"Discovering" is the beginning of an investigation

- **Video:** Welcome to week 2
  3 min
- **Video:** Yaser: Understand data to drive value
  2 min
- **Video:** Where the data comes from
  10 min
- **Reading:** Reference guide: Import datasets with Python
  20 min
- **Reading:** Reference guide: Pandas methods for the discovery of a dataset
  20 min
- **Reading:** Follow-along instructions: EDA using basic data functions with Python
  10 min
- **Lab:** Annotated follow-along resource: EDA using basic functions with Python
  20 min
- **Video:** EDA using basic data functions with Python
  10 min
- **Lab:** Activity: Discover what is in your dataset
  1h
- **Lab:** Exemplar: Discover what is in your dataset
  20 min
- **Practice Quiz:** Test your knowledge: Discovering is the beginning of an investigation
  3 questions

Understand data format

Create structure from raw data

Review: Explore raw data

Reference guide: Pandas methods for the discovery of a dataset

Python reference guide for EDA: Discovering

Use the following pandas methods and attributes to help you learn about a dataset when you encounter it for the first time.

Save this course item

You may want to save a copy of this guide for future reference. You can use it as a resource for additional practice or in your future professional projects. To access a downloadable version of this course item, click the link below and select "Use Template."

Reference guide: Pandas methods for the discovery of a dataset

OR

If you don't have a Google account, you can download the item directly from the attachment below.

Reference guide_ Python functions for the discovery of a dataset
DOCX File

DataFrame.head()

- The **`head()`** method will display the first *n* rows of the dataframe.
- In the argument field, input the number of rows you want displayed in a Python notebook. The default is 5 rows.
- Once executed, the **`head()`** method returns something like this:

`df.head(10)`

index	date	number of strikes	center point geom
0	2018-01-03	194	POINT(-75 27)
1	2018-01-03	41	POINT(-78.4 29)
2	2018-01-03	33	POINT(-73.9 27)
3	2018-01-03	38	POINT(-73.8 27)
4	2018-01-03	92	POINT(-79 28)
5	2018-01-03	119	POINT(-78 28)
6	2018-01-03	35	POINT(-79.3 28)
7	2018-01-03	60	POINT(-79.1 28)
8	2018-01-03	41	POINT(-78.7 28)
9	2018-01-03	119	POINT(-78.6 28)

Note: In a Python notebook, the results of **`head()`** will not include a table with visible grid lines.

DataFrame.info(X)

- The **`info()`** method will display a summary of the dataframe, including the range index, dtypes, column headers, and memory usage.
- Leaving the argument field blank will return a full summary. As an option, in the argument field you can type in **`show_counts=True`**, which will return the count of non-null values for each column.
- Once executed, the **`info()`** method returns something like this:

Note: The following code block is not interactive.

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex:3401012 entries, 0 to 3401011
Data columns (total 3 columns):
#     Column                  Dtype
--    ----                    -----
0     date                    object
1     number_of_strikes       int64
2     center_point_geom       object
Dtypes: int64(1), object(2)
Memory usage 77.8+ MB
```

DataFrame.describe()

- The **`describe()`** method will return descriptive statistics of the entire dataset, including total count, mean, minimum, maximum, dispersion, and distribution.
- Leaving the argument field blank will default to returning a summary of the data frame's statistics. As an option, you can use "include=[X]" and "exclude=[X]" which will limit the results to specific data types, depending on what you input in the brackets.
- Once executed, the **`describe()`** method returns something like this:

`df_joined.describe()`

N/A	longitude	latitude	number_of_strikes_x	number_of_strikes_y
count	717530.00	717530.00	717530.00	323700.00000
mean	-90.875445	33.328572	21.637081	25.410587
std	13.648429	7.938831	48.02952	57.421824
min	-133.9000	16.600000	1.00000	1.000000
25%	-102.80000	26.900000	3.00000	3.000000
50%	-90.300000	33.200000	6.00000	8.000000
75%	-80.900000	39.400000	21.00000	24.000000
max	-43.800000	51.700000	2211.00000	2211.000000

Note: In a Python notebook, the results of **`describe()`** will not include a table with visible grid lines.

DataFrame.shape

- **`shape`** is an attribute that returns a tuple representing the dimensions of the dataframe by number of rows and columns. Remember that attributes are not followed by parentheses. The code will look something like this:

Note: The following code block is not interactive.

```
df.shape
(3401012, 3)
```

Key takeaways

`head()`, **`info()`**, **`describe()`**, and **`shape`** are pandas tools that data scientists can use to understand a dataset at a high level. The information learned from using these tools will serve to inform the remainder of your EDA work when you use pandas to analyze data throughout your career.

Resources for more information

For more information on the EDA discovering functions above and others like it, you can use the online Pandas reference guide:

- A list of Pandas dataframe functions

Mark as completed

Like Dislike Report an issue