        08:09
48    49             X2.6  2005/01/15     720      22:25        23:02
49    50             X2.6  2001/09/24    9632      09:32        10:38

   end_time              movie
0     20:06  MovieView archive
1     22:03  MovieView archive
2     11:24  MovieView archive
3     18:03  MovieView archive
4     13:55  MovieView archive
```

```
5     21:01  MovieView archive
6     12:01  MovieView archive
7     12:10  MovieView archive
8     10:45  MovieView archive
9     17:39  MovieView archive
10    16:31  MovieView archive
11    07:26  MovieView archive
12    08:08  MovieView archive
13    19:00  MovieView archive
14    20:36  MovieView archive
15    14:35  MovieView archive
16    10:43  MovieView archive
17    19:31  MovieView archive
18    00:40  MovieView archive
19    21:17  MovieView archive
20    08:49  MovieView archive
21    17:04  MovieView archive
22    01:03  MovieView archive
23    22:28       View archive
24    00:47  MovieView archive
25    16:56  MovieView archive
26    10:19  MovieView archive
27    21:50       View archive
28    10:07  MovieView archive
29    06:49  MovieView archive
30    10:08  MovieView archive
31    14:01  MovieView archive
32    00:39  MovieView archive
33    02:57  MovieView archive
34    21:32  MovieView archive
35    22:15  MovieView archive
36    21:54  MovieView archive
37    06:13  MovieView archive
38    01:20  MovieView archive
39    22:13  MovieView archive
40    01:31  MovieView archive
41    20:14  MovieView archive
42    16:16  MovieView archive
43    08:14  MovieView archive
44    08:32       View archive
45    22:15  MovieView archive
46    01:45  MovieView archive
47    08:20  MovieView archive
48    23:31  MovieView archive
49    11:09  MovieView archive
```

Part 1 Step 1: Using a BeautifulSoup object, I get the HTML content from the cmsc320 backup of the SpaceWeatherLive site. I then scrape the table from the site and convert it into a pandas DataFrame. I rename the columns for clarity,

```
[453]: #Step 2: Tidy the top 50 solar flare data
       df_swl = df_swl.drop(columns=['movie'])
       for index, row in df_swl.iterrows():
           startdate_str = str(row['date']) + " " + str(row['start_time']) + ":00"
           maxdate_str = str(row['date']) + " " + str(row['maximum_time']) + ":00"
           enddate_str = str(row['date']) + " " + str(row['end_time']) + ":00"
           startdate_str = startdate_str.replace('/', '-')
           maxdate_str = maxdate_str.replace('/', '-')
           enddate_str = enddate_str.replace('/', '-')
           startdate_obj = datetime.datetime.strptime(startdate_str, '%Y-%m-%d %H:%M:
        ↪%S')
           maxdate_obj = datetime.datetime.strptime(maxdate_str, '%Y-%m-%d %H:%M:%S')
           enddate_obj = datetime.datetime.strptime(enddate_str, '%Y-%m-%d %H:%M:%S')
           df_swl.at[index, 'date'] = startdate_obj
           df_swl.at[index, 'start_time'] = startdate_obj
           df_swl.at[index, 'maximum_time'] = maxdate_obj
           df_swl.at[index, 'end_time'] = enddate_obj
           if df_swl.at[index, 'region'] == '-':
               df_swl.replace(to_replace = 'region', value = np.nan)
       df_swl = df_swl.drop(columns=['date'])
       df_swl
```

