Fall24UPDATEDchicago_crime652

October 7, 2024

1 Structured data analysis example

This example will use subsets of a dataset of reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago in 2021. Source – Chicago data portal. URL https://data.cityofchicago.org/Public-Safety/Crimes-2021/dwme-t96c

Only a portion of the data entries in the dataset will be used to illustrate how to work with structured data using Pandas.

1.0.1 Let us begin by bringing in the libraries we are interested in using for this exercise. We are using pandas, numpy and Matplotlib. We have have set up the plots to render in our notebook by using *%matplotlib inline*.

[1]:
```
%matplotlib inline

import pandas as pd
import numpy as np

#import requests
#from io import StringIO
```

1.1 Now to bring the data into our notebook. Our data set is a .csv file stored on the class server in the datasets folder. You are given the file path. We will get the data from a data repository (inside SU) and bring it into the notebook.

1.2 Read the data using *pd.read_csv* . This is a small version on the dataset we are bringing into our notebook.

[2]:
```
#Defining location of dataset
filepath="/opt/datasets/ist652/Crimes/Crimes_small.csv"

crimes_small=pd.read_csv(filepath)
```

1.3 Let's see the data from the CSV file looks like... we use the head function to get the top 5 rows. There are 22 columns. This is a good point to go to the data source to see what these columns mean. See URL https://data.cityofchicago.org/Public-Safety/Crimes-2021/dwme-t96c

[3]:
```
crimes_small.head()
```

[3]:

	ID	Case Number	Date	Block	IUCR	\
0	12571973	JE482457	12/19/2021 7:23	042XX S MOZART ST	460	
1	12343475	JE202728	4/16/2021 20:45	056XX N RIDGE AVE	820	
2	12602803	JF125633	10/21/2021 11:00	083XX S STONY ISLAND AVE	500E	
3	12540388	JE444591	11/14/2021 6:00	086XX S COTTAGE GROVE AVE	850	
4	12541139	JE445494	11/14/2021 4:00	034XX W 38TH ST	486	

	Primary Type	Description	Location Description	Arrest	\
0	BATTERY	SIMPLE	SIDEWALK	True	
1	THEFT	$500 AND UNDER	OTHER (SPECIFY)	False	
2	OTHER OFFENSE	EAVESDROPPING	OTHER (SPECIFY)	False	
3	THEFT	ATTEMPT THEFT	CONVENIENCE STORE	False	
4	BATTERY	DOMESTIC BATTERY SIMPLE	RESIDENCE	False	

	Domestic	…	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	\
0	True	…	15	58	08B	1158067.0	1876425.0	
1	False	…	48	77	6	NaN	NaN	
2	False	…	8	45	26	1188260.0	1849805.0	
3	False	…	6	44	6	1183071.0	1847869.0	
4	True	…	12	58	08B	1154073.0	1879187.0	

	Year	Updated On	Latitude	Longitude	Location
0	2021	9/12/2022 16:45	41.816657	-87.695689	(41.81665685, -87.695688608)
1	2021	4/23/2021 16:51	NaN	NaN	NaN
2	2021	2/27/2022 15:46	41.742941	-87.585783	(41.74294124, -87.585783412)
3	2021	11/21/2021 15:48	41.737751	-87.604856	(41.737750767, -87.604855911)
4	2021	11/21/2021 15:48	41.824317	-87.710266	(41.824316537, -87.710266215)

[5 rows x 22 columns]

2 DATE DATATYPE AND SETTING INDEXES

2.1 To get more infomation on the dataframe we use the *.info()* function. Inspect the data types.

[4]:
```
crimes_small.info() # about the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 22 columns):
```

```
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    9 non-null      int64
 1   Case Number           9 non-null      object
 2   Date                  9 non-null      object
 3   Block                 9 non-null      object
 4   IUCR                  9 non-null      object
 5   Primary Type          9 non-null      object
 6   Description           9 non-null      object
 7   Location Description  9 non-null      object
 8   Arrest                9 non-null      bool
 9   Domestic              9 non-null      bool
 10  Beat                  9 non-null      int64
 11  District              9 non-null      int64
 12  Ward                  9 non-null      int64
 13  Community Area        9 non-null      int64
 14  FBI Code              9 non-null      object
 15  X Coordinate          8 non-null      float64
 16  Y Coordinate          8 non-null      float64
 17  Year                  9 non-null      int64
 18  Updated On            9 non-null      object
 19  Latitude              8 non-null      float64
 20  Longitude             8 non-null      float64
 21  Location              8 non-null      object
dtypes: bool(2), float64(4), int64(6), object(10)
memory usage: 1.6+ KB
```

2.2 The immediate red flag is the "Date" column does not have the correct data type to make good use of it as a date column. It is an object type therefore a string, this is the default when parsing. If you recall Pandas has a datatype for dates and time.

2.3 We need to parse that column so that it gets correctly formatted. (See this link for more tricks with dates: https://towardsdatascience.com/4-tricks-you-should-know-to-parse-date-columns-with-pandas-read-csv-27355bb2ad0e)

2.4 We use ***parse_dates*** which by default is month/day/year. You can customize the manner in which the date is parsed. Here we are specifying that the date/time column is located in the third column, hence ***parse_dates=[2]***.

[5]:
```
crimes_small=pd.read_csv(filepath,parse_dates=[2]) #getting the data into a␣
↪pandas dataframe with the correct format for dates
```

[6]:
```
crimes_small.info()
```

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

```
RangeIndex: 9 entries, 0 to 8
Data columns (total 22 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    9 non-null      int64
 1   Case Number           9 non-null      object
 2   Date                  9 non-null      datetime64[ns]
 3   Block                 9 non-null      object
 4   IUCR                  9 non-null      object
 5   Primary Type          9 non-null      object
 6   Description           9 non-null      object
 7   Location Description  9 non-null      object
 8   Arrest                9 non-null      bool
 9   Domestic              9 non-null      bool
 10  Beat                  9 non-null      int64
 11  District              9 non-null      int64
 12  Ward                  9 non-null      int64
 13  Community Area        9 non-null      int64
 14  FBI Code              9 non-null      object
 15  X Coordinate          8 non-null      float64
 16  Y Coordinate          8 non-null      float64
 17  Year                  9 non-null      int64
 18  Updated On            9 non-null      object
 19  Latitude              8 non-null      float64
 20  Longitude             8 non-null      float64
 21  Location              8 non-null      object
dtypes: bool(2), datetime64[ns](1), float64(4), int64(6), object(9)
memory usage: 1.6+ KB
```

2.4.1 The dataset in indexed automatically... 0, 1, 2. Let's index the dataset by date. It will make it easier to work with. To do this we use set_index and set the inplace parameter as True to make the change to our dataset. This parameter is set as False by default. You can add your own index series with *set_index*

[7]:
```
crimes_small.set_index("Date",inplace=True) # setting datetime as the index
```

[8]:
```
crimes_small.index
```

[8]:
```
DatetimeIndex(['2021-12-19 07:23:00', '2021-04-16 20:45:00',
               '2021-10-21 11:00:00', '2021-11-14 06:00:00',
               '2021-11-14 04:00:00', '2021-11-14 09:00:00',
               '2021-11-14 14:00:00', '2021-11-14 00:00:00',
               '2021-01-09 15:59:00'],
              dtype='datetime64[ns]', name='Date', freq=None)
```

[9]:
```
crimes_small.head()
```

[9]:

Date	ID	Case Number	Block	IUCR
2021-12-19 07:23:00	12571973	JE482457	042XX S MOZART ST	460
2021-04-16 20:45:00	12343475	JE202728	056XX N RIDGE AVE	820
2021-10-21 11:00:00	12602803	JF125633	083XX S STONY ISLAND AVE	500E
2021-11-14 06:00:00	12540388	JE444591	086XX S COTTAGE GROVE AVE	850
2021-11-14 04:00:00	12541139	JE445494	034XX W 38TH ST	486

Date	Primary Type	Description
2021-12-19 07:23:00	BATTERY	SIMPLE
2021-04-16 20:45:00	THEFT	$500 AND UNDER
2021-10-21 11:00:00	OTHER OFFENSE	EAVESDROPPING
2021-11-14 06:00:00	THEFT	ATTEMPT THEFT
2021-11-14 04:00:00	BATTERY	DOMESTIC BATTERY SIMPLE

Date	Location Description	Arrest	Domestic	Beat	…	Ward
2021-12-19 07:23:00	SIDEWALK	True	True	921	…	15
2021-04-16 20:45:00	OTHER (SPECIFY)	False	False	2013	…	48
2021-10-21 11:00:00	OTHER (SPECIFY)	False	False	412	…	8
2021-11-14 06:00:00	CONVENIENCE STORE	False	False	632	…	6
2021-11-14 04:00:00	RESIDENCE	False	True	911	…	12

Date	Community Area	FBI Code	X Coordinate	Y Coordinate
2021-12-19 07:23:00	58	08B	1158067.0	1876425.0
2021-04-16 20:45:00	77	6	NaN	NaN
2021-10-21 11:00:00	45	26	1188260.0	1849805.0
2021-11-14 06:00:00	44	6	1183071.0	1847869.0
2021-11-14 04:00:00	58	08B	1154073.0	1879187.0

Date	Year	Updated On	Latitude	Longitude
2021-12-19 07:23:00	2021	9/12/2022 16:45	41.816657	-87.695689
2021-04-16 20:45:00	2021	4/23/2021 16:51	NaN	NaN
2021-10-21 11:00:00	2021	2/27/2022 15:46	41.742941	-87.585783
2021-11-14 06:00:00	2021	11/21/2021 15:48	41.737751	-87.604856
2021-11-14 04:00:00	2021	11/21/2021 15:48	41.824317	-87.710266

