1 Using Pandas to merge or lookup data

Sometimes we may want to cross-reference data between different data tables. This may be in order to perform a full merge of data, or just to produce a summary lookup table referencing across different tables.

```
import numpy as np
import pandas as pd

# Set up the first data frame

df1 = pd.DataFrame()

names = ['Gandolf','Gimli','Frodo','Legolas','Bilbo']
types = ['Wizard','Dwarf','Hobbit','Elf','Hobbit']
magic = [10, 1, 4, 6, 4]
aggression = [7, 10, 2, 5, 1]
stealth = [8, 2, 5, 10, 5]

df1['names'] = names
df1['type'] = types
df1['magic_power'] = magic
df1['aggression'] = aggression
df1['stealth'] = stealth

# Set up the second dataframe

names = ['Gandolf','Gimli','Frodo','Aragorn','Sauron']
popularity = ['High','Medium','High','Medium','Low']

df2 = pd.DataFrame()

df2['name'] = names
df2['popularity'] = popularity
```

We will look here at where the reference fields are not the index fields. We are going to want to merge using 'names' in df1 and 'name' in df2, and we are going to keep all records from df1, and add data from df2 where it is available.

```
merged_df = pd.merge(df1, df2,
                     left_on = 'names',
                     right_on = 'name',
                     how='left')

print (merged_df)

OUT:

     names    type  magic_power  aggression  stealth     name popularity
0  Gandolf  Wizard           10           7        8  Gandolf       High
1    Gimli   Dwarf            1          10        2    Gimli     Medium
2    Frodo  Hobbit            4           2        5    Frodo       High
3  Legolas     Elf            6           5       10      NaN        NaN
4    Bilbo  Hobbit            4           1        5      NaN        NaN
```

We can keep all data from both databases by using *how=outer* (an outer database join).

```
merged_df = pd.merge(df1, df2,
                     left_on = 'names',
                     right_on = 'name',
```

```
                    how='outer')

print (merged_df)
```

OUT:

```
     names    type  magic_power  aggression  stealth     name popularity
0  Gandolf  Wizard         10.0         7.0      8.0  Gandolf       High
1    Gimli   Dwarf          1.0        10.0      2.0    Gimli     Medium
2    Frodo  Hobbit          4.0         2.0      5.0    Frodo       High
3  Legolas     Elf          6.0         5.0     10.0      NaN        NaN
4    Bilbo  Hobbit          4.0         1.0      5.0      NaN        NaN
5      NaN     NaN          NaN         NaN      NaN  Aragorn     Medium
6      NaN     NaN          NaN         NaN      NaN   Sauron        Low
```

Or we could use *how=inner* to produce a dataframe for only the rows with reference columns in both dataframes, or *how=right* to keep all rows in df2 and add data from df1 where it exists.

In the above examples we have returned all fields both both dataframes. We can choose to select just the fields we wish to return (though we need the reference field from both dataframes):

```
merged_df = pd.merge(df1[['names','type']],
                     df2[['name','popularity']],
                     left_on = 'names',
                     right_on = 'name',
                     how='left')

print (merged_df)
```

OUT:

```
     names    type     name popularity
0  Gandolf  Wizard  Gandolf       High
1    Gimli   Dwarf    Gimli     Medium
2    Frodo  Hobbit    Frodo       High
3  Legolas     Elf      NaN        NaN
4    Bilbo  Hobbit      NaN        NaN
```

Using this method we need our reference fields, used to join data, as part of the dataframe and not as the index. To use this method where there is an index column that you wish to use as a joining field you will need to reset the index to a new numbered column. That is done by the command *df.reset_index()*. Alternatively you can use a method that joins using the index fields of each dataframe, as described in the link below.

There are many ways of joining and merging Pandas dataframes. I have set out just one method here. See `https://pandas.pydata.org/pandas-docs/stable/merging.html` for all methods.