```
[453]:     rank x_classification  region          start_time        maximum_time  \
       0      1            X28+     486 2003-11-04 19:29:00 2003-11-04 19:53:00
       1      2            X20+    9393 2001-04-02 21:32:00 2001-04-02 21:51:00
       2      3          X17.2+     486 2003-10-28 09:51:00 2003-10-28 11:10:00
       3      4            X17+     808 2005-09-07 17:17:00 2005-09-07 17:40:00
       4      5           X14.4    9415 2001-04-15 13:19:00 2001-04-15 13:50:00
       5      6             X10     486 2003-10-29 20:37:00 2003-10-29 20:49:00
       6      7            X9.4    8100 1997-11-06 11:49:00 1997-11-06 11:55:00
       7      8            X9.3    2673 2017-09-06 11:53:00 2017-09-06 12:02:00
       8      9              X9     930 2006-12-05 10:18:00 2006-12-05 10:35:00
       9     10            X8.3     486 2003-11-02 17:03:00 2003-11-02 17:25:00
       10    11            X8.2    2673 2017-09-10 15:35:00 2017-09-10 16:06:00
       11    12            X7.1     720 2005-01-20 06:36:00 2005-01-20 07:01:00
       12    13            X6.9    1263 2011-08-09 07:48:00 2011-08-09 08:05:00
       13    14            X6.5     930 2006-12-06 18:29:00 2006-12-06 18:47:00
       14    15            X6.2     808 2005-09-09 19:13:00 2005-09-09 20:04:00
       15    16            X6.2    9733 2001-12-13 14:20:00 2001-12-13 14:30:00
       16    17            X5.7    9077 2000-07-14 10:03:00 2000-07-14 10:24:00
       17    18            X5.6    9415 2001-04-06 19:10:00 2001-04-06 19:21:00
       18    19            X5.4    1429 2012-03-07 00:02:00 2012-03-07 00:24:00
       19    20            X5.4     808 2005-09-08 20:52:00 2005-09-08 21:06:00
       20    21            X5.4     486 2003-10-23 08:19:00 2003-10-23 08:35:00
```

```
21    22               X5.3   9591  2001-08-25 16:23:00  2001-08-25 16:45:00
22    23               X4.9   1990  2014-02-25 00:39:00  2014-02-25 00:49:00
23    24               X4.9   8307  1998-08-18 22:10:00  1998-08-18 22:19:00
24    25               X4.8     39  2002-07-23 00:18:00  2002-07-23 00:35:00
25    26                 X4   9236  2000-11-26 16:34:00  2000-11-26 16:48:00
26    27               X3.9    488  2003-11-03 09:43:00  2003-11-03 09:55:00
27    28               X3.9   8307  1998-08-19 21:35:00  1998-08-19 21:45:00
28    29               X3.8    720  2005-01-17 06:59:00  2005-01-17 09:52:00
29    30               X3.7   8384  1998-11-22 06:30:00  1998-11-22 06:42:00
30    31               X3.6    808  2005-09-09 09:42:00  2005-09-09 09:59:00
31    32               X3.6    649  2004-07-16 13:49:00  2004-07-16 13:55:00
32    33               X3.6    365  2003-05-28 00:17:00  2003-05-28 00:27:00
33    34               X3.4    930  2006-12-13 02:14:00  2006-12-13 02:40:00
34    35               X3.4   9767  2001-12-28 20:02:00  2001-12-28 20:45:00
35    36               X3.3   1890  2013-11-05 22:07:00  2013-11-05 22:12:00
36    37               X3.3     39  2002-07-20 21:04:00  2002-07-20 21:30:00
37    38               X3.3   8395  1998-11-28 04:54:00  1998-11-28 05:52:00
38    39               X3.2   1748  2013-05-14 00:00:00  2013-05-14 01:11:00
39    40               X3.1   2192  2014-10-24 21:07:00  2014-10-24 21:41:00
40    41               X3.1     69  2002-08-24 00:49:00  2002-08-24 01:12:00
41    42                 X3     30  2002-07-15 19:59:00  2002-07-15 20:08:00
42    43               X2.8   1748  2013-05-13 15:48:00  2013-05-13 16:05:00
43    44               X2.8   9733  2001-12-11 07:58:00  2001-12-11 08:08:00
44    45               X2.8   8307  1998-08-18 08:14:00  1998-08-18 08:24:00
45    46               X2.7   2339  2015-05-05 22:05:00  2015-05-05 22:11:00
46    47               X2.7    488  2003-11-03 01:09:00  2003-11-03 01:30:00
47    48               X2.7   8210  1998-05-06 07:58:00  1998-05-06 08:09:00
48    49               X2.6    720  2005-01-15 22:25:00  2005-01-15 23:02:00
49    50               X2.6   9632  2001-09-24 09:32:00  2001-09-24 10:38:00

               end_time
0   2003-11-04 20:06:00
1   2001-04-02 22:03:00
2   2003-10-28 11:24:00
3   2005-09-07 18:03:00
4   2001-04-15 13:55:00
5   2003-10-29 21:01:00
6   1997-11-06 12:01:00
7   2017-09-06 12:10:00
8   2006-12-05 10:45:00
9   2003-11-02 17:39:00
10  2017-09-10 16:31:00
11  2005-01-20 07:26:00
12  2011-08-09 08:08:00
13  2006-12-06 19:00:00
14  2005-09-09 20:36:00
15  2001-12-13 14:35:00
```

```
16   2000-07-14 10:43:00
17   2001-04-06 19:31:00
18   2012-03-07 00:40:00
19   2005-09-08 21:17:00
20   2003-10-23 08:49:00
21   2001-08-25 17:04:00
22   2014-02-25 01:03:00
23   1998-08-18 22:28:00
24   2002-07-23 00:47:00
25   2000-11-26 16:56:00
26   2003-11-03 10:19:00
27   1998-08-19 21:50:00
28   2005-01-17 10:07:00
29   1998-11-22 06:49:00
30   2005-09-09 10:08:00
31   2004-07-16 14:01:00
32   2003-05-28 00:39:00
33   2006-12-13 02:57:00
34   2001-12-28 21:32:00
35   2013-11-05 22:15:00
36   2002-07-20 21:54:00
37   1998-11-28 06:13:00
38   2013-05-14 01:20:00
39   2014-10-24 22:13:00
40   2002-08-24 01:31:00
41   2002-07-15 20:14:00
42   2013-05-13 16:16:00
43   2001-12-11 08:14:00
44   1998-08-18 08:32:00
45   2015-05-05 22:15:00
46   2003-11-03 01:45:00
47   1998-05-06 08:20:00
48   2005-01-15 23:31:00
49   2001-09-24 11:09:00
```

Part 1 Step 2: I begin cleaning the data by dropping the movies column from the DataFrame. I then create datetime objects using the 'date' column and the 'start_time', 'maximum_time', and 'end_time' columns. This datetime object replaces whatever is at the 'index' of the 'date', 'start_time', 'maximum_time', and 'end_time' columns. Additionally, if the cell at 'index' in column 'Region' contains a '-' character I replace the cell with NaN.

[454]:

```
#Step 3: Scrape the NASA data

r = requests.get('http://www.hcbravo.org/IntroDataSci/misc/waves_type2.html')
root = BeautifulSoup(r.content)
rawtext = root.find("pre").get_text()
rawtext = rawtext.split('\n')
```

```
table = []

for item in rawtext[12:-3]:
    item = item.split()
    table.append(item)
df_nasa =  pd.DataFrame(table)
df_nasa = df_nasa.iloc[: , : -9]
df_nasa = df_nasa.set_axis(['start_date', 'start_time', 'end_date', 'end_time',␣
 ↪'start_freq', 'end_freq', 'Loc', 'NOAA', 'Imp', 'cme_date', 'cme_time',␣
 ↪'CPA', 'cme_width', 'cme_speed', 'plots'], axis=1, inplace=False)
#df_nasa
```

Part 1 Step 3: I begin by using ‘requests’ and BeautifulSoup to get the HTML content from the hcbravo site. Next, I find the text of the page and ‘split’ it into a list of strings where each entry is a row of the table. From there, I iterate over the rows that make up the table and ‘split’ again to get each value by itself. Each value gets appended to a separate list and the whole list is converted into a DataFrame. I drop extra columns picked up from extraneous text from the site and set the column names.

[455]:

```
#Step 4: Tidy the NASA table
df_nasa['start_freq'] = df_nasa.start_freq.apply(lambda x: x if x != "????"␣
 ↪else np.nan)
df_nasa['end_freq'] = df_nasa.end_freq.apply(lambda x: x if x != "????" else np.
 ↪nan)
df_nasa['NOAA'] = df_nasa.NOAA.apply(lambda x: x if x != "-----" else np.nan)
df_nasa['Imp'] = df_nasa.Imp.apply(lambda x: x if x != "----" else np.nan)
df_nasa['is_halo'] = df_nasa.CPA.apply(lambda x: True if x == "Halo" else False)
df_nasa['CPA'] = df_nasa.CPA.apply(lambda x: x if x != "----" else np.nan)
df_nasa['CPA'] = df_nasa.CPA.apply(lambda x: x if x != "Halo" else np.nan)
df_nasa['width_lower_bound'] = df_nasa.cme_width.apply(lambda x: True if '>' in␣
 ↪str(x) else False)
df_nasa['cme_width'] = df_nasa.cme_width.apply(lambda x: str(x)[1:] if '>' in␣
 ↪str(x) else str(x))
df_nasa['cme_width'] = df_nasa.cme_width.apply(lambda x: x if x != "----" else␣
 ↪np.nan)
df_nasa['cme_speed'] = df_nasa.cme_speed.apply(lambda x: x if x != "----" else␣
 ↪np.nan)
df_nasa['Loc'] = df_nasa.Loc.apply(lambda x: np.nan if "back" in str(x).lower()␣
 ↪else str(x))
df_nasa['cme_date'] = df_nasa.cme_date.apply(lambda x: "01/01" if str(x) == "--/
 ↪--" else str(x))
df_nasa['cme_time'] = df_nasa.cme_time.apply(lambda x: "00:00" if str(x) == "--:
 ↪--" else str(x))

startdate_list = []
for index, row in df_nasa.iterrows():
    if "24:00" in str(row['start_time']):
        row['start_time'] = "00:00"
    if "24:00" in str(row['cme_time']):
        row['cme_time'] = "00:00"
    if "24:00" in str(row['end_time']):
        row['end_time'] = "00:00"
    startdate_str = str(row['start_date']) + " " + str(row['start_time'])
    startdate_list.append(startdate_str)
    cmedate_str = startdate_str[0:5] + str(row['cme_date']) + " " +␣
 ↪str(row['cme_time'])
    enddate_str = startdate_str[0:5] + str(row['end_date']) + " " +␣
 ↪str(row['end_time'])
    startdate_str = startdate_str.replace('/', '-')
    cmedate_str = cmedate_str.replace('/', '-')
    enddate_str = enddate_str.replace('/', '-')

    startdate_obj = datetime.datetime.strptime(startdate_str, '%Y-%m-%d %H:%M')
    cmedate_obj = datetime.datetime.strptime(cmedate_str, '%Y-%m-%d %H:%M')
    enddate_obj = datetime.datetime.strptime(enddate_str, '%Y-%m-%d %H:%M')
    df_nasa.at[index, 'start_datetime'] = startdate_obj
    df_nasa.at[index, 'end_datetime'] = cmedate_obj
    df_nasa.at[index, 'cme_datetime'] = enddate_obj

df_nasa = df_nasa.drop(columns = ['start_time', 'start_date', 'end_date',␣
 ↪'end_time', 'cme_date', 'cme_time'])
df_nasa
```

[455]:

```
    start_freq end_freq     Loc   NOAA  Imp  CPA cme_width cme_speed plots  \
0         8000     4000  S25E16   8026  M1.3   74        79       312   PHTX
1        11000     1000  S28E19   8027  C6.8  NaN       360       878   PHTX
2        12000       80  N21W08   8038  C1.3  NaN       360       464   PHTX
3         5000      500  N05W12   8040  M1.3  263       165       296   PHTX
4         6000     2000  S29E25   8088  C1.4  133       155       712   PHTX
..         …      …     …    …   …  …       …       …   …
477      14000     3900    W90b    NaN   NaN  NaN       360      2222   PHTX
478       2900     2100  S11E33  12241  M1.1  107       108       869   PHTX
479      14000    11500  S20E09  12242  M8.7  NaN       360       587   PHTX
480       5100     1300  S11E15  12241  M6.9  NaN       360      1195   PHTX
481      14000     7400  S14W25  12241  M1.0  NaN       360       669   PHTX