Date	Location
2021-12-19 07:23:00	(41.81665685, -87.695688608)
2021-04-16 20:45:00	NaN
2021-10-21 11:00:00	(41.74294124, -87.585783412)
2021-11-14 06:00:00	(41.737750767, -87.604855911)
2021-11-14 04:00:00	(41.824316537, -87.710266215)

```
[5 rows x 21 columns]
```

2.4.2 Confirming date is set as our index

[15]:
```
crimes_small.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9 entries, 2021-12-19 07:23:00 to 2021-01-09 15:59:00
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    9 non-null      int64
 1   Case Number           9 non-null      object
 2   Block                 9 non-null      object
 3   IUCR                  9 non-null      object
 4   Primary Type          9 non-null      object
 5   Description           9 non-null      object
 6   Location Description  9 non-null      object
 7   Arrest                9 non-null      bool
 8   Domestic              9 non-null      bool
 9   Beat                  9 non-null      int64
 10  District              9 non-null      int64
 11  Ward                  9 non-null      int64
 12  Community Area        9 non-null      int64
 13  FBI Code              9 non-null      object
 14  X Coordinate          8 non-null      float64
 15  Y Coordinate          8 non-null      float64
 16  Year                  9 non-null      int64
 17  Updated On            9 non-null      object
 18  Latitude              8 non-null      float64
 19  Longitude             8 non-null      float64
 20  Location              8 non-null      object
dtypes: bool(2), float64(4), int64(6), object(9)
memory usage: 1.4+ KB
```

2.4.3 If I want to see the values for a specific date. I can use my index value.

[17]:
```
crimes_small.sort_index().loc['2021-12-19']
```

[17]:
```
                           ID Case Number              Block IUCR  \
Date
2021-12-19 07:23:00  12571973    JE482457  042XX S MOZART ST  460

                     Primary Type Description Location Description  Arrest  \
Date
2021-12-19 07:23:00      BATTERY      SIMPLE             SIDEWALK    True
```

```
                     Domestic  Beat  …  Ward  Community Area  FBI Code  \
Date                                 …
2021-12-19 07:23:00      True   921  …    15              58       08B

                    X Coordinate  Y Coordinate  Year       Updated On  \
Date
2021-12-19 07:23:00    1158067.0     1876425.0  2021  9/12/2022 16:45

                     Latitude  Longitude                     Location
Date
2021-12-19 07:23:00  41.816657 -87.695689  (41.81665685, -87.695688608)

[1 rows x 21 columns]
```

2.4.4 I can also use a date range

[21]:
```
crimes_small.sort_index().loc['2021-10-19' : '2021-12-30']
```

[21]:
```
                          ID Case Number                       Block  IUCR  \
Date
2021-10-21 11:00:00  12602803    JF125633   083XX S STONY ISLAND AVE  500E
2021-11-14 00:00:00  12541098    JE444580         072XX S HAMLIN AVE   486
2021-11-14 04:00:00  12541139    JE445494            034XX W 38TH ST   486
2021-11-14 06:00:00  12540388    JE444591  086XX S COTTAGE GROVE AVE   850
2021-11-14 09:00:00  12540496    JE444717        070XX S INDIANA AVE   820
2021-11-14 14:00:00  12542477    JE447028         021XX N BINGHAM ST   820
2021-12-19 07:23:00  12571973    JE482457          042XX S MOZART ST   460

                      Primary Type              Description  \
Date
2021-10-21 11:00:00  OTHER OFFENSE            EAVESDROPPING
2021-11-14 00:00:00        BATTERY  DOMESTIC BATTERY SIMPLE
2021-11-14 04:00:00        BATTERY  DOMESTIC BATTERY SIMPLE
2021-11-14 06:00:00          THEFT            ATTEMPT THEFT
2021-11-14 09:00:00          THEFT           $500 AND UNDER
2021-11-14 14:00:00          THEFT           $500 AND UNDER
2021-12-19 07:23:00        BATTERY                   SIMPLE

                    Location Description  Arrest  Domestic  Beat  …  Ward  \
Date                                                              …
2021-10-21 11:00:00      OTHER (SPECIFY)   False     False   412  …     8
2021-11-14 00:00:00            RESIDENCE   False      True   833  …    13
2021-11-14 04:00:00            RESIDENCE   False      True   911  …    12
2021-11-14 06:00:00    CONVENIENCE STORE   False     False   632  …     6
2021-11-14 09:00:00            APARTMENT   False      True   322  …     6
2021-11-14 14:00:00            RESIDENCE   False     False  1431  …     1
```

```
2021-12-19 07:23:00             SIDEWALK    True      True   921  …    15

                    Community Area  FBI Code X Coordinate  Y Coordinate  \
Date
2021-10-21 11:00:00              45        26    1188260.0     1849805.0
2021-11-14 00:00:00              65       08B    1152286.0     1856407.0
2021-11-14 04:00:00              58       08B    1154073.0     1879187.0
2021-11-14 06:00:00              44         6    1183071.0     1847869.0
2021-11-14 09:00:00              69         6    1178811.0     1858376.0
2021-11-14 14:00:00              22         6    1158583.0     1913745.0
2021-12-19 07:23:00              58       08B    1158067.0     1876425.0

                    Year        Updated On   Latitude  Longitude  \
Date
2021-10-21 11:00:00  2021   2/27/2022 15:46  41.742941 -87.585783
2021-11-14 00:00:00  2021  11/21/2021 15:48  41.761840 -87.717421
2021-11-14 04:00:00  2021  11/21/2021 15:48  41.824317 -87.710266
2021-11-14 06:00:00  2021  11/21/2021 15:48  41.737751 -87.604856
2021-11-14 09:00:00  2021  11/21/2021 15:48  41.766681 -87.620144
2021-11-14 14:00:00  2021  11/21/2021 15:48  41.919056 -87.692774
2021-12-19 07:23:00  2021   9/12/2022 16:45  41.816657 -87.695689

                                       Location
Date
2021-10-21 11:00:00   (41.74294124, -87.585783412)
2021-11-14 00:00:00  (41.761840209, -87.717420956)
2021-11-14 04:00:00  (41.824316537, -87.710266215)
2021-11-14 06:00:00  (41.737750767, -87.604855911)
2021-11-14 09:00:00   (41.766681066, -87.62014422)
2021-11-14 14:00:00  (41.919056144, -87.692774252)
2021-12-19 07:23:00   (41.81665685, -87.695688608)

[7 rows x 21 columns]
```

2.4.5 Now, let's get the larger dataset into our notebook

[22]:
```
filepath="/opt/datasets/ist652/Crimes/Crimes_20k.csv"
```

2.4.6 We can get the pandas dataframe correctly structured in one command (thanks to the work on the small version of the dataset). Let us parse the date and set date as a index. Here we are specifying with column to index using *index_col*

[23]:
```
crimes=pd.read_csv(filepath,parse_dates=[2], index_col=[2])
```

[24]:
```
crimes.tail()
```

[24]:

Date	ID	Case Number	Block	IUCR
2021-01-02 12:32:00	12259624	JE101261	046XX W VAN BUREN ST	4650
2021-01-06 12:18:00	12262739	JE104970	013XX S FAIRFIELD AVE	143A
2021-01-07 10:52:00	12263546	JE105815	036XX W POLK ST	143A
2021-01-01 01:00:00	12259461	JE100546	076XX S RHODES AVE	460
2021-01-02 23:02:00	12260076	JE101741	075XX S COLES AVE	460

Date	Primary Type	Description
2021-01-02 12:32:00	OTHER OFFENSE	SEX OFFENDER - FAIL TO REGISTER
2021-01-06 12:18:00	WEAPONS VIOLATION	UNLAWFUL POSSESSION - HANDGUN
2021-01-07 10:52:00	WEAPONS VIOLATION	UNLAWFUL POSSESSION - HANDGUN
2021-01-01 01:00:00	BATTERY	SIMPLE
2021-01-02 23:02:00	BATTERY	SIMPLE

Date	Location Description	Arrest	Domestic	Beat	…	Ward
2021-01-02 12:32:00	STREET	True	False	1131	…	24.0
2021-01-06 12:18:00	STREET	True	False	1023	…	28.0
2021-01-07 10:52:00	STREET	True	False	1133	…	24.0
2021-01-01 01:00:00	RESIDENCE	False	False	624	…	6.0
2021-01-02 23:02:00	APARTMENT	False	False	421	…	7.0

Date	Community Area	FBI Code	X Coordinate	Y Coordinate
2021-01-02 12:32:00	25	26	1145550.0	1897622.0
2021-01-06 12:18:00	29	15	1158247.0	1893603.0
2021-01-07 10:52:00	27	15	1152328.0	1896125.0
2021-01-01 01:00:00	69	08B	1181227.0	1854479.0
2021-01-02 23:02:00	43	08B	1195976.0	1855570.0

Date	Year	Updated On	Latitude	Longitude
2021-01-02 12:32:00	2021	1/16/2021 15:49	41.875070	-87.741068
2021-01-06 12:18:00	2021	1/16/2021 15:49	41.863792	-87.694560
2021-01-07 10:52:00	2021	1/16/2021 15:49	41.870831	-87.716222
2021-01-01 01:00:00	2021	1/16/2021 15:49	41.755932	-87.611409
2021-01-02 23:02:00	2021	1/16/2021 15:49	41.758573	-87.557322

Date	Location
2021-01-02 12:32:00	(41.875069995, -87.741068317)
2021-01-06 12:18:00	(41.863791576, -87.694559683)
2021-01-07 10:52:00	(41.870831024, -87.716221583)
2021-01-01 01:00:00	(41.755931978, -87.611408601)
2021-01-02 23:02:00	(41.758573421, -87.557321761)

```
[5 rows x 21 columns]
```

