Learn

Now that we've combined our dataframes, we're almost at a place where we can perform some kind of analysis! First, though, we'll have to clean up the `institute_service` column. This column is tricky to clean because it currently contains values in a couple different forms:

```
NaN                   164
Less than 1 year       77
3-4                    72
1-2                    68
11-20                  49
More than 20 years     48
5-6                    36
7-10                   30
5.0                    30
3.0                    28
```

To analyze the data, we'll convert these numbers into categories. We'll base our anlaysis on this article, which makes the argument that understanding employee's needs according to career stage instead of age is more effective.

We'll use the slightly modified definitions below:

- New: Less than 3 years at a company
- Experienced: 3-6 years at a company
- Established: 7-10 years at a company
- Veteran: 11 or more years at a company

Let's categorize the values in the `institute_service` column using the definitions above.

Instructions

- First, we'll extract the years of service from each value in the `institute_service` column.
  - Use the `Series.astype()` method to change the type to `'str'`.
  - Use vectorized string methods to extract the years of service from each pattern. You can find the full list of vectorized string methods here.
  - Double check that you didn't miss extracting any digits.
  - Use the `Series.astype()` method to change the type to `'float'`.
- Next, we'll map each value to one of the career stage definitions above.
  - Create a function that maps each year value to one of the career stages above.
    - Remember that you'll have to handle missing values separately. You can use the following code to check if a value is `NaN` where `val` is the name of the value: `pd.isnull(val)`.
  - Use the `Series.apply()` method to apply the function to the `institute_service` column. Assign the result to a new column named `service_cat`.
- Write a markdown paragraph explaining the changes you made and why.

Get Help

Basics.ipynb Refresh Download

jupyter Basics

File Edit View Insert Cell Kernel Help Python 3

Markdown CellToolbar

```
3.0     20
4.0     16
5.0     23
6.0     17
7.0     13
8.0      8
9.0     14
10.0     6
11.0     4
12.0     6
13.0     8
14.0     6
15.0     7
16.0     5
17.0     6
18.0     5
19.0     3
20.0     7
21.0     3
22.0     6
23.0     4
24.0     4
25.0     2
26.0     2
27.0     1
28.0     2
29.0     1
30.0     2
31.0     1
32.0     3
33.0     1
34.0     1
35.0     1
36.0     2
38.0     1
39.0     3
41.0     1
42.0     1
```