     is_halo  width_lower_bound      start_datetime        end_datetime  \
0      False              False 1997-04-01 14:00:00 1997-04-01 15:18:00
1       True              False 1997-04-07 14:30:00 1997-04-07 14:27:00
2       True              False 1997-05-12 05:15:00 1997-05-12 05:30:00
3      False              False 1997-05-21 20:20:00 1997-05-21 21:00:00
4      False              False 1997-09-23 21:53:00 1997-09-23 22:02:00
..       …                …                 …                 …
477     True              False 2014-12-13 14:27:00 2014-12-13 14:24:00
```

```
478    False              False 2014-12-17 04:09:00 2014-12-17 02:00:00
479     True              False 2014-12-17 05:00:00 2014-12-17 05:00:00
480     True              False 2014-12-18 22:31:00 2014-12-19 01:04:00
481     True              False 2014-12-21 12:05:00 2014-12-21 12:12:00

           cme_datetime
0   1997-04-01 14:15:00
1   1997-04-07 17:30:00
2   1997-05-14 16:00:00
3   1997-05-21 22:00:00
4   1997-09-23 22:16:00
..                  …
477 2014-12-13 14:51:00
478 2014-12-17 04:19:00
479 2014-12-17 05:09:00
480 2014-12-18 22:54:00
481 2014-12-21 12:28:00

[482 rows x 14 columns]
```

Part 1 Step 4: I complete steps 1, 2, and 3 listed on the github page using a series of lambda functions involving the columns of the DataFrame. For step 4, I iterate through the rows of the DataFrame and convert datetime objects for the start, maximum, and CME date and time columns. Then, I drop said columns as I stored the datetime objects into new columns.

[456]:

```python
#Part 2: Analysis
#-----------------------------------------------------
#Question 1: Replication
def classify(x):
    if x == x and x != "FILA" and str(x)[0] == 'X':
        return str(x)[1:]
    else:
        return "NaN"

df_tmp1 = df_nasa.copy(deep=True)
df_tmp1['Imp_Classify'] = df_tmp1.Imp.apply(classify)
df_tmp1['Imp_Classify'] = df_tmp1['Imp_Classify'].astype(float)
df_tmp1.sort_values('Imp_Classify',inplace=True, ascending=False)
imps = df_tmp1.head(50)
print(type(imps))
#df_tmp1['Imp_Classify'].head(50)
```

```
<class 'pandas.core.frame.DataFrame'>
```

Question 1: Replication

The data coming from NASA is incomplete in some instances (e.g. 10/28/2003 NASA has X17. whereas SWL has X17.2) or incorrect (e.g. 09/07/2005 NASA has X1.7 whereas SWL has X17.0). Therefore you cannot get the table from SWL exactly without manipulating the data extracted from NASA. Given the relatively small size of the SWL data (n = 50), this could be a meaningful investment.

[ ]:

```

```

[473]:

```
#Question 2: Integration
swl_datetimes = []
for index, row in df_swl.iterrows():
    swl_datetimes.append((df_swl.at[index,'start_time']))
nasa_datetimes = []
for index, row in df_nasa.iterrows():
    nasa_datetimes.append((df_nasa.at[index, 'start_datetime']))

df_match = pd.DataFrame(columns=['swl_date', "nasa_date", "diff", "swl_rank"])
count = 0
for i in range(len(swl_datetimes)):
    diff = timedelta(weeks=39)
    swl_dt = swl_datetimes[i]
    df_match.at[count, "swl_date"] = swl_dt
    for j in range(len(nasa_datetimes)):
        nasa_dt = nasa_datetimes[j]
        dt_diff = abs(swl_dt - nasa_dt)
        if dt_diff < diff:
            diff = dt_diff
            df_match.at[count, "nasa_date"] = nasa_dt
            df_match.at[count, "diff"] = diff
    df_match.at[count, "swl_rank"] = count + 1
    count += 1