[25]:
```
crimes.index # Confirming our index column
```

[25]:
```
DatetimeIndex(['2021-12-19 07:23:00', '2021-04-16 20:45:00',
               '2021-10-21 11:00:00', '2021-11-14 06:00:00',
               '2021-11-14 04:00:00', '2021-11-14 09:00:00',
               '2021-11-14 14:00:00', '2021-11-14 00:00:00',
               '2021-01-09 15:59:00', '2021-04-28 20:18:00',
               …
               '2021-01-04 18:30:00', '2021-01-04 11:00:00',
               '2021-01-04 09:00:00', '2021-01-09 04:19:00',
               '2021-01-05 00:46:00', '2021-01-02 12:32:00',
               '2021-01-06 12:18:00', '2021-01-07 10:52:00',
               '2021-01-01 01:00:00', '2021-01-02 23:02:00'],
              dtype='datetime64[ns]', name='Date', length=20000, freq=None)
```

[26]:
```
crimes.info(verbose=True) # setting verbose to true gives us detailed
↪information.
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 20000 entries, 2021-12-19 07:23:00 to 2021-01-02 23:02:00
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    20000 non-null  int64
 1   Case Number           20000 non-null  object
 2   Block                 20000 non-null  object
 3   IUCR                  20000 non-null  object
 4   Primary Type          20000 non-null  object
 5   Description           20000 non-null  object
 6   Location Description  19936 non-null  object
 7   Arrest                20000 non-null  bool
 8   Domestic              20000 non-null  bool
 9   Beat                  20000 non-null  int64
 10  District              20000 non-null  int64
 11  Ward                  19999 non-null  float64
 12  Community Area        20000 non-null  int64
 13  FBI Code              20000 non-null  object
 14  X Coordinate          19789 non-null  float64
 15  Y Coordinate          19789 non-null  float64
 16  Year                  20000 non-null  int64
 17  Updated On            20000 non-null  object
 18  Latitude              19789 non-null  float64
 19  Longitude             19789 non-null  float64
 20  Location              19789 non-null  object
```

```
dtypes: bool(2), float64(5), int64(5), object(9)
memory usage: 3.1+ MB
```

3 RESAMPLING

3.0.1 Resampling is used to enhance your analysis. You can leverage if you have a datetime type index in your dataset. You can think of this as a sampling distribution. You can resample by time series frequecies like day, week, month ... (upsampling). Let's resample the dataset by day for the categories of crimes(the series "Primary Type")

3.0.2 (For more information on re-sampling, look at https://towardsdatascience.com/using-the-pandas-resample-function-a231144194c4)

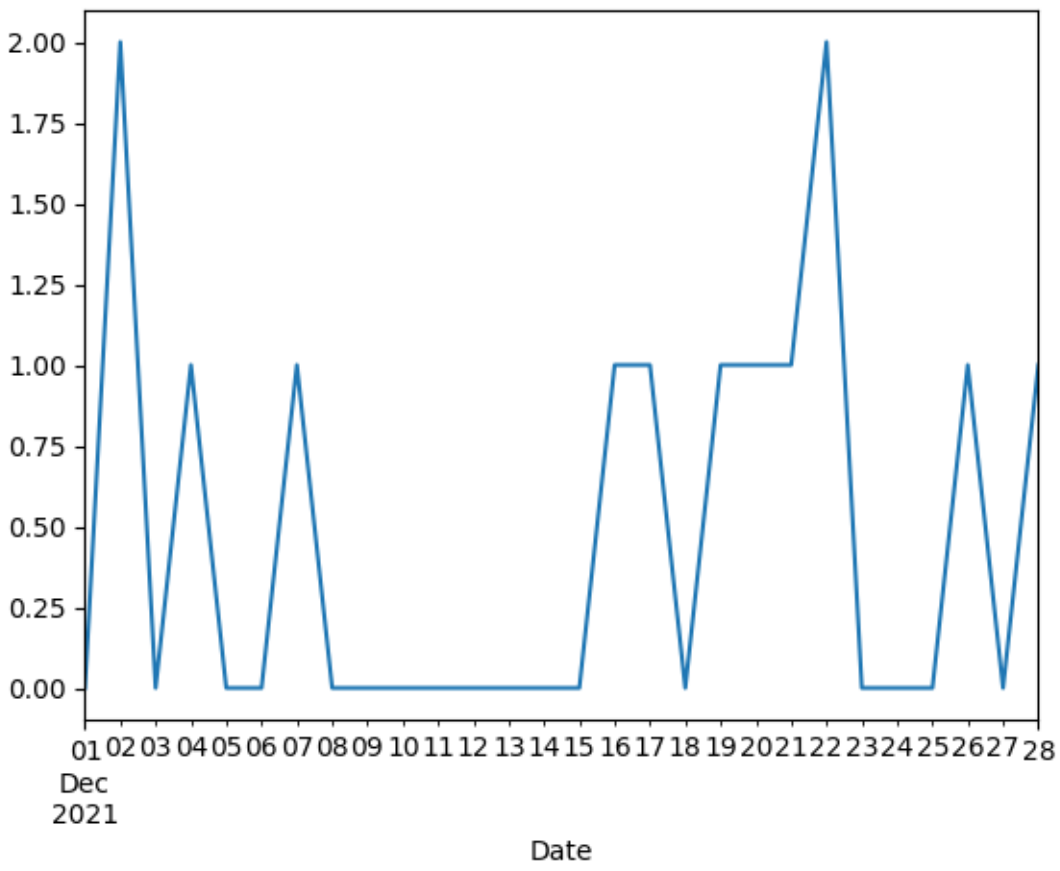
[27]:
```
daily_crimes = crimes['Primary Type'].resample('D').count()  #resample, count␣
 ↪crimes per day
daily_crimes.sort_index(inplace=True) # set your inplace as True
```

[28]:
```
daily_crimes.tail()
```

[28]:
```
Date
2021-12-24    0
2021-12-25    0
2021-12-26    1
2021-12-27    0
2021-12-28    1
Freq: D, Name: Primary Type, dtype: int64
```

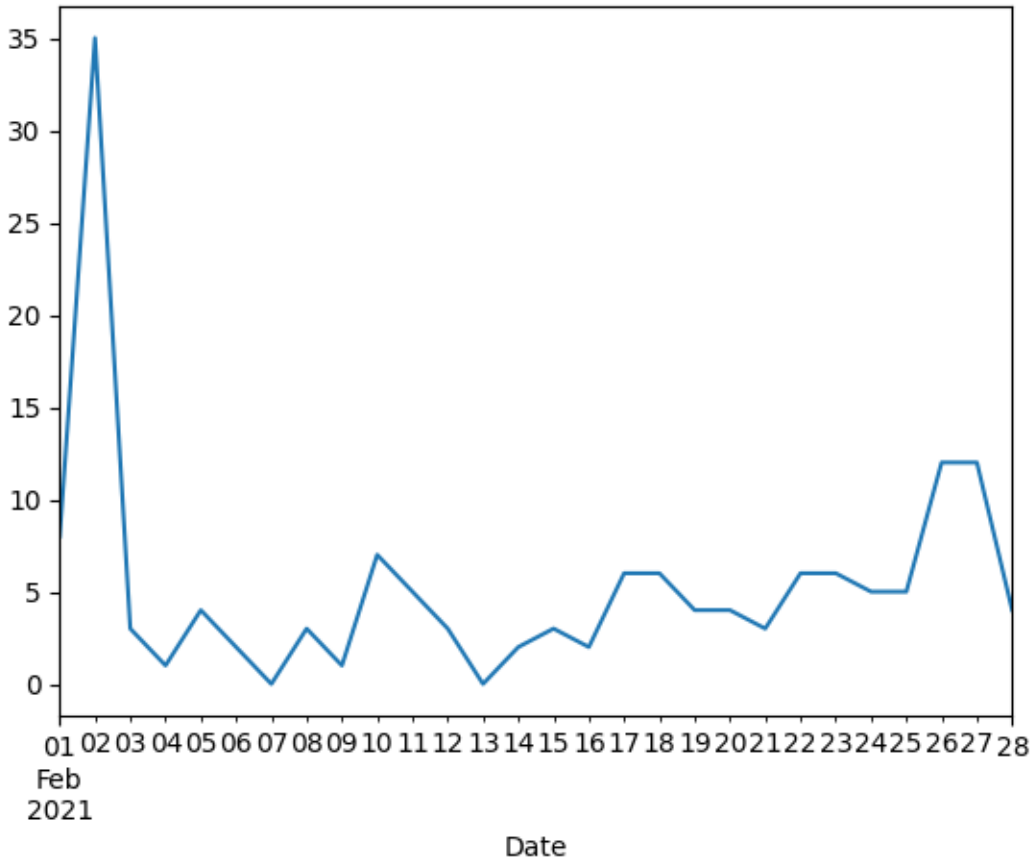
[29]:
```
daily_crimesDec = daily_crimes['2021-12']
daily_crimesDec.plot()  # Let us narrow to the month of December and plot
```

[29]:
```
<Axes: xlabel='Date'>
```



[30]:
```
daily_crimesFeb = daily_crimes['2021-02']
daily_crimesFeb.plot()   # let us look at the month of Feb and plot
```

[30]:
```
<Axes: xlabel='Date'>
```



4 PANDAS DATA STRUCTURES

4.0.1 Let's explain them on the data we have...

4.0.2 Series. Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.). So, the variable `crime_types` is a series of strings designating types of crimes and labeled with the crime's time of occurrence.

[31]:
```
crime_types = crimes['Primary Type']
crime_types
```

[31]:
```
Date
2021-12-19 07:23:00             BATTERY
2021-04-16 20:45:00               THEFT
2021-10-21 11:00:00       OTHER OFFENSE
2021-11-14 06:00:00               THEFT
2021-11-14 04:00:00             BATTERY
                                  ...
```

```
2021-01-02 12:32:00        OTHER OFFENSE
2021-01-06 12:18:00    WEAPONS VIOLATION
2021-01-07 10:52:00    WEAPONS VIOLATION
2021-01-01 01:00:00              BATTERY
2021-01-02 23:02:00              BATTERY
Name: Primary Type, Length: 20000, dtype: object
```

4.0.3 On the other hand, `daily_crimesDec` is a series with the number of crimes per day.

[32]:
```
daily_crimesDec
```

[32]:
```
Date
2021-12-01    0
2021-12-02    2
2021-12-03    0
2021-12-04    1
2021-12-05    0
2021-12-06    0
2021-12-07    1
2021-12-08    0
2021-12-09    0
2021-12-10    0
2021-12-11    0
2021-12-12    0
2021-12-13    0
2021-12-14    0
2021-12-15    0
2021-12-16    1
2021-12-17    1
2021-12-18    0
2021-12-19    1
2021-12-20    1
2021-12-21    1
2021-12-22    2
2021-12-23    0
2021-12-24    0
2021-12-25    0
2021-12-26    1
2021-12-27    0
2021-12-28    1
Freq: D, Name: Primary Type, dtype: int64
```

4.0.4 crime_type has an index which is a NumPy array. You will recall we set out date as our index

[33]:
```
crime_types.index
```

[33]:
```
DatetimeIndex(['2021-12-19 07:23:00', '2021-04-16 20:45:00',
               '2021-10-21 11:00:00', '2021-11-14 06:00:00',
               '2021-11-14 04:00:00', '2021-11-14 09:00:00',
               '2021-11-14 14:00:00', '2021-11-14 00:00:00',
               '2021-01-09 15:59:00', '2021-04-28 20:18:00',
               …
               '2021-01-04 18:30:00', '2021-01-04 11:00:00',
               '2021-01-04 09:00:00', '2021-01-09 04:19:00',
               '2021-01-05 00:46:00', '2021-01-02 12:32:00',
               '2021-01-06 12:18:00', '2021-01-07 10:52:00',
               '2021-01-01 01:00:00', '2021-01-02 23:02:00'],
              dtype='datetime64[ns]', name='Date', length=20000, freq=None)
```

4.0.5 … and array of values too

[34]:
```
crime_types.values
```

[34]:
```
array(['BATTERY', 'THEFT', 'OTHER OFFENSE', …, 'WEAPONS VIOLATION',
       'BATTERY', 'BATTERY'], dtype=object)
```

4.0.6 Remember, data alignment is intrinsic. If we sort the index, values are still correctly matched to corresponding dates.

[19]:
```
crime_types.sort_index(ascending=True).head(10)
```

[19]:
```
Date
2021-01-01 00:00:00                         THEFT
2021-01-01 00:00:00                       ASSAULT
2021-01-01 00:00:00                       BATTERY
2021-01-01 00:00:00                 OTHER OFFENSE
2021-01-01 00:00:00             DECEPTIVE PRACTICE
2021-01-01 00:00:00    OFFENSE INVOLVING CHILDREN
2021-01-01 00:00:00                CRIMINAL DAMAGE
2021-01-01 00:00:00             DECEPTIVE PRACTICE
2021-01-01 00:00:00                CRIMINAL DAMAGE
2021-01-01 00:01:00                       BATTERY
Name: Primary Type, dtype: object
```

[20]:
```
crime_types.sort_index(ascending=False).head()
```

[20]:
```
Date
2021-12-28 00:00:00                ROBBERY
```

```
2021-12-26 12:00:00    DECEPTIVE PRACTICE
2021-12-22 09:00:00    DECEPTIVE PRACTICE
2021-12-22 00:01:00    DECEPTIVE PRACTICE
2021-12-21 18:30:00               BATTERY
Name: Primary Type, dtype: object
```

4.0.7 The other date structure...

4.0.8 DataFrame

A table-like data structure. Essentially the whole crimes object is a DataFrame.

[35]:
```
crimes.head()
```

[35]:
```
                          ID Case Number                       Block  IUCR  \
Date
2021-12-19 07:23:00  12571973    JE482457          042XX S MOZART ST   460
2021-04-16 20:45:00  12343475    JE202728          056XX N RIDGE AVE   820
2021-10-21 11:00:00  12602803    JF125633   083XX S STONY ISLAND AVE  500E
2021-11-14 06:00:00  12540388    JE444591  086XX S COTTAGE GROVE AVE   850
2021-11-14 04:00:00  12541139    JE445494            034XX W 38TH ST   486

                      Primary Type              Description  \
Date
2021-12-19 07:23:00       BATTERY                   SIMPLE
2021-04-16 20:45:00         THEFT           $500 AND UNDER
2021-10-21 11:00:00  OTHER OFFENSE            EAVESDROPPING
2021-11-14 06:00:00         THEFT            ATTEMPT THEFT
2021-11-14 04:00:00       BATTERY  DOMESTIC BATTERY SIMPLE

                    Location Description  Arrest  Domestic  Beat  …  Ward  \
Date                                                              …
2021-12-19 07:23:00             SIDEWALK    True      True   921  …  15.0
2021-04-16 20:45:00      OTHER (SPECIFY)   False     False  2013  …  48.0
2021-10-21 11:00:00      OTHER (SPECIFY)   False     False   412  …   8.0
2021-11-14 06:00:00    CONVENIENCE STORE   False     False   632  …   6.0
2021-11-14 04:00:00            RESIDENCE   False      True   911  …  12.0

                     Community Area  FBI Code X Coordinate  Y Coordinate  \
Date
2021-12-19 07:23:00              58       08B    1158067.0     1876425.0
2021-04-16 20:45:00              77         6          NaN           NaN
2021-10-21 11:00:00              45        26    1188260.0     1849805.0
2021-11-14 06:00:00              44         6    1183071.0     1847869.0
2021-11-14 04:00:00              58       08B    1154073.0     1879187.0

                     Year        Updated On   Latitude  Longitude  \
Date
```

```
2021-12-19 07:23:00  2021   9/12/2022 16:45  41.816657 -87.695689
2021-04-16 20:45:00  2021   4/23/2021 16:51        NaN        NaN
2021-10-21 11:00:00  2021   2/27/2022 15:46  41.742941 -87.585783
2021-11-14 06:00:00  2021  11/21/2021 15:48  41.737751 -87.604856
2021-11-14 04:00:00  2021  11/21/2021 15:48  41.824317 -87.710266

                                          Location
Date
2021-12-19 07:23:00   (41.81665685, -87.695688608)
2021-04-16 20:45:00                            NaN
2021-10-21 11:00:00   (41.74294124, -87.585783412)
2021-11-14 06:00:00  (41.737750767, -87.604855911)
2021-11-14 04:00:00  (41.824316537, -87.710266215)

[5 rows x 21 columns]
```

It also has an index of rows:

[22]:
```
crimes.index
```

[22]:
```
DatetimeIndex(['2021-12-19 07:23:00', '2021-04-16 20:45:00',
               '2021-10-21 11:00:00', '2021-11-14 06:00:00',
               '2021-11-14 04:00:00', '2021-11-14 09:00:00',
               '2021-11-14 14:00:00', '2021-11-14 00:00:00',
               '2021-01-09 15:59:00', '2021-04-28 20:18:00',
               …
               '2021-01-04 18:30:00', '2021-01-04 11:00:00',
               '2021-01-04 09:00:00', '2021-01-09 04:19:00',
               '2021-01-05 00:46:00', '2021-01-02 12:32:00',
               '2021-01-06 12:18:00', '2021-01-07 10:52:00',
               '2021-01-01 01:00:00', '2021-01-02 23:02:00'],
              dtype='datetime64[ns]', name='Date', length=20000, freq=None)
```

… but also an index of columns

[23]:
```
crimes.columns
```

[23]:
```
Index(['ID', 'Case Number', 'Block', 'IUCR', 'Primary Type', 'Description',
       'Location Description', 'Arrest', 'Domestic', 'Beat', 'District',
       'Ward', 'Community Area', 'FBI Code', 'X Coordinate', 'Y Coordinate',
       'Year', 'Updated On', 'Latitude', 'Longitude', 'Location'],
      dtype='object')
```

4.1 TANGENT : CREATING PANDA OBJECTS

4.1.1 So far we've only been creating Pandas objects from CSV files using the `pd.read_csv` function, but we can also create new ones from other Python data structures (which will come in handy later when we get to scraping data from the web).

4.1.2 Let us create a series with a Python object

[24]:
```
s = pd.Series({'A': 15, 'B': 8, 'C': 6, 'D': 2, 'E': 10})
s
```

[24]:
```
A    15
B     8
C     6
D     2
E    10
dtype: int64
```

[25]:
```
df = pd.DataFrame({'age': s, 'test': {'A': 2.6, 'B': 69.27, 'C': 14.2, 'D': 8.
↪0, 'E': 5.93}})
df  # we set age to the series s
```

[25]:

	age	test
A	15	2.60
B	8	69.27
C	6	14.20
D	2	8.00
E	10	5.93

4.2 SELECTING VALUES

4.2.1 Now let's give a very quick overview of the many ways of data selection in Pandas.

4.2.2 Selecting by label ... Label-based using `.loc`

4.2.3 By single value – this actually returns a series when we select a single column/row in a DataFrame

[26]:
```
df.loc['A'] #selecting from the dataframe
```

[26]:
```
age     15.0
test     2.6
Name: A, dtype: float64
```

If we're selecting in a Series, we get back a scalar. Note the difference.

