Becoming a Data Analyst using Python, from scratch

An Introduction to Data Analysis

Najib Mozahem

May 14, 2023

1 Introduction

There is a lot of hype about data analysts in the media today, and rightfully so. Many companies are looking to recruit people who can analyse large quantities of data. Over the past years, the data that these companies have managed to collect has grown very quickly. Hidden deep in these vast amounts of data are insights that can help managers make better decisions. As such, companies are turning to data analysts to help then uncover these insights.

Data analysts need two skills. First, they need to write code. This does not mean that they need to be programmers. A common mistake is thinking that data analysts need to come from a computer science background. This is not true. Today, there are many libraries and packages that make the job of the analyst much easier when it comes to writing code. Most courses on data analysis start by providing the student with a background in programming. This course is different. The student will learn the coding as we go along. The course will focus on analytical skills. The coding will be introduced when needed, and even then, we will rely on libraries instead of writing complicated code.

Another skill that data analysts need is statistics. Unfortunately, it has been my experience that many data analysts do not have a good background in statistics. To them, data analysis is just about plotting graphs and training models. The result is that the work done by these analysts does not make much sense. The only way to understand which tool to use is to have a good grasp of these tools. This requires an understanding of statistics. Fortunately, you do not need to be a mathematician to understand statistics, at least not at the level required of a data analyst. Just like the coding part, this course will introduce the students to the various statistical concepts as we move along. In other words, as the course progresses, we will be talking about coding and statistics at the same time while introducing new concepts when they are needed.

The goal of this course is to provide the begining students with the necessary foundation that will allow them to start analysing data on their own using python. This course assumes no previous knowledge of python, statistics, or data structures. This course will not make you an expert. No single course can take you from A to Z, although many courses claim to do that. Instead, if you are thinking about pursuing a career as a data analyst, then this course is a good place to start. Concepts will be explained in a simple way by relying on examples. Concepts will be introduced on a need to know basis. There is no separate chapter about python. There is no separate chapter about statistics. Instead, the student will learn both at the same time. The further you go in this course, the more you will learn about python and statistics.

2 What You Need to Install

In order to follow this course you will need to be able to execute the code. You can do that using Anaconda. You have two options. Either install Anaconda on your computer and work locally, or you can work on the cloud without installing anything. Work on the cloud is the easiest option. You do not need to install or configure anything. All you have to do is go to the website https://www.anaconda.com/code-in-the-cloud, create an account, and start coding. That's all.

If you would rather download what you need and work on your computer, you can also do that easily. Go to https://www.anaconda.com/products/distribution and download and install Anaconda. Once you do that, start Anaconda by searching for "Anaconda" in your computer search bar. The symbol is a green hollow circle. Click on it and wait a little as your computer launches it. You should then see a screen similar to the one shown in the figure below.

In this navigator, launch JupyterLab and wait a few seconds. The notebook should open in your web browser.

3 Libraries

On your local computer, you can install libraries by going to the Environments tab in Anaconda navigator, as shown in the figure below.

Here you can see a list of installed libraries. As you notice, Anaconda comes with a long list of libraries that are already installed. In the event that you want to use a library that is not installed, you can simply choose the Not installed selection from the drop down box and then search for the library that you wish to install using the search bar on the right. Select that library and then click on Apply. This will install the library. Once a library is installed you can import it and use it in your notebook. We will be imorting the pandas library soon since it is the main library that we will use. To see that thi slibrary is already installed, select the Installed selection in the drop down box in Anaconda navigator and search for pandas. You should see pandas in the list.

4 A First Look at the Pandas Library

The reason why you do not need to be a professional programmer to become a data analyst is libraries. Libraries do a lot of the work for you. Over the past few years, many useful libraries have been developed. These libraries have made the job of the data analyst less about programming and more about analyzing the data.

Let us take a look at our first library, which is one of the most important. This library is called Pandas:

[1]:
```
import pandas as pd
```

In python, whenever you want to use a library, you will have to import it. This is accomplished using the import command. In the above command, we told python that we want to import the Pandas library and that we want to call it pd. We could have just typed import pandas but since whenever we use a library we will have to type its name, it is much easier to use a shorter name.

Once you execute the above code (press control+enter) you will have access to all the useful functions of this library. Let us use one of these useful functions to read a data set.

[2]:
```
first_data_set = pd.read_csv("https://data.cityofchicago.org/api/views/9hwr-2zxp/
↪rows.csv?accessType=DOWNLOAD&bom=true&format=true")
```

The read_csv() function is a very useful function that is part of the pandas library. Notice that to use a function in a library, you will have to type the name of the library (or the short alias that we chose for it), followed by a '.', and then followed by the function that you want to use. In our case the function is read_csv(). From the name, we know that this function allows us to read a data set that is in CSV format. The data set is stored in the web address that we provided to the function as input (inside the brackets). So basically, we are using the pandas library to download a data set that is stored as a CSV file.

In python, to save something we simply assign it to a variable. In the above command, the read_csv() function is going to return a data set. We saved the data set into the variable first_data_set. We now use this variable to refer to the data set. The data set has been stored as a panda dataframe that is called first_data_set. We can now use many functions associated with panda dataframes in order to look at and to analyse the data. One of these functions is the head() function.

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

[3]:

	ID	Case Number	Date	Block	IUCR	\
0	12789052	JF350580	08/09/2022 04:07:00 PM	014XX W ELMDALE AVE	0325	
1	12790581	JF352712	08/10/2022 04:00:00 PM	062XX S ARTESIAN AVE	0810	
2	12790652	JF352659	08/11/2022 10:00:00 AM	094XX S STATE ST	0810	
3	12796135	JF359082	08/15/2022 09:14:00 PM	048XX S KARLOV AVE	0560	
4	12795972	JF359058	08/16/2022 04:10:00 PM	015XX S HALSTED ST	0820	

	Primary Type	Description	Location Description	Arrest	Domestic	\
0	ROBBERY	VEHICULAR HIJACKING	STREET	True	False	
1	THEFT	OVER $500	STREET	False	False	
2	THEFT	OVER $500	STREET	False	True	
3	ASSAULT	SIMPLE	RESIDENCE	False	False	
4	THEFT	$500 AND UNDER	SIDEWALK	False	False	

	...	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	\
0	...	48.0	77	03	1165640.0	1939961.0	2022	
1	...	16.0	66	06	1161110.0	1863210.0	2022	
2	...	9.0	49	06	1177962.0	1842197.0	2022	
3	...	14.0	57	08A	1149844.0	1872244.0	2022	
4	...	11.0	28	06	1171290.0	1892413.0	2022	

```
              Updated On   Latitude  Longitude                      Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
3  01/03/2023 03:46:28 PM  41.805347 -87.725961  (41.805347066, -87.725961264)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)

[5 rows x 22 columns]
```

The head() function displays the first five rows of the dataframe. This is useful because as of yet we do not know what the data set contains. Looking at the output above, we see that the data set has 22 columns. We also see the names of each column. Take a look at the column named Primary Type. You can see that the data stored in this column describes crimes (ROBBERY, THEFT, ASSAULT,. . . etc). It seems that this data set has something to do with crimes. We also see a column called FBI Code. There is also a column named Location. Perhaps this columns provides the location where the crime took place. Maybe.

A very important thing to notice here is that some of the columns have numerical values while others have text values. This is important because different types of columns have different types, and the kind of analysis that we can perform depends on the values of the columns. We can look at the column names and types using the dtypes attribute:

[4]:
```
first_data_set.dtypes
```

[4]:
```
ID                        int64
Case Number              object
Date                     object
Block                    object
IUCR                     object
Primary Type             object
Description              object
Location Description     object
Arrest                     bool
Domestic                   bool
Beat                      int64
District                  int64
Ward                    float64
Community Area            int64
FBI Code                 object
X Coordinate            float64
Y Coordinate            float64
Year                      int64
Updated On               object
Latitude                float64
Longitude               float64
Location                 object
dtype: object
```

Notice that we called dtypes an attribute. As you can see from the command, there are no brack-

ets after dtypes. Functions must be followed by brackets, while attributes are not. The dtypes attribute of the dataframe lists the names of all columns and the type of each column. ID for example is of type int64. This simply means an integer (a number with no decimal points). The Case Number column is of type object which means that it is composed of text. The Arrest column is of type bool which means that it takes on two values, True or False. The X Coordinate column is of type float64 which is to say a number with a decimal point.

Let us take another look at the data set, but this time instead of displaying the top five rows, let us display the last five rows so that we can introduce a new function:

[5]:
```
first_data_set.tail()
```

[5]:
```
             ID Case Number                    Date                 Block  \
238312  12936285    JF526139  06/27/2022 10:05:00 AM    025XX N HALSTED ST
238313  12936301    JF526810  12/22/2022 06:00:00 PM  020XX W CORNELIA AVE
238314  12936397    JF526745  12/19/2022 02:00:00 PM   044XX N ROCKWELL ST
238315  12935341    JF525383  12/20/2022 06:45:00 AM  027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM     021XX W DEVON AVE

        IUCR        Primary Type                          Description  \
238312  1153  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT OVER $ 300
238313  1320     CRIMINAL DAMAGE                            TO VEHICLE
238314  0620            BURGLARY                        UNLAWFUL ENTRY
238315  0810               THEFT                             OVER $500
238316  0810               THEFT                             OVER $500

                       Location Description  Arrest  Domestic  ...  Ward  \
238312                                  NaN   False     False  ...  43.0
238313                               STREET   False     False  ...  32.0
238314                            APARTMENT   False     False  ...  47.0
238315                               STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
238312              7        11     1170513.0    1917030.0  2022
238313              5        14     1161968.0    1923233.0  2022
238314              4        05     1158237.0    1929586.0  2022
238315             29        06     1158071.0    1894595.0  2022
238316              2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
```

```
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[5 rows x 22 columns]
```

We see that the ID number is an integer, the Case Number is a combination of letters and numbers, the Arrest column takes on the values of True and False, and the X Coordinate column is numerical with a decimal point. We can now understand what different data types are stored in each column.

So far we have looked at the first and last five rows. We now know that the dataframe contains data about crimes. Actually, the data is about crimes in Chicago (https://data.cityofchicago.org), so let us give the dataframe a more meanigful name.

[6]:
```
chicago_crimes = first_data_set.copy()
```

Here, we used the copy() function in order to create a new copy of the dataframe and save it to the variable chicago_crimes. From now on we will use this new variable since the name makes more sense.

5 Looking More Closely at the Data

So far we have learned how to use the pandas library to download a data set from the internet, how to look at the column types of the data set, and how to display some of the rows in this data set. Notice that the coding has been very simple. This is because the pandas library makes our life much easier. This is why I said at the start that being a data analyst does not mean that you have to be an expert programmer.

We will now take further steps to understanding what this data set is about. It is useful to get some information about the data set, like the size of the data. We previously saw how to use the dtypes attribute to get the names and types of columns. There is a useful function that will provide us with this information in addition to the number of rows:

[7]:
```
chicago_crimes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238317 entries, 0 to 238316
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   ID                    238317 non-null  int64
 1   Case Number           238317 non-null  object
 2   Date                  238317 non-null  object
 3   Block                 238317 non-null  object
 4   IUCR                  238317 non-null  object
 5   Primary Type          238317 non-null  object
 6   Description           238317 non-null  object
```

```
 7   Location Description  237540 non-null  object
 8   Arrest                238317 non-null  bool
 9   Domestic              238317 non-null  bool
 10  Beat                  238317 non-null  int64
 11  District              238317 non-null  int64
 12  Ward                  238307 non-null  float64
 13  Community Area        238317 non-null  int64
 14  FBI Code              238317 non-null  object
 15  X Coordinate          233118 non-null  float64
 16  Y Coordinate          233118 non-null  float64
 17  Year                  238317 non-null  int64
 18  Updated On            238317 non-null  object
 19  Latitude              233118 non-null  float64
 20  Longitude             233118 non-null  float64
 21  Location              233118 non-null  object
dtypes: bool(2), float64(5), int64(5), object(10)
memory usage: 36.8+ MB
```

Let us look at the outout of the info() function. At the top we see that the RangeIndex is 237604 entries. This is the number of rows in the data set. Below that we see the number of columns (22). Below that we see a list of the columns. In this list, we see the number of the column (#), the name of the column (Column), and the type of the column (dtype). We also see something called Non-Null Count. This is an interesting column because it gives us the number of entries in each column that have a non-null value. But what is a null-value? A null-value is simply when the value is empty. It is like having an empty cell in an Excel sheet. It is usually the case that the data that we are dealing with is not complete. Not every single row will have all the information for every single column. Sometimes the information is missing, and the value is just empty, it is a null-value. Therefore, the Non-Null Count is the number of entries in the column that are not null, i.e. not missing. For the column ID for example, we see that there are 237604 non-null values. We know that the total number of rows in the dataframe is 237604. This means that for this column there are no missing values. Now look at the entry for the column X Coordinate. We see that there are 233156 non-null values. This means that there are 237604 - 233156 = 4448 null-values.

As data analysts, it is very important that we understand how much missing values exist in our data. The above output tells us the number of non-null values. We can calculate the number of null-values by subtracting the total number of rows and the number of null-values, as we did for the column X Coordinate. To make things simpler, we can use pandas functions to calculate the number of missing values for each column, instead of calculating the number of non-missing. To do that, we should first understand the isna() method:

[8]:
```
chicago_crimes.isna()
```

[8]:

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	\
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	

```
...       ...          ...    ...    ...    ...          ...          ...
238312  False        False  False  False  False        False        False
238313  False        False  False  False  False        False        False
238314  False        False  False  False  False        False        False
238315  False        False  False  False  False        False        False
238316  False        False  False  False  False        False        False

        Location Description  Arrest  Domestic  ...   Ward  Community Area  \
0                      False   False     False  ...  False           False
1                      False   False     False  ...  False           False
2                      False   False     False  ...  False           False
3                      False   False     False  ...  False           False
4                      False   False     False  ...  False           False
...                      ...     ...       ...  ...    ...             ...
238312                  True   False     False  ...  False           False
238313                 False   False     False  ...  False           False
238314                 False   False     False  ...  False           False
238315                 False   False     False  ...  False           False
238316                 False   False     False  ...  False           False

        FBI Code  X Coordinate  Y Coordinate   Year  Updated On  Latitude  \
0          False         False         False  False       False     False
1          False         False         False  False       False     False
2          False         False         False  False       False     False
3          False         False         False  False       False     False
4          False         False         False  False       False     False
...          ...           ...           ...    ...         ...       ...
238312     False         False         False  False       False     False
238313     False         False         False  False       False     False
238314     False         False         False  False       False     False
238315     False         False         False  False       False     False
238316     False         False         False  False       False     False

        Longitude  Location
0           False     False
1           False     False
2           False     False
3           False     False
4           False     False
...           ...       ...
238312      False     False
238313      False     False
238314      False     False
238315      False     False
238316      False     False

[238317 rows x 22 columns]
```

Looking at the output above, we see that all cells are now either True or False. This is because the isna() method displays a False when the cell is not missing (it has a value) and a True if the cell is null, i.e. empty. But how is this useful? Well, it is not useful by itself, but it becomes useful when we combine it with another function:

[9]:
```
chicago_crimes.isna().sum()
```

[9]:
```
ID                        0
Case Number               0
Date                      0
Block                     0
IUCR                      0
Primary Type              0
Description               0
Location Description    777
Arrest                    0
Domestic                  0
Beat                      0
District                  0
Ward                     10
Community Area            0
FBI Code                  0
X Coordinate           5199
Y Coordinate           5199
Year                      0
Updated On                0
Latitude               5199
Longitude              5199
Location               5199
dtype: int64
```

The sum() function is used to simply add all values in a column together. True is evaluated as 1 and False is evaluated as 0. So when we add the True and False values, we get the number of True values, and True represents a null-value (missing value). Looking at the output we see that for the ID column the sum is 0. This means that all values were False, so we were just adding zeros. This column has no null values. Now look at the column X Cordinate. The sum is 4448. This means that there were 4448 True values after using the isna() function. Each one of these True values was treated as a 1. So when we added them we got the total number of missing values.

As a summary, the isna() function returns the dataframe with True for missing values and False for non-missing values. The sum() method adds all values in each column together, with True being 1 and False being 0. Therefore, the sum for each column gives us the total number of null values in that column. In our case, some columns have no null values, while other columns have some null values.

So now we know how many rows are in the dataframe. We also know how many null values are in each of the columns. Let us now start looking at specific rows or specific columns.

5.1 Zooming in to a specific row

We know by now that the head() function displays the first five rows while the tail() function displays the last five rows. What if I wanted to look at a specific row? Let us look at the tenth row for example.

[10]:
```
chicago_crimes.loc[10]
```

[10]:
```
ID                                    12798495
Case Number                           JF361883
Date                      08/18/2022 06:00:00 PM
Block                            030XX E 80TH ST
IUCR                                       2820
Primary Type                     OTHER OFFENSE
Description                   TELEPHONE THREAT
Location Description                 RESIDENCE
Arrest                                   False
Domestic                                  True
Beat                                       422
District                                     4
Ward                                       7.0
Community Area                              46
FBI Code                                   08A
X Coordinate                         1197715.0
Y Coordinate                         1852504.0
Year                                      2022
Updated On                01/03/2023 03:46:28 PM
Latitude                             41.750117
Longitude                           -87.551051
Location          (41.75011688, -87.551050773)
Name: 10, dtype: object
```

The loc[] method allows us to look at a specific row. Here we used the function to retireve the information in row number 10. Notice that panda displays the output vertically because this makes looking at the values much easier. It is important to note that pandas starts numering rows with zero. So the first row is assigned the number 0, not 1. This means that when we look at the row with number 10, we are actually looking at the 11th row. To look at the 10th row we would use the following:

[11]:
```
chicago_crimes.loc[9]
```

[11]:
```
ID                                    12793378
Case Number                           JF355848
Date                      08/13/2022 10:45:00 PM
Block                        118XX S SANGAMON ST
IUCR                                       0560
Primary Type                           ASSAULT
Description                             SIMPLE
Location Description                    STREET
```

```
Arrest                                          False
Domestic                                         True
Beat                                              524
District                                            5
Ward                                             34.0
Community Area                                     53
FBI Code                                          08A
X Coordinate                                1172121.0
Y Coordinate                                1826361.0
Year                                             2022
Updated On                    01/03/2023 03:46:28 PM
Latitude                                    41.678977
Longitude                                  -87.645603
Location             (41.678976937, -87.645603032)
Name: 9, dtype: object
```

We can even use this method to look at more than one row. Using the ":" we can display the rows numbered 10, 11, and 12:

[12]:
```
chicago_crimes.loc[10:12]
```

[12]:
```
          ID Case Number                    Date               Block  IUCR  \
10  12798495    JF361883  08/18/2022 06:00:00 PM     030XX E 80TH ST  2820
11  12796933    JF360155  08/17/2022 07:00:00 AM     001XX E 42ND ST  0820
12  12794155    JF356683  08/14/2022 04:57:00 PM  050XX W BELDEN AVE  041A

     Primary Type           Description Location Description  Arrest  \
10  OTHER OFFENSE      TELEPHONE THREAT            RESIDENCE   False
11          THEFT        $500 AND UNDER               STREET   False
12        BATTERY  AGGRAVATED - HANDGUN             SIDEWALK   False

    Domestic  ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  \
10      True  ...   7.0              46       08A     1197715.0    1852504.0
11     False  ...   3.0              38        06     1177979.0    1877198.0
12     False  ...  36.0              19       04B     1142223.0    1914820.0

    Year              Updated On   Latitude  Longitude  \
10  2022  01/03/2023 03:46:28 PM  41.750117 -87.551051
11  2022  01/03/2023 03:46:28 PM  41.818349 -87.622623
12  2022  01/03/2023 03:46:28 PM  41.922326 -87.752857

                       Location
10  (41.75011688, -87.551050773)
11  (41.81834934, -87.622623461)
12  (41.922325648, -87.75285662)

[3 rows x 22 columns]
```

Here we used the loc[] method to display the three consecutive rows which are numbered 10, 11, and 12. The ":" is used to tell the method that we want to display a series of rows starting with the number to the left of the ":" and ending with the number to the right of the ":".

5.2 Zooming into a specific column

We can also look at a specific column:

[13]:
```
chicago_crimes["Primary Type"]
```

[13]:
```
0                    ROBBERY
1                      THEFT
2                      THEFT
3                    ASSAULT
4                      THEFT
                   ...
238312    DECEPTIVE PRACTICE
238313       CRIMINAL DAMAGE
238314              BURGLARY
238315                 THEFT
238316                 THEFT
Name: Primary Type, Length: 238317, dtype: object
```

We can also look at more than one column:

[14]:
```
chicago_crimes[["Primary Type", "Description"]]
```

[14]:
```
              Primary Type                          Description
0                  ROBBERY                    VEHICULAR HIJACKING
1                    THEFT                              OVER $500
2                    THEFT                              OVER $500
3                  ASSAULT                                 SIMPLE
4                    THEFT                        $500 AND UNDER
...                    ...                                   ...
238312  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT OVER $ 300
238313     CRIMINAL DAMAGE                           TO VEHICLE
238314            BURGLARY                        UNLAWFUL ENTRY
238315               THEFT                              OVER $500
238316               THEFT                              OVER $500

[238317 rows x 2 columns]
```

Notice that we used double square brackets. This is a very important point because it has to do with a very important data structure in python, and this structure is arrays.

5.2.1 Arrays

An array is a collection of items. Certain methods require an input. This input can be one value, or a collection of values. An array is a collection of values. It is how we group items together to

use them in a method. When we wanted to display only the column Primary Type, we just typed "Primary Type" between the brackets. But when we want to display more than one column type, we will have to send a collection of the names of the columns. This collection is the array. To collect items inside an array, we use the square brackets []. So to create an array that contains the names of two columns, we need to create an array that contains both names:

[15]:
```
my_array = ["Primary Type", "Description"]
```

We can then use this as the input to display the columns specified above:

[16]:
```
chicago_crimes[my_array]
```

[16]:

	Primary Type	Description
0	ROBBERY	VEHICULAR HIJACKING
1	THEFT	OVER $500
2	THEFT	OVER $500
3	ASSAULT	SIMPLE
4	THEFT	$500 AND UNDER
...	...	...
238312	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300
238313	CRIMINAL DAMAGE	TO VEHICLE
238314	BURGLARY	UNLAWFUL ENTRY
238315	THEFT	OVER $500
238316	THEFT	OVER $500

[238317 rows x 2 columns]

If we wanted to do this in one step, i.e. without creating an array, we can do the following:

[17]:
```
chicago_crimes[["Primary Type", "Description"]]
```

[17]:

	Primary Type	Description
0	ROBBERY	VEHICULAR HIJACKING
1	THEFT	OVER $500
2	THEFT	OVER $500
3	ASSAULT	SIMPLE
4	THEFT	$500 AND UNDER
...	...	...
238312	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300
238313	CRIMINAL DAMAGE	TO VEHICLE
238314	BURGLARY	UNLAWFUL ENTRY
238315	THEFT	OVER $500
238316	THEFT	OVER $500

[238317 rows x 2 columns]

Let us take another example. If we wanted to display only the column Date, then we can use the following:

[18]:
```
chicago_crimes["Date"]
```

[18]:
```
0         08/09/2022 04:07:00 PM
1         08/10/2022 04:00:00 PM
2         08/11/2022 10:00:00 AM
3         08/15/2022 09:14:00 PM
4         08/16/2022 04:10:00 PM
                  ...
238312    06/27/2022 10:05:00 AM
238313    12/22/2022 06:00:00 PM
238314    12/19/2022 02:00:00 PM
238315    12/20/2022 06:45:00 AM
238316    12/26/2022 10:30:00 PM
Name: Date, Length: 238317, dtype: object
```

If we also wanted to display the column Block as well, then we need to combine both columns in an array and use that array as the input:

[19]:
```
chicago_crimes[["Date", "Block"]]
```

[19]:
```
                        Date                 Block
0       08/09/2022 04:07:00 PM   014XX W ELMDALE AVE
1       08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE
2       08/11/2022 10:00:00 AM      094XX S STATE ST
3       08/15/2022 09:14:00 PM    048XX S KARLOV AVE
4       08/16/2022 04:10:00 PM    015XX S HALSTED ST
...                        ...                   ...
238312  06/27/2022 10:05:00 AM    025XX N HALSTED ST
238313  12/22/2022 06:00:00 PM  020XX W CORNELIA AVE
238314  12/19/2022 02:00:00 PM   044XX N ROCKWELL ST
238315  12/20/2022 06:45:00 AM  027XX W ROOSEVELT RD
238316  12/26/2022 10:30:00 PM     021XX W DEVON AVE

[238317 rows x 2 columns]
```

5.3 Zooming in to a certain row and column

We can combine all of the above together. Let us say that we want to look at the values of Date and Description in row 13. We can do the following:

[20]:
```
chicago_crimes[["Date", "Description"]].loc[13]
```

[20]:
```
Date           08/11/2022 06:00:00 PM
Description      AGGRAVATED - HANDGUN
Name: 13, dtype: object
```

Now we want to do the same for the rows numbered 15 to 20:

[21]:
```
chicago_crimes[["Date", "Description"]].loc[15:20]
```

[21]:

	Date	Description
15	08/18/2022 02:25:00 PM	OVER $500
16	08/18/2022 10:30:00 PM	SIMPLE
17	08/10/2022 10:55:00 PM	ARMED - HANDGUN
18	08/14/2022 01:45:00 PM	RETAIL THEFT
19	08/14/2022 10:00:00 PM	OVER $500
20	08/13/2022 01:30:00 AM	AUTOMOBILE

Now that we know how to zoom into a column, let us count the number of null values in a specific column. Previously, we saw how to count the number of null values for all columns, like so:

[22]:
```
chicago_crimes.isna().sum()
```

[22]:
```
ID                        0
Case Number               0
Date                      0
Block                     0
IUCR                      0
Primary Type              0
Description               0
Location Description    777
Arrest                    0
Domestic                  0
Beat                      0
District                  0
Ward                     10
Community Area            0
FBI Code                  0
X Coordinate           5199
Y Coordinate           5199
Year                      0
Updated On                0
Latitude               5199
Longitude              5199
Location               5199
dtype: int64
```

If we were just interested in a certain column, then we can do the following:

[23]:
```
chicago_crimes["Location Description"].isna().sum()
```

[23]:
```
777
```

If we wanted to do this for three columns:

[24]:
```
chicago_crimes[["Date", "Location Description", "Ward"]].isna().sum()
```

[24]:
```
Date                      0
Location Description    777
```

```
Ward                    10
dtype: int64
```

This brings us to a very important point. When we choose a subset of columns using the square brackets [], we are still dealing with a dataframe. This means that the functions that we used on the whole dataframe can also be used on a subset of the dataframe. This is why the isna() function worked here. The variable chicago_crimes is a pandas dataframe. The command `chicago_crimes[["Date", "Location Description", "Ward"]]` also returns a pandas dataframe. If you want to double check this, then create a new variable that we will use to store the subset of the data:

[25]:
```
chicago_crimes_subset = chicago_crimes[["Date", "Location Description", "Ward"]]
chicago_crimes_subset.head()
```

[25]:

	Date	Location Description	Ward
0	08/09/2022 04:07:00 PM	STREET	48.0
1	08/10/2022 04:00:00 PM	STREET	16.0
2	08/11/2022 10:00:00 AM	STREET	9.0
3	08/15/2022 09:14:00 PM	RESIDENCE	14.0
4	08/16/2022 04:10:00 PM	SIDEWALK	11.0

See? A subset of a dataframe is also a dataframe.

6 Removing Rows and Columns

Data analysts do not just look at data, they also need to modify the data. One of the most common things that a data analyst does is to delete rows and columns. Pandas comes with a useful function which allows us to delete either rows or columns. This function is drop(). For example, assume that we want to delete row number 3. In this case, we can simply delete it as follows:

[26]:
```
chicago_crimes.drop(3)
```

[26]:

	ID	Case Number	Date	Block	\
0	12789052	JF350580	08/09/2022 04:07:00 PM	014XX W ELMDALE AVE	
1	12790581	JF352712	08/10/2022 04:00:00 PM	062XX S ARTESIAN AVE	
2	12790652	JF352659	08/11/2022 10:00:00 AM	094XX S STATE ST	
4	12795972	JF359058	08/16/2022 04:10:00 PM	015XX S HALSTED ST	
5	12796817	JF359932	08/15/2022 03:00:00 PM	075XX S PHILLIPS AVE	
...	...	...	...	...	
238312	12936285	JF526139	06/27/2022 10:05:00 AM	025XX N HALSTED ST	
238313	12936301	JF526810	12/22/2022 06:00:00 PM	020XX W CORNELIA AVE	
238314	12936397	JF526745	12/19/2022 02:00:00 PM	044XX N ROCKWELL ST	
238315	12935341	JF525383	12/20/2022 06:45:00 AM	027XX W ROOSEVELT RD	
238316	12938501	JF523997	12/26/2022 10:30:00 PM	021XX W DEVON AVE	

	IUCR	Primary Type	Description	\
0	0325	ROBBERY	VEHICULAR HIJACKING	
1	0810	THEFT	OVER $500	

```
2       0810               THEFT                            OVER $500
4       0820               THEFT                       $500 AND UNDER
5       0810               THEFT                            OVER $500
...      ...                 ...                                   ...
238312  1153  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT OVER $ 300
238313  1320     CRIMINAL DAMAGE                           TO VEHICLE
238314  0620            BURGLARY                       UNLAWFUL ENTRY
238315  0810               THEFT                            OVER $500
238316  0810               THEFT                            OVER $500

                         Location Description  Arrest  Domestic  ...  Ward  \
0                                      STREET    True     False  ...  48.0
1                                      STREET   False     False  ...  16.0
2                                      STREET   False      True  ...   9.0
4                                    SIDEWALK   False     False  ...  11.0
5                                   RESIDENCE   False     False  ...   7.0
...                                       ...     ...       ...  ...   ...
238312                                    NaN   False     False  ...  43.0
238313                                 STREET   False     False  ...  32.0
238314                              APARTMENT   False     False  ...  47.0
238315                                 STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
...                ...       ...           ...          ...   ...
238312               7        11     1170513.0    1917030.0  2022
238313               5        14     1161968.0    1923233.0  2022
238314               4        05     1158237.0    1929586.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...        ...        ...
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
```

```
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                              ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[238316 rows x 22 columns]
```

Notice that we no longer have a row that is numbered 3. Let us now use the head() function to display the first five rows:

[27]:
```
chicago_crimes.head()
```

[27]:
```
         ID Case Number                    Date                 Block  IUCR  \
0  12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE  0325
1  12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE  0810
2  12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST  0810
3  12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE  0560
4  12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST  0820

  Primary Type          Description Location Description  Arrest  Domestic  \
0      ROBBERY  VEHICULAR HIJACKING               STREET    True     False
1        THEFT            OVER $500               STREET   False     False
2        THEFT            OVER $500               STREET   False      True
3      ASSAULT               SIMPLE            RESIDENCE   False     False
4        THEFT       $500 AND UNDER             SIDEWALK   False     False

   ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0  ...  48.0              77        03     1165640.0    1939961.0  2022
1  ...  16.0              66        06     1161110.0    1863210.0  2022
2  ...   9.0              49        06     1177962.0    1842197.0  2022
3  ...  14.0              57       08A     1149844.0    1872244.0  2022
4  ...  11.0              28        06     1171290.0    1892413.0  2022

              Updated On   Latitude  Longitude                     Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
3  01/03/2023 03:46:28 PM  41.805347 -87.725961  (41.805347066, -87.725961264)
```

```
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)

[5 rows x 22 columns]
```

Row number 3 is still there. What happened? What happened is that we did not "save" the drop action. We did not overwrite the chicago_crimes variable. When we used the `chicago_crimes.drop(3)` command, the result was the dataframe without row number 3, but the variable chicago_crimes is still the same. If we wanted to save our work, then we have two options. The first option is the following:

[28]:
```
chicago_crimes_new = chicago_crimes.drop(3)
chicago_crimes_new.head()
```

[28]:
```
         ID Case Number                    Date                  Block  IUCR  \
0  12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE  0325
1  12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE  0810
2  12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST  0810
4  12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST  0820
5  12796817    JF359932  08/15/2022 03:00:00 PM  075XX S PHILLIPS AVE  0810

  Primary Type          Description Location Description  Arrest  Domestic  \
0      ROBBERY  VEHICULAR HIJACKING               STREET    True     False
1        THEFT            OVER $500               STREET   False     False
2        THEFT            OVER $500               STREET   False      True
4        THEFT       $500 AND UNDER             SIDEWALK   False     False
5        THEFT            OVER $500            RESIDENCE   False     False

   ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0  ...  48.0              77        03     1165640.0    1939961.0  2022
1  ...  16.0              66        06     1161110.0    1863210.0  2022
2  ...   9.0              49        06     1177962.0    1842197.0  2022
4  ...  11.0              28        06     1171290.0    1892413.0  2022
5  ...   7.0              43        06     1193842.0    1855434.0  2022

               Updated On   Latitude  Longitude                      Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)
5  01/03/2023 03:46:28 PM  41.758253 -87.565147  (41.758252785, -87.565147001)

[5 rows x 22 columns]
```

Here, we created a new variable to save the returned dataframe.

Another option is to tell the drop() function to save the action. This is accomplished as follows:

[29]:
```
chicago_crimes.drop(3, inplace=True)
```

In the above command, we are telling drop() to remove row number 3, and we are also telling it to set the parameter inplace to True, which means that we want to save the returned dataframe in place of the old one. Let is now look at the dataframe:

[30]:
```
chicago_crimes.head()
```

[30]:
```
         ID Case Number                    Date                  Block  IUCR  \
0  12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE  0325
1  12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE  0810
2  12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST  0810
4  12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST  0820
5  12796817    JF359932  08/15/2022 03:00:00 PM  075XX S PHILLIPS AVE  0810

  Primary Type          Description Location Description  Arrest  Domestic  \
0      ROBBERY  VEHICULAR HIJACKING               STREET    True     False
1        THEFT             OVER $500               STREET   False     False
2        THEFT             OVER $500               STREET   False      True
4        THEFT       $500 AND UNDER             SIDEWALK   False     False
5        THEFT             OVER $500            RESIDENCE   False     False

   ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0  ...  48.0              77        03     1165640.0    1939961.0  2022
1  ...  16.0              66        06     1161110.0    1863210.0  2022
2  ...   9.0              49        06     1177962.0    1842197.0  2022
4  ...  11.0              28        06     1171290.0    1892413.0  2022
5  ...   7.0              43        06     1193842.0    1855434.0  2022

               Updated On   Latitude  Longitude                     Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)
5  01/03/2023 03:46:28 PM  41.758253 -87.565147  (41.758252785, -87.565147001)

[5 rows x 22 columns]
```

We now see that the row which is numbered 3 is no longer part of the dataframe.

We can also use the drop() method to delete more than one row:

[31]:
```
chicago_crimes.drop([4, 5])
```

[31]:
```
             ID Case Number                    Date  \
0       12789052    JF350580  08/09/2022 04:07:00 PM
1       12790581    JF352712  08/10/2022 04:00:00 PM
2       12790652    JF352659  08/11/2022 10:00:00 AM
6       12796316    JF358737  08/16/2022 11:57:00 AM
7       12795474    JF358356  08/15/2022 10:00:00 PM
...          ...         ...                     ...
```

```
238312  12936285    JF526139  06/27/2022 10:05:00 AM
238313  12936301    JF526810  12/22/2022 06:00:00 PM
238314  12936397    JF526745  12/19/2022 02:00:00 PM
238315  12935341    JF525383  12/20/2022 06:45:00 AM
238316  12938501    JF523997  12/26/2022 10:30:00 PM

                           Block  IUCR        Primary Type  \
0              014XX W ELMDALE AVE  0325             ROBBERY
1             062XX S ARTESIAN AVE  0810               THEFT
2                094XX S STATE ST  0810               THEFT
6              012XX N LA SALLE DR  0460             BATTERY
7       061XX S PARK SHORE EAST CT  0810               THEFT
...                            ...   ...                 ...
238312          025XX N HALSTED ST  1153  DECEPTIVE PRACTICE
238313        020XX W CORNELIA AVE  1320     CRIMINAL DAMAGE
238314         044XX N ROCKWELL ST  0620            BURGLARY
238315        027XX W ROOSEVELT RD  0810               THEFT
238316           021XX W DEVON AVE  0810               THEFT

                                Description  \
0                       VEHICULAR HIJACKING
1                                  OVER $500
2                                  OVER $500
6                                     SIMPLE
7                                  OVER $500
...                                      ...
238312  FINANCIAL IDENTITY THEFT OVER $ 300
238313                          TO VEHICLE
238314                      UNLAWFUL ENTRY
238315                             OVER $500
238316                             OVER $500

                         Location Description  Arrest  Domestic  ...  Ward  \
0                                      STREET    True     False  ...  48.0
1                                      STREET   False     False  ...  16.0
2                                      STREET   False      True  ...   9.0
6                                   APARTMENT   False     False  ...   2.0
7                                      STREET   False     False  ...   5.0
...                                       ...     ...       ...  ...   ...
238312                                    NaN   False     False  ...  43.0
238313                                 STREET   False     False  ...  32.0
238314                              APARTMENT   False     False  ...  47.0
238315                                 STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
```

```
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
6                    8       08B     1174910.0    1908582.0  2022
7                   42        06     1187557.0    1864569.0  2022
...                ...       ...           ...          ...   ...
238312               7        11     1170513.0    1917030.0  2022
238313               5        14     1161968.0    1923233.0  2022
238314               4        05     1158237.0    1929586.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
6       01/03/2023 03:46:28 PM  41.904538 -87.632942
7       01/03/2023 03:46:28 PM  41.783472 -87.587890
...                        ...        ...        ...
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                           Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
6       (41.904538325, -87.632942313)
7       (41.783471704, -87.587890396)
...                               ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[238314 rows x 22 columns]
```

Notice that we now use the square brackets []. As you recall, when we want to pass more than one value, then we need to combine them into an array. We can see from the output that the rows numbered 4 and 5 are no longer there (in addition to the row numbered 3 which we had previously deleted). Remember, the data frame chicago_crimes still contains these rows since we had not saved the drop action by using the inplace parameter.

We can also delete columns using the drop() function:

[32]:
```
chicago_crimes.drop(columns="ID")
```

[32]:

	Case Number	Date	Block	IUCR	\
0	JF350580	08/09/2022 04:07:00 PM	014XX W ELMDALE AVE	0325	
1	JF352712	08/10/2022 04:00:00 PM	062XX S ARTESIAN AVE	0810	
2	JF352659	08/11/2022 10:00:00 AM	094XX S STATE ST	0810	
4	JF359058	08/16/2022 04:10:00 PM	015XX S HALSTED ST	0820	
5	JF359932	08/15/2022 03:00:00 PM	075XX S PHILLIPS AVE	0810	
...	...	...	...	...	
238312	JF526139	06/27/2022 10:05:00 AM	025XX N HALSTED ST	1153	
238313	JF526810	12/22/2022 06:00:00 PM	020XX W CORNELIA AVE	1320	
238314	JF526745	12/19/2022 02:00:00 PM	044XX N ROCKWELL ST	0620	
238315	JF525383	12/20/2022 06:45:00 AM	027XX W ROOSEVELT RD	0810	
238316	JF523997	12/26/2022 10:30:00 PM	021XX W DEVON AVE	0810	

	Primary Type	Description	\
0	ROBBERY	VEHICULAR HIJACKING	
1	THEFT	OVER $500	
2	THEFT	OVER $500	
4	THEFT	$500 AND UNDER	
5	THEFT	OVER $500	
...	...	...	
238312	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	
238313	CRIMINAL DAMAGE	TO VEHICLE	
238314	BURGLARY	UNLAWFUL ENTRY	
238315	THEFT	OVER $500	
238316	THEFT	OVER $500	

	Location Description	Arrest	Domestic	Beat	...	\
0	STREET	True	False	2013	...	
1	STREET	False	False	825	...	
2	STREET	False	True	634	...	
4	SIDEWALK	False	False	1232	...	
5	RESIDENCE	False	False	421	...	
...	...	...	...	...	...	
238312	NaN	False	False	1935	...	
238313	STREET	False	False	1921	...	
238314	APARTMENT	False	False	1911	...	
238315	STREET	False	False	1135	...	
238316	PARKING LOT / GARAGE (NON RESIDENTIAL)	False	False	2413	...	

	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	\
0	48.0	77	03	1165640.0	1939961.0	2022	
1	16.0	66	06	1161110.0	1863210.0	2022	
2	9.0	49	06	1177962.0	1842197.0	2022	
4	11.0	28	06	1171290.0	1892413.0	2022	
5	7.0	43	06	1193842.0	1855434.0	2022	
...	...	...	...	...	...	...	
238312	43.0	7	11	1170513.0	1917030.0	2022	

```
238313  32.0              5        14    1161968.0     1923233.0  2022
238314  47.0              4        05    1158237.0     1929586.0  2022
238315  28.0             29        06    1158071.0     1894595.0  2022
238316  50.0              2        06    1160681.0     1942466.0  2022

                  Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...        ...        ...
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[238316 rows x 21 columns]
```

Notice that we instructued the function to drop columns and not rows but using the columns parameter. We can also drop more than one column. As you should know by now, to do that we need to pass the names as an array:

[33]:
```
chicago_crimes.drop(columns=["Date", "Block"])
```

[33]:
```
              ID Case Number  IUCR        Primary Type  \
0       12789052    JF350580  0325             ROBBERY
1       12790581    JF352712  0810               THEFT
2       12790652    JF352659  0810               THEFT
4       12795972    JF359058  0820               THEFT
5       12796817    JF359932  0810               THEFT
...          ...         ...   ...                 ...
238312  12936285    JF526139  1153  DECEPTIVE PRACTICE
238313  12936301    JF526810  1320     CRIMINAL DAMAGE
```

```
238314  12936397    JF526745  0620            BURGLARY
238315  12935341    JF525383  0810               THEFT
238316  12938501    JF523997  0810               THEFT

                              Description  \
0                       VEHICULAR HIJACKING
1                                OVER $500
2                                OVER $500
4                           $500 AND UNDER
5                                OVER $500
...                                    ...
238312  FINANCIAL IDENTITY THEFT OVER $ 300
238313                           TO VEHICLE
238314                       UNLAWFUL ENTRY
238315                           OVER $500
238316                           OVER $500

                         Location Description  Arrest  Domestic  Beat  \
0                                      STREET    True     False  2013
1                                      STREET   False     False   825
2                                      STREET   False      True   634
4                                    SIDEWALK   False     False  1232
5                                   RESIDENCE   False     False   421
...                                       ...     ...       ...   ...
238312                                    NaN   False     False  1935
238313                                 STREET   False     False  1921
238314                              APARTMENT   False     False  1911
238315                                 STREET   False     False  1135
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  2413

        District  Ward  Community Area FBI Code  X Coordinate  Y Coordinate  \
0             20  48.0              77       03     1165640.0     1939961.0
1              8  16.0              66       06     1161110.0     1863210.0
2              6   9.0              49       06     1177962.0     1842197.0
4             12  11.0              28       06     1171290.0     1892413.0
5              4   7.0              43       06     1193842.0     1855434.0
...          ...   ...             ...      ...           ...           ...
238312        19  43.0               7       11     1170513.0     1917030.0
238313        19  32.0               5       14     1161968.0     1923233.0
238314        19  47.0               4       05     1158237.0     1929586.0
238315        11  28.0              29       06     1158071.0     1894595.0
238316        24  50.0               2       06     1160681.0     1942466.0

        Year              Updated On   Latitude  Longitude  \
0       2022  01/03/2023 03:46:28 PM  41.990846 -87.666096
1       2022  01/03/2023 03:46:28 PM  41.780331 -87.684892
2       2022  01/03/2023 03:46:28 PM  41.722303 -87.623745
```

```
4       2022  01/03/2023 03:46:28 PM  41.860250 -87.646715
5       2022  01/03/2023 03:46:28 PM  41.758253 -87.565147
...      ...                      ...        ...        ...
238312  2022  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  2022  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  2022  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  2022  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  2022  03/22/2023 04:47:43 PM  41.997825 -87.684267

                          Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[238316 rows x 20 columns]
```

The above commands have not been saved. Remember, we were just viewing the result after the drop, but we have not been saving these actions. To make sure, display the first five rows of the dataframe:

[34]:
```
chicago_crimes.head()
```

[34]:
```
         ID Case Number                    Date                 Block  IUCR  \
0  12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE  0325
1  12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE  0810
2  12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST  0810
4  12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST  0820
5  12796817    JF359932  08/15/2022 03:00:00 PM  075XX S PHILLIPS AVE  0810

  Primary Type          Description Location Description  Arrest  Domestic  \
0      ROBBERY  VEHICULAR HIJACKING               STREET    True     False
1        THEFT            OVER $500               STREET   False     False
2        THEFT            OVER $500               STREET   False      True
4        THEFT       $500 AND UNDER             SIDEWALK   False     False
5        THEFT            OVER $500            RESIDENCE   False     False

   ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0  ...  48.0              77        03     1165640.0    1939961.0  2022
1  ...  16.0              66        06     1161110.0    1863210.0  2022
2  ...   9.0              49        06     1177962.0    1842197.0  2022
```

```
4  ...  11.0              28        06     1171290.0    1892413.0  2022
5  ...   7.0              43        06     1193842.0    1855434.0  2022

                Updated On   Latitude  Longitude                      Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)
5  01/03/2023 03:46:28 PM  41.758253 -87.565147  (41.758252785, -87.565147001)

[5 rows x 22 columns]
```

The columns ID, Date, and Block are still there. If we wanted to drop these columns and to overwrite the dataframe, then we need to use the inplace parameter:

[35]:
```
chicago_crimes.drop(columns=["ID","Date", "Block"], inplace=True)
chicago_crimes.head()
```

[35]:
```
  Case Number  IUCR Primary Type          Description Location Description  \
0    JF350580  0325      ROBBERY  VEHICULAR HIJACKING                STREET
1    JF352712  0810        THEFT            OVER $500                STREET
2    JF352659  0810        THEFT            OVER $500                STREET
4    JF359058  0820        THEFT       $500 AND UNDER              SIDEWALK
5    JF359932  0810        THEFT            OVER $500             RESIDENCE

   Arrest  Domestic  Beat  District  Ward  Community Area FBI Code  \
0    True     False  2013        20  48.0              77       03
1   False     False   825         8  16.0              66       06
2   False      True   634         6   9.0              49       06
4   False     False  1232        12  11.0              28       06
5   False     False   421         4   7.0              43       06

   X Coordinate  Y Coordinate  Year              Updated On   Latitude  \
0     1165640.0     1939961.0  2022  01/03/2023 03:46:28 PM  41.990846
1     1161110.0     1863210.0  2022  01/03/2023 03:46:28 PM  41.780331
2     1177962.0     1842197.0  2022  01/03/2023 03:46:28 PM  41.722303
4     1171290.0     1892413.0  2022  01/03/2023 03:46:28 PM  41.860250
5     1193842.0     1855434.0  2022  01/03/2023 03:46:28 PM  41.758253

   Longitude                      Location
0 -87.666096  (41.990846423, -87.666096144)
1 -87.684892  (41.780330681, -87.684891779)
2 -87.623745  (41.722303228, -87.623745129)
4 -87.646715   (41.860249838, -87.64671467)
5 -87.565147  (41.758252785, -87.565147001)
```

At this point I would like to reload the original data set. We were only deleting rows and columns just as an illustration. How do we do that? We can simply just re-download the data set using the

read_csv() method that is part of the pandas package:

[36]:
```
chicago_crimes = pd.read_csv("https://data.cityofchicago.org/api/views/9hwr-2zxp/
↪rows.csv?accessType=DOWNLOAD&bom=true&format=true")
chicago_crimes.head()
```

[36]:
```
         ID Case Number                    Date                 Block  IUCR  \
0  12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE  0325
1  12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE  0810
2  12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST  0810
3  12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE  0560
4  12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST  0820

  Primary Type          Description Location Description  Arrest  Domestic  \
0      ROBBERY  VEHICULAR HIJACKING               STREET    True     False
1        THEFT            OVER $500               STREET   False     False
2        THEFT            OVER $500               STREET   False      True
3      ASSAULT               SIMPLE            RESIDENCE   False     False
4        THEFT       $500 AND UNDER             SIDEWALK   False     False

   ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0  ...  48.0              77        03     1165640.0    1939961.0  2022
1  ...  16.0              66        06     1161110.0    1863210.0  2022
2  ...   9.0              49        06     1177962.0    1842197.0  2022
3  ...  14.0              57       08A     1149844.0    1872244.0  2022
4  ...  11.0              28        06     1171290.0    1892413.0  2022

              Updated On   Latitude  Longitude                     Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
3  01/03/2023 03:46:28 PM  41.805347 -87.725961  (41.805347066, -87.725961264)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)

[5 rows x 22 columns]
```

7 Using Conditions to Perform Operations

7.1 One Condition

Notice that so far, with very few lines of code, we are able to download data sets, look at the column types, check the total number of rows in the data set, check the number of missing values for each column, look at specific rows and columns, and drop certain rows and columns. Each of the above can be accomplished using a single line of code. We have not had to write long lines of codes. This is the beauty of libraries, particularly the pandas library.

While what we have done so far is useful and important, it definately is not enough. Yes, it is useful to look at a subset of the data, but do we really need to look at the 4th row? The 20th row?

A more useful thing to do would be to look at rows that satisfy a certain condition. This is perhaps one of the most common tasks that a data analyst does.

As an example, let us consider the type of of crime. We know that in our data set there is a column called Primary Type:

[37]:
```
chicago_crimes["Primary Type"]
```

[37]:
```
0                    ROBBERY
1                      THEFT
2                      THEFT
3                    ASSAULT
4                      THEFT
                   ...
238312    DECEPTIVE PRACTICE
238313       CRIMINAL DAMAGE
238314              BURGLARY
238315                 THEFT
238316                 THEFT
Name: Primary Type, Length: 238317, dtype: object
```

We see that the types of crimes vary. What if we just wanted to look at the records that were of type THEFT? This is possible using the following command:

[38]:
```
chicago_crimes[chicago_crimes["Primary Type"] == "THEFT"]
```

[38]:
```
               ID Case Number                    Date  \
1        12790581    JF352712  08/10/2022 04:00:00 PM
2        12790652    JF352659  08/11/2022 10:00:00 AM
4        12795972    JF359058  08/16/2022 04:10:00 PM
5        12796817    JF359932  08/15/2022 03:00:00 PM
7        12795474    JF358356  08/15/2022 10:00:00 PM
...           ...         ...                     ...
238307   12936322    JF526096  12/24/2022 01:30:00 PM
238310   12935276    JF524804  12/21/2022 11:05:00 PM
238311   12937237    JF527290  12/23/2022 08:41:00 AM
238315   12935341    JF525383  12/20/2022 06:45:00 AM
238316   12938501    JF523997  12/26/2022 10:30:00 PM

                             Block  IUCR Primary Type     Description  \
1             062XX S ARTESIAN AVE  0810        THEFT       OVER $500
2                 094XX S STATE ST  0810        THEFT       OVER $500
4               015XX S HALSTED ST  0820        THEFT  $500 AND UNDER
5             075XX S PHILLIPS AVE  0810        THEFT       OVER $500
7       061XX S PARK SHORE EAST CT  0810        THEFT       OVER $500
...                            ...   ...          ...             ...
238307         001XX W RANDOLPH ST  0860        THEFT    RETAIL THEFT
238310           042XX N MARINE DR  0810        THEFT       OVER $500
238311      064XX W Irving Park Rd  0860        THEFT    RETAIL THEFT
```

```
238315        027XX W ROOSEVELT RD  0810        THEFT       OVER $500
238316           021XX W DEVON AVE  0810        THEFT       OVER $500

                        Location Description  Arrest  Domestic  ...  Ward  \
1                                      STREET   False     False  ...  16.0
2                                      STREET   False      True  ...   9.0
4                                   SIDEWALK   False     False  ...  11.0
5                                   RESIDENCE   False     False  ...   7.0
7                                      STREET   False     False  ...   5.0
...                                       ...     ...       ...  ...   ...
238307                              DRUG STORE   False     False  ...  42.0
238310                                 STREET   False     False  ...  46.0
238311                    SMALL RETAIL STORE   False     False  ...  38.0
238315                                 STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
7                   42        06     1187557.0    1864569.0  2022
...                ...       ...           ...          ...   ...
238307              32        06     1175414.0    1901275.0  2022
238310               3        06     1171054.0    1928302.0  2022
238311              17        06     1132626.0    1925898.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                  Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
7       01/03/2023 03:46:28 PM  41.783472 -87.587890
...                        ...        ...        ...
238307  01/03/2023 03:46:28 PM  41.884476 -87.631311
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                           Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
```

```
7         (41.783471704, -87.587890396)
...                                 ...
238307    (41.884476226, -87.631310613)
238310    (41.958736385, -87.646526104)
238311    (41.952897791, -87.787860507)
238315    (41.866517317, -87.695178701)
238316    (41.997824802, -87.684266677)

[54757 rows x 22 columns]
```

There are several things to explain about the above command. First the double "=" sign. In programming languages, we use a double "=" sign to check if two things are equal. So to check if 2 is equal to 2 we type:

[39]:
```
2==2
```

[39]:
```
True
```

Therefore, when we type `chicago_crimes["Primary Type"] == "THEFT"` we are basically saying that we want only the rows that satisfy this condition:

[40]:
```
chicago_crimes["Primary Type"] == "THEFT"
```

[40]:
```
0         False
1          True
2          True
3         False
4          True
          ...
238312    False
238313    False
238314    False
238315     True
238316     True
Name: Primary Type, Length: 238317, dtype: bool
```

Notice the output. We have a True or a False for each row. True is for rows that have a value of "THEFT" in the column Primary Type, and False is for rows that have some other values. To see this, let us look at the first five rows:

[41]:
```
chicago_crimes.head()
```

[41]:
```
          ID Case Number                  Date                 Block  IUCR  \
0   12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE  0325
1   12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE  0810
2   12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST  0810
3   12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE  0560
4   12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST  0820
```

```
  Primary Type          Description Location Description  Arrest  Domestic  \
0      ROBBERY  VEHICULAR HIJACKING               STREET    True     False
1        THEFT            OVER $500               STREET   False     False
2        THEFT            OVER $500               STREET   False      True
3      ASSAULT               SIMPLE            RESIDENCE   False     False
4        THEFT       $500 AND UNDER             SIDEWALK   False     False

    ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0   ...  48.0              77        03     1165640.0    1939961.0  2022
1   ...  16.0              66        06     1161110.0    1863210.0  2022
2   ...   9.0              49        06     1177962.0    1842197.0  2022
3   ...  14.0              57       08A     1149844.0    1872244.0  2022
4   ...  11.0              28        06     1171290.0    1892413.0  2022

                Updated On   Latitude  Longitude                     Location
0  01/03/2023 03:46:28 PM  41.990846 -87.666096  (41.990846423, -87.666096144)
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
3  01/03/2023 03:46:28 PM  41.805347 -87.725961  (41.805347066, -87.725961264)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)

[5 rows x 22 columns]
```

See the primary type values? We have ROBBERY, THEFT, THEFT, ASSULT, THEFT, which results in FALSE, TRUE, TRUE, FALSE, TRUE.

We then use this to pick the subset of our dataframe. When the value is True, then the corresponding row will be part of the subset. When the value is False then the row will not be part of the subset:

[42]:
```
chicago_crimes[chicago_crimes["Primary Type"]=="THEFT"]
```

[42]:
```
              ID Case Number                    Date  \
1       12790581    JF352712  08/10/2022 04:00:00 PM
2       12790652    JF352659  08/11/2022 10:00:00 AM
4       12795972    JF359058  08/16/2022 04:10:00 PM
5       12796817    JF359932  08/15/2022 03:00:00 PM
7       12795474    JF358356  08/15/2022 10:00:00 PM
...          ...         ...                     ...
238307  12936322    JF526096  12/24/2022 01:30:00 PM
238310  12935276    JF524804  12/21/2022 11:05:00 PM
238311  12937237    JF527290  12/23/2022 08:41:00 AM
238315  12935341    JF525383  12/20/2022 06:45:00 AM
238316  12938501    JF523997  12/26/2022 10:30:00 PM

                           Block  IUCR Primary Type     Description  \
1             062XX S ARTESIAN AVE  0810        THEFT       OVER $500
2                 094XX S STATE ST  0810        THEFT       OVER $500
```

```
4               015XX S HALSTED ST  0820        THEFT  $500 AND UNDER
5             075XX S PHILLIPS AVE  0810        THEFT       OVER $500
7       061XX S PARK SHORE EAST CT  0810        THEFT       OVER $500
...                            ...   ...          ...             ...
238307         001XX W RANDOLPH ST  0860        THEFT    RETAIL THEFT
238310           042XX N MARINE DR  0810        THEFT       OVER $500
238311      064XX W Irving Park Rd  0860        THEFT    RETAIL THEFT
238315        027XX W ROOSEVELT RD  0810        THEFT       OVER $500
238316           021XX W DEVON AVE  0810        THEFT       OVER $500

                        Location Description  Arrest  Domestic  ...  Ward  \
1                                      STREET   False     False  ...  16.0
2                                      STREET   False      True  ...   9.0
4                                   SIDEWALK   False     False  ...  11.0
5                                   RESIDENCE   False     False  ...   7.0
7                                      STREET   False     False  ...   5.0
...                                       ...     ...       ...  ...   ...
238307                              DRUG STORE   False     False  ...  42.0
238310                                 STREET   False     False  ...  46.0
238311                     SMALL RETAIL STORE   False     False  ...  38.0
238315                                 STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
7                   42        06     1187557.0    1864569.0  2022
...                ...       ...           ...          ...   ...
238307              32        06     1175414.0    1901275.0  2022
238310               3        06     1171054.0    1928302.0  2022
238311              17        06     1132626.0    1925898.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                   Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
7       01/03/2023 03:46:28 PM  41.783472 -87.587890
...                        ...       ...        ...
238307  01/03/2023 03:46:28 PM  41.884476 -87.631311
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
```

```
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
7       (41.783471704, -87.587890396)
...                               ...
238307  (41.884476226, -87.631310613)
238310  (41.958736385, -87.646526104)
238311  (41.952897791, -87.787860507)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[54757 rows x 22 columns]
```

We can save this result in a new dataframe that contains only theft crimes. Remember, a subset of a dataframe is a dataframe:

[43]:
```
chicago_crimes_thefts = chicago_crimes[chicago_crimes["Primary Type"]=="THEFT"]
chicago_crimes_thefts.head()
```

[43]:
```
         ID Case Number                    Date                        Block  \
1  12790581    JF352712  08/10/2022 04:00:00 PM          062XX S ARTESIAN AVE
2  12790652    JF352659  08/11/2022 10:00:00 AM              094XX S STATE ST
4  12795972    JF359058  08/16/2022 04:10:00 PM            015XX S HALSTED ST
5  12796817    JF359932  08/15/2022 03:00:00 PM          075XX S PHILLIPS AVE
7  12795474    JF358356  08/15/2022 10:00:00 PM  061XX S PARK SHORE EAST CT

   IUCR Primary Type     Description Location Description  Arrest  Domestic  \
1  0810        THEFT       OVER $500               STREET   False     False
2  0810        THEFT       OVER $500               STREET   False      True
4  0820        THEFT  $500 AND UNDER             SIDEWALK   False     False
5  0810        THEFT       OVER $500            RESIDENCE   False     False
7  0810        THEFT       OVER $500               STREET   False     False

   ...  Ward  Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1  ...  16.0              66        06     1161110.0    1863210.0  2022
2  ...   9.0              49        06     1177962.0    1842197.0  2022
4  ...  11.0              28        06     1171290.0    1892413.0  2022
5  ...   7.0              43        06     1193842.0    1855434.0  2022
7  ...   5.0              42        06     1187557.0    1864569.0  2022

              Updated On   Latitude  Longitude                      Location
1  01/03/2023 03:46:28 PM  41.780331 -87.684892  (41.780330681, -87.684891779)
2  01/03/2023 03:46:28 PM  41.722303 -87.623745  (41.722303228, -87.623745129)
4  01/03/2023 03:46:28 PM  41.860250 -87.646715   (41.860249838, -87.64671467)
```

```
5  01/03/2023 03:46:28 PM  41.758253 -87.565147  (41.758252785, -87.565147001)
7  01/03/2023 03:46:28 PM  41.783472 -87.587890  (41.783471704, -87.587890396)

[5 rows x 22 columns]
```

Let us look at another example. What if I wanted a dataframe that contained only those crimes that resulted in an Arrest. To do that, we might type the following:

[44]:
```
chicago_crimes[chicago_crimes["Arrest"]=="True"]
```

[44]:
```
Empty DataFrame
Columns: [ID, Case Number, Date, 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]
Index: []

[0 rows x 22 columns]
```

This seems strange. Is it possible that the data set does not contain any crime that resulted in an arrest? Actually, we have done a simple but serious mistake. In the condition above, we are checking to see if the value of Arrest is equal to the text "True", but if you remember well, this column is of type bool (which means it is either traue or false). Let us see this:

[45]:
```
chicago_crimes.dtypes
```

[45]:
```
ID                      int64
Case Number            object
Date                   object
Block                  object
IUCR                   object
Primary Type           object
Description            object
Location Description   object
Arrest                   bool
Domestic                 bool
Beat                    int64
District                int64
Ward                  float64
Community Area          int64
FBI Code               object
X Coordinate          float64
Y Coordinate          float64
Year                    int64
Updated On             object
Latitude              float64
Longitude             float64
Location               object
```

```
dtype: object
```

Since we are just interested in the Arrest column, we could have typed the following:

[46]:
```
chicago_crimes["Arrest"].dtypes
```

[46]:
```
dtype('bool')
```

In either case, notice that the type is bool. This is not an object. This is why the condition should have been written the following way:

[47]:
```
chicago_crimes[chicago_crimes["Arrest"]==True]
```

[47]:

	ID	Case Number	Date	Block	\
0	12789052	JF350580	08/09/2022 04:07:00 PM	014XX W ELMDALE AVE	
37	12796098	JF359198	08/16/2022 05:20:00 PM	033XX W ROOSEVELT RD	
61	12801503	JF365550	08/22/2022 05:50:00 AM	086XX S KARLOV AVE	
67	12802877	JF367040	08/23/2022 08:45:00 AM	056XX N MOZART ST	
72	12812318	JF378344	09/01/2022 12:25:00 AM	006XX N MICHIGAN AVE	
...	...	...	...	...	
237927	12931560	JF521115	12/23/2022 07:06:00 PM	001XX E SUPERIOR ST	
237936	12925588	JF513886	12/17/2022 11:22:00 AM	039XX W FILLMORE ST	
237967	12925155	JF513324	12/16/2022 08:12:00 PM	045XX S WESTERN BLVD	
237999	12938989	JG100842	12/04/2022 07:00:00 PM	011XX N CLARK ST	
238187	12935231	JF519782	10/29/2022 03:30:00 AM	014XX W LAKE ST	

	IUCR	Primary Type	Description	\
0	0325	ROBBERY	VEHICULAR HIJACKING	
37	0560	ASSAULT	SIMPLE	
61	0910	MOTOR VEHICLE THEFT	AUTOMOBILE	
67	0560	ASSAULT	SIMPLE	
72	2022	NARCOTICS	POSSESS - COCAINE	
...	...	...	...	
237927	1812	NARCOTICS	POSSESS - CANNABIS MORE THAN 30 GRAMS	
237936	2026	NARCOTICS	POSSESS - PCP	
237967	0860	THEFT	RETAIL THEFT	
237999	2090	NARCOTICS	ALTER / FORGE PRESCRIPTION	
238187	2250	LIQUOR LAW VIOLATION	LIQUOR LICENSE VIOLATION	

	Location Description	Arrest	Domestic	...	Ward	Community Area	\
0	STREET	True	False	...	48.0	77	
37	RESTAURANT	True	False	...	24.0	29	
61	STREET	True	False	...	18.0	70	
67	SCHOOL - PUBLIC GROUNDS	True	False	...	40.0	2	
72	STREET	True	False	...	42.0	8	
...	...	...	...	...	...	...	
237927	STREET	True	False	...	42.0	8	
237936	VEHICLE NON-COMMERCIAL	True	False	...	24.0	29	

```
237967          APPLIANCE STORE    True     False  ...  12.0              61
237999               DRUG STORE    True     False  ...   2.0               8
238187             BAR OR TAVERN    True     False  ...  27.0              28

        FBI Code  X Coordinate Y Coordinate  Year              Updated On  \
0             03     1165640.0    1939961.0  2022  01/03/2023 03:46:28 PM
37           08A     1154087.0    1894508.0  2022  01/03/2023 03:46:28 PM
61            07     1150583.0    1847019.0  2022  01/03/2023 03:46:28 PM
67           08A     1156333.0    1937407.0  2022  01/03/2023 03:46:28 PM
72            18     1177309.0    1904938.0  2022  01/03/2023 03:46:28 PM
...          ...           ...          ...   ...                     ...
237927        18     1177164.0    1905392.0  2022  01/03/2023 03:46:28 PM
237936        18     1150188.0    1895080.0  2022  01/03/2023 03:46:28 PM
237967        06     1161289.0    1874276.0  2022  01/03/2023 03:46:28 PM
237999        18     1175324.0    1908199.0  2022  01/03/2023 03:46:28 PM
238187        22     1166633.0    1901539.0  2022  01/03/2023 03:46:28 PM

         Latitude  Longitude                      Location
0       41.990846 -87.666096  (41.990846423, -87.666096144)
37      41.866359 -87.709807  (41.866358918, -87.709806763)
61      41.736111 -87.723906  (41.736111204, -87.723906429)
67      41.984032 -87.700399  (41.984032004, -87.700398992)
72      41.894485 -87.624241   (41.894484934, -87.62424085)
...           ...        ...                           ...
237927  41.895734 -87.624760   (41.89573402, -87.624759616)
237936  41.868005 -87.724106  (41.868005372, -87.724105561)
237967  41.810694 -87.683929  (41.810693516, -87.683929054)
237999  41.903478 -87.631433  (41.903478069, -87.631433102)
238187  41.885393 -87.663548  (41.885393253, -87.663547848)

[27524 rows x 22 columns]
```

Notice the difference in the commands? The difference is that the word True is not in quotation marks. When the column is of type object, then we use quotation marks. When the column is of type bool then we do not use the quotation marks.

Let us try another example:

[48]:
```
chicago_crimes[chicago_crimes["Location Description"]=="STREET"]
```

[48]:
```
              ID Case Number                    Date  \
0       12789052    JF350580  08/09/2022 04:07:00 PM
1       12790581    JF352712  08/10/2022 04:00:00 PM
2       12790652    JF352659  08/11/2022 10:00:00 AM
7       12795474    JF358356  08/15/2022 10:00:00 PM
8       12799341    JF363039  08/19/2022 11:00:00 AM
...          ...         ...                     ...
238295  12935727    JF526034  12/25/2022 03:00:00 PM
```

```
238305  12935459    JF525819  12/22/2022 12:01:00 AM
238310  12935276    JF524804  12/21/2022 11:05:00 PM
238313  12936301    JF526810  12/22/2022 06:00:00 PM
238315  12935341    JF525383  12/20/2022 06:45:00 AM

                           Block  IUCR         Primary Type  \
0              014XX W ELMDALE AVE  0325              ROBBERY
1            062XX S ARTESIAN AVE  0810                THEFT
2               094XX S STATE ST  0810                THEFT
7       061XX S PARK SHORE EAST CT  0810                THEFT
8                010XX E 46TH ST  0910  MOTOR VEHICLE THEFT
...                          ...   ...                  ...
238295       061XX N WASHTENAW AVE  0810                THEFT
238305       044XX N MAPLEWOOD AVE  0910  MOTOR VEHICLE THEFT
238310          042XX N MARINE DR  0810                THEFT
238313        020XX W CORNELIA AVE  1320      CRIMINAL DAMAGE
238315        027XX W ROOSEVELT RD  0810                THEFT

               Description Location Description  Arrest  Domestic  ...  Ward  \
0       VEHICULAR HIJACKING               STREET    True     False  ...  48.0
1                 OVER $500               STREET   False     False  ...  16.0
2                 OVER $500               STREET   False      True  ...   9.0
7                 OVER $500               STREET   False     False  ...   5.0
8                AUTOMOBILE               STREET   False     False  ...   4.0
...                     ...                  ...     ...       ...  ...   ...
238295            OVER $500               STREET   False     False  ...  40.0
238305           AUTOMOBILE               STREET   False     False  ...  47.0
238310            OVER $500               STREET   False     False  ...  46.0
238313           TO VEHICLE               STREET   False     False  ...  32.0
238315            OVER $500               STREET   False     False  ...  28.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
7                   42        06     1187557.0    1864569.0  2022
8                   39        07     1183968.0    1874723.0  2022
...                ...       ...           ...          ...   ...
238295               2        06     1157219.0    1940724.0  2022
238305               4        07     1158578.0    1929302.0  2022
238310               3        06     1171054.0    1928302.0  2022
238313               5        14     1161968.0    1923233.0  2022
238315              29        06     1158071.0    1894595.0  2022

                    Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
```

```
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
7       01/03/2023 03:46:28 PM  41.783472 -87.587890
8       01/03/2023 03:46:28 PM  41.811420 -87.600731
...                        ...        ...        ...
238295  01/03/2023 03:46:28 PM  41.993116 -87.697050
238305  01/03/2023 03:46:28 PM  41.961746 -87.692365
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179

                            Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
7       (41.783471704, -87.587890396)
8       (41.811419728, -87.600731459)
...                               ...
238295  (41.993115987, -87.697049774)
238305  (41.961745665, -87.692365237)
238310  (41.958736385, -87.646526104)
238313  (41.945021752, -87.680071764)
238315  (41.866517317, -87.695178701)

[67590 rows x 22 columns]
```

Here, we are getting the rows where the value of Location Description is STREET. Notice that we used quotation marks. This is because this column contains text, i.e. it is of type object. What if we wanted to get the records where the value of Domestic was False? Is the Domestic column of type bool or object? Let's check:

[49]:
```
chicago_crimes["Domestic"].dtype
```

[49]:
```
dtype('bool')
```

It is of type bool. We therefore use the following command:

[50]:
```
chicago_crimes[chicago_crimes["Domestic"]==False]
```

[50]:
```
              ID Case Number                    Date                 Block  \
0       12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE
1       12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE
3       12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST
5       12796817    JF359932  08/15/2022 03:00:00 PM  075XX S PHILLIPS AVE
...          ...         ...                     ...                   ...
238312  12936285    JF526139  06/27/2022 10:05:00 AM    025XX N HALSTED ST
238313  12936301    JF526810  12/22/2022 06:00:00 PM  020XX W CORNELIA AVE
238314  12936397    JF526745  12/19/2022 02:00:00 PM   044XX N ROCKWELL ST
```

```
238315  12935341    JF525383  12/20/2022 06:45:00 AM  027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM     021XX W DEVON AVE

        IUCR        Primary Type                          Description  \
0       0325             ROBBERY                  VEHICULAR HIJACKING
1       0810               THEFT                             OVER $500
3       0560             ASSAULT                                SIMPLE
4       0820               THEFT                       $500 AND UNDER
5       0810               THEFT                             OVER $500
...      ...                 ...                                   ...
238312  1153  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT OVER $ 300
238313  1320     CRIMINAL DAMAGE                           TO VEHICLE
238314  0620            BURGLARY                       UNLAWFUL ENTRY
238315  0810               THEFT                             OVER $500
238316  0810               THEFT                             OVER $500

                        Location Description  Arrest  Domestic  ...  Ward  \
0                                      STREET    True     False  ...  48.0
1                                      STREET   False     False  ...  16.0
3                                   RESIDENCE   False     False  ...  14.0
4                                    SIDEWALK   False     False  ...  11.0
5                                   RESIDENCE   False     False  ...   7.0
...                                       ...     ...       ...  ...   ...
238312                                    NaN   False     False  ...  43.0
238313                                 STREET   False     False  ...  32.0
238314                              APARTMENT   False     False  ...  47.0
238315                                 STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
1                   66        06     1161110.0    1863210.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
...                ...       ...           ...          ...   ...
238312               7        11     1170513.0    1917030.0  2022
238313               5        14     1161968.0    1923233.0  2022
238314               4        05     1158237.0    1929586.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                   Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
```

```
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...        ...        ...
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[195854 rows x 22 columns]
```

As a final example, let us look at crimes that took place in Community Area 29:

[52]:
```
chicago_crimes[chicago_crimes["Community Area"]==29]
```

[52]:
```
              ID Case Number                    Date  \
22      12788778    JF350628  08/09/2022 09:00:00 AM
35      12796489    JF359613  08/17/2022 02:00:00 AM
37      12796098    JF359198  08/16/2022 05:20:00 PM
86      12804684    JF369234  08/24/2022 06:30:00 PM
144     12818261    JF385387  08/10/2022 09:35:00 PM
...          ...         ...                    ...
237936  12925588    JF513886  12/17/2022 11:22:00 AM
237976  12924648    JF512358  12/16/2022 06:14:00 AM
238110  12935262    JF524783  12/23/2022 06:00:00 AM
238188  12936468    JF526997  11/05/2022 03:30:00 PM
238315  12935341    JF525383  12/20/2022 06:45:00 AM

                            Block  IUCR        Primary Type  \
22                 033XX W 13TH ST  0890               THEFT
35                039XX W OGDEN AVE  0460             BATTERY
37            033XX W ROOSEVELT RD  0560             ASSAULT
86                035XX W CERMAK RD  0870               THEFT
144           013XX S KOMENSKY AVE  0880               THEFT
...                            ...   ...                 ...
237936         039XX W FILLMORE ST  2026           NARCOTICS
```

```
237976  013XX S INDEPENDENCE BLVD  1310     CRIMINAL DAMAGE
238110      014XX S SPAULDING AVE  1153  DECEPTIVE PRACTICE
238188          012XX S AVERS AVE  1153  DECEPTIVE PRACTICE
238315       027XX W ROOSEVELT RD  0810               THEFT

                                Description         Location Description  \
22                           FROM BUILDING                      LIBRARY
35                                  SIMPLE                  GAS STATION
37                                  SIMPLE                   RESTAURANT
86                         POCKET-PICKING                     SIDEWALK
144                       PURSE-SNATCHING                        ALLEY
...                                   ...                          ...
237936                       POSSESS - PCP       VEHICLE NON-COMMERCIAL
237976                        TO PROPERTY  RESIDENCE - PORCH / HALLWAY
238110  FINANCIAL IDENTITY THEFT OVER $ 300                         NaN
238188  FINANCIAL IDENTITY THEFT OVER $ 300                   APARTMENT
238315                           OVER $500                      STREET

        Arrest  Domestic  ...  Ward  Community Area  FBI Code  X Coordinate  \
22       False     False  ...  24.0              29        06     1154098.0
35       False     False  ...  22.0              29       08B     1150286.0
37        True     False  ...  24.0              29       08A     1154087.0
86       False     False  ...  24.0              29        06     1152918.0
144      False     False  ...  24.0              29        06     1149595.0
...        ...       ...  ...   ...             ...       ...           ...
237936    True     False  ...  24.0              29        18     1150188.0
237976   False     False  ...  24.0              29        14     1151464.0
238110   False     False  ...  24.0              29        11     1154598.0
238188   False     False  ...  24.0              29        11     1150909.0
238315   False     False  ...  28.0              29        06     1158071.0

       Y Coordinate  Year              Updated On   Latitude  Longitude  \
22        1893840.0  2022  01/03/2023 03:46:28 PM  41.864526 -87.709784
35        1889027.0  2022  01/03/2023 03:46:28 PM  41.851393 -87.723904
37        1894508.0  2022  01/03/2023 03:46:28 PM  41.866359 -87.709807
86        1889147.0  2022  01/03/2023 03:46:28 PM  41.851671 -87.714240
144       1893535.0  2022  01/03/2023 03:46:28 PM  41.863777 -87.726323
...             ...   ...                     ...        ...        ...
237936    1895080.0  2022  01/03/2023 03:46:28 PM  41.868005 -87.724106
237976    1893655.0  2022  01/03/2023 03:46:28 PM  41.864070 -87.719458
238110    1892865.0  2022  01/03/2023 03:46:28 PM  41.861840 -87.707975
238188    1894078.0  2022  01/03/2023 03:46:28 PM  41.865242 -87.721485
238315    1894595.0  2022  01/03/2023 03:46:28 PM  41.866517 -87.695179

                          Location
22      (41.864525633, -87.709784193)
35      (41.851393305, -87.723903515)
```

```
37       (41.866358918, -87.709806763)
86       (41.851670934, -87.714240208)
144        (41.863777237, -87.7263227)
...                                ...
237936   (41.868005372, -87.724105561)
237976   (41.864070063, -87.719458499)
238110   (41.861840145, -87.707974761)
238188   (41.865241697, -87.721484832)
238315   (41.866517317, -87.695178701)

[6737 rows x 22 columns]
```

Why did we not put the number 29 in quotation marks? You guessed it. The column is not of type object. What type is it?

[51]:
```
chicago_crimes["Community Area"].dtypes
```

[51]:
```
dtype('int64')
```

It is an integer. Numbers, just like bool values do not go in a quotation.

7.2 Two Conditions

How do we get data that satisfies two conditions? Here, we need to introduce the OR and AND operators.

7.2.1 The AND Operator

In pandas dataframes, when we want to say "and" we use the symbols "&". So, if we want to check if two is equal to two and if two is greater than zero, we type:

[53]:
```
(2==2) & (2>0)
```

[53]:
```
True
```

Let us now use this syntax to get a subset of the crimes data where the crime was of type THEFT and an arrest was made:

[54]:
```
chicago_crimes[(chicago_crimes["Primary Type"]=="THEFT") &␣
 ↪(chicago_crimes["Arrest"]==True)]
```

[54]:
```
              ID Case Number                    Date                  Block  \
140     12817625    JF384566  09/05/2022 11:27:00 PM        023XX W 21ST ST
461     12848814    JF421820  10/04/2022 03:42:00 PM       031XX N CLARK ST
654     12867355    JF444181  10/22/2022 06:50:00 AM       035XX N BROADWAY
849     12891196    JF472493  10/25/2022 12:00:00 AM     035XX S PULASKI RD
857     12886022    JF466544  11/08/2022 02:56:00 AM     017XX N LINDER AVE
...          ...         ...                     ...                    ...
237663  12931318    JF520805  12/23/2022 11:50:00 AM       031XX W 103RD ST
```

```
237674  12928904    JF517643  12/20/2022 04:30:00 PM      0000X E GRAND AVE
237846  12933440    JF523488  12/26/2022 08:37:00 PM    045XX S WESTERN AVE
237868  12931059    JF520508  12/23/2022 03:40:00 AM  020XX N MILWAUKEE AVE
237967  12925155    JF513324  12/16/2022 08:12:00 PM   045XX S WESTERN BLVD

        IUCR Primary Type   Description Location Description  Arrest  \
140     0810        THEFT     OVER $500    CONVENIENCE STORE    True
461     0860        THEFT  RETAIL THEFT   SMALL RETAIL STORE    True
654     0860        THEFT  RETAIL THEFT   GROCERY FOOD STORE    True
849     0810        THEFT     OVER $500            WAREHOUSE    True
857     0810        THEFT     OVER $500               STREET    True
...      ...          ...           ...                  ...     ...
237663  0860        THEFT  RETAIL THEFT   SMALL RETAIL STORE    True
237674  0860        THEFT  RETAIL THEFT   SMALL RETAIL STORE    True
237846  0860        THEFT  RETAIL THEFT      APPLIANCE STORE    True
237868  0860        THEFT  RETAIL THEFT     DEPARTMENT STORE    True
237967  0860        THEFT  RETAIL THEFT      APPLIANCE STORE    True

        Domestic  ...  Ward  Community Area  FBI Code  X Coordinate  \
140        False  ...  25.0              31        06     1161190.0
461        False  ...  44.0               6        06     1170283.0
654        False  ...  46.0               6        06     1171076.0
849        False  ...  22.0              30        06     1150261.0
857        False  ...  37.0              25        06     1139463.0
...          ...  ...   ...             ...       ...           ...
237663     False  ...  19.0              74        06     1157026.0
237674     False  ...  42.0               8        06     1176780.0
237846     False  ...  15.0              61        06     1161131.0
237868     False  ...   1.0              22        06     1159682.0
237967     False  ...  12.0              61        06     1161289.0

       Y Coordinate  Year              Updated On   Latitude  Longitude  \
140       1890017.0  2022  01/03/2023 03:46:28 PM  41.853891 -87.683856
461       1920956.0  2022  01/03/2023 03:46:28 PM  41.938596 -87.649576
654       1923738.0  2022  01/03/2023 03:46:28 PM  41.946212 -87.646580
849       1881205.0  2022  01/03/2023 03:46:28 PM  41.829929 -87.724199
857       1911028.0  2022  01/03/2023 03:46:28 PM  41.911971 -87.763090
...             ...   ...                     ...        ...        ...
237663    1836163.0  2022  01/03/2023 03:46:28 PM  41.706193 -87.700594
237674    1903916.0  2022  01/03/2023 03:46:28 PM  41.891692 -87.626215
237846    1874302.0  2022  01/03/2023 03:46:28 PM  41.810768 -87.684508
237868    1913243.0  2022  01/03/2023 03:46:28 PM  41.917656 -87.688750
237967    1874276.0  2022  01/03/2023 03:46:28 PM  41.810694 -87.683929

                          Location
140     (41.853890639, -87.683855655)
461      (41.93859561, -87.649576033)
```

```
654     (41.946212132, -87.646579678)
849     (41.829929212, -87.724198877)
857      (41.911970795, -87.76309048)
...                              ...
237663  (41.706192826, -87.700593626)
237674  (41.891692496, -87.626214622)
237846  (41.810768138, -87.684507864)
237868  (41.917656022, -87.688750258)
237967  (41.810693516, -87.683929054)

[1995 rows x 22 columns]
```

An important point to note here is that the two conditions must be placed in separate parantheses.

7.2.2 The OR Operator

In python, the OR operator is written using the symbol "|":

[55]:
```
(2==2) | (2 < 0)
```

[55]:
```
True
```

Notice that the expression evaluated to True even though 2 is not les sthan 0. This is because we want the first or second condition to be true. When one of the conditions is true, then the result will be true.

Going to our dataframe, let us get the crimes that resulted in an arrest or that were domestic:

[56]:
```
chicago_crimes[(chicago_crimes["Arrest"]==True) |␣
 ↪(chicago_crimes["Domestic"]==True)]
```

[56]:
```
              ID Case Number                    Date                  Block  \
0       12789052    JF350580  08/09/2022 04:07:00 PM    014XX W ELMDALE AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM       094XX S STATE ST
9       12793378    JF355848  08/13/2022 10:45:00 PM    118XX S SANGAMON ST
10      12798495    JF361883  08/18/2022 06:00:00 PM        030XX E 80TH ST
13      12790997    JF353231  08/11/2022 06:00:00 PM         073XX S MAY ST
...          ...         ...                     ...                     ...
238262  12938984    JG100896  12/25/2022 01:00:00 PM  075XX S EGGLESTON AVE
238263  12936415    JF518127  12/21/2022 01:15:00 AM        010XX W 14TH ST
238288  12937066    JF522595  12/25/2022 08:06:00 PM       0000X W 109TH ST
238308  12937577    JF515781  12/19/2022 06:00:00 AM      039XX W MONROE ST
238309  12935640    JF526043  12/23/2022 09:00:00 AM      010XX N AVERS AVE

        IUCR     Primary Type  \
0       0325          ROBBERY
2       0810            THEFT
9       0560          ASSAULT
10      2820    OTHER OFFENSE
```

```
13      051A          ASSAULT
...      ...              ...
238262  0560          ASSAULT
238263  0497          BATTERY
238288  1310  CRIMINAL DAMAGE
238308  0498          BATTERY
238309  2820    OTHER OFFENSE

                                                Description  \
0                                     VEHICULAR HIJACKING
2                                               OVER $500
9                                                  SIMPLE
10                                        TELEPHONE THREAT
13                                      AGGRAVATED - HANDGUN
...                                                      ...
238262                                             SIMPLE
238263  AGGRAVATED DOMESTIC BATTERY - OTHER DANGEROUS ...
238288                                          TO PROPERTY
238308  AGG. DOMESTIC BATTERY - HANDS, FISTS, FEET, SE...
238309                                     TELEPHONE THREAT

       Location Description  Arrest  Domestic  ...  Ward  Community Area  \
0                    STREET    True     False  ...  48.0              77
2                    STREET   False      True  ...   9.0              49
9                    STREET   False      True  ...  34.0              53
10                RESIDENCE   False      True  ...   7.0              46
13                RESIDENCE   False      True  ...  17.0              68
...                     ...     ...       ...  ...   ...             ...
238262            RESIDENCE   False      True  ...   6.0              69
238263            APARTMENT   False      True  ...  25.0              28
238288               STREET   False      True  ...  34.0              49
238308            APARTMENT   False      True  ...  28.0              26
238309            RESIDENCE   False      True  ...  37.0              23

        FBI Code  X Coordinate Y Coordinate  Year              Updated On  \
0             03     1165640.0    1939961.0  2022  01/03/2023 03:46:28 PM
2             06     1177962.0    1842197.0  2022  01/03/2023 03:46:28 PM
9            08A     1172121.0    1826361.0  2022  01/03/2023 03:46:28 PM
10           08A     1197715.0    1852504.0  2022  01/03/2023 03:46:28 PM
13           04A     1169922.0    1856159.0  2022  01/03/2023 03:46:28 PM
...          ...           ...          ...   ...                     ...
238262       08A     1174599.0    1854938.0  2022  01/03/2023 03:46:28 PM
238263       04B     1169755.0    1893571.0  2022  01/03/2023 03:46:28 PM
238288        14     1177923.0    1832690.0  2022  01/03/2023 03:46:28 PM
238308       04B     1150117.0    1899370.0  2022  01/03/2023 03:46:28 PM
238309       08A     1150527.0    1906726.0  2022  01/03/2023 03:46:28 PM
```

```
         Latitude  Longitude                      Location
0       41.990846 -87.666096  (41.990846423, -87.666096144)
2       41.722303 -87.623745  (41.722303228, -87.623745129)
9       41.678977 -87.645603  (41.678976937, -87.645603032)
10      41.750117 -87.551051   (41.75011688, -87.551050773)
13      41.760795 -87.652790    (41.76079497, -87.65279005)
...           ...        ...                           ...
238262  41.757342 -87.635685   (41.757341587, -87.63568495)
238263  41.863461 -87.652316  (41.863461031, -87.652315509)
238288  41.696216 -87.624175  (41.696215599, -87.624174764)
238308  41.879779 -87.724254  (41.879779001, -87.724254458)
238309  41.899957 -87.722557  (41.899956643, -87.722556723)

[64568 rows x 22 columns]
```

Looking at the result, we see that in each of these records, either Arrest is True or Domestic is True.

Let us now get the crimes that were commited in Community Area 23 or that were of type BATTERY:

[ ]:
```
chicago_crimes[(chicago_crimes["Community Area"]==23) | (chicago_crimes["Primary␣
↪Type"]=="BATTERY")]
```

Examining these records, we see that all of them are either of type BATTERY, or they have been commited in Community Area 23.

7.2.3 Less Than or Equal to and Greater than or Equal to

Let us get the crimes that were committed in a Community Area with a number that is at least 23, and the Ward number is less than 30:

[57]:
```
chicago_crimes[(chicago_crimes["Community Area"] >= 23) &␣
↪(chicago_crimes["Ward"] < 30)]
```

[57]:
```
              ID Case Number                    Date                 Block  \
1       12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM      094XX S STATE ST
3       12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST
5       12796817    JF359932  08/15/2022 03:00:00 PM  075XX S PHILLIPS AVE
...          ...         ...                     ...                   ...
238300  12935244    JF505474  12/10/2022 01:00:00 PM       017XX W 86TH ST
238301  12935201    JF523453  12/26/2022 02:00:00 PM       013XX W 76TH ST
238306  12936699    JF524860  12/26/2022 12:00:00 PM       017XX W 79TH ST
238308  12937577    JF515781  12/19/2022 06:00:00 AM     039XX W MONROE ST
238315  12935341    JF525383  12/20/2022 06:45:00 AM  027XX W ROOSEVELT RD

        IUCR       Primary Type  \
1       0810              THEFT
```

```
2       0810              THEFT
3       0560            ASSAULT
4       0820              THEFT
5       0810              THEFT
...      ...                ...
238300  1365  CRIMINAL TRESPASS
238301  0610           BURGLARY
238306  0890              THEFT
238308  0498            BATTERY
238315  0810              THEFT

                                          Description  \
1                                           OVER $500
2                                           OVER $500
3                                              SIMPLE
4                                      $500 AND UNDER
5                                           OVER $500
...                                               ...
238300                                   TO RESIDENCE
238301                                  FORCIBLE ENTRY
238306                                   FROM BUILDING
238308  AGG. DOMESTIC BATTERY - HANDS, FISTS, FEET, SE...
238315                                      OVER $500

               Location Description  Arrest  Domestic  ...  Ward  \
1                            STREET   False     False  ...  16.0
2                            STREET   False      True  ...   9.0
3                         RESIDENCE   False     False  ...  14.0
4                          SIDEWALK   False     False  ...  11.0
5                         RESIDENCE   False     False  ...   7.0
...                             ...     ...       ...  ...   ...
238300                    APARTMENT   False     False  ...  21.0
238301                    APARTMENT   False     False  ...  17.0
238306  COMMERCIAL / BUSINESS OFFICE   False     False  ...  17.0
238308                    APARTMENT   False      True  ...  28.0
238315                       STREET   False     False  ...  28.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
...                ...       ...           ...          ...   ...
238300              71        26     1166366.0    1847669.0  2022
238301              71        05     1168669.0    1854374.0  2022
238306              71        06     1166290.0    1852314.0  2022
```

```
238308              26       04B     1150117.0    1899370.0  2022
238315              29        06     1158071.0    1894595.0  2022

                    Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...        ...        ...
238300  01/03/2023 03:46:28 PM  41.737574 -87.666064
238301  01/03/2023 03:46:28 PM  41.755924 -87.657434
238306  01/03/2023 03:46:28 PM  41.750322 -87.666211
238308  01/03/2023 03:46:28 PM  41.879779 -87.724254
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179

                           Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238300  (41.737573664, -87.666064301)
238301   (41.755923803, -87.65743376)
238306  (41.750321833, -87.666210811)
238308  (41.879779001, -87.724254458)
238315  (41.866517317, -87.695178701)

[154032 rows x 22 columns]
```

The phrase "at least" means greater than or equal, which is written using the operator ">=". The condition "less than" is simply written using the operator "<". If we wanted to say "less than or equal", we would use the operator "<=". As an example, get the crimes where the value of X Coordinate is less than or equal to 1177962:

[58]:
```
chicago_crimes[chicago_crimes["X Coordinate"] <= 1177962]
```

[58]:
```
              ID Case Number                    Date                 Block  \
0       12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE
1       12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM     094XX S STATE ST
3       12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST
...          ...         ...                     ...                   ...
238312  12936285    JF526139  06/27/2022 10:05:00 AM    025XX N HALSTED ST
238313  12936301    JF526810  12/22/2022 06:00:00 PM  020XX W CORNELIA AVE
238314  12936397    JF526745  12/19/2022 02:00:00 PM   044XX N ROCKWELL ST
```

```
238315  12935341    JF525383  12/20/2022 06:45:00 AM  027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM     021XX W DEVON AVE

        IUCR        Primary Type                          Description  \
0       0325             ROBBERY                  VEHICULAR HIJACKING
1       0810               THEFT                             OVER $500
2       0810               THEFT                             OVER $500
3       0560             ASSAULT                                SIMPLE
4       0820               THEFT                      $500 AND UNDER
...      ...                 ...                                 ...
238312  1153  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT OVER $ 300
238313  1320     CRIMINAL DAMAGE                          TO VEHICLE
238314  0620            BURGLARY                      UNLAWFUL ENTRY
238315  0810               THEFT                             OVER $500
238316  0810               THEFT                             OVER $500

                       Location Description  Arrest  Domestic  ...  Ward  \
0                                     STREET    True     False  ...  48.0
1                                     STREET   False     False  ...  16.0
2                                     STREET   False      True  ...   9.0
3                                   RESIDENCE  False     False  ...  14.0
4                                    SIDEWALK  False     False  ...  11.0
...                                       ...     ...       ...  ...   ...
238312                                   NaN   False     False  ...  43.0
238313                                STREET   False     False  ...  32.0
238314                             APARTMENT   False     False  ...  47.0
238315                                STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
...                ...       ...           ...          ...   ...
238312               7        11     1170513.0    1917030.0  2022
238313               5        14     1161968.0    1923233.0  2022
238314               4        05     1158237.0    1929586.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
```

```
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
...                        ...        ...        ...
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
...                               ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[183338 rows x 22 columns]
```

7.2.4 Not Equal to

There is another operator that we have not yet covered, and that is the "not equal to" operator. What if we wanted the crimes which did not take place in Community Area 23? In other words, we want to get all crimes where the value of Community Area is not equal to 23. This can be accomplished using the "!=" symbols, which basically means "not equal to":

[59]:
```
chicago_crimes[chicago_crimes["Community Area"]!=23]
```

[59]:
```
              ID Case Number                    Date                  Block  \
0       12789052    JF350580  08/09/2022 04:07:00 PM   014XX W ELMDALE AVE
1       12790581    JF352712  08/10/2022 04:00:00 PM  062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM     094XX S STATE ST
3       12796135    JF359082  08/15/2022 09:14:00 PM    048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM    015XX S HALSTED ST
...          ...         ...                     ...                   ...
238312  12936285    JF526139  06/27/2022 10:05:00 AM    025XX N HALSTED ST
238313  12936301    JF526810  12/22/2022 06:00:00 PM  020XX W CORNELIA AVE
238314  12936397    JF526745  12/19/2022 02:00:00 PM   044XX N ROCKWELL ST
238315  12935341    JF525383  12/20/2022 06:45:00 AM  027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM     021XX W DEVON AVE

        IUCR        Primary Type                         Description  \
0       0325             ROBBERY                 VEHICULAR HIJACKING
```

```
1       0810               THEFT                              OVER $500
2       0810               THEFT                              OVER $500
3       0560             ASSAULT                                 SIMPLE
4       0820               THEFT                        $500 AND UNDER
...      ...                 ...                                   ...
238312  1153  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT OVER $ 300
238313  1320     CRIMINAL DAMAGE                           TO VEHICLE
238314  0620            BURGLARY                       UNLAWFUL ENTRY
238315  0810               THEFT                              OVER $500
238316  0810               THEFT                              OVER $500

                       Location Description  Arrest  Domestic  ...  Ward  \
0                                     STREET    True     False  ...  48.0
1                                     STREET   False     False  ...  16.0
2                                     STREET   False      True  ...   9.0
3                                   RESIDENCE  False     False  ...  14.0
4                                    SIDEWALK  False     False  ...  11.0
...                                       ...     ...       ...  ...   ...
238312                                   NaN   False     False  ...  43.0
238313                                STREET   False     False  ...  32.0
238314                             APARTMENT   False     False  ...  47.0
238315                                STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)  False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
...                ...       ...           ...          ...   ...
238312               7        11     1170513.0    1917030.0  2022
238313               5        14     1161968.0    1923233.0  2022
238314               4        05     1158237.0    1929586.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
...                        ...        ...        ...
238312  01/03/2023 03:46:28 PM  41.927817 -87.648846
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
```

```
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
...                              ...
238312  (41.927817456, -87.648845932)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[232222 rows x 22 columns]
```

7.2.5 The isin() Method

Assume that wa want the crimes that were either THEFT or BATTERY. The following command will do the job:

[60]:
```
chicago_crimes[(chicago_crimes["Primary Type"]=="THEFT") |␣
 ↪(chicago_crimes["Primary Type"]=="BATTERY")]
```

[60]:
```
              ID Case Number                    Date                   Block  \
1       12790581    JF352712  08/10/2022 04:00:00 PM    062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM        094XX S STATE ST
4       12795972    JF359058  08/16/2022 04:10:00 PM      015XX S HALSTED ST
5       12796817    JF359932  08/15/2022 03:00:00 PM    075XX S PHILLIPS AVE
6       12796316    JF358737  08/16/2022 11:57:00 AM     012XX N LA SALLE DR
...          ...         ...                     ...                     ...
238308  12937577    JF515781  12/19/2022 06:00:00 AM       039XX W MONROE ST
238310  12935276    JF524804  12/21/2022 11:05:00 PM       042XX N MARINE DR
238311  12937237    JF527290  12/23/2022 08:41:00 AM  064XX W Irving Park Rd
238315  12935341    JF525383  12/20/2022 06:45:00 AM    027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM       021XX W DEVON AVE

        IUCR Primary Type                                        Description  \
1       0810        THEFT                                          OVER $500
2       0810        THEFT                                          OVER $500
4       0820        THEFT                                  $500 AND UNDER
5       0810        THEFT                                          OVER $500
6       0460      BATTERY                                             SIMPLE
...      ...          ...                                                ...
238308  0498      BATTERY  AGG. DOMESTIC BATTERY - HANDS, FISTS, FEET, SE...
```

```
238310  0810        THEFT                                          OVER $500
238311  0860        THEFT                                       RETAIL THEFT
238315  0810        THEFT                                          OVER $500
238316  0810        THEFT                                          OVER $500

                        Location Description  Arrest  Domestic  ...  Ward  \
1                                     STREET   False     False  ...  16.0
2                                     STREET   False      True  ...   9.0
4                                   SIDEWALK   False     False  ...  11.0
5                                  RESIDENCE   False     False  ...   7.0
6                                  APARTMENT   False     False  ...   2.0
...                                      ...     ...       ...  ...   ...
238308                             APARTMENT   False      True  ...  28.0
238310                                STREET   False     False  ...  46.0
238311                  SMALL RETAIL STORE   False     False  ...  38.0
238315                                STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
6                    8       08B     1174910.0    1908582.0  2022
...                ...       ...           ...          ...   ...
238308              26       04B     1150117.0    1899370.0  2022
238310               3        06     1171054.0    1928302.0  2022
238311              17        06     1132626.0    1925898.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
6       01/03/2023 03:46:28 PM  41.904538 -87.632942
...                        ...       ...        ...
238308  01/03/2023 03:46:28 PM  41.879779 -87.724254
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                          Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
```

```
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
6       (41.904538325, -87.632942313)
...                               ...
238308  (41.879779001, -87.724254458)
238310  (41.958736385, -87.646526104)
238311  (41.952897791, -87.787860507)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[95644 rows x 22 columns]
```

What about getting the crimes that are either THEFT, or BATTERY, or ASSUALT? Again, we can just use the same logic as above:

[61]:
```
chicago_crimes[(chicago_crimes["Primary Type"]=="THEFT") |␣
↪(chicago_crimes["Primary Type"]=="BATTERY") | (chicago_crimes["Primary␣
↪Type"]=="ASSAULT")]
```

[61]:
```
              ID Case Number                    Date                   Block  \
1       12790581    JF352712  08/10/2022 04:00:00 PM    062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM        094XX S STATE ST
3       12796135    JF359082  08/15/2022 09:14:00 PM      048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM      015XX S HALSTED ST
5       12796817    JF359932  08/15/2022 03:00:00 PM    075XX S PHILLIPS AVE
...          ...         ...                     ...                    ...
238308  12937577    JF515781  12/19/2022 06:00:00 AM       039XX W MONROE ST
238310  12935276    JF524804  12/21/2022 11:05:00 PM       042XX N MARINE DR
238311  12937237    JF527290  12/23/2022 08:41:00 AM  064XX W Irving Park Rd
238315  12935341    JF525383  12/20/2022 06:45:00 AM    027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM       021XX W DEVON AVE

        IUCR Primary Type                                        Description  \
1       0810        THEFT                                          OVER $500
2       0810        THEFT                                          OVER $500
3       0560      ASSAULT                                             SIMPLE
4       0820        THEFT                                   $500 AND UNDER
5       0810        THEFT                                          OVER $500
...      ...          ...                                                ...
238308  0498      BATTERY  AGG. DOMESTIC BATTERY - HANDS, FISTS, FEET, SE...
238310  0810        THEFT                                          OVER $500
238311  0860        THEFT                                      RETAIL THEFT
238315  0810        THEFT                                          OVER $500
238316  0810        THEFT                                          OVER $500

                      Location Description  Arrest  Domestic  ...  Ward  \
1                                   STREET   False     False  ...  16.0
2                                   STREET   False      True  ...   9.0
```

```
3                                    RESIDENCE   False     False  ...  14.0
4                                     SIDEWALK   False     False  ...  11.0
5                                    RESIDENCE   False     False  ...   7.0
...                                        ...     ...       ...  ...   ...
238308                               APARTMENT   False      True  ...  28.0
238310                                  STREET   False     False  ...  46.0
238311                      SMALL RETAIL STORE   False     False  ...  38.0
238315                                  STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
...                ...       ...           ...          ...   ...
238308              26       04B     1150117.0    1899370.0  2022
238310               3        06     1171054.0    1928302.0  2022
238311              17        06     1132626.0    1925898.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                  Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...        ...        ...
238308  01/03/2023 03:46:28 PM  41.879779 -87.724254
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238308  (41.879779001, -87.724254458)
238310  (41.958736385, -87.646526104)
238311  (41.952897791, -87.787860507)
238315  (41.866517317, -87.695178701)
```

```
238316  (41.997824802, -87.684266677)

[116428 rows x 22 columns]
```

We can put as many conditions as we want. However, this is getting tedious. Luckily, the pandas library has a very useful function for this. This function is the isin() function. As the name suggests, it checks whether a value is in a certain set. Let us take a look at it:

[62]:
```
chicago_crimes[chicago_crimes["Primary Type"].isin(["THEFT", "BATTERY",␣
↪"ASSAULT"])]
```

[62]:

	ID	Case Number	Date	Block	\
1	12790581	JF352712	08/10/2022 04:00:00 PM	062XX S ARTESIAN AVE	
2	12790652	JF352659	08/11/2022 10:00:00 AM	094XX S STATE ST	
3	12796135	JF359082	08/15/2022 09:14:00 PM	048XX S KARLOV AVE	
4	12795972	JF359058	08/16/2022 04:10:00 PM	015XX S HALSTED ST	
5	12796817	JF359932	08/15/2022 03:00:00 PM	075XX S PHILLIPS AVE	
...	...	...	...	...	
238308	12937577	JF515781	12/19/2022 06:00:00 AM	039XX W MONROE ST	
238310	12935276	JF524804	12/21/2022 11:05:00 PM	042XX N MARINE DR	
238311	12937237	JF527290	12/23/2022 08:41:00 AM	064XX W Irving Park Rd	
238315	12935341	JF525383	12/20/2022 06:45:00 AM	027XX W ROOSEVELT RD	
238316	12938501	JF523997	12/26/2022 10:30:00 PM	021XX W DEVON AVE	

	IUCR	Primary Type	Description	\
1	0810	THEFT	OVER $500	
2	0810	THEFT	OVER $500	
3	0560	ASSAULT	SIMPLE	
4	0820	THEFT	$500 AND UNDER	
5	0810	THEFT	OVER $500	
...	...	...	...	
238308	0498	BATTERY	AGG. DOMESTIC BATTERY - HANDS, FISTS, FEET, SE...	
238310	0810	THEFT	OVER $500	
238311	0860	THEFT	RETAIL THEFT	
238315	0810	THEFT	OVER $500	
238316	0810	THEFT	OVER $500	

	Location Description	Arrest	Domestic	...	Ward	\
1	STREET	False	False	...	16.0	
2	STREET	False	True	...	9.0	
3	RESIDENCE	False	False	...	14.0	
4	SIDEWALK	False	False	...	11.0	
5	RESIDENCE	False	False	...	7.0	
...	...	...	...	...	...	
238308	APARTMENT	False	True	...	28.0	
238310	STREET	False	False	...	46.0	
238311	SMALL RETAIL STORE	False	False	...	38.0	
238315	STREET	False	False	...	28.0	

```
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
5                   43        06     1193842.0    1855434.0  2022
...                ...       ...           ...          ...   ...
238308              26       04B     1150117.0    1899370.0  2022
238310               3        06     1171054.0    1928302.0  2022
238311              17        06     1132626.0    1925898.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                    Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...        ...        ...
238308  01/03/2023 03:46:28 PM  41.879779 -87.724254
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                           Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238308  (41.879779001, -87.724254458)
238310  (41.958736385, -87.646526104)
238311  (41.952897791, -87.787860507)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[116428 rows x 22 columns]
```

Here, we ae just telling pandas to return the records where the value of the column Primary Type is in the given array. Notice that we use the square brackets. This is because, as you remember, we are passing more than one value to the function. We could have performed the above in two steps:

[63]:
```
crime_types = ["THEFT", "BATTERY", "ASSAULT"]
chicago_crimes[chicago_crimes["Primary Type"].isin(crime_types)]
```

[63]:

```
              ID Case Number                    Date                    Block  \
1       12790581    JF352712  08/10/2022 04:00:00 PM    062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM       094XX S STATE ST
3       12796135    JF359082  08/15/2022 09:14:00 PM     048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM     015XX S HALSTED ST
5       12796817    JF359932  08/15/2022 03:00:00 PM    075XX S PHILLIPS AVE
...          ...         ...                     ...                    ...
238308  12937577    JF515781  12/19/2022 06:00:00 AM      039XX W MONROE ST
238310  12935276    JF524804  12/21/2022 11:05:00 PM      042XX N MARINE DR
238311  12937237    JF527290  12/23/2022 08:41:00 AM  064XX W Irving Park Rd
238315  12935341    JF525383  12/20/2022 06:45:00 AM    027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM      021XX W DEVON AVE

        IUCR Primary Type                                        Description  \
1       0810        THEFT                                          OVER $500
2       0810        THEFT                                          OVER $500
3       0560      ASSAULT                                             SIMPLE
4       0820        THEFT                                    $500 AND UNDER
5       0810        THEFT                                          OVER $500
...      ...          ...                                                ...
238308  0498      BATTERY  AGG. DOMESTIC BATTERY - HANDS, FISTS, FEET, SE...
238310  0810        THEFT                                          OVER $500
238311  0860        THEFT                                       RETAIL THEFT
238315  0810        THEFT                                          OVER $500
238316  0810        THEFT                                          OVER $500

                          Location Description  Arrest  Domestic  ...  Ward  \
1                                       STREET   False     False  ...  16.0
2                                       STREET   False      True  ...   9.0
3                                    RESIDENCE   False     False  ...  14.0
4                                     SIDEWALK   False     False  ...  11.0
5                                    RESIDENCE   False     False  ...   7.0
...                                        ...     ...       ...  ...   ...
238308                               APARTMENT   False      True  ...  28.0
238310                                  STREET   False     False  ...  46.0
238311                    SMALL RETAIL STORE   False     False  ...  38.0
238315                                  STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
```

```
5                  43        06     1193842.0    1855434.0  2022
...               ...       ...           ...          ...   ...
238308             26       04B     1150117.0    1899370.0  2022
238310              3        06     1171054.0    1928302.0  2022
238311             17        06     1132626.0    1925898.0  2022
238315             29        06     1158071.0    1894595.0  2022
238316              2        06     1160681.0    1942466.0  2022

                   Updated On   Latitude  Longitude  \
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
5       01/03/2023 03:46:28 PM  41.758253 -87.565147
...                        ...       ...        ...
238308  01/03/2023 03:46:28 PM  41.879779 -87.724254
238310  01/03/2023 03:46:28 PM  41.958736 -87.646526
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                            Location
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
5       (41.758252785, -87.565147001)
...                               ...
238308  (41.879779001, -87.724254458)
238310  (41.958736385, -87.646526104)
238311  (41.952897791, -87.787860507)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[116428 rows x 22 columns]
```

We can combine the isin() function with other operators. For example, let us get all crimes of types THEFT, BATTERY, and ASSAULT that have been commited in Community Area 29:

[64]:
```
chicago_crimes[(chicago_crimes["Primary Type"].isin(["THEFT", "BATTERY",␣
↪"ASSAULT"])) & (chicago_crimes["Community Area"]==29)]
```

[64]:
```
              ID Case Number                    Date  \
22      12788778    JF350628  08/09/2022 09:00:00 AM
35      12796489    JF359613  08/17/2022 02:00:00 AM
37      12796098    JF359198  08/16/2022 05:20:00 PM
86      12804684    JF369234  08/24/2022 06:30:00 PM
144     12818261    JF385387  08/10/2022 09:35:00 PM
```

```
...          ...         ...                      ...
237679  12934815    JF525187  12/15/2022 12:00:00 AM
237734  12925997    JF514296  12/17/2022 06:28:00 PM
237877  12923943    JF511793  12/15/2022 03:58:00 PM
237904  12930839    JF520302  12/22/2022 06:59:00 PM
238315  12935341    JF525383  12/20/2022 06:45:00 AM

                         Block  IUCR Primary Type              Description  \
22              033XX W 13TH ST  0890        THEFT            FROM BUILDING
35            039XX W OGDEN AVE  0460      BATTERY                   SIMPLE
37         033XX W ROOSEVELT RD  0560      ASSAULT                   SIMPLE
86            035XX W CERMAK RD  0870        THEFT           POCKET-PICKING
144        013XX S KOMENSKY AVE  0880        THEFT          PURSE-SNATCHING
...                         ...   ...          ...                      ...
237679          027XX W 15TH ST  0820        THEFT           $500 AND UNDER
237734        036XX W OGDEN AVE  051A      ASSAULT     AGGRAVATED - HANDGUN
237877        011XX S HOMAN AVE  0560      ASSAULT                   SIMPLE
237904  011XX S CENTRAL PARK AVE  0486      BATTERY  DOMESTIC BATTERY SIMPLE
238315     027XX W ROOSEVELT RD  0810        THEFT                OVER $500

             Location Description  Arrest  Domestic  ...  Ward  \
22                         LIBRARY   False     False  ...  24.0
35                     GAS STATION   False     False  ...  22.0
37                      RESTAURANT    True     False  ...  24.0
86                        SIDEWALK   False     False  ...  24.0
144                          ALLEY   False     False  ...  24.0
...                            ...     ...       ...  ...   ...
237679  HOSPITAL BUILDING / GROUNDS   False     False  ...  28.0
237734                       STREET   False     False  ...  24.0
237877    SCHOOL - PUBLIC BUILDING    True     False  ...  24.0
237904                   RESIDENCE   False      True  ...  24.0
238315                       STREET   False     False  ...  28.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
22                  29        06     1154098.0    1893840.0  2022
35                  29       08B     1150286.0    1889027.0  2022
37                  29       08A     1154087.0    1894508.0  2022
86                  29        06     1152918.0    1889147.0  2022
144                 29        06     1149595.0    1893535.0  2022
...                ...       ...           ...          ...   ...
237679              29        06     1158464.0    1892617.0  2022
237734              29       04A     1152598.0    1890161.0  2022
237877              29       08A     1153879.0    1894754.0  2022
237904              29       08B     1152544.0    1894972.0  2022
238315              29        06     1158071.0    1894595.0  2022

                    Updated On   Latitude  Longitude  \
```

```
22      01/03/2023 03:46:28 PM  41.864526 -87.709784
35      01/03/2023 03:46:28 PM  41.851393 -87.723904
37      01/03/2023 03:46:28 PM  41.866359 -87.709807
86      01/03/2023 03:46:28 PM  41.851671 -87.714240
144     01/03/2023 03:46:28 PM  41.863777 -87.726323
...                        ...        ...        ...
237679  01/03/2023 03:46:28 PM  41.861081 -87.693790
237734  01/03/2023 03:46:28 PM  41.854460 -87.715388
237877  01/03/2023 03:46:28 PM  41.867038 -87.710564
237904  01/03/2023 03:46:28 PM  41.867663 -87.715459
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179

                            Location
22      (41.864525633, -87.709784193)
35      (41.851393305, -87.723903515)
37      (41.866358918, -87.709806763)
86      (41.851670934, -87.714240208)
144       (41.863777237, -87.7263227)
...                              ...
237679  (41.861081457, -87.693790049)
237734  (41.854459796, -87.715387915)
237877  (41.867038112, -87.710563803)
237904    (41.8676628, -87.715459036)
238315  (41.866517317, -87.695178701)

[3020 rows x 22 columns]
```

7.2.6 The isna() Method

If you recall, we had previously used the isna() function to get the values that were null. This function can also be used as a condition:

[65]:
```
chicago_crimes[chicago_crimes["Location Description"].isna()]
```

[65]:
```
              ID Case Number                    Date                    Block  \
749     12814348    JF380344  08/27/2022 08:35:00 PM   018XX N MILWAUKEE AVE
1258    12624817    JF152109  02/21/2022 01:00:00 AM    044XX S BERKELEY AVE
1305    12941312    JG102678  12/22/2022 10:35:00 AM      032XX W MADISON ST
1316    12624795    JF152108  02/21/2022 12:40:00 AM    044XX S BERKELEY AVE
1373    12624799    JF152283  02/22/2022 05:00:00 AM    048XX N LAVERGNE AVE
...          ...         ...                     ...                     ...
238182  12939131    JG100330  08/31/2022 01:00:00 PM  017XX W IRVING PARK RD
238196  12936303    JF526006  12/16/2022 08:30:00 AM     0000X N ABERDEEN ST
238204  12935269    JF524782  12/06/2022 04:20:00 PM         001XX E 44TH ST
238259  12935258    JF524780  12/12/2022 06:20:00 PM    074XX W PALATINE AVE
238312  12936285    JF526139  06/27/2022 10:05:00 AM      025XX N HALSTED ST
```

```
        IUCR        Primary Type                              Description  \
749     1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
1258    1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
1305    1154  DECEPTIVE PRACTICE  FINANCIAL IDENTITY THEFT $300 AND UNDER
1316    1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
1373    1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
...      ...                 ...                                      ...
238182  1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
238196  1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
238204  1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
238259  1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300
238312  1153  DECEPTIVE PRACTICE      FINANCIAL IDENTITY THEFT OVER $ 300

       Location Description  Arrest  Domestic  ...  Ward  Community Area  \
749                     NaN   False     False  ...  32.0              22
1258                    NaN   False     False  ...   4.0              39
1305                    NaN   False     False  ...  28.0              27
1316                    NaN   False     False  ...   4.0              39
1373                    NaN   False     False  ...  45.0              11
...                     ...     ...       ...  ...   ...             ...
238182                  NaN   False     False  ...  47.0               6
238196                  NaN   False     False  ...  27.0              28
238204                  NaN   False     False  ...   3.0              38
238259                  NaN   False     False  ...  41.0              10
238312                  NaN   False     False  ...  43.0               7

       FBI Code  X Coordinate Y Coordinate  Year              Updated On  \
749          11           NaN          NaN  2022  09/03/2022 04:48:29 PM
1258         11           NaN          NaN  2022  02/28/2022 03:47:25 PM
1305         11           NaN          NaN  2022  01/04/2023 03:49:45 PM
1316         11           NaN          NaN  2022  02/28/2022 03:47:25 PM
1373         11           NaN          NaN  2022  03/01/2022 03:51:28 PM
...         ...           ...          ...   ...                     ...
238182       11     1163856.0    1926604.0  2022  01/03/2023 03:46:28 PM
238196       11     1169140.0    1900400.0  2022  01/03/2023 03:46:28 PM
238204       11     1178021.0    1875881.0  2022  01/03/2023 03:46:28 PM
238259       11     1125839.0    1941361.0  2022  01/03/2023 03:46:28 PM
238312       11     1170513.0    1917030.0  2022  01/03/2023 03:46:28 PM

         Latitude  Longitude                      Location
749           NaN        NaN                           NaN
1258          NaN        NaN                           NaN
1305          NaN        NaN                           NaN
1316          NaN        NaN                           NaN
1373          NaN        NaN                           NaN
...           ...        ...                           ...
238182  41.954232 -87.673037  (41.954232251, -87.673036778)
```

```
238196  41.882214 -87.654375  (41.882213693, -87.654374823)
238204  41.814734 -87.622509  (41.814734427, -87.622509357)
238259  41.995446 -87.812465  (41.995445746, -87.812464832)
238312  41.927817 -87.648846  (41.927817456, -87.648845932)

[777 rows x 22 columns]
```

Notice that the value for Location Description for all returned rows is null. This is because the function isna() returns a True when the value is 0, and returns a False otherwise. True values satisfy the condition and therefore the rows are shown.

What if we wanted to return the rows where the value was not missing? Once again, pandas has what we need:

[66]:
```
chicago_crimes[chicago_crimes["Location Description"].notna()]
```

[66]:
```
              ID Case Number                  Date                    Block  \
0       12789052    JF350580  08/09/2022 04:07:00 PM     014XX W ELMDALE AVE
1       12790581    JF352712  08/10/2022 04:00:00 PM    062XX S ARTESIAN AVE
2       12790652    JF352659  08/11/2022 10:00:00 AM        094XX S STATE ST
3       12796135    JF359082  08/15/2022 09:14:00 PM      048XX S KARLOV AVE
4       12795972    JF359058  08/16/2022 04:10:00 PM      015XX S HALSTED ST
...          ...         ...                     ...                     ...
238311  12937237    JF527290  12/23/2022 08:41:00 AM  064XX W Irving Park Rd
238313  12936301    JF526810  12/22/2022 06:00:00 PM    020XX W CORNELIA AVE
238314  12936397    JF526745  12/19/2022 02:00:00 PM     044XX N ROCKWELL ST
238315  12935341    JF525383  12/20/2022 06:45:00 AM    027XX W ROOSEVELT RD
238316  12938501    JF523997  12/26/2022 10:30:00 PM       021XX W DEVON AVE

        IUCR     Primary Type          Description  \
0       0325          ROBBERY  VEHICULAR HIJACKING
1       0810            THEFT            OVER $500
2       0810            THEFT            OVER $500
3       0560          ASSAULT               SIMPLE
4       0820            THEFT       $500 AND UNDER
...      ...              ...                  ...
238311  0860            THEFT         RETAIL THEFT
238313  1320  CRIMINAL DAMAGE           TO VEHICLE
238314  0620         BURGLARY       UNLAWFUL ENTRY
238315  0810            THEFT            OVER $500
238316  0810            THEFT            OVER $500

                       Location Description  Arrest  Domestic  ...  Ward  \
0                                    STREET    True     False  ...  48.0
1                                    STREET   False     False  ...  16.0
2                                    STREET   False      True  ...   9.0
3                                 RESIDENCE   False     False  ...  14.0
4                                  SIDEWALK   False     False  ...  11.0
```

```
...                                        ...     ...       ...  ...   ...
238311                     SMALL RETAIL STORE   False     False  ...  38.0
238313                                 STREET   False     False  ...  32.0
238314                              APARTMENT   False     False  ...  47.0
238315                                 STREET   False     False  ...  28.0
238316  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  ...  50.0

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
0                   77        03     1165640.0    1939961.0  2022
1                   66        06     1161110.0    1863210.0  2022
2                   49        06     1177962.0    1842197.0  2022
3                   57       08A     1149844.0    1872244.0  2022
4                   28        06     1171290.0    1892413.0  2022
...                ...       ...           ...          ...   ...
238311              17        06     1132626.0    1925898.0  2022
238313               5        14     1161968.0    1923233.0  2022
238314               4        05     1158237.0    1929586.0  2022
238315              29        06     1158071.0    1894595.0  2022
238316               2        06     1160681.0    1942466.0  2022

                   Updated On   Latitude  Longitude  \
0       01/03/2023 03:46:28 PM  41.990846 -87.666096
1       01/03/2023 03:46:28 PM  41.780331 -87.684892
2       01/03/2023 03:46:28 PM  41.722303 -87.623745
3       01/03/2023 03:46:28 PM  41.805347 -87.725961
4       01/03/2023 03:46:28 PM  41.860250 -87.646715
...                        ...        ...        ...
238311  01/03/2023 03:46:28 PM  41.952898 -87.787861
238313  01/03/2023 03:46:28 PM  41.945022 -87.680072
238314  01/03/2023 03:46:28 PM  41.962532 -87.693611
238315  01/03/2023 03:46:28 PM  41.866517 -87.695179
238316  03/22/2023 04:47:43 PM  41.997825 -87.684267

                           Location
0       (41.990846423, -87.666096144)
1       (41.780330681, -87.684891779)
2       (41.722303228, -87.623745129)
3       (41.805347066, -87.725961264)
4        (41.860249838, -87.64671467)
...                               ...
238311  (41.952897791, -87.787860507)
238313  (41.945021752, -87.680071764)
238314  (41.962531969, -87.693611152)
238315  (41.866517317, -87.695178701)
238316  (41.997824802, -87.684266677)

[237540 rows x 22 columns]
```