df_match = df_match.sort_values(by=['nasa_date'])
df_match = df_match.reset_index(drop=True)
temp = [np.nan for i in range(482)]
df_nasa = df_nasa.assign(approx_SWLRank = temp)
df_nasa = df_nasa.sort_values(by=['start_datetime'])

match_list = []
nasa_list = df_match['nasa_date'].tolist()
swl_list = df_match['swl_rank'].tolist()
for i in range(50):
    match_list.append(tuple((nasa_list[i], swl_list[i])))

for index in range(len(df_nasa['start_datetime'])):
    t1 = pd.Timestamp(df_nasa.at[index, "start_datetime"])
    d1 = t1.to_pydatetime()
    s1 = d1.strftime("%m/%d/%Y")
    for j in range(50):
        temp = match_list[j]
        t2 = pd.Timestamp(temp[0])
```

```python
        d2 = t2.to_pydatetime()
        s2 = d1.strftime("%m/%d/%Y")
        if s2 == s1:
            #print(temp[1])
            df_nasa.at[index, "approx_SWLRank"] = temp[1]

df_nasa.head(25)
```

[473]:

```
   start_freq end_freq     Loc  NOAA   Imp  CPA cme_width cme_speed plots  \
0        8000     4000  S25E16  8026  M1.3   74        79       312 PHTX
1       11000     1000  S28E19  8027  C6.8  NaN       360       878 PHTX
2       12000       80  N21W08  8038  C1.3  NaN       360       464 PHTX
3        5000      500  N05W12  8040  M1.3  263       165       296 PHTX
4        6000     2000  S29E25  8088  C1.4  133       155       712 PHTX
5       14000      250  S20W13  8100  C8.6  240       109       227 PHTX
6       14000     5000  S16W21  8100  M4.2  233       122       352 PHTX
7       14000      100  S14W33  8100  X2.1  NaN       360       785 PHTX
8       14000      100  S18W63  8100  X9.4  NaN       360      1556 PHTX
9       14000     7000  N17E63  8113  X2.6   98        91       441 PHTX
10      14000     8000  N25W52  8116  B9.4  278        73       191 PHTX
11      14000    10000  N21E25  8141  C1.1  NaN       360       693 PHTX
12      14000     7000    SW90   NaN   NaN  NaN       360      1397 PHTX
13      10000       35  S22W90  8194  M1.4  284       165      1863 PHTX
14      14000      200  S17E90  8210  X1.2  NaN       360      1691 PHTX
15       4700     2600  S10E90  8210  C8.9  100        84      1184 PHTX
16      10000     1000  S16E50  8210  X1.0  NaN       360      1385 PHTX
17      10000     2000  S18E20  8210  M6.8  NaN       360      1374 PHTX
18       5000     3000  S15W15  8210  X1.1  NaN       360       938 PHTX
19      14000     5000  S11W65  8210  X2.7  309       190      1099 PHTX
20       9000      400  S14W89  8210  M7.7  262       178      2331 PHTX
21      10000     1000  N32W90  8214  B6.6  208       301       830 PHTX
22      14000     3000  N29W46  8222  B7.9  268       139       801 PHTX
23       4000     1000  N19W62  8226  C7.5  175       268       878 PHTX
24       8000     4000  N16E86  8243  M1.4  123       177      1223 PHTX

    is_halo  width_lower_bound      start_datetime        end_datetime  \
0     False              False 1997-04-01 14:00:00 1997-04-01 15:18:00
1      True              False 1997-04-07 14:30:00 1997-04-07 14:27:00
2      True              False 1997-05-12 05:15:00 1997-05-12 05:30:00
3     False              False 1997-05-21 20:20:00 1997-05-21 21:00:00
4     False              False 1997-09-23 21:53:00 1997-09-23 22:02:00
5     False              False 1997-11-03 05:15:00 1997-11-03 05:28:00
6     False              False 1997-11-03 10:30:00 1997-11-03 11:11:00
7      True              False 1997-11-04 06:00:00 1997-11-04 06:10:00
8      True              False 1997-11-06 12:20:00 1997-11-06 12:10:00
9     False              False 1997-11-27 13:30:00 1997-11-27 13:56:00
10    False              False 1997-12-12 22:45:00 1997-12-13 00:26:00
```

```
11     True             False 1998-01-25 15:03:00 1998-01-25 15:26:00
12     True             False 1998-03-29 03:40:00 1998-03-29 03:48:00
13    False             False 1998-04-20 10:25:00 1998-04-20 10:07:00
14     True             False 1998-04-23 06:00:00 1998-04-23 05:55:00
15    False             False 1998-04-24 09:17:00 1998-04-24 08:55:00
16     True             False 1998-04-27 09:20:00 1998-04-27 08:56:00
17     True             False 1998-04-29 16:30:00 1998-04-29 16:58:00
18     True             False 1998-05-02 14:25:00 1998-05-02 14:06:00
19    False             False 1998-05-06 08:25:00 1998-05-06 08:29:00
20    False             False 1998-05-09 03:35:00 1998-05-09 03:35:00
21    False              True 1998-05-11 21:40:00 1998-05-11 21:55:00
22    False             False 1998-05-19 10:00:00 1998-05-19 10:27:00
23    False             False 1998-05-27 13:30:00 1998-05-27 13:45:00
24    False             False 1998-06-11 10:15:00 1998-06-11 10:28:00
```

```
          cme_datetime  approx_SWLRank
0  1997-04-01 14:15:00            11.0
1  1997-04-07 17:30:00            11.0
2  1997-05-14 16:00:00            11.0
3  1997-05-21 22:00:00            11.0
4  1997-09-23 22:16:00            11.0
5  1997-11-03 12:00:00            11.0
6  1997-11-03 11:30:00            11.0
7  1997-11-05 04:30:00            11.0
8  1997-11-07 08:30:00            11.0
9  1997-11-27 14:00:00            11.0
10 1997-12-12 23:20:00            11.0
11 1998-01-25 15:18:00            11.0
12 1998-03-29 03:52:00            11.0
13 1998-04-22 06:00:00            11.0
14 1998-04-23 15:30:00            11.0
15 1998-04-24 09:25:00            11.0
16 1998-04-27 10:00:00            11.0
17 1998-04-29 17:00:00            11.0
18 1998-05-02 14:50:00            11.0
19 1998-05-06 08:35:00            11.0
20 1998-05-09 10:00:00            11.0
21 1998-05-11 22:00:00            11.0
22 1998-05-19 11:30:00            11.0
23 1998-05-27 14:20:00            11.0
24 1998-06-11 10:20:00            11.0
```

Question 2: Integration

I chose to use the starting date of the type II burst (first column NASA data, second column SWL data) as a the way to replicate the SWL data using the NASA data. However, I only used MM-DD-YYYY format because anything more specific (e.g. hour, minute, etc.) wouldn't match between the two datasets exactly. Furthermore, since neither the X classification (column nine NASA data) nor the region (column eight NASA data) were consistent between the datasets, I couldn't use either of those indicators. Thus, the only consistent way to compare datapoints between the two sites is the date. However, sometimes there were multiple entries in the NASA dataset on the same day. For example, on 04/02/2001 there were two recordings. The first recording had an x classification of X1.1 whereas the second had an x classification of X20. Thus, whichever recording had the higher x classification is the one I'd use.

[ ]:

```
#Question 3: Analysis
datetime_list = []
for item in startdate_list:
    item = item.replace('/', '-')
    item = datetime.datetime.strptime(item, '%Y-%m-%d %H:%M')
    datetime_list.append(item)
X_plot = matplotlib.dates.date2num(datetime_list)
Y_plot = df_nasa['NOAA'].values
Y_plot = Y_plot.astype(str)
plt.plot_date(X_plot, Y_plot)
plt.show()
```

Question 3:

I plotted location over time. This shows how quickly new solar flares got added to the database.

[ ]:

[ ]: