Pandas_Tutorial

July 27, 2019

1 Pandas for Data Science

This tutorial covers enough content to get started with Python Pandas library. This library contains functionalities that can be extremely useful for data analysis. Let's dive into this topic.

This tutorial is organized as follows: A sequence of tasks is presented, followed by the presentation of the code segment completes that task using the Pandas library. For data analysis, a simple dataset of Pokemon is considered that can be found using the link: https://www.kaggle.com/abcsds/pokemon/downloads/pokemon.zip/2.

1.1 Task 1:

Import the Pandas library, read the downloaded CSV file from the website as mentioned earlier, and then display the first ten rows of the dataset.

[2]:

```
import pandas as pd
poke_data = pd.read_csv('Pokemon.csv')
poke_data.head(10)
```

[2]:

```
   #                       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
5  5                 Charmeleon   Fire     NaN    405  58      64       58
6  6                  Charizard   Fire  Flying    534  78      84       78
7  6  CharizardMega Charizard X   Fire  Dragon    634  78     130      111
8  6  CharizardMega Charizard Y   Fire  Flying    634  78     104       78
9  7                   Squirtle  Water     NaN    314  44      48       65

   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       80       65     80           1      False
6      109       85    100           1      False
```

```
7    130   85  100           1    False
8    159  115  100           1    False
9     50   64   43           1    False
```

1.2 Task 2:

Display the column names separately and then count the number of columns/features/attributes in the dataset. Display the number of rows/samples in the dataset as well as show a high-level description of the dataset.

[81]:

```
import numpy as np
print("The name of the columns are as follows: ")
for val in poke_data.columns.values:
    print(val)
print("The number of columns is: ", len(poke_data.columns.values))
print("The number of samples is: ", np.shape(poke_data.values)[0])
poke_data.describe()
```

```
The name of the columns are as follows:
#
Name
Type 1
Type 2
Total
HP
Attack
Defense
Sp. Atk
Sp. Def
Speed
Generation
Legendary
The number of columns is:  13
The number of samples is:  800
```

[81]:

```
                #       Total          HP      Attack     Defense     Sp. Atk  \
count  800.000000  800.00000  800.000000  800.000000  800.000000  800.000000
mean   362.813750  435.10250   69.258750   79.001250   73.842500   72.820000
std    208.343798  119.96304   25.534669   32.457366   31.183501   32.722294
min      1.000000  180.00000    1.000000    5.000000    5.000000   10.000000
25%    184.750000  330.00000   50.000000   55.000000   50.000000   49.750000
50%    364.500000  450.00000   65.000000   75.000000   70.000000   65.000000
75%    539.250000  515.00000   80.000000  100.000000   90.000000   95.000000
max    721.000000  780.00000  255.000000  190.000000  230.000000  194.000000

          Sp. Def       Speed  Generation
count  800.000000  800.000000   800.00000
mean    71.902500   68.277500     3.32375
```

```
std     27.828916   29.060474     1.66129
min     20.000000    5.000000     1.00000
25%     50.000000   45.000000     2.00000
50%     70.000000   65.000000     3.00000
75%     90.000000   90.000000     5.00000
max    230.000000  180.000000     6.00000
```

1.3 Task 3

Display the last ten observations from the dataset. Only show the name of the Pokemon in the first case. In the second case, show separately the following columns: Name, Attack, and Speed.

[93]:

```
print(poke_data['Name'].tail(10))
poke_data[['Name', 'Attack', 'Speed']].tail(10)
```

```
790                  Noibat
791                 Noivern
792                 Xerneas
793                 Yveltal
794        Zygarde50% Forme
795                 Diancie
796     DiancieMega Diancie
797     HoopaHoopa Confined
798      HoopaHoopa Unbound
799               Volcanion
Name: Name, dtype: object
```

[93]:

```
                    Name  Attack  Speed
790               Noibat      30     55
791              Noivern      70    123
792              Xerneas     131     99
793              Yveltal     131     99
794     Zygarde50% Forme     100     95
795              Diancie     100     50
796  DiancieMega Diancie     160    110
797  HoopaHoopa Confined     110     70
798   HoopaHoopa Unbound     160     80
799            Volcanion     110     70
```

1.4 Task 4:

Display rows from 15 to 20. Show the names of the Pokemons. Show the following columns: Name, Type 1, Type 2. Show rows from hp to legendary.

[101]:

```
print(poke_data.loc[15:20, 'Name'])
print(poke_data.loc[15:20, ['Name', 'Type 1', 'Type 2']])
print(poke_data.loc[15:20, 'HP': 'Legendary'])
```

```
15                 Butterfree
16                     Weedle
17                     Kakuna
18                   Beedrill
19    BeedrillMega Beedrill
20                     Pidgey
Name: Name, dtype: object
                     Name  Type 1  Type 2
15             Butterfree     Bug  Flying
16                 Weedle     Bug  Poison
17                 Kakuna     Bug  Poison
18               Beedrill     Bug  Poison
19  BeedrillMega Beedrill     Bug  Poison
20                 Pidgey  Normal  Flying
    HP  Attack  Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
15  60      45       50       90       80     70           1      False
16  40      35       30       20       20     50           1      False
17  45      25       50       25       25     35           1      False
18  65      90       40       45       80     75           1      False
19  65     150       40       15       80    145           1      False
20  40      45       40       35       35     56           1      False
```

1.5 Task 5:

Complete the previous task using iloc function.

[106]:
```
print(poke_data.iloc[15:20, 1])
print(poke_data.iloc[15:20, [1, 2, 3]])
print(poke_data.iloc[15:20, 5:12])
```

```
15                 Butterfree
16                     Weedle
17                     Kakuna
18                   Beedrill
19    BeedrillMega Beedrill
Name: Name, dtype: object
                     Name Type 1  Type 2
15             Butterfree    Bug  Flying
16                 Weedle    Bug  Poison
17                 Kakuna    Bug  Poison
18               Beedrill    Bug  Poison
19  BeedrillMega Beedrill    Bug  Poison
    HP  Attack  Defense  Sp. Atk  Sp. Def  Speed  Generation
15  60      45       50       90       80     70           1
16  40      35       30       20       20     50           1
17  45      25       50       25       25     35           1
18  65      90       40       45       80     75           1
19  65     150       40       15       80    145           1
```

1.6 Task 6:

Create a new data frame using the first 5 attributes and the last 10 samples. Display the new data frame. Show the data frame seperately after resetting the index.

[110]:

```
new_poke = poke_data.iloc[-10:, 0:5]
print(new_poke)
new_poke = new_poke.reset_index()
print(new_poke)
```

```
       #                  Name   Type 1  Type 2  Total
790  714                Noibat   Flying  Dragon    245
791  715               Noivern   Flying  Dragon    535
792  716               Xerneas    Fairy     NaN    680
793  717               Yveltal     Dark  Flying    680
794  718      Zygarde50% Forme   Dragon  Ground    600
795  719               Diancie     Rock   Fairy    600
796  719   DiancieMega Diancie     Rock   Fairy    700
797  720   HoopaHoopa Confined  Psychic   Ghost    600
798  720    HoopaHoopa Unbound  Psychic    Dark    680
799  721             Volcanion     Fire   Water    600
   index    #                  Name   Type 1  Type 2  Total
0    790  714                Noibat   Flying  Dragon    245
1    791  715               Noivern   Flying  Dragon    535
2    792  716               Xerneas    Fairy     NaN    680
3    793  717               Yveltal     Dark  Flying    680
4    794  718      Zygarde50% Forme   Dragon  Ground    600
5    795  719               Diancie     Rock   Fairy    600
6    796  719   DiancieMega Diancie     Rock   Fairy    700
7    797  720   HoopaHoopa Confined  Psychic   Ghost    600
8    798  720    HoopaHoopa Unbound  Psychic    Dark    680
9    799  721             Volcanion     Fire   Water    600
```

1.7 Task 7:

Iterate through the samples of the dataset and display the hp and attack power of the first two dark dragon Pokemon.

[119]:

```
# Method 1
count = 1
for index, row in poke_data.iterrows():
    if row['Type 1'] == "Dark" and row['Type 2'] == "Dragon" and count <= 2:
        print(index, row['Name'], row['HP'], row['Attack'], row['Type 1'],␣
 ↪row['Type 2'])
        count += 1

#Method 2
poke_data.loc[(poke_data['Type 1'] == "Dark") & (poke_data['Type 2'] ==␣
 ↪"Dragon")].head(2)
```

```
694 Deino 52 65 Dark Dragon
695 Zweilous 72 85 Dark Dragon
```

```
[119]:        #      Name Type 1  Type 2  Total  HP  Attack  Defense  Sp. Atk  \
       694  633     Deino   Dark  Dragon    300  52      65       50       45
       695  634  Zweilous   Dark  Dragon    420  72      85       70       65

            Sp. Def  Speed  Generation  Legendary
       694       50     38           5      False
       695       70     58           5      False
```

1.8 Task 8:

Sort the Pokemons based on their Type 1 category in descending order. When there are conflicts, sort the values based on their Type 2 attribute in ascending order. Show only the first 20 rows.

```
[123]: poke_data.sort_values(['Type 1', 'Type 2'], ascending=[False, True]).head(20)
```

```
[123]:        #                    Name Type 1    Type 2  Total   HP  Attack  Defense  \
       141  130   GyaradosMega Gyarados  Water      Dark    640   95     155      109
       347  318                Carvanha  Water      Dark    305   45      90       20
       348  319                Sharpedo  Water      Dark    460   70     120       40
       349  319   SharpedoMega Sharpedo  Water      Dark    560   70     140       70
       374  342               Crawdaunt  Water      Dark    468   63     120       85
       726  658                Greninja  Water      Dark    530   72      95       67
       249  230                 Kingdra  Water    Dragon    540   75      95       95
       541  484                  Palkia  Water    Dragon    680   90     120      100
       184  170                Chinchou  Water  Electric    330   75      38       38
       185  171                 Lanturn  Water  Electric    460  125      58       58
       198  183                  Marill  Water     Fairy    250   70      20       50
       199  184               Azumarill  Water     Fairy    420  100      50       80
       67    62               Poliwrath  Water  Fighting    510   90      95       95
       713  647    KeldeoOrdinary Forme  Water  Fighting    580   91      72       90
       714  647    KeldeoResolute Forme  Water  Fighting    580   91      72       90
       140  130                Gyarados  Water    Flying    540   95     125       79
       244  226                 Mantine  Water    Flying    465   65      40       70
       301  278                 Wingull  Water    Flying    270   40      30       30
       302  279                Pelipper  Water    Flying    430   60      50      100
       508  458                 Mantyke  Water    Flying    345   45      20       50

            Sp. Atk  Sp. Def  Speed  Generation  Legendary
       141       70      130     81           1      False
       347       65       20     65           3      False
       348       95       40     95           3      False
       349      110       65    105           3      False
       374       90       55     55           3      False
```

```
726       103       71    122           6      False
249        95       95     85           2      False
541       150      120    100           4       True
184        56       56     67           2      False
185        76       76     67           2      False
198        20       50     40           2      False
199        60       80     50           2      False
67         70       90     70           1      False
713       129       90    108           5      False
714       129       90    108           5      False
140        60      100     81           1      False
244        80      140     70           2      False
301        55       30     85           3      False
302        85       70     65           3      False
508        60      120     50           4      False
```

1.9 Task 9:

What is the HP of the Pokemon named 'Butterfree'? Is the 240th Pokemon legendary?

[134]:
```
print("The HP of the Pokemon named 'Butterfree' is: ", poke_data.
 ↪loc[poke_data['Name'] == "Butterfree"]['HP'].values[0])
if poke_data.iloc[240, -1]:
    print("The 240th Pokemon is", "legendary")
else:
    print("The 240th Pokemon is", "not Legendary")
```

```
The HP of the Pokemon named 'Butterfree' is:  60
The 240th Pokemon is not Legendary
```

1.10 Task 10:

Create a new column called 'Power'. The column assigns the average attack and defense of the corresponding pokemon. Drop the newly created column and display the dataset again. Show only the first five rows.

[136]:
```
poke_data['Power'] = (poke_data['Attack'] + poke_data['Defense']) / 2
print(poke_data.head(5))
poke_data = poke_data.drop(columns=['Power'])
print(poke_data.head(5))
```

```
   #                   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  Power
0       65       65     45           1      False   49.0
1       80       80     60           1      False   62.5
2      100      100     80           1      False   82.5
3      122      120     80           1      False  111.5
4       60       50     65           1      False   47.5
   #                   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
```

1.11 Task 11:

Create a new column called ‘Average Power’ that computes the average score of all the attributes from hp to speed. Show the names and average powers of the first 5 rows. Save the modified data into two files using two seperators: comma and tab. Avoid writing the index into the newly created files. In the end, drop the column for further processing.

[148]:

```python
print('Method 1')
poke_data['Average Power'] = poke_data.iloc[:, [5, 6, 7, 8, 9, 10]].sum(axis=1) / 6
print(poke_data[['Name', 'Average Power']].head(5))
print('Method 2')
poke_data['Average Power'] = poke_data.iloc[:, 5:11].sum(axis=1) / 6
print(poke_data[['Name', 'Average Power']].head(5))
print('Method 3')
poke_data['Average Power'] = poke_data.loc[:, 'HP':'Speed'].sum(axis=1) / 6
print(poke_data[['Name', 'Average Power']].head(5))

poke_data.to_csv('modified_comma', index=False)
poke_data.to_csv('modified_tab', index=False, sep='\t')
poke_data = poke_data.drop(columns=['Average Power'])
print(poke_data.head(5))
```

```
Method 1
                    Name  Average Power
0              Bulbasaur      53.000000
1                Ivysaur      67.500000
2               Venusaur      87.500000
```

```
3  VenusaurMega Venusaur     104.166667
4             Charmander      51.500000
Method 2
                    Name  Average Power
0              Bulbasaur      53.000000
1                Ivysaur      67.500000
2               Venusaur      87.500000
3  VenusaurMega Venusaur     104.166667
4             Charmander      51.500000
Method 3
                    Name  Average Power
0              Bulbasaur      53.000000
1                Ivysaur      67.500000
2               Venusaur      87.500000
3  VenusaurMega Venusaur     104.166667
4             Charmander      51.500000
   #                   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
```

1.12 Task 12:

Display the list of the Pokemon with the following names: name containing the term ‘cute’, name containing the word ‘cute’ or ‘Mega’, name that starts with the letter’S’ and ends with the letter ‘a’, and names the substring ‘ri’ in the middle (can’t be start or end with ‘ri’).

[181]:
```python
# Displaying only the first 5 rows and the first 5 columns
print(poke_data[poke_data['Name'].str.contains('cute')])
print(poke_data[poke_data['Name'].str.contains('cute|Mega')].head(5).iloc[:, 0:
 ↪5])
print(poke_data[poke_data['Name'].str.contains('^S.*a+$', regex=True)].head(5).
 ↪iloc[:, 0:5])
print(poke_data[poke_data['Name'].str.contains('.+ri[a-z].+', regex=True)].
 ↪head(5).iloc[:, 0:5])
```

```
       #       Name Type 1   Type 2  Total  HP  Attack  Defense  Sp. Atk  \
110  102  Exeggcute  Grass  Psychic    325  60      40       80       60
```

```
     Sp. Def  Speed  Generation  Legendary
110       45     40           1      False
     #                       Name Type 1  Type 2  Total
3    3      VenusaurMega Venusaur  Grass  Poison    625
7    6  CharizardMega Charizard X   Fire  Dragon    634
8    6  CharizardMega Charizard Y   Fire  Flying    634
12   9    BlastoiseMega Blastoise  Water     NaN    630
19  15      BeedrillMega Beedrill    Bug  Poison    495
       #      Name  Type 1 Type 2  Total
126  117    Seadra   Water    NaN    440
207  192  Sunflora   Grass    NaN    425
236  218    Slugma    Fire    NaN    250
316  292  Shedinja     Bug  Ghost    236
358  327    Spinda  Normal    NaN    360
     #                       Name Type 1  Type 2  Total
6    6                  Charizard   Fire  Flying    534
7    6  CharizardMega Charizard X   Fire  Dragon    634
8    6  CharizardMega Charizard Y   Fire  Flying    634
18  15                   Beedrill    Bug  Poison    395
19  15      BeedrillMega Beedrill    Bug  Poison    495
```

1.13 Task 13:

Calculate the highest and lowest used term in 'Type 2'. Then, replace all the 'NaN' value terms found in the 'Type 2' column. Display both before and after modifications of these values.

[242]:

```
# highest/maximum type 2 value
max_val = poke_data.loc[~poke_data['Type 2'].isnull()].groupby('Type 2').
 ↪count().sort_values('Name').index.values[-1]
# lowest/minimum type 2 value
min_val = poke_data.loc[~poke_data['Type 2'].isnull()].groupby('Type 2').
 ↪count().sort_values('Name').index.values[0]

modified = poke_data.copy(deep=True)
print(modified.head(10))
modified.loc[modified['Type 2'].isnull(), 'Type 2'] = max_val

print(modified.head(10))
```

```
   #                       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
5  5                 Charmeleon   Fire     NaN    405  58      64       58
6  6                  Charizard   Fire  Flying    534  78      84       78
7  6  CharizardMega Charizard X   Fire  Dragon    634  78     130      111
```

```
8  6  CharizardMega Charizard Y   Fire  Flying    634  78     104       78
9  7                   Squirtle  Water     NaN    314  44      48       65

   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       80       65     80           1      False
6      109       85    100           1      False
7      130       85    100           1      False
8      159      115    100           1      False
9       50       64     43           1      False
   #                       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  Flying    309  39      52       43
5  5                 Charmeleon   Fire  Flying    405  58      64       58
6  6                  Charizard   Fire  Flying    534  78      84       78
7  6  CharizardMega Charizard X   Fire  Dragon    634  78     130      111
8  6  CharizardMega Charizard Y   Fire  Flying    634  78     104       78
9  7                   Squirtle  Water  Flying    314  44      48       65

   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       80       65     80           1      False
6      109       85    100           1      False
7      130       85    100           1      False
8      159      115    100           1      False
9       50       64     43           1      False
```

1.14 Task 14:

Find all Pokemon with Type 1 equals to 'Grass'. Increase their attack by 5% and decrease their defense by 2%. Show before and after modifications.

[277]:
```python
# create a completely new copy
# N.B. without the copy function, the assignment operation would create a mere reference to the original data
# changing the modified data would be reflected on the original data
modi = poke_data.copy(deep=True)
```

```
print(modi.loc[modi['Type 1'] == "Grass"].head(5))
prev_attack = modi.loc[modi['Type 1'] == "Grass"]['Attack'].values
prev_def = modi.loc[modi['Type 1'] == "Grass"]['Defense'].values
modi.loc[modi['Type 1'] == "Grass", ['Attack', 'Defense']] = pd.
 ↪DataFrame({'Attack' : [x + x * 0.05 for x in prev_attack],
                                                                         ␣
 ↪'Defense' : [x - x * 0.02 for x in prev_def]}, index=modi.loc[modi['Type 1']␣
 ↪== "Grass"].index)
print(modi.loc[modi['Type 1'] == "Grass"].head(5))
```

```
     #                  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
48  43                Oddish  Grass  Poison    320  45      50       55

    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
48       75       65     30           1      False
     #                  Name Type 1  Type 2  Total  HP  Attack  Defense  \
0    1             Bulbasaur  Grass  Poison    318  45   51.45    48.02
1    2               Ivysaur  Grass  Poison    405  60   65.10    61.74
2    3              Venusaur  Grass  Poison    525  80   86.10    81.34
3    3 VenusaurMega Venusaur  Grass  Poison    625  80  105.00   120.54
48  43                Oddish  Grass  Poison    320  45   52.50    53.90

    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
48       75       65     30           1      False
```

1.15 Task 15:

Split the dataset into 5 chunks and display it.

[10]:

```
r, c = poke_data.shape
splits = []
for df in pd.read_csv('Pokemon.csv', chunksize=r//5):
    splits.append(df)

for i in range(len(splits)):
    print('Chunk Number:', i + 1)
    print(splits[i].iloc[:, 0:5].head(5))
```

```
Chunk Number: 1
   #                   Name Type 1  Type 2  Total
0  1              Bulbasaur  Grass  Poison    318
1  2                Ivysaur  Grass  Poison    405
2  3               Venusaur  Grass  Poison    525
3  3  VenusaurMega Venusaur  Grass  Poison    625
4  4             Charmander   Fire     NaN    309
Chunk Number: 2
       #                 Name   Type 1    Type 2  Total
160  148            Dragonair   Dragon       NaN    420
161  149            Dragonite   Dragon    Flying    600
162  150               Mewtwo  Psychic       NaN    680
163  150  MewtwoMega Mewtwo X  Psychic  Fighting    780
164  150  MewtwoMega Mewtwo Y  Psychic       NaN    780
Chunk Number: 3
       #      Name    Type 1 Type 2  Total
320  296  Makuhita  Fighting    NaN    237
321  297  Hariyama  Fighting    NaN    474
322  298   Azurill    Normal  Fairy    190
323  299  Nosepass      Rock    NaN    375
324  300    Skitty    Normal    NaN    260
Chunk Number: 4
       #       Name   Type 1   Type 2  Total
480  432    Purugly   Normal      NaN    452
481  433  Chingling  Psychic      NaN    285
482  434     Stunky   Poison     Dark    329
483  435   Skuntank   Poison     Dark    479
484  436    Bronzor    Steel  Psychic    300
Chunk Number: 5
       #       Name   Type 1  Type 2  Total
640  579  Reuniclus  Psychic     NaN    490
641  580   Ducklett    Water  Flying    305
642  581     Swanna    Water  Flying    473
643  582  Vanillite      Ice     NaN    305
644  583  Vanillish      Ice     NaN    395
```

[ ]: