Day7_Start_Pandas

August 2, 2021

Day 7: Intro to Pandas

Goals for the day:

- Learn how to import & export CSVs in pandas
- First glances at the data: Head, keys, sort
- Learn how to index, add, and remove data within a dataset: (.iloc, .loc), set_index

Functions Learned:

- Make an empty data frame: `pd.DataFrame()`
- View top lines of a dataframe: `head()`
- View last lines of a dataframe: `tail()`
- Select data based on position: `df.iloc[row,column]`
- Select data based on value: `.loc['value']`
- Set an index: `set_index()`
- Sort by a specific value: `sort_values()`

1.Loading Pandas

1. Now we are going to use pandas. Pandas is the Python Data Analysis Library and is popular because it allows the user to manipulate and clean large amount of data.

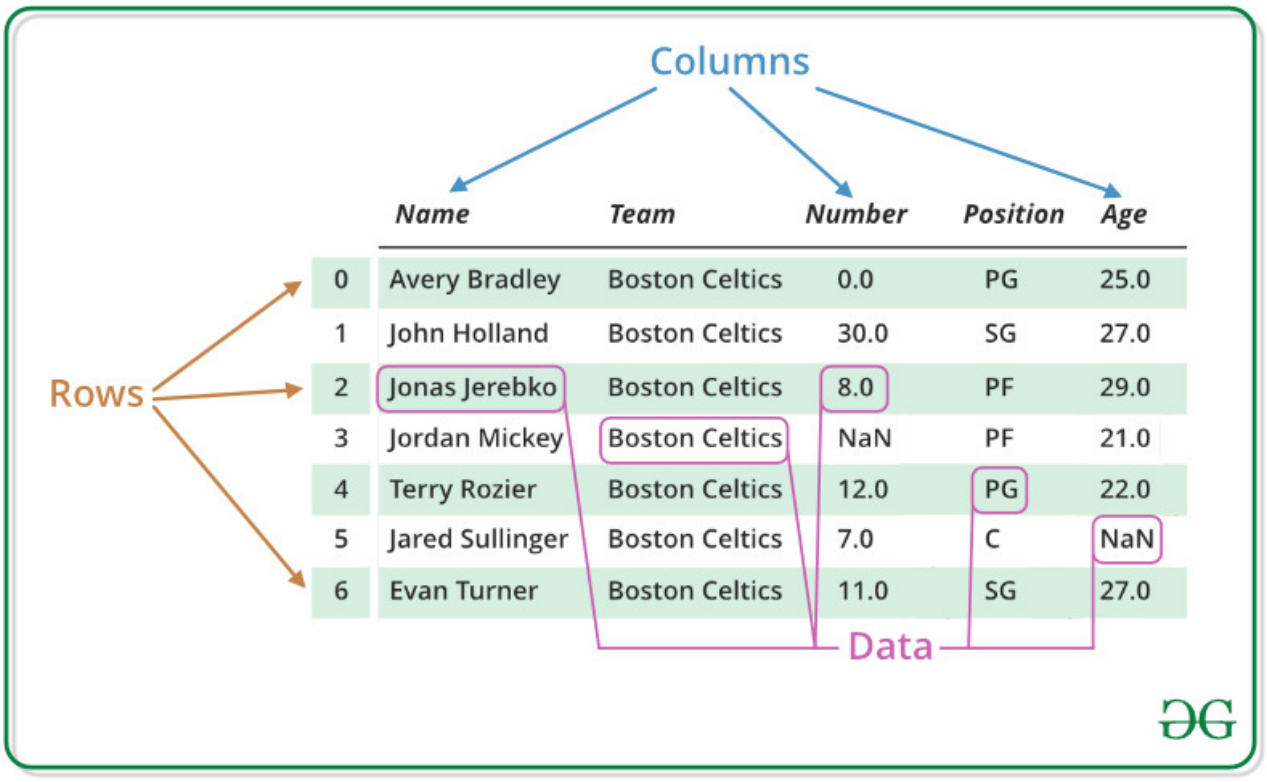
Pandas and numpy come from the SciPy library and much of the Pandas data analysis is similar to Numpy. While numpy works with numerical arrays, Pandas works with series and DataFrames that can have mixed datatypes. Pandas lets us take complicated datasets (dates, long names, missing data) and anlyze them.

You can think of it like a supercharged excel where you combine the organization of excel with the power of a programming language.

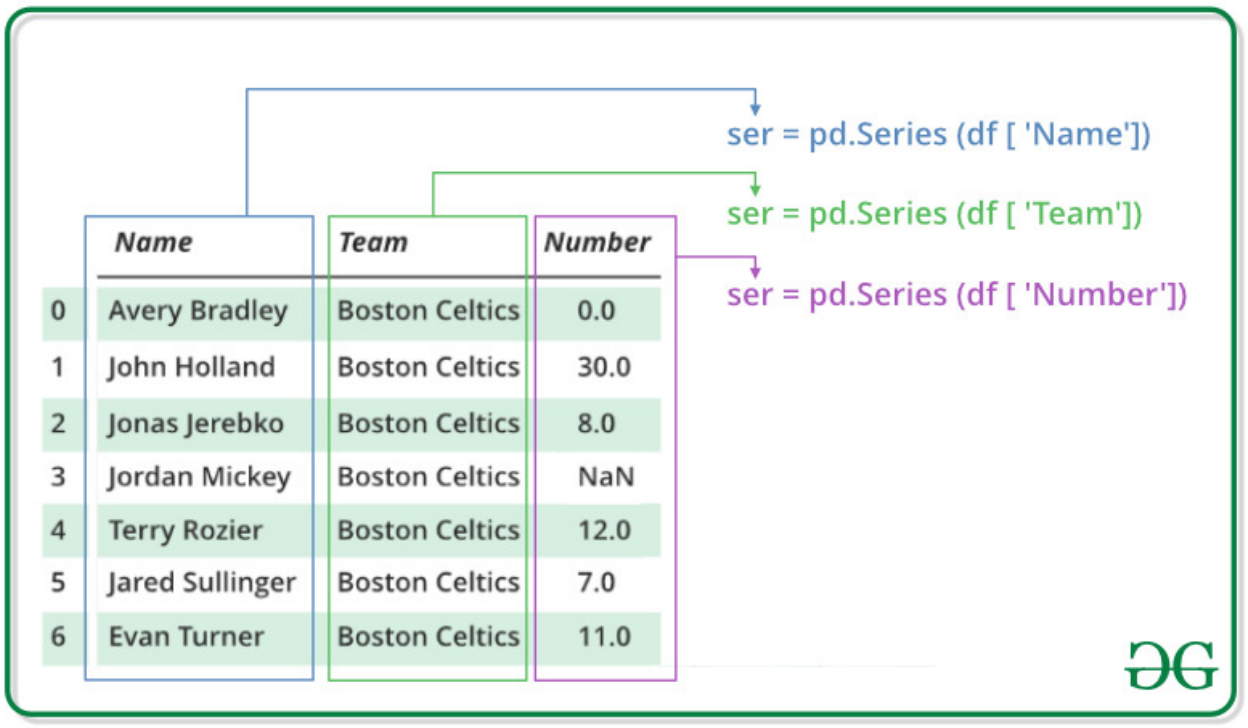
2. Just like we use np as a shortcut for numpy, we use pd for pandas -import pandas as pd
3. On a final note, you can see I made a numbered list in markdown. To do that, you type a number, a period, and then two spaces.

[30]:
```
import pandas as pd
import numpy as np
```

1.1 DataFrame Intro



Columns are name of series and must usually contain the same data type, when this is not the case you get into many issues.



2. DataFrame from scratch

While most of the time you will work with data that is already in a tabular format, it is important that you know how to construct a dataframe from scratch.

[2]:
```
##make an empty dataframe
my_df=pd.DataFrame()
my_df
```

[2]:
```
Empty DataFrame
Columns: []
Index: []
```

Each column and row in a dataframe can be considered as a series and can be str or numeric, or, if you are evil, a mix of datatypes. So we can add columns/rows by adding series, lists, sets, you name it.

[3]:
```
## create a list with your first and last name and add this to your df
my_info=['Maria','Hernandez']

my_df['Name']=my_info
my_df
```

[3]:
```
        Name
0      Maria
1  Hernandez
```

[4]:
```
## let's add a row with my middle initail
my_df=my_df.append({'Name':'D'},ignore_index=True)
my_df
```

[4]:

	Name
0	Maria
1	Hernandez
2	D

In the previous cell we said to ignore the index. The index is the name pandas gives to the rows. The index always contains a unique identifier for every row. When we tell a function to ignore the index, we are ignoring the current labels and adding one more value. However, our new value will have an index and won't mess up pandas labeling.

2.1 More dataframe making tricks

[5]:
```
### make an empty frame in one step by specifing the col and index
my_df = pd.DataFrame(1,columns=['User_ID', 'UserName', 'Action'], index=['a',␣
 ↪'b', 'c'])
my_df
```

[5]:

	User_ID	UserName	Action
a	1	1	1
b	1	1	1
c	1	1	1

[6]:
```
##make a dataframe using a dictionary

my_df= pd.DataFrame({'Col1':[100,200,300],'Col2':['A','B','C']})
my_df
```

[6]:

	Col1	Col2
0	100	A
1	200	B
2	300	C

[7]:
```
## make dataframe using list

#define your lists, this will be the columns
my_list1=['Mercury','Venus','Earth']
my_list2=['hot','hot','perfect']

#call the dataframe construction
#the first item is your list zipped together as one
#the second is the index labels you want
#the third is the column names
my_df=pd.DataFrame(list(zip(my_list1, my_list2)),index =[0, 1, 2],columns␣
 ↪=['Planet','Temp'])
```

```
my_df
```

[7]:

	Planet	Temp
0	Mercury	hot
1	Venus	hot
2	Earth	perfect

We will talk more about data manipulation tomorrow. You can find more info on working with empty dataframes here.

2.1 Skills practice

Make a dataframe with two columns, one column with your favorite three names and a second column with the number of letters in those names. You can use whatever method you want.

[8]:
```
#### your work here
##tip: copy the code for your favorite method from above, and edit your code
```

2.1 Answer

[9]:
```
## this is my favorite method
my_new_df= pd.DataFrame({'Col1':['Gohan','Naruto','Luffy'],'Col2':[5,6,5]})
my_new_df
```

[9]:

	Col1	Col2
0	Gohan	5
1	Naruto	6
2	Luffy	5

3. Read in Data

3.0 Set directory: Showing Pandas where the files are

Our data files are in the folder you downloaded called data. We can tell python where that data is once so you don't have to type the path everytime.

[10]:
```
#this is the specific directory where the data we want to use is stored
datadirectory = '../data/'

#this is the directory where we want to store the data we finish analyzing
data_out_directory='../output/'
```

Pandas has many built in function that we can call by doing pd.(function we want). Here is a list of functions we can use to read in (input) a document based on the kind of data you are working with. We can also save (output) new tables we create.

[11]:
```
#type help(pd.read_csv) to learn more about the options
#help(pd.read_csv)
```

```
#if you encounter any errors running help(pd.read_csv) rerun the following␣
 ↪command:
#import pandas as pd
```

Load Pokedex

[12]:
```
#stop and make sure everyone can load the data

#to read in a csv call read_csv from pd which looks like pd.read_csv
pokemon_csv=pd.read_csv(datadirectory+'Pokemon.csv')

#to see it we can call print
print (pokemon_csv)
```

```
     Number                     Name   Type 1  Type 2  Total  HP  Attack  \
0         1                Bulbasaur    Grass  Poison    318  45      49
1         2                  Ivysaur    Grass  Poison    405  60      62
2         3                 Venusaur    Grass  Poison    525  80      82
3         3  VenusaurMega Venusaur      Grass  Poison    625  80     100
4         4                Charmander     Fire     NaN    309  39      52
..      ...                       ...      ...     ...    ...  ..     ...
795     719                  Diancie     Rock   Fairy    600  50     100
796     719     DiancieMega Diancie      Rock   Fairy    700  50     160
797     720     HoopaHoopa Confined   Psychic   Ghost    600  80     110
798     720      HoopaHoopa Unbound   Psychic    Dark    680  80     160
799     721                Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         49       65       65     45           1      False
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
..       ...      ...      ...    ...         ...        ...
795      150      100      150     50           6       True
796      110      160      110    110           6       True
797       60      150      130     70           6       True
798       60      170      130     80           6       True
799      120      130       90     70           6       True

[800 rows x 13 columns]
```

[13]:
```
#or we can just have the name alone as the last line
pokemon_csv
```

[13]:
```
     Number                   Name   Type 1  Type 2  Total  HP  Attack  \
0         1              Bulbasaur    Grass  Poison    318  45      49
```

```
1         2                Ivysaur    Grass  Poison    405  60      62
2         3               Venusaur    Grass  Poison    525  80      82
3         3  VenusaurMega Venusaur    Grass  Poison    625  80     100
4         4             Charmander     Fire     NaN    309  39      52
..      ...                    ...      ...     ...    ...  ..     ...
795     719                Diancie     Rock   Fairy    600  50     100
796     719    DiancieMega Diancie     Rock   Fairy    700  50     160
797     720    HoopaHoopa Confined  Psychic   Ghost    600  80     110
798     720     HoopaHoopa Unbound  Psychic    Dark    680  80     160
799     721              Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         49       65       65     45           1      False
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
..       ...      ...      ...    ...         ...        ...
795      150      100      150     50           6       True
796      110      160      110    110           6       True
797       60      150      130     70           6       True
798       60      170      130     80           6       True
799      120      130       90     70           6       True

[800 rows x 13 columns]
```

3.1 Pro-tip: How to read txt files, or files with non ',' delimeters

[14]:
```
#let's try reading in the .txt file with pd.read_csv and see what happens
pokemon_txt=pd.read_csv(datadirectory+'Pokemon.txt')
pokemon_txt.head()
```

[14]:
```
  Number\tName\tType 1\tType 2\tTotal\tHP\tAttack\tDefense\tSp. Atk\tSp.
Def\tSpeed\tGeneration\tLegendary
0  1\tBulbasaur\tGrass\tPoison\t318\t45\t49\t49\t...
1  2\tIvysaur\tGrass\tPoison\t405\t60\t62\t63\t80...
2  3\tVenusaur\tGrass\tPoison\t525\t80\t82\t83\t1...
3  3\tVenusaurMega Venusaur\tGrass\tPoison\t625\t...
4  4\tCharmander\tFire\t\t309\t39\t52\t43\t60\t50...
```

Because the delimiter (the symbol that separates data entries) is , in a csv reading a txt that is delimited does not work. There are two ways around this:

[15]:
```
# method 1, call pd.read_csv and set the delimiter to '\t' to override the ','␣
 ↪default
pokemon_txt=pd.read_csv(datadirectory+'Pokemon.txt',delimiter='\t')
pokemon_txt
```

[15]:
```
     Number                    Name   Type 1  Type 2  Total  HP  Attack  \
0         1               Bulbasaur    Grass  Poison    318  45      49
1         2                 Ivysaur    Grass  Poison    405  60      62
2         3                Venusaur    Grass  Poison    525  80      82
3         3  VenusaurMega Venusaur    Grass  Poison    625  80     100
4         4              Charmander     Fire     NaN    309  39      52
..      ...                     ...      ...     ...    ...  ..     ...
795     719                 Diancie     Rock   Fairy    600  50     100
796     719    DiancieMega Diancie     Rock   Fairy    700  50     160
797     720    HoopaHoopa Confined  Psychic   Ghost    600  80     110
798     720     HoopaHoopa Unbound  Psychic    Dark    680  80     160
799     721              Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         49       65       65     45           1      False
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
..       ...      ...      ...    ...         ...        ...
795      150      100      150     50           6       True
796      110      160      110    110           6       True
797       60      150      130     70           6       True
798       60      170      130     80           6       True
799      120      130       90     70           6       True

[800 rows x 13 columns]
```

[16]:
```
#or call pd.read_table that has '\t' as the default delimeter
pokemon_txt=pd.read_table(datadirectory+'Pokemon.txt')
pokemon_txt
```

[16]:
```
     Number                    Name   Type 1  Type 2  Total  HP  Attack  \
0         1               Bulbasaur    Grass  Poison    318  45      49
1         2                 Ivysaur    Grass  Poison    405  60      62
2         3                Venusaur    Grass  Poison    525  80      82
3         3  VenusaurMega Venusaur    Grass  Poison    625  80     100
4         4              Charmander     Fire     NaN    309  39      52
..      ...                     ...      ...     ...    ...  ..     ...
795     719                 Diancie     Rock   Fairy    600  50     100
796     719    DiancieMega Diancie     Rock   Fairy    700  50     160
797     720    HoopaHoopa Confined  Psychic   Ghost    600  80     110
798     720     HoopaHoopa Unbound  Psychic    Dark    680  80     160
799     721              Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         49       65       65     45           1      False
```

```
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
..       ...      ...      ...    ...         ...        ...
795      150      100      150     50           6       True
796      110      160      110    110           6       True
797       60      150      130     70           6       True
798       60      170      130     80           6       True
799      120      130       90     70           6       True

[800 rows x 13 columns]
```

3.2 Pro-tip 2: Data input options

As you may expect, the data you will work with may not always look this neat, you may have to skip rows or change delimeters. There are many ways to do that. I won't spend time discussing how to do that here but I want you to know that you can find that information in the pandas documentation.

[17]:
```
## the dataset that we are dealing with is 75KB, sometimes you'll have lots of␣
 ↪data but you don't need it all
## redicing the data you read in will speed up pandas
# pd.info() gives you the information for each column and the total memory usage
pokemon_selected=pd.read_csv(datadirectory+'Pokemon.csv')
pokemon_selected.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Number      800 non-null    int64
 1   Name        800 non-null    object
 2   Type 1      800 non-null    object
 3   Type 2      414 non-null    object
 4   Total       800 non-null    int64
 5   HP          800 non-null    int64
 6   Attack      800 non-null    int64
 7   Defense     800 non-null    int64
 8   Sp. Atk     800 non-null    int64
 9   Sp. Def     800 non-null    int64
 10  Speed       800 non-null    int64
 11  Generation  800 non-null    int64
 12  Legendary   800 non-null    bool
dtypes: bool(1), int64(9), object(3)
memory usage: 75.9+ KB
```

[18]:
```
### lets call in only call the columns you may use like 'Name' and 'Speed',
pokemon_selected=pd.read_csv(datadirectory+'Pokemon.
 ↪csv',usecols=['Name','Speed'])
pokemon_selected
```

[18]:

	Name	Speed
0	Bulbasaur	45
1	Ivysaur	60
2	Venusaur	80
3	VenusaurMega Venusaur	80
4	Charmander	65
..	...	...
795	Diancie	50
796	DiancieMega Diancie	110
797	HoopaHoopa Confined	70
798	HoopaHoopa Unbound	80
799	Volcanion	70

[800 rows x 2 columns]

[19]:
```
#this reduces the memory usage and will speed up your analysis
pokemon_selected.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    800 non-null    object
 1   Speed   800 non-null    int64
dtypes: int64(1), object(1)
memory usage: 12.6+ KB
```

[20]:
```
### let's only read in the first 5 rows
pokemon_selected=pd.read_csv(datadirectory+'Pokemon.
 ↪csv',usecols=['Name','Speed'],nrows=5)
pokemon_selected
```

[20]:

	Name	Speed
0	Bulbasaur	45
1	Ivysaur	60
2	Venusaur	80
3	VenusaurMega Venusaur	80
4	Charmander	65

4. Viewing the dataframe

I'm going to make a copy of the pokemon_csv with a shorter name unless you make a copy of a dataframe changes that you make will happen to your original dataframe. This is really important when we talk about indexing! Pro-tip: when making changes to your dataframe always make a copy

in python 'A.B' means that B belongs to A

View head

[21]:
```
#to view top lines we use .head() notice this is also a fucntion so this means you have options
pokemon_df=pokemon_csv.copy()
pokemon_df.head()
```

[21]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

View tail

[22]:
```
# to view the last lines we use .tail()
pokemon_df.tail()
```

[22]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
795	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	True
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	True
797	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	True
798	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	True
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	True

4.1 Skill check

I want to see the top 7 and then the bottom 3 lines

[23]:
```
### your code here

##head



###tail
```

Answer

[24]:
```
##answer
pokemon_df.head(7)
pokemon_df.tail(3)
```

[24]:
```
     Number                  Name   Type 1 Type 2  Total  HP  Attack  Defense  \
797     720  HoopaHoopa Confined  Psychic  Ghost    600  80     110       60
798     720   HoopaHoopa Unbound  Psychic   Dark    680  80     160       60
799     721            Volcanion     Fire  Water    600  80     110      120

     Sp. Atk  Sp. Def  Speed  Generation  Legendary
797      150      130     70           6       True
798      170      130     80           6       True
799      130       90     70           6       True
```

More viewing

[25]:
```
pokemon_df.head()
```

[25]:
```
   Number                    Name Type 1  Type 2  Total  HP  Attack  Defense  \
0       1               Bulbasaur  Grass  Poison    318  45      49       49
1       2                 Ivysaur  Grass  Poison    405  60      62       63
2       3                Venusaur  Grass  Poison    525  80      82       83
3       3  VenusaurMega Venusaur  Grass  Poison    625  80     100      123
4       4              Charmander   Fire     NaN    309  39      52       43

   Sp. Atk  Sp. Def  Speed  Generation  Legendary
0       65       65     45           1      False
1       80       80     60           1      False
2      100      100     80           1      False
3      122      120     80           1      False
4       60       50     65           1      False
```

view a column

[26]:
```
## can use . notation
pokemon_df.Name
```

[26]:
```
0                  Bulbasaur
1                    Ivysaur
2                   Venusaur
3      VenusaurMega Venusaur
4                 Charmander
               ...
795                  Diancie
796      DiancieMega Diancie
797      HoopaHoopa Confined
798       HoopaHoopa Unbound
799                Volcanion
Name: Name, Length: 800, dtype: object
```

[27]:
```
## but this gets messy when the names have symbols, like spaces
pokemon_df.Type 1
```

```
  File "<ipython-input-27-ba86f7d208d3>", line 2
    pokemon_df.Type 1
                    ^
SyntaxError: invalid syntax
```

[31]:
```
# we can use []
pokemon_df['Type 1']
```

[31]:
```
0        Grass
1        Grass
2        Grass
3        Grass
4         Fire
        ...
795       Rock
796       Rock
797    Psychic
798    Psychic
799       Fire
Name: Type 1, Length: 800, dtype: object
```

[32]:
```
### print Attack

print (pokemon_df['Defense'])
print(pokemon_df['Attack'])
```

```
0       49
```

```
1       63
2       83
3      123
4       43
      ...
795    150
796    110
797     60
798     60
799    120
Name: Defense, Length: 800, dtype: int64
0       49
1       62
2       82
3      100
4       52
      ...
795    100
796    160
797    110
798    160
799    110
Name: Attack, Length: 800, dtype: int64
```

!!Using brackets to call your columns is the cleanest way!

get column names

[33]:
```
#we can get the names of the colums with multiple ways
# list(df), df.keys(), df.columns
#print (list(pokemon_df))

pokemon_df.columns
```

[33]:
```
Index(['Number', 'Name', 'Type 1', 'Type 2', 'Total', 'HP', 'Attack',
       'Defense', 'Sp. Atk', 'Sp. Def', 'Speed', 'Generation', 'Legendary'],
      dtype='object')
```

get the row or 'index' names

[34]:
```
pokemon_df.index
```

[34]:
```
RangeIndex(start=0, stop=800, step=1)
```

set the index

[35]:
```
##change the index to name and since we want the change in our current dataframe␣
 ↪we use inplace=True
pokemon_df.set_index('Name',inplace=True)
pokemon_df.head()
```

[35]:
```
                        Number Type 1  Type 2  Total  HP  Attack  Defense  \
Name
Bulbasaur                    1  Grass  Poison    318  45      49       49
Ivysaur                      2  Grass  Poison    405  60      62       63
Venusaur                     3  Grass  Poison    525  80      82       83
VenusaurMega Venusaur        3  Grass  Poison    625  80     100      123
Charmander                   4   Fire     NaN    309  39      52       43

                        Sp. Atk  Sp. Def  Speed  Generation  Legendary
Name
Bulbasaur                    65       65     45           1      False
Ivysaur                      80       80     60           1      False
Venusaur                    100      100     80           1      False
VenusaurMega Venusaur       122      120     80           1      False
Charmander                   60       50     65           1      False
```

[36]:
```
##view index
pokemon_df.index
```

[36]:
```
Index(['Bulbasaur', 'Ivysaur', 'Venusaur', 'VenusaurMega Venusaur',
       'Charmander', 'Charmeleon', 'Charizard', 'CharizardMega Charizard X',
       'CharizardMega Charizard Y', 'Squirtle',
       ...
       'Noibat', 'Noivern', 'Xerneas', 'Yveltal', 'Zygarde50% Forme',
       'Diancie', 'DiancieMega Diancie', 'HoopaHoopa Confined',
       'HoopaHoopa Unbound', 'Volcanion'],
      dtype='object', name='Name', length=800)
```

[37]:
```
### RESET the index to avoid complications later
pokemon_df.reset_index(inplace=True)
pokemon_df.head()
```

[37]:
```
                   Name  Number Type 1  Type 2  Total  HP  Attack  Defense  \
0             Bulbasaur       1  Grass  Poison    318  45      49       49
1               Ivysaur       2  Grass  Poison    405  60      62       63
2              Venusaur       3  Grass  Poison    525  80      82       83
3  VenusaurMega Venusaur       3  Grass  Poison    625  80     100      123
4            Charmander       4   Fire     NaN    309  39      52       43

   Sp. Atk  Sp. Def  Speed  Generation  Legendary
0       65       65     45           1      False
1       80       80     60           1      False
2      100      100     80           1      False
3      122      120     80           1      False
4       60       50     65           1      False
```

5. Selecting specific data

You can select data from a pandas dataframe by selecting rows or columns. You can also select data based on position with .iloc or with condiitons using .loc. [or a combination of both using .ix, but this is archaic and strongly discouraged in the pandas community]

Sort by numeric values

[38]:
```
## view based on some sorted values. .sort_values
## tell ascending True if you want values to increase or False if you want them␣
 ↪to decrease
## tell inplace True if you want the changes to be made in our original df or␣
 ↪leave it blank
## but if you do this you must create a new variable that stores the the sorted␣
 ↪dataframe

pokemon_df.sort_values(by='Total', ascending=True, inplace=True)
pokemon_df.head()

# or
# pokemon_sorted=pokemon_df.sort_values(by='Total', ascending=True)
# pokemon_sorted.head()
```

[38]:

	Name	Number	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	\
206	Sunkern	191	Grass	NaN	180	30	30	30	30	
322	Azurill	298	Normal	Fairy	190	50	20	40	20	
446	Kricketot	401	Bug	NaN	194	37	25	41	25	
288	Wurmple	265	Bug	NaN	195	45	45	35	20	
16	Weedle	13	Bug	Poison	195	40	35	30	20	

	Sp. Def	Speed	Generation	Legendary
206	30	30	2	False
322	40	20	3	False
446	41	25	4	False
288	30	20	3	False
16	20	50	1	False

Sort by alphabetical order

[39]:
```
## mechanics are the same as sorting by numerical values, except that here you␣
 ↪must select a column
## with string to sort alphabetically
pokemon_df.sort_values(by='Name', ascending=False, inplace=True)
pokemon_df.head()
```

[39]:

	Name	Number	Type 1	Type 2	Total	HP	Attack	Defense	\
794	Zygarde50% Forme	718	Dragon	Ground	600	108	100	121	
695	Zweilous	634	Dark	Dragon	420	72	85	70	

```
46              Zubat      41  Poison  Flying    245   40      45       35
631             Zorua     570    Dark     NaN    330   40      65       40
632           Zoroark     571    Dark     NaN    510   60     105       60

     Sp. Atk  Sp. Def  Speed  Generation  Legendary
794       81       95     95           6       True
695       65       70     58           5      False
46        30       40     55           1      False
631       80       40     65           5      False
632      120       60    105           5      False
```

5.0 Skill check

Sort by increasing Attack

[40]:
```
#### your code here
```

Answer

[41]:
```
pokemon_df.sort_values(by='Attack', ascending=True, inplace=True)
pokemon_df.head()
```

[41]:
```
          Name  Number  Type 1 Type 2  Total   HP  Attack  Defense  Sp. Atk  \
488    Happiny     440  Normal    NaN    220  100       5        5       15
121    Chansey     113  Normal    NaN    450  250       5        5       35
261    Blissey     242  Normal    NaN    540  255      10       10       75
230    Shuckle     213     Bug   Rock    505   20      10      230       10
139   Magikarp     129   Water    NaN    200   20      10       55       15

     Sp. Def  Speed  Generation  Legendary
488       65     30           4      False
121      105     50           1      False
261      135     55           2      False
230      230      5           2      False
139       20     80           1      False
```

5.1 Select based on position using *.iloc*

.iloc uses intergers, the specifc locations of the rows and columns. To select data we run df.iloc[row,column]. The values here are inclusive of the first value but exlusive of the second so 0:3, would select 0,1,2. The i in front of loc means interger, so we are indexing with the interger location.

[42]:
```
#how do we see specific rows
print (pokemon_df.iloc[0,0])

pokemon_df.head()
```

Happiny

[42]:
```
          Name  Number  Type 1 Type 2  Total   HP  Attack  Defense  Sp. Atk  \
488    Happiny     440  Normal    NaN    220  100       5        5       15
121    Chansey     113  Normal    NaN    450  250       5        5       35
261    Blissey     242  Normal    NaN    540  255      10       10       75
230    Shuckle     213     Bug   Rock    505   20      10      230       10
139   Magikarp     129   Water    NaN    200   20      10       55       15

     Sp. Def  Speed  Generation  Legendary
488       65     30           4      False
121      105     50           1      False
261      135     55           2      False
230      230      5           2      False
139       20     80           1      False
```

5.1 Skill Practice:

-Select rows 10-20 and columns [Name, Attack, Speed] using iloc and save as a new dataframe called 'my_selected_frame'

[43]:
```
##### answer the skill practice here
```

5.1 answer

[44]:
```
## Indexing practice
## How would you select rows 10-20 and columns [Name, Attack, Speed]

my_selected_frame=pokemon_df.iloc[10:21,[1,6,10]].copy()
my_selected_frame
```

[44]:
```
     Number  Attack  Speed
198     183      20     40
322     298      20     20
189     175      20     20
254     235      20     75
733     665      22     29
394     360      23     23
484     436      24     23
303     280      25     40
17       14      25     35
356     325      25     60
187     173      25     15
```

[45]:
```
### frame that has rows 1,5,8, columns Name to total

my_selected_frame2=pokemon_df.iloc[[1,5,8],0:5]
my_selected_frame2
```

[45]:
```
        Name  Number  Type 1  Type 2  Total
121  Chansey     113  Normal     NaN    450
381   Feebas     349   Water     NaN    200
508  Mantyke     458   Water  Flying    345
```

[46]:
```
####reset index here

pokemon_df.reset_index(inplace=True)
pokemon_df
```

[46]:
```
     index                     Name  Number   Type 1    Type 2  Total   HP  \
0      488                  Happiny     440   Normal       NaN    220  100
1      121                  Chansey     113   Normal       NaN    450  250
2      261                  Blissey     242   Normal       NaN    540  255
3      230                  Shuckle     213      Bug      Rock    505   20
4      139                 Magikarp     129    Water       NaN    200   20
..     ...                      ...     ...      ...       ...    ...  ...
795    426   RayquazaMega Rayquaza     384   Dragon    Flying    780  105
796    429      DeoxysAttack Forme     386  Psychic       NaN    600   50
797    424   GroudonPrimal Groudon     383   Ground      Fire    770  100
798    232  HeracrossMega Heracross     214      Bug  Fighting    600   80
799    163     MewtwoMega Mewtwo X     150  Psychic  Fighting    780  106

     Attack  Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         5        5       15       65     30           4      False
1         5        5       35      105     50           1      False
2        10       10       75      135     55           2      False
3        10      230       10      230      5           2      False
4        10       55       15       20     80           1      False
..      ...      ...      ...      ...    ...         ...        ...
795     180      100      180      100    115           3       True
796     180       20      180       20    150           3       True
797     180      160      150       90     90           3       True
798     185      115       40      105     75           2      False
799     190      100      154      100    130           1       True

[800 rows x 14 columns]
```

[47]:
```
pokemon_df=pd.read_csv(datadirectory+'Pokemon.csv')
pokemon_df
```

[47]:
```
     Number                     Name   Type 1  Type 2  Total  HP  Attack  \
0         1                Bulbasaur    Grass  Poison    318  45      49
1         2                  Ivysaur    Grass  Poison    405  60      62
2         3                 Venusaur    Grass  Poison    525  80      82
3         3  VenusaurMega Venusaur    Grass  Poison    625  80     100
4         4               Charmander     Fire     NaN    309  39      52
```

```
..      ...                    ...      ...     ...    ...  ..     ...
795     719                Diancie     Rock   Fairy    600  50     100
796     719    DiancieMega Diancie     Rock   Fairy    700  50     160
797     720    HoopaHoopa Confined  Psychic   Ghost    600  80     110
798     720     HoopaHoopa Unbound  Psychic    Dark    680  80     160
799     721              Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         49       65       65     45           1      False
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
..       ...      ...      ...    ...         ...        ...
795      150      100      150     50           6       True
796      110      160      110    110           6       True
797       60      150      130     70           6       True
798       60      170      130     80           6       True
799      120      130       90     70           6       True

[800 rows x 13 columns]
```

5.2 Select based on condition using *.loc*

This works the same as loc in that you need to specify a row and column value. The difference is that instead of using position you specify a value.

[48]:
```
##change the index to Name so that we can refer to pokemon by name.
## This is why being able to change the index is so useful.

pokemon_by_Name=pokemon_df.set_index('Name')
#pokemon_by_Name.head()

pokemon_by_Name.loc['Pichu':'Lugia','Total':'Defense']
```

[48]:
```
                         Total   HP  Attack  Defense
Name
Pichu                      205   20      40       15
Cleffa                     218   50      25       28
Igglybuff                  210   90      30       15
Togepi                     245   35      20       65
Togetic                    405   55      40       85
...                        ...  ...     ...      ...
Larvitar                   300   50      64       50
Pupitar                    410   70      84       70
Tyranitar                  600  100     134      110
TyranitarMega Tyranitar    700  100     164      150
```

```
Lugia                      680  106      90      130

[84 rows x 4 columns]
```

[49]:
```
##change the index to Name so that we can refer to pokemon by name.
## This is why being able to change the index is so useful.

## Be ready for it to not work (remeber the sorting we did)

pokemon_by_Name=pokemon_df.set_index('Name')

pokemon_by_Name.loc['Lugia':'Pichu','Total':'Defense']
```

[49]:
```
Empty DataFrame
Columns: [Total, HP, Attack, Defense]
Index: []
```

We can alaso use conditional statements to select values.

[50]:
```
### print the stats of your favorite pokemon? let's try Pichu
#here we are telling pandas to select the rows where the column 'Name' is␣
 ↪'Pichu', and all the columns

pokemon_df.loc[pokemon_df['Name']=='Pichu',]
```

[50]:
```
     Number   Name    Type 1 Type 2  Total  HP  Attack  Defense  Sp. Atk  \
186     172  Pichu  Electric    NaN    205  20      40       15       35

     Sp. Def  Speed  Generation  Legendary
186       35     60           2      False
```

[51]:
```
pichu=pokemon_df.loc[pokemon_df['Name']=='Pichu',].copy()
pichu
```

[51]:
```
     Number   Name    Type 1 Type 2  Total  HP  Attack  Defense  Sp. Atk  \
186     172  Pichu  Electric    NaN    205  20      40       15       35

     Sp. Def  Speed  Generation  Legendary
186       35     60           2      False
```

[52]:
```
## view based on boolean, where legendary is true
#here we are telling pandas to select the rows where the column 'Legendary' is␣
 ↪'True',and all the columns
pokemon_df.loc[pokemon_df['Legendary']==True,].head()
```

[52]:
```
     Number                Name    Type 1    Type 2  Total  HP  Attack  \
156     144            Articuno       Ice    Flying    580  90      85
157     145              Zapdos  Electric    Flying    580  90      90
```

```
158     146              Moltres      Fire    Flying    580   90     100
162     150               Mewtwo   Psychic       NaN    680  106     110
163     150  MewtwoMega Mewtwo X   Psychic  Fighting    780  106     190

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
156      100       95      125     85           1       True
157       85      125       90    100           1       True
158       90      125       85     90           1       True
162       90      154       90    130           1       True
163      100      154      100    130           1       True
```

[53]:
```
##select Fire or grass in Type 1
#here we are telling pandas to select the rows where the column 'Type 1'␣
 ↪contains the words Fire or Grass,and all the columns
# the | means 'or', while & means 'and'

pokemon_df.loc[pokemon_df['Type 1'].str.contains('Fire|Grass'),]
```

[53]:
```
     Number                  Name Type 1  Type 2  Total   HP  Attack  \
0         1             Bulbasaur  Grass  Poison    318   45      49
1         2               Ivysaur  Grass  Poison    405   60      62
2         3              Venusaur  Grass  Poison    525   80      82
3         3  VenusaurMega Venusaur  Grass  Poison    625   80     100
4         4            Charmander   Fire     NaN    309   39      52
..      ...                   ...    ...     ...    ...  ...     ...
735     667               Litleo   Fire  Normal    369   62      50
736     668               Pyroar   Fire  Normal    507   86      68
740     672               Skiddo  Grass     NaN    350   66      65
741     673               Gogoat  Grass     NaN    531  123     100
799     721            Volcanion   Fire   Water    600   80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
0         49       65       65     45           1      False
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
..       ...      ...      ...    ...         ...        ...
735       58       73       54     72           6      False
736       72      109       66    106           6      False
740       48       62       57     52           6      False
741       62       97       81     68           6      False
799      120      130       90     70           6       True

[122 rows x 13 columns]
```

[54]:
```
## we can also select based on specifc numerical conditions
#here we are telling pandas to select the rows where the column 'Speed' is␣
 ↪greater than 50,and all the columns
pokemon_df.loc[pokemon_df['Speed']>50,:]
```

[54]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	\
1	2	Ivysaur	Grass	Poison	405	60	62	
2	3	Venusaur	Grass	Poison	525	80	82	
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	
4	4	Charmander	Fire	NaN	309	39	52	
5	5	Charmeleon	Fire	NaN	405	58	64	
..	...	...	...	...	...	...	...	
794	718	Zygarde50% Forme	Dragon	Ground	600	108	100	
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	
797	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	
798	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	
799	721	Volcanion	Fire	Water	600	80	110	

	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
1	63	80	80	60	1	False
2	83	100	100	80	1	False
3	123	122	120	80	1	False
4	43	60	50	65	1	False
5	58	80	65	80	1	False
..	...	...	...	...	...	...
794	121	81	95	95	6	True
796	110	160	110	110	6	True
797	60	150	130	70	6	True
798	60	170	130	80	6	True
799	120	130	90	70	6	True

[537 rows x 13 columns]

[55]:
```
## how would you select the data where the speed is between 2 values, (without␣
 ↪between)
## notice I left the column section empty, when you do this pandas assumes you␣
 ↪want all the columns
pokemon_df.loc[((pokemon_df['Speed']>50) & (pokemon_df['Speed']<80)),]

#or
pokemon_df.loc[pokemon_df['Speed'].between(50,80),]
```

[55]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	\
1	2	Ivysaur	Grass	Poison	405	60	62	
2	3	Venusaur	Grass	Poison	525	80	82	
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	
4	4	Charmander	Fire	NaN	309	39	52	

```
5         5             Charmeleon     Fire     NaN    405  58      64
..      ...                    ...      ...     ...    ...  ..     ...
790     714                Noibat   Flying  Dragon    245  40      30
795     719               Diancie     Rock   Fairy    600  50     100
797     720    HoopaHoopa Confined  Psychic   Ghost    600  80     110
798     720     HoopaHoopa Unbound  Psychic    Dark    680  80     160
799     721              Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
1         63       80       80     60           1      False
2         83      100      100     80           1      False
3        123      122      120     80           1      False
4         43       60       50     65           1      False
5         58       80       65     80           1      False
..       ...      ...      ...    ...         ...        ...
790       35       45       40     55           6      False
795      150      100      150     50           6       True
797       60      150      130     70           6       True
798       60      170      130     80           6       True
799      120      130       90     70           6       True

[320 rows x 13 columns]
```

[56]:
```
## what if we wanted Speed or attack over 100?
pokemon_df.loc[((pokemon_df['Speed']>100) | (pokemon_df['Attack']>100)),]
```

[56]:
```
     Number                       Name   Type 1  Type 2  Total   HP  Attack  \
7         6  CharizardMega Charizard X     Fire  Dragon    634   78     130
8         6  CharizardMega Charizard Y     Fire  Flying    634   78     104
12        9    BlastoiseMega Blastoise    Water     NaN    630   79     103
19       15      BeedrillMega Beedrill      Bug  Poison    495   65     150
22       18                    Pidgeot   Normal  Flying    479   83      80
..      ...                        ...      ...     ...    ...  ...     ...
793     717                    Yveltal     Dark  Flying    680  126     131
796     719        DiancieMega Diancie     Rock   Fairy    700   50     160
797     720        HoopaHoopa Confined  Psychic   Ghost    600   80     110
798     720         HoopaHoopa Unbound  Psychic    Dark    680   80     160
799     721                  Volcanion     Fire   Water    600   80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
7        111      130       85    100           1      False
8         78      159      115    100           1      False
12       120      135      115     78           1      False
19        40       15       80    145           1      False
22        75       70       70    101           1      False
..       ...      ...      ...    ...         ...        ...
793       95      131       98     99           6       True
```

```
796      110      160      110      110           6       True
797       60      150      130       70           6       True
798       60      170      130       80           6       True
799      120      130       90       70           6       True

[238 rows x 13 columns]
```

5.2 Skill Practice:

Print the pokemons that have Attack between 90 and 120 and Type 1 contains Dark or Dragon. We only want the columns Name, Attack, Type 1. Hint: Be careful with parentheses

[57]:
```
##### answer the skill practice here
```

5.2 answer

[58]:
```
#Print the pokemons that have Attack is between 20 and 100, and speed is >80,␣
 ↪and we only want their names, type, attack

pokemon_df.loc[(pokemon_df['Attack'].between(90,120) & (pokemon_df['Type 1'].str.
 ↪contains('Dark|Dragon'))),['Name','Attack','Type 1']]
```

[58]:

	Name	Attack	Type 1
233	Sneasel	95	Dark
247	Houndoom	90	Dark
248	HoundoomMega Houndoom	90	Dark
285	Mightyena	90	Dark
366	AltariaMega Altaria	110	Dragon
407	Shelgon	95	Dragon
418	LatiasMega Latias	100	Dragon
419	Latios	90	Dragon
492	Gabite	90	Dragon
512	Weavile	120	Dark
549	Darkrai	90	Dark
621	Scrafty	90	Dark
632	Zoroark	105	Dark
672	Fraxure	117	Dragon
682	Druddigon	120	Dragon
696	Hydreigon	105	Dark
706	Reshiram	120	Dragon
712	KyuremWhite Kyurem	120	Dragon
757	Malamar	92	Dark
776	Goodra	100	Dragon
794	Zygarde50% Forme	100	Dragon

BREAK

6. Making some changes to the data

The data we often work with is a starting point for our analysis and we often have to add/remove or manipulate the data.

add columns

[59]:
```
## let's add a column
pokemon_df['test']=0
pokemon_df.head()
```

[59]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	test
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False	0
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False	0
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False	0
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False	0
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False	0

[60]:
```
## let's do some fancy column additions
## add a column named Boss, if attack is >100 using .loc
#notice what happens when attack is not >100

pokemon_df.loc[pokemon_df['Attack']>100,'Boss']=1
pokemon_df.head()
```

[60]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	test	Boss
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False	0	NaN
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False	0	NaN
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False	0	NaN
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False	0	NaN
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False	0	NaN

[61]:
```
## let's do some fancy column additions
## add a column named Boss, if attack is >100 using np.where
```

```
## notice what happend when attack is <100

pokemon_df['Boss2']=np.where(pokemon_df['Attack']>100,1,0)
pokemon_df
```

[61]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	test	Boss	Boss2
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False	0	NaN	0
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False	0	NaN	0
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False	0	NaN	0
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False	0	NaN	0
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False	0	NaN	0
..	...	...	...	...	...	..	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	True	0	NaN	0
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	True	0	1.0	1
797	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	True	0	1.0	1
798	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	True	0	1.0	1
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	True	0	1.0	1

[800 rows x 16 columns]

remove columns

[62]:
```
## lets remove the Boss column with del df[col]

del pokemon_df['Boss']
pokemon_df.head()
```

[62]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	test	Boss2
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False	0	0
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False	0	0
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False	0	0
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False	0	0
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False	0	0

remove multiple columns

[63]:
```
pokemon3=pokemon_df.drop(['Type 1', 'Type 2', 'Total','test'], axis=1)
pokemon3.head()
```

[63]:

	Number	Name	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Boss2
0	1	Bulbasaur	45	49	49	65	65	45	1	False	0
1	2	Ivysaur	60	62	63	80	80	60	1	False	0
2	3	Venusaur	80	82	83	100	100	80	1	False	0
3	3	VenusaurMega Venusaur	80	100	123	122	120	80	1	False	0
4	4	Charmander	39	52	43	60	50	65	1	False	0

[64]:
```
pokemon_df
```

[64]:

	Number	Name	Type 1	Type 2	Total	HP	Attack
0	1	Bulbasaur	Grass	Poison	318	45	49
1	2	Ivysaur	Grass	Poison	405	60	62
2	3	Venusaur	Grass	Poison	525	80	82
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100
4	4	Charmander	Fire	NaN	309	39	52
..	...	...	...	...	...	..	...
795	719	Diancie	Rock	Fairy	600	50	100

```
796     719    DiancieMega Diancie     Rock   Fairy    700  50     160
797     720    HoopaHoopa Confined  Psychic   Ghost    600  80     110
798     720     HoopaHoopa Unbound  Psychic    Dark    680  80     160
799     721              Volcanion     Fire   Water    600  80     110

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary  test  Boss2
0         49       65       65     45           1      False     0      0
1         63       80       80     60           1      False     0      0
2         83      100      100     80           1      False     0      0
3        123      122      120     80           1      False     0      0
4         43       60       50     65           1      False     0      0
..       ...      ...      ...    ...         ...        ...   ...    ...
795      150      100      150     50           6       True     0      0
796      110      160      110    110           6       True     0      1
797       60      150      130     70           6       True     0      1
798       60      170      130     80           6       True     0      1
799      120      130       90     70           6       True     0      1

[800 rows x 15 columns]
```

remove duplicate values

[65]:
```
### let's remove duplicated data so that we only get one pokemon of each type 1
pokemon_df.drop_duplicates(subset=['Type 1'],keep='first', inplace=True)
pokemon_df
```

[65]:
```
     Number                     Name    Type 1   Type 2  Total  HP  Attack  \
0         1                Bulbasaur     Grass   Poison    318  45      49
4         4               Charmander      Fire      NaN    309  39      52
9         7                 Squirtle     Water      NaN    314  44      48
13       10                 Caterpie       Bug      NaN    195  45      30
20       16                   Pidgey    Normal   Flying    251  40      45
28       23                    Ekans    Poison      NaN    288  35      60
30       25                  Pikachu  Electric      NaN    320  35      55
32       27                Sandshrew    Ground      NaN    300  50      75
40       35                Clefairy     Fairy      NaN    323  70      45
61       56                  Mankey  Fighting      NaN    305  40      80
68       63                    Abra   Psychic      NaN    310  25      20
80       74                 Geodude      Rock   Ground    300  40      80
99       92                  Gastly     Ghost   Poison    310  30      35
133     124                    Jynx       Ice  Psychic    455  65      50
159     147                 Dratini    Dragon      NaN    300  41      64
212     197                 Umbreon      Dark      NaN    525  95      65
223     208                 Steelix     Steel   Ground    510  75      85
702     641  TornadusIncarnate Forme    Flying      NaN    580  79     115

     Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary  test  Boss2
0         49       65       65     45           1      False     0      0
```

```
4          43        60        50      65             1       False      0       0
9          65        50        64      43             1       False      0       0
13         35        20        20      45             1       False      0       0
20         40        35        35      56             1       False      0       0
28         44        40        54      55             1       False      0       0
30         40        50        50      90             1       False      0       0
32         85        20        30      40             1       False      0       0
40         48        60        65      35             1       False      0       0
61         35        35        45      70             1       False      0       0
68         15       105        55      90             1       False      0       0
80        100        30        30      20             1       False      0       0
99         30       100        35      80             1       False      0       0
133        35       115        95      95             1       False      0       0
159        45        50        50      50             1       False      0       0
212       110        60       130      65             2       False      0       0
223       200        55        65      30             2       False      0       0
702        70       125        80     111             5        True      0       1
```

rename columns

[66]:
```
### Let's rename the total columns to 'Total_Stats'
pokemon_df.rename(columns={'Total':'Total_Stats'},inplace=True)
pokemon_df.head()
```

[66]:
```
    Number        Name  Type 1  Type 2  Total_Stats  HP  Attack  Defense  \
0        1   Bulbasaur   Grass  Poison          318  45      49       49
4        4  Charmander    Fire     NaN          309  39      52       43
9        7    Squirtle   Water     NaN          314  44      48       65
13      10    Caterpie     Bug     NaN          195  45      30       35
20      16      Pidgey  Normal  Flying          251  40      45       40

    Sp. Atk  Sp. Def  Speed  Generation  Legendary  test  Boss2
0        65       65     45           1      False     0      0
4        60       50     65           1      False     0      0
9        50       64     43           1      False     0      0
13       20       20     45           1      False     0      0
20       35       35     56           1      False     0      0
```

replace NaN values

[67]:
```
#select the colum you want then call .fillna
pokemon_df['Type 2'].fillna(value='no type',inplace=True)
pokemon_df.head()
```

[67]:
```
    Number        Name  Type 1   Type 2  Total_Stats  HP  Attack  Defense  \
0        1   Bulbasaur   Grass   Poison          318  45      49       49
4        4  Charmander    Fire  no type          309  39      52       43
9        7    Squirtle   Water  no type          314  44      48       65
```

```
13      10    Caterpie     Bug  no type          195  45      30       35
20      16      Pidgey  Normal   Flying          251  40      45       40

    Sp. Atk  Sp. Def  Speed  Generation  Legendary  test  Boss2
0        65       65     45           1      False     0      0
4        60       50     65           1      False     0      0
9        50       64     43           1      False     0      0
13       20       20     45           1      False     0      0
20       35       35     56           1      False     0      0
```

7. Export new table

[68]:
```
### let's save our table with only 1 pokemon of type 1

pokemon_unique=pokemon_df.drop_duplicates(subset=['Type 1'],keep='first')
pokemon_unique.to_csv(data_out_directory+'Pokemon_uniqueType1.csv')
pokemon_unique
```

[68]:
```
     Number                      Name    Type 1   Type 2  Total_Stats  HP  \
0         1                 Bulbasaur     Grass   Poison          318  45
4         4                Charmander      Fire  no type          309  39
9         7                  Squirtle     Water  no type          314  44
13       10                  Caterpie       Bug  no type          195  45
20       16                    Pidgey    Normal   Flying          251  40
28       23                     Ekans    Poison  no type          288  35
30       25                   Pikachu  Electric  no type          320  35
32       27                 Sandshrew    Ground  no type          300  50
40       35                 Clefairy     Fairy  no type          323  70
61       56                    Mankey  Fighting  no type          305  40
68       63                     Abra   Psychic  no type          310  25
80       74                  Geodude      Rock   Ground          300  40
99       92                   Gastly     Ghost   Poison          310  30
133     124                     Jynx       Ice  Psychic          455  65
159     147                  Dratini    Dragon  no type          300  41
212     197                  Umbreon      Dark  no type          525  95
223     208                  Steelix     Steel   Ground          510  75
702     641  TornadusIncarnate Forme    Flying  no type          580  79

     Attack  Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary  test  \
0        49       49       65       65     45           1      False     0
4        52       43       60       50     65           1      False     0
9        48       65       50       64     43           1      False     0
13       30       35       20       20     45           1      False     0
20       45       40       35       35     56           1      False     0
28       60       44       40       54     55           1      False     0
30       55       40       50       50     90           1      False     0
```

```
32        75        85        20        30        40             1       False      0
40        45        48        60        65        35             1       False      0
61        80        35        35        45        70             1       False      0
68        20        15       105        55        90             1       False      0
80        80       100        30        30        20             1       False      0
99        35        30       100        35        80             1       False      0
133       50        35       115        95        95             1       False      0
159       64        45        50        50        50             1       False      0
212       65       110        60       130        65             2       False      0
223       85       200        55        65        30             2       False      0
702      115        70       125        80       111             5        True      0
```

```
     Boss2
0        0
4        0
9        0
13       0
20       0
28       0
30       0
32       0
40       0
61       0
68       0
80       0
99       0
133      0
159      0
212      0
223      0
702      1
```

Note: Make a thread about what kinds of plots students want to learn how to make.

Homework

1. Create a new variable that holds a data frame with only the data for pokemon from Generation 1 or 3 that are also Legendary and Psychic.
2. Export and save this table to your working folder.

Answer (2 ways)

[75]:
```
###keep in mind that I use Pandas for everything so I can do stuff in one step␣
 ↪but that
### was not always the case. It is okay to start with multiple steps until you␣
 ↪master the material
### don't feel frustrated if you don't get all this immediately
```

```
## call in my data
pokemon_df=pd.read_csv(datadirectory+'Pokemon.csv')
## Remove the # in front of the line you want to run

#answer 1 with multiple steps

##STEP 1: get the generations you wnat first (1)
pokemon_df_selected_1=pokemon_df.loc[((pokemon_df['Generation']==1)
 ↪|(pokemon_df['Generation']==3)),:].copy()
##STEP 2: get the Legendary pokemon next, from the previous df (1)
pokemon_df_selected_2=pokemon_df_selected_1.
 ↪loc[pokemon_df_selected_1['Legendary']==True,:].copy()
## STEP 3: the the Psychic next, from the previous def (2)
pokemon_df_selected=pokemon_df_selected_2.loc[pokemon_df_selected_2['Type
 ↪1']=='Psychic',:].copy()

###export the table, note I wrote all my answers the same so this line will work
 ↪regardless of the method you use
pokemon_df_selected.to_csv(data_out_directory+'pokemon_hw.csv')

pokemon_df_selected
```

[75]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
162	150	Mewtwo	Psychic	NaN	680	106	110	90	154	90	130	1	True
163	150	MewtwoMega Mewtwo X	Psychic	Fighting	780	106	190	100	154	100	130	1	True
164	150	MewtwoMega Mewtwo Y	Psychic	NaN	780	106	150	70	194	120	140	1	True
428	386	DeoxysNormal Forme	Psychic	NaN	600	50	150	50	150	50	150	3	True
429	386	DeoxysAttack Forme	Psychic	NaN	600	50	180	20	180	20	150	3	True
430	386	DeoxysDefense Forme	Psychic	NaN	600	50	70	160	70	160	90	3	True
431	386	DeoxysSpeed Forme	Psychic	NaN	600	50	95	90	95	90	180	3	True

[76]:
```
#answer 2

pokemon_df=pd.read_csv(datadirectory+'Pokemon.csv')
##one step, but A lot of ()
```

```
pokemon_df_selected_2=pokemon_df.
 ↪loc[(((pokemon_df['Generation']==1)|(pokemon_df['Generation']==3))&((pokemon_df['Legendary']=
 ↪True) & (pokemon_df['Type 1']=='Psychic'))),:].copy()

###export the table, note I wrote all my answers the same so this line will work␣
 ↪regardless of the method you use
pokemon_df_selected_2.to_csv(data_out_directory+'pokemon_hw_2.csv')

pokemon_df_selected_2
```

[76]:

	Number	Name	Type 1	Type 2	Total	HP	Attack	\
162	150	Mewtwo	Psychic	NaN	680	106	110	
163	150	MewtwoMega Mewtwo X	Psychic	Fighting	780	106	190	
164	150	MewtwoMega Mewtwo Y	Psychic	NaN	780	106	150	
428	386	DeoxysNormal Forme	Psychic	NaN	600	50	150	
429	386	DeoxysAttack Forme	Psychic	NaN	600	50	180	
430	386	DeoxysDefense Forme	Psychic	NaN	600	50	70	
431	386	DeoxysSpeed Forme	Psychic	NaN	600	50	95	

	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
162	90	154	90	130	1	True
163	100	154	100	130	1	True
164	70	194	120	140	1	True
428	50	150	50	150	3	True
429	20	180	20	150	3	True
430	160	70	160	90	3	True
431	90	95	90	180	3	True