[27]:
```
s.loc['A'] #seleting from the series
```

[27]:
```
15
```

Or we can select by passing a list of labels to select (which keeps the same dimensionality)

[28]:
```
# using lists
df.loc[['A', 'B']]
```

[28]:
```
   age   test
A   15   2.60
B    8  69.27
```

[29]:
```
s.loc[['A', 'B']]
```

[29]:
```
A    15
B     8
dtype: int64
```

We can also select columns

[ ]:
```
df.loc[:, ['age']] # o select all rows (:) but only the age column from the␣
↪DataFrame df
```

[ ]:
```
   age
A   15
B    8
C    6
D    2
E   10
```

... or both by row and column

[30]:
```
df.loc[['A','B'], ['age']]
```

[30]:
```
   age
A   15
B    8
```

Shorthand we've been using already for selecting a whole column

[32]:
```
df['age']
```

[32]:
```
A    15
B     8
C     6
D     2
E    10
Name: age, dtype: int64
```

If the column name doesn't contain spaces or doesn't clash with any object method names we can also use the following syntax

[33]:
```
df.age
```

[33]:
```
A    15
B     8
C     6
D     2
E    10
Name: age, dtype: int64
```

4.3 Index-based using `.iloc`

4.3.1 Here we select using index numbers

[34]:
```
df.iloc[[0, 2, 3]]
```

[34]:

	age	test
A	15	2.6
C	6	14.2
D	2	8.0

[36]:
```
crimes.iloc[:10, [5]]  # give me the first 10 rows and the 5th column
```

[36]:

Date	Description
2021-12-19 07:23:00	SIMPLE
2021-04-16 20:45:00	$500 AND UNDER
2021-10-21 11:00:00	EAVESDROPPING
2021-11-14 06:00:00	ATTEMPT THEFT
2021-11-14 04:00:00	DOMESTIC BATTERY SIMPLE
2021-11-14 09:00:00	$500 AND UNDER
2021-11-14 14:00:00	$500 AND UNDER
2021-11-14 00:00:00	DOMESTIC BATTERY SIMPLE
2021-01-09 15:59:00	BY FIRE
2021-04-28 20:18:00	VIOLATE ORDER OF PROTECTION

4.3.2 Using iloc for slicing

Selecting by using ranges. Back to our small dataframe.

[36]:
```
df.iloc[2:5] # give me rows 2 to 4
```

[36]:

	age	test
C	6	14.20
D	2	8.00
E	10	5.93

Or for short

[37]:
```
df[2:5]
```

[37]:
```
   age   test
C    6  14.20
D    2   8.00
E   10   5.93
```

4.3.3 An awesome feature – slicing is datetime-aware

[38]:
```
daily_crimesDec['2021-12-20':].head(10) # give me crimes from Dec 20 to the end
```

[38]:
```
Date
2021-12-20    1
2021-12-21    1
2021-12-22    2
2021-12-23    0
2021-12-24    0
2021-12-25    0
2021-12-26    1
2021-12-27    0
2021-12-28    1
Freq: D, Name: Primary Type, dtype: int64
```

[39]:
```
daily_crimesDec['2021-12-01':] # crimes for the month of December
```

[39]:
```
Date
2021-12-01    0
2021-12-02    2
2021-12-03    0
2021-12-04    1
2021-12-05    0
2021-12-06    0
2021-12-07    1
2021-12-08    0
2021-12-09    0
2021-12-10    0
2021-12-11    0
2021-12-12    0
2021-12-13    0
2021-12-14    0
2021-12-15    0
2021-12-16    1
2021-12-17    1
2021-12-18    0
2021-12-19    1
```

```
2021-12-20    1
2021-12-21    1
2021-12-22    2
2021-12-23    0
2021-12-24    0
2021-12-25    0
2021-12-26    1
2021-12-27    0
2021-12-28    1
Freq: D, Name: Primary Type, dtype: int64
```

5 BOOLEAN INDEXING

5.0.1 Filter based on boolean expressions. Can be used on any expression that returns boolean values...

[40]:
```
daily_crimesDec > 1 # Here we are trying to determine the crimes for
```

[40]:
```
Date
2021-12-01    False
2021-12-02     True
2021-12-03    False
2021-12-04    False
2021-12-05    False
2021-12-06    False
2021-12-07    False
2021-12-08    False
2021-12-09    False
2021-12-10    False
2021-12-11    False
2021-12-12    False
2021-12-13    False
2021-12-14    False
2021-12-15    False
2021-12-16    False
2021-12-17    False
2021-12-18    False
2021-12-19    False
2021-12-20    False
2021-12-21    False
2021-12-22     True
2021-12-23    False
2021-12-24    False
2021-12-25    False
2021-12-26    False
2021-12-27    False
2021-12-28    False
```

```
Freq: D, Name: Primary Type, dtype: bool
```

… can also be used to index values where the boolean expression yields true.

[ ]:
```
daily_crimesDec[daily_crimesDec > 1].count() # days in Dec when crimes was > 1
```

[ ]:
```
2
```

6 RESHAPING

6.0.1 The input data format is often not the most useful for actually processing and visualising the information we're most interested in. Pandas' offers many useful methods for reshaping the data:

6.0.2 - `pivot_table` – select exactly the rows/columns you want

6.0.3 - `stack` / `unstack` – append to the index, adding more levels of a hierarchical Multindex

6.0.4 - `groupby` – similar to the SQL `Group By` command

6.0.5 - `resample` – like `groupby`, but for creating groups from time intervals (hours, days, weeks, etc.)

6.0.6 There are lots of operations you can express using these commands – see more documentation on them here. See https://pandas.pydata.org/pandas-docs/stable/user_guide/reshaping.html

[37]:
```
crimes_small.head() #using our small data set
```

[37]:
```
                          ID Case Number                       Block  IUCR  \
Date
2021-12-19 07:23:00  12571973    JE482457          042XX S MOZART ST   460
2021-04-16 20:45:00  12343475    JE202728          056XX N RIDGE AVE   820
2021-10-21 11:00:00  12602803    JF125633   083XX S STONY ISLAND AVE  500E
2021-11-14 06:00:00  12540388    JE444591  086XX S COTTAGE GROVE AVE   850
2021-11-14 04:00:00  12541139    JE445494            034XX W 38TH ST   486

                       Primary Type              Description  \
Date
2021-12-19 07:23:00        BATTERY                   SIMPLE
2021-04-16 20:45:00          THEFT           $500 AND UNDER
2021-10-21 11:00:00  OTHER OFFENSE            EAVESDROPPING
2021-11-14 06:00:00          THEFT            ATTEMPT THEFT
2021-11-14 04:00:00        BATTERY  DOMESTIC BATTERY SIMPLE

                     Location Description  Arrest  Domestic  Beat  …  Ward  \
Date                                                               …
2021-12-19 07:23:00             SIDEWALK    True      True   921  …    15
```

```
2021-04-16 20:45:00      OTHER (SPECIFY)   False     False  2013  …    48
2021-10-21 11:00:00      OTHER (SPECIFY)   False     False   412  …     8
2021-11-14 06:00:00   CONVENIENCE STORE    False     False   632  …     6
2021-11-14 04:00:00           RESIDENCE    False      True   911  …    12

                     Community Area  FBI Code X Coordinate  Y Coordinate  \
Date
2021-12-19 07:23:00              58       08B    1158067.0     1876425.0
2021-04-16 20:45:00              77         6           NaN           NaN
2021-10-21 11:00:00              45        26    1188260.0     1849805.0
2021-11-14 06:00:00              44         6    1183071.0     1847869.0
2021-11-14 04:00:00              58       08B    1154073.0     1879187.0

                     Year        Updated On   Latitude  Longitude  \
Date
2021-12-19 07:23:00  2021   9/12/2022 16:45  41.816657 -87.695689
2021-04-16 20:45:00  2021   4/23/2021 16:51        NaN        NaN
2021-10-21 11:00:00  2021   2/27/2022 15:46  41.742941 -87.585783
2021-11-14 06:00:00  2021  11/21/2021 15:48  41.737751 -87.604856
2021-11-14 04:00:00  2021  11/21/2021 15:48  41.824317 -87.710266

                                        Location
Date
2021-12-19 07:23:00   (41.81665685, -87.695688608)
2021-04-16 20:45:00                            NaN
2021-10-21 11:00:00   (41.74294124, -87.585783412)
2021-11-14 06:00:00  (41.737750767, -87.604855911)
2021-11-14 04:00:00  (41.824316537, -87.710266215)

[5 rows x 21 columns]
```

6.0.7 Adding another index

[38]:
```
crimes_small_mi = crimes_small.set_index(['Primary Type'], append=True)  #adds another index
crimes_small_mi  #the dataframe has two indices
```

[38]:
```
                                        ID Case Number  \
Date                 Primary Type
2021-12-19 07:23:00 BATTERY        12571973    JE482457
2021-04-16 20:45:00 THEFT          12343475    JE202728
2021-10-21 11:00:00 OTHER OFFENSE  12602803    JF125633
2021-11-14 06:00:00 THEFT          12540388    JE444591
2021-11-14 04:00:00 BATTERY        12541139    JE445494
2021-11-14 09:00:00 THEFT          12540496    JE444717
2021-11-14 14:00:00 THEFT          12542477    JE447028
2021-11-14 00:00:00 BATTERY        12541098    JE444580
```

```
2021-01-09 15:59:00 ARSON          12265790    JE108071

                                                      Block  IUCR  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY                  042XX S MOZART ST   460
2021-04-16 20:45:00 THEFT                    056XX N RIDGE AVE   820
2021-10-21 11:00:00 OTHER OFFENSE   083XX S STONY ISLAND AVE  500E
2021-11-14 06:00:00 THEFT          086XX S COTTAGE GROVE AVE   850
2021-11-14 04:00:00 BATTERY                    034XX W 38TH ST   486
2021-11-14 09:00:00 THEFT                   070XX S INDIANA AVE   820
2021-11-14 14:00:00 THEFT                    021XX N BINGHAM ST   820
2021-11-14 00:00:00 BATTERY                  072XX S HAMLIN AVE   486
2021-01-09 15:59:00 ARSON                  017XX W WINNEMAC AVE  1020

                                                 Description  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY                           SIMPLE
2021-04-16 20:45:00 THEFT                      $500 AND UNDER
2021-10-21 11:00:00 OTHER OFFENSE               EAVESDROPPING
2021-11-14 06:00:00 THEFT                       ATTEMPT THEFT
2021-11-14 04:00:00 BATTERY        DOMESTIC BATTERY SIMPLE
2021-11-14 09:00:00 THEFT                      $500 AND UNDER
2021-11-14 14:00:00 THEFT                      $500 AND UNDER
2021-11-14 00:00:00 BATTERY        DOMESTIC BATTERY SIMPLE
2021-01-09 15:59:00 ARSON                          BY FIRE

                                      Location Description  Arrest  Domestic  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY                        SIDEWALK    True      True
2021-04-16 20:45:00 THEFT                   OTHER (SPECIFY)   False     False
2021-10-21 11:00:00 OTHER OFFENSE           OTHER (SPECIFY)   False     False
2021-11-14 06:00:00 THEFT                CONVENIENCE STORE   False     False
2021-11-14 04:00:00 BATTERY                      RESIDENCE   False      True
2021-11-14 09:00:00 THEFT                        APARTMENT   False      True
2021-11-14 14:00:00 THEFT                        RESIDENCE   False     False
2021-11-14 00:00:00 BATTERY                      RESIDENCE   False      True
2021-01-09 15:59:00 ARSON          VEHICLE NON-COMMERCIAL    True      True

                                     Beat  District  Ward  Community Area  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY          921         9    15              58
2021-04-16 20:45:00 THEFT           2013        20    48              77
2021-10-21 11:00:00 OTHER OFFENSE    412         4     8              45
2021-11-14 06:00:00 THEFT            632         6     6              44
2021-11-14 04:00:00 BATTERY          911         9    12              58
2021-11-14 09:00:00 THEFT            322         3     6              69
2021-11-14 14:00:00 THEFT           1431        14     1              22
```

```
2021-11-14 00:00:00 BATTERY         833         8    13              65
2021-01-09 15:59:00 ARSON          2032        20    40               3

                                    FBI Code  X Coordinate  Y Coordinate  Year  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY              08B     1158067.0     1876425.0  2021
2021-04-16 20:45:00 THEFT                  6           NaN           NaN  2021
2021-10-21 11:00:00 OTHER OFFENSE         26     1188260.0     1849805.0  2021
2021-11-14 06:00:00 THEFT                  6     1183071.0     1847869.0  2021
2021-11-14 04:00:00 BATTERY              08B     1154073.0     1879187.0  2021
2021-11-14 09:00:00 THEFT                  6     1178811.0     1858376.0  2021
2021-11-14 14:00:00 THEFT                  6     1158583.0     1913745.0  2021
2021-11-14 00:00:00 BATTERY              08B     1152286.0     1856407.0  2021
2021-01-09 15:59:00 ARSON                  9     1164039.0     1933591.0  2021

                                          Updated On   Latitude  Longitude  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY          9/12/2022 16:45  41.816657 -87.695689
2021-04-16 20:45:00 THEFT            4/23/2021 16:51        NaN        NaN
2021-10-21 11:00:00 OTHER OFFENSE    2/27/2022 15:46  41.742941 -87.585783
2021-11-14 06:00:00 THEFT           11/21/2021 15:48  41.737751 -87.604856
2021-11-14 04:00:00 BATTERY         11/21/2021 15:48  41.824317 -87.710266
2021-11-14 09:00:00 THEFT           11/21/2021 15:48  41.766681 -87.620144
2021-11-14 14:00:00 THEFT           11/21/2021 15:48  41.919056 -87.692774
2021-11-14 00:00:00 BATTERY         11/21/2021 15:48  41.761840 -87.717421
2021-01-09 15:59:00 ARSON            8/12/2021 16:59  41.973401 -87.672166

                                                        Location
Date                Primary Type
2021-12-19 07:23:00 BATTERY          (41.81665685, -87.695688608)
2021-04-16 20:45:00 THEFT                                    NaN
2021-10-21 11:00:00 OTHER OFFENSE    (41.74294124, -87.585783412)
2021-11-14 06:00:00 THEFT           (41.737750767, -87.604855911)
2021-11-14 04:00:00 BATTERY         (41.824316537, -87.710266215)
2021-11-14 09:00:00 THEFT            (41.766681066, -87.62014422)
2021-11-14 14:00:00 THEFT           (41.919056144, -87.692774252)
2021-11-14 00:00:00 BATTERY         (41.761840209, -87.717420956)
2021-01-09 15:59:00 ARSON           (41.973401006, -87.672165851)
```

[39]:
```
#Preparation for crime counting analysis
crimes_small_mi.ID   # Access the ID column from the crimes_small_mi DataFrame
#crimes_small_mi['ID'] Another syntax
```

[39]:
```
Date                Primary Type
2021-12-19 07:23:00  BATTERY          12571973
2021-04-16 20:45:00  THEFT            12343475
2021-10-21 11:00:00  OTHER OFFENSE    12602803
```

```
2021-11-14 06:00:00  THEFT            12540388
2021-11-14 04:00:00  BATTERY          12541139
2021-11-14 09:00:00  THEFT            12540496
2021-11-14 14:00:00  THEFT            12542477
2021-11-14 00:00:00  BATTERY          12541098
2021-01-09 15:59:00  ARSON            12265790
Name: ID, dtype: int64
```

[ ]:

6.0.8 Similar to when we use a pivot table we use unstack to create new columns. To do this we need to have a multiindex. See https://www.w3resource.com/pandas/dataframe/dataframe-unstack.php

6.0.9 You can think of this as transposing

[40]:
```
# pivot or "unstack" a multi-indexed DataFrame by moving one of the levels of␣
 ↪the index (in this case, level=1) into the columns

crimes_small_mi.ID.unstack(level= 1)  # primary type and transposed to create␣
 ↪new columns.We set the level to 1... the second index
```

[40]:
```
Primary Type              ARSON     BATTERY  OTHER OFFENSE       THEFT
Date
2021-01-09 15:59:00  12265790.0         NaN            NaN         NaN
2021-04-16 20:45:00         NaN         NaN            NaN  12343475.0
2021-10-21 11:00:00         NaN         NaN     12602803.0         NaN
2021-11-14 00:00:00         NaN  12541098.0            NaN         NaN
2021-11-14 04:00:00         NaN  12541139.0            NaN         NaN
2021-11-14 06:00:00         NaN         NaN            NaN  12540388.0
2021-11-14 09:00:00         NaN         NaN            NaN  12540496.0
2021-11-14 14:00:00         NaN         NaN            NaN  12542477.0
2021-12-19 07:23:00         NaN  12571973.0            NaN         NaN
```

[41]:
```
crimes_small_mi.ID.unstack(level=1).resample('h').count()
```

[41]:
```
Primary Type         ARSON  BATTERY  OTHER OFFENSE  THEFT
Date
2021-01-09 15:00:00      1        0              0      0
2021-01-09 16:00:00      0        0              0      0
2021-01-09 17:00:00      0        0              0      0
2021-01-09 18:00:00      0        0              0      0
2021-01-09 19:00:00      0        0              0      0
…                      …        …              …      …
2021-12-19 03:00:00      0        0              0      0
2021-12-19 04:00:00      0        0              0      0
2021-12-19 05:00:00      0        0              0      0
```

```
2021-12-19 06:00:00      0        0              0      0
2021-12-19 07:00:00      0        1              0      0

[8249 rows x 4 columns]
```

6.0.10 Now let's do this for the big dataset

[42]:
```
crimes_mi = crimes.set_index(['Primary Type'], append=True) # setting the␣
↪second index
```

6.0.11 Bummer turns out the same command we used before would yield a ValueError: Index contains duplicate entries, cannot reshape

[43]:
```
crimes_mi.ID.unstack(level=1).resample('h').count().plot()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[43], line 1
----> 1 crimes_mi.ID.unstack(level=1).resample('h').count().plot()

File /opt/conda/lib/python3.11/site-packages/pandas/core/series.py:4459, in␣
↪Series.unstack(self, level, fill_value, sort)
   4414 """
   4415 Unstack, also known as pivot, Series with MultiIndex to produce␣
↪DataFrame.
   4416
   (…)
   4455 b    2    4
   4456 """
   4457 from pandas.core.reshape.reshape import unstack
-> 4459 return unstack(self, level, fill_value, sort)

File /opt/conda/lib/python3.11/site-packages/pandas/core/reshape/reshape.py:517␣
↪in unstack(obj, level, fill_value, sort)
    515 if is_1d_only_ea_dtype(obj.dtype):
    516     return _unstack_extension_series(obj, level, fill_value, sort=sort)
--> 517 unstacker = _Unstacker(
    518␣
↪    obj.index, level=level, constructor=obj._constructor_expanddim, sort=sort
    519 )
    520 return unstacker.get_result(
    521     obj._values, value_columns=None, fill_value=fill_value
    522 )

File /opt/conda/lib/python3.11/site-packages/pandas/core/reshape/reshape.py:154␣
↪in _Unstacker.__init__(self, index, level, constructor, sort)
```

```
    146 if num_cells > np.iinfo(np.int32).max:
    147     warnings.warn(
    148         f"The following operation may generate {num_cells} cells "
    149         f"in the resulting pandas object.",
    150         PerformanceWarning,
    151         stacklevel=find_stack_level(),
    152     )
--> 154 self._make_selectors()

File /opt/conda/lib/python3.11/site-packages/pandas/core/reshape/reshape.py:210
 ↪in _Unstacker._make_selectors(self)
    207 mask.put(selector, True)
    209 if mask.sum() < len(self.index):
--> 210     raise ValueError("Index contains duplicate entries, cannot reshape"
    212 self.group_index = comp_index
    213 self.mask = mask

ValueError: Index contains duplicate entries, cannot reshape
```

How to deal with duplicates – the same crime noted under the same minute. Let's see if we can sum those up first...

We use numpy to create a new array of ones.

[58]:
```
crimes_mi['ocurrences'] = np.ones(len(crimes_mi), dtype=int) # create and array
 ↪of ones call occurences
crimes_mi
```

[58]:

Date	Primary Type	ID	Case Number	Block	IUCR
2021-12-19 07:23:00	BATTERY	12571973	JE482457	042XX S MOZART ST	460
2021-04-16 20:45:00	THEFT	12343475	JE202728	056XX N RIDGE AVE	820
2021-10-21 11:00:00	OTHER OFFENSE	12602803	JF125633	083XX S STONY ISLAND AVE	500E
2021-11-14 06:00:00	THEFT	12540388	JE444591		
2021-11-14 04:00:00	BATTERY	12541139	JE445494		
…		…	…		
2021-01-02 12:32:00	OTHER OFFENSE	12259624	JE101261		
2021-01-06 12:18:00	WEAPONS VIOLATION	12262739	JE104970		
2021-01-07 10:52:00	WEAPONS VIOLATION	12263546	JE105815		
2021-01-01 01:00:00	BATTERY	12259461	JE100546		
2021-01-02 23:02:00	BATTERY	12260076	JE101741		

```
2021-11-14 06:00:00 THEFT              086XX S COTTAGE GROVE AVE   850
2021-11-14 04:00:00 BATTERY                    034XX W 38TH ST   486
…                                                           …   …
2021-01-02 12:32:00 OTHER OFFENSE           046XX W VAN BUREN ST  4650
2021-01-06 12:18:00 WEAPONS VIOLATION      013XX S FAIRFIELD AVE  143A
2021-01-07 10:52:00 WEAPONS VIOLATION            036XX W POLK ST  143A
2021-01-01 01:00:00 BATTERY                   076XX S RHODES AVE   460
2021-01-02 23:02:00 BATTERY                    075XX S COLES AVE   460

                                                        Description  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY                                    SIMPLE
2021-04-16 20:45:00 THEFT                                $500 AND UNDER
2021-10-21 11:00:00 OTHER OFFENSE                         EAVESDROPPING
2021-11-14 06:00:00 THEFT                                 ATTEMPT THEFT
2021-11-14 04:00:00 BATTERY                     DOMESTIC BATTERY SIMPLE
…                                                                    …
2021-01-02 12:32:00 OTHER OFFENSE      SEX OFFENDER - FAIL TO REGISTER
2021-01-06 12:18:00 WEAPONS VIOLATION    UNLAWFUL POSSESSION - HANDGUN
2021-01-07 10:52:00 WEAPONS VIOLATION    UNLAWFUL POSSESSION - HANDGUN
2021-01-01 01:00:00 BATTERY                                    SIMPLE
2021-01-02 23:02:00 BATTERY                                    SIMPLE

                                      Location Description  Arrest  Domestic  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY                         SIDEWALK    True      True
2021-04-16 20:45:00 THEFT                     OTHER (SPECIFY)   False     False
2021-10-21 11:00:00 OTHER OFFENSE             OTHER (SPECIFY)   False     False
2021-11-14 06:00:00 THEFT                   CONVENIENCE STORE   False     False
2021-11-14 04:00:00 BATTERY                         RESIDENCE   False      True
…                                                         …     …       …
2021-01-02 12:32:00 OTHER OFFENSE                     STREET    True     False
2021-01-06 12:18:00 WEAPONS VIOLATION                 STREET    True     False
2021-01-07 10:52:00 WEAPONS VIOLATION                 STREET    True     False
2021-01-01 01:00:00 BATTERY                         RESIDENCE   False     False
2021-01-02 23:02:00 BATTERY                         APARTMENT   False     False

                                       Beat  District  …  Community Area  \
Date                Primary Type                     …
2021-12-19 07:23:00 BATTERY             921         9  …              58
2021-04-16 20:45:00 THEFT              2013        20  …              77
2021-10-21 11:00:00 OTHER OFFENSE       412         4  …              45
2021-11-14 06:00:00 THEFT               632         6  …              44
2021-11-14 04:00:00 BATTERY             911         9  …              58
…                                       …       …  …               …
2021-01-02 12:32:00 OTHER OFFENSE      1131        11  …              25
2021-01-06 12:18:00 WEAPONS VIOLATION  1023        10  …              29
```

```
2021-01-07 10:52:00 WEAPONS VIOLATION  1133        11  …              27
2021-01-01 01:00:00 BATTERY             624         6  …              69
2021-01-02 23:02:00 BATTERY             421         4  …              43

                                    FBI Code X Coordinate  Y Coordinate  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY              08B    1158067.0     1876425.0
2021-04-16 20:45:00 THEFT                  6          NaN           NaN
2021-10-21 11:00:00 OTHER OFFENSE         26    1188260.0     1849805.0
2021-11-14 06:00:00 THEFT                  6    1183071.0     1847869.0
2021-11-14 04:00:00 BATTERY              08B    1154073.0     1879187.0
…                                          …          …             …
2021-01-02 12:32:00 OTHER OFFENSE         26    1145550.0     1897622.0
2021-01-06 12:18:00 WEAPONS VIOLATION     15    1158247.0     1893603.0
2021-01-07 10:52:00 WEAPONS VIOLATION     15    1152328.0     1896125.0
2021-01-01 01:00:00 BATTERY              08B    1181227.0     1854479.0
2021-01-02 23:02:00 BATTERY              08B    1195976.0     1855570.0

                                    Year        Updated On   Latitude  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY         2021   9/12/2022 16:45  41.816657
2021-04-16 20:45:00 THEFT           2021   4/23/2021 16:51        NaN
2021-10-21 11:00:00 OTHER OFFENSE   2021   2/27/2022 15:46  41.742941
2021-11-14 06:00:00 THEFT           2021  11/21/2021 15:48  41.737751
2021-11-14 04:00:00 BATTERY         2021  11/21/2021 15:48  41.824317
…                                      …                 …         …
2021-01-02 12:32:00 OTHER OFFENSE   2021   1/16/2021 15:49  41.875070
2021-01-06 12:18:00 WEAPONS VIOLATION  2021   1/16/2021 15:49  41.863792
2021-01-07 10:52:00 WEAPONS VIOLATION  2021   1/16/2021 15:49  41.870831
2021-01-01 01:00:00 BATTERY         2021   1/16/2021 15:49  41.755932
2021-01-02 23:02:00 BATTERY         2021   1/16/2021 15:49  41.758573

                                    Longitude  \
Date                Primary Type
2021-12-19 07:23:00 BATTERY          -87.695689
2021-04-16 20:45:00 THEFT                   NaN
2021-10-21 11:00:00 OTHER OFFENSE    -87.585783
2021-11-14 06:00:00 THEFT            -87.604856
2021-11-14 04:00:00 BATTERY          -87.710266
…                                         …
2021-01-02 12:32:00 OTHER OFFENSE    -87.741068
2021-01-06 12:18:00 WEAPONS VIOLATION -87.694560
2021-01-07 10:52:00 WEAPONS VIOLATION -87.716222
2021-01-01 01:00:00 BATTERY          -87.611409
2021-01-02 23:02:00 BATTERY          -87.557322

                                                          Location  \
```

```
Date                Primary Type
2021-12-19 07:23:00 BATTERY             (41.81665685, -87.695688608)
2021-04-16 20:45:00 THEFT                                        NaN
2021-10-21 11:00:00 OTHER OFFENSE       (41.74294124, -87.585783412)
2021-11-14 06:00:00 THEFT              (41.737750767, -87.604855911)
2021-11-14 04:00:00 BATTERY            (41.824316537, -87.710266215)
…                                                                …
2021-01-02 12:32:00 OTHER OFFENSE      (41.875069995, -87.741068317)
2021-01-06 12:18:00 WEAPONS VIOLATION  (41.863791576, -87.694559683)
2021-01-07 10:52:00 WEAPONS VIOLATION  (41.870831024, -87.716221583)
2021-01-01 01:00:00 BATTERY            (41.755931978, -87.611408601)
2021-01-02 23:02:00 BATTERY            (41.758573421, -87.557321761)

                                    ocurrences
Date                Primary Type
2021-12-19 07:23:00 BATTERY                  1
2021-04-16 20:45:00 THEFT                    1
2021-10-21 11:00:00 OTHER OFFENSE            1
2021-11-14 06:00:00 THEFT                    1
2021-11-14 04:00:00 BATTERY                  1
…                                          …
2021-01-02 12:32:00 OTHER OFFENSE            1
2021-01-06 12:18:00 WEAPONS VIOLATION        1
2021-01-07 10:52:00 WEAPONS VIOLATION        1
2021-01-01 01:00:00 BATTERY                  1
2021-01-02 23:02:00 BATTERY                  1

[20000 rows x 21 columns]
```

Add a new column ocurrences to the DataFrame crimes_mi, where every value in this column is set to 1. This is done using numpy to create an array of ones with the same length as the number of rows in crimes_mi. We group the ocurrences column by the first two levels of its index (levels 0 and 1), and then sum the occurrences for each group. Useful when you have multiIndex

This approach is often used when you want to aggregate counts later, e.g., for summing occurrences across different groups.

[60]:
```
crimes_mi['ocurrences'] = np.ones(len(crimes_mi), dtype=int) # np.ones()␣
 ↪returns a new array
crimes_grouped = crimes_mi.ocurrences.groupby(level=[0,1]).sum() #group by the␣
 ↪times in the index (level 0) and
                                                                  #also use the␣
 ↪columns (level 1)
crimes_grouped
```

[60]:
```
Date                 Primary Type
2021-01-01 00:00:00  ASSAULT                          1
                     BATTERY                          1
                     CRIMINAL DAMAGE                  2
                     DECEPTIVE PRACTICE               2
                     OFFENSE INVOLVING CHILDREN       1
                                                     ..
2021-12-21 18:30:00  BATTERY                          1
2021-12-22 00:01:00  DECEPTIVE PRACTICE               1
2021-12-22 09:00:00  DECEPTIVE PRACTICE               1
2021-12-26 12:00:00  DECEPTIVE PRACTICE               1
2021-12-28 00:00:00  ROBBERY                          1
Name: ocurrences, Length: 17134, dtype: int64
```

6.1 Now that the duplicates are addressed we can transpose the categories from the series 'Primary Type' into columns. So we *unstack at level =1* . This method transforms the second level of the index (level 1) of crimes_grouped into columns. After unstacking, each unique value from level 1 becomes a separate column. We resample by hour. Any values that are na we interpolate with 0 and make those integers

[62]:
```
crimes_by_type = crimes_grouped.unstack(level=1).fillna(0).resample('h').sum().
↪fillna(0).astype(int)
crimes_by_type.head()
```

[62]:
```
Primary Type         ARSON  ASSAULT  BATTERY  BURGLARY  \
Date
2021-01-01 00:00:00      0        1        9         0
2021-01-01 01:00:00      0        0       10         1
2021-01-01 02:00:00      0        0        5         1
2021-01-01 03:00:00      0        0        7         1
2021-01-01 04:00:00      0        1        6         0

Primary Type         CONCEALED CARRY LICENSE VIOLATION  CRIMINAL DAMAGE  \
Date
2021-01-01 00:00:00                                  0                9
2021-01-01 01:00:00                                  0                1
2021-01-01 02:00:00                                  0                0
2021-01-01 03:00:00                                  0                2
2021-01-01 04:00:00                                  0                2

Primary Type         CRIMINAL SEXUAL ASSAULT  CRIMINAL TRESPASS  \
Date
2021-01-01 00:00:00                        0                  0
2021-01-01 01:00:00                        0                  0
2021-01-01 02:00:00                        1                  0
```

```
2021-01-01 03:00:00                         0                 0
2021-01-01 04:00:00                         0                 0

Primary Type         DECEPTIVE PRACTICE  GAMBLING  …  OBSCENITY  \
Date                                               …
2021-01-01 00:00:00                   2         0  …          0
2021-01-01 01:00:00                   0         0  …          0
2021-01-01 02:00:00                   0         0  …          0
2021-01-01 03:00:00                   0         0  …          0
2021-01-01 04:00:00                   0         0  …          0

Primary Type         OFFENSE INVOLVING CHILDREN  OTHER OFFENSE  PROSTITUTION  \
Date
2021-01-01 00:00:00                           1              4             0
2021-01-01 01:00:00                           0              1             0
2021-01-01 02:00:00                           0              0             0
2021-01-01 03:00:00                           0              0             0
2021-01-01 04:00:00                           0              2             0

Primary Type         PUBLIC PEACE VIOLATION  ROBBERY  SEX OFFENSE  STALKING  \
Date
2021-01-01 00:00:00                       0        0            0         0
2021-01-01 01:00:00                       0        1            0         0
2021-01-01 02:00:00                       0        0            0         0
2021-01-01 03:00:00                       1        0            0         0
2021-01-01 04:00:00                       0        0            0         0

Primary Type         THEFT  WEAPONS VIOLATION
Date
2021-01-01 00:00:00      1                  6
2021-01-01 01:00:00      0                  2
2021-01-01 02:00:00      0                  0
2021-01-01 03:00:00      1                  1
2021-01-01 04:00:00      0                  0

[5 rows x 28 columns]
```

[74]:
```
crimes_by_type.loc[:, ['THEFT', 'HOMICIDE', 'ASSAULT']]
```

[74]:
```
Primary Type         THEFT  HOMICIDE  ASSAULT
Date
2021-01-01 00:00:00      1         0        1
2021-01-01 01:00:00      0         0        0
2021-01-01 02:00:00      0         0        0
2021-01-01 03:00:00      1         0        0
2021-01-01 04:00:00      0         0        1
…                      …         …        …
```

```
2021-12-27 20:00:00      0         0         0
2021-12-27 21:00:00      0         0         0
2021-12-27 22:00:00      0         0         0
2021-12-27 23:00:00      0         0         0
2021-12-28 00:00:00      0         0         0

[8665 rows x 3 columns]
```

6.1.1 Let us select by label resample by day and find the total of these crimes categories.

[92]:
```
crimes_by_type.loc[:, ['THEFT', 'HOMICIDE', 'ASSAULT']].resample('D').sum()
```

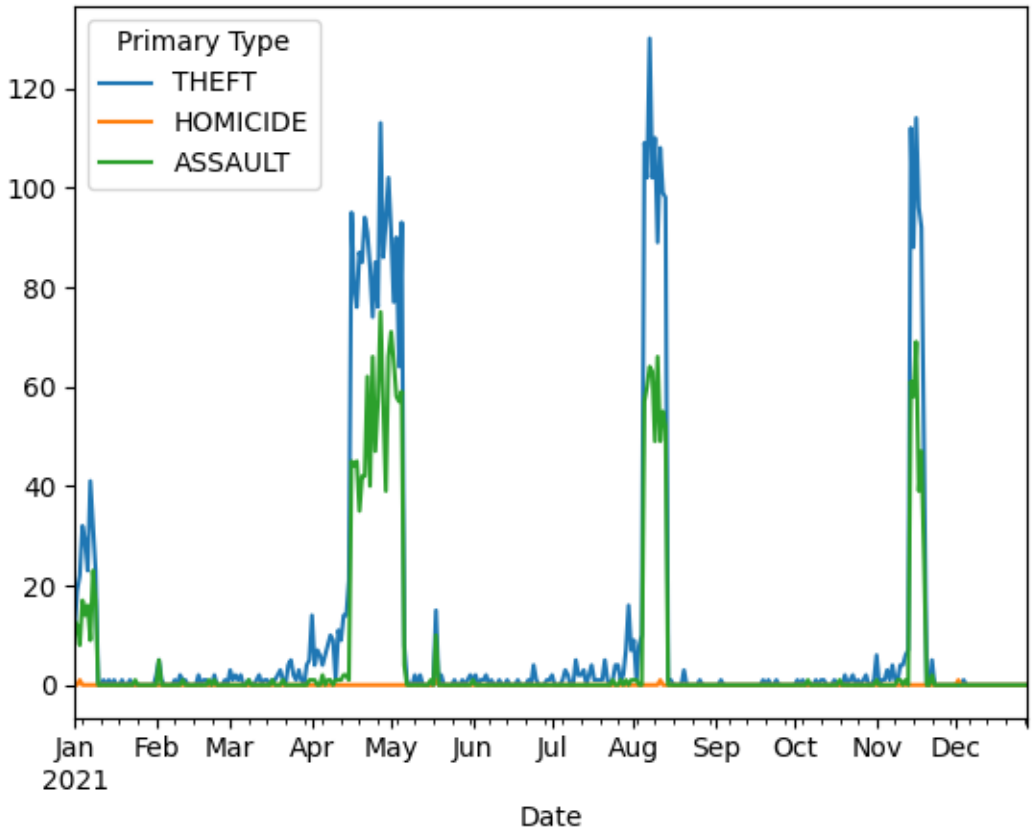
[92]:
```
Primary Type  THEFT  HOMICIDE  ASSAULT
Date
2021-01-01       11         0       13
2021-01-02       19         0       12
2021-01-03       22         1        8
2021-01-04       32         0       17
2021-01-05       29         0       14
…               …         …        …
2021-12-24        0         0        0
2021-12-25        0         0        0
2021-12-26        0         0        0
2021-12-27        0         0        0
2021-12-28        0         0        0

[362 rows x 3 columns]
```

[93]:
```
daily_crimes = crimes_by_type.loc[:, ['THEFT', 'HOMICIDE', 'ASSAULT']].
 ↪resample('D').sum() # let us look at some specific crimes by a day
selected_crimes_nov_19 = daily_crimes.loc['2021-11-19']
selected_crimes_nov_19
```

[93]:
```
Primary Type
THEFT       47
HOMICIDE     0
ASSAULT     25
Name: 2021-11-19 00:00:00, dtype: int64
```

[78]:
```
crimes_by_type.loc[:, ['THEFT', 'HOMICIDE', 'ASSAULT']].resample('D').sum().
 ↪plot() # let us plot this
```

[78]:
```
<Axes: xlabel='Date'>
```



7 USING INTERACT

7.1 Create an interactive widget

7.1.1 Just a taste – see the docs for more options.

[ ]:

[80]:
```
from ipywidgets import interact
```

[81]:
```
@interact(crime_type = crimes_by_type.columns) # dropdown
def plot_chart(crime_type='THEFT'): # theft is the default
    crimes_by_type.loc[:, crime_type]['2021'].resample('D').sum().
 ↪plot(title=crime_type)
```

```
interactive(children=(Dropdown(description='crime_type', index=26,␣
 ↪options=('ARSON', 'ASSAULT', 'BATTERY', 'BU…
```

[82]:
```
@interact(crime_type = crimes_by_type.columns) # dropdown save to a variable␣
 ↪crime_type
```

```
def plot_chart(crime_type): # Sort alphabetically
    crimes_by_type.loc[:, crime_type]['2021'].resample('D').sum().
 ↪plot(title=crime_type)
```

```
interactive(children=(Dropdown(description='crime_type', options=('ARSON',␣
 ↪'ASSAULT', 'BATTERY', 'BURGLARY', '…
```

[ ]: