Apply,mapply

November 20, 2016

```
In [1]: import pandas as pd

In [2]: train = pd.read_csv('http://bit.ly/kaggletrain')

In [3]: train.head(2)
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C

```
In [5]: train ['Sex_num'] = train.Sex.map({'female': 0, 'male': 1})

In [6]: train.loc[0:4, ['Sex', 'Sex_num']] # show me columns Sex and Sex_num, where
```

Out[6]:

	Sex	Sex_num
0	male	1
1	female	0
2	female	0
3	female	0
4	male	1

```
In [7]: train['Name_length'] = train.Name.apply(len) # applies length to Name

In [8]: train.loc[0:4, ['Name', 'Name_length']]
```

Out[8]:

	Name	Name_length
0	Braund, Mr. Owen Harris	23
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	51
2	Heikkinen, Miss. Laina	22
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	44
4	Allen, Mr. William Henry	24

```
In [9]: import numpy as np

In [10]: train['Fare_ceil'] = train.Fare.apply(np.ceil)

In [12]: train.loc[0:4, ['Fare', 'Fare_ceil']] # rounds up

Out[12]:       Fare  Fare_ceil
         0   7.2500        8.0
         1  71.2833       72.0
         2   7.9250        8.0
         3  53.1000       54.0
         4   8.0500        9.0

In [13]: train.Name.str.split(',').head() # list of strings

Out[13]: 0                            [Braund,  Mr. Owen Harris]
         1    [Cumings,  Mrs. John Bradley (Florence Briggs ...
         2                             [Heikkinen,  Miss. Laina]
         3      [Futrelle,  Mrs. Jacques Heath (Lily May Peel)]
         4                           [Allen,  Mr. William Henry]
         Name: Name, dtype: object

In [14]: def get_element(my_list, position):
             return my_list[position]

In [15]: train.Name.str.split(',').apply(get_element, position=0).head() # => just
         # => index 0 of list of strings => last names

Out[15]: 0       Braund
         1      Cumings
         2    Heikkinen
         3     Futrelle
         4        Allen
         Name: Name, dtype: object

In [16]: train.Name.str.split(',').apply(lambda x: x[0]).head() # rewritten as lamb
         # lambda functions used a lot with apply funcs

Out[16]: 0       Braund
         1      Cumings
         2    Heikkinen
         3     Futrelle
         4        Allen
         Name: Name, dtype: object

In [17]: drinks = pd.read_csv('http://bit.ly/drinksbycountry')

In [18]: drinks.head(2)
```

```
Out[18]:        country  beer_servings  spirit_servings  wine_servings  \
         0  Afghanistan              0                0              0
         1      Albania             89              132             54

            total_litres_of_pure_alcohol continent
         0                           0.0      Asia
         1                           4.9    Europe
```

```
In [20]: drinks.loc[:, 'beer_servings': 'wine_servings'].apply(max, axis=0)
         # figured out max value in each columns by operating over axis = 0
```

```
Out[20]: beer_servings      376
         spirit_servings    438
         wine_servings      370
         dtype: int64
```

```
In [21]: drinks.loc[:, 'beer_servings': 'wine_servings'].apply(max, axis=1)
```

```
Out[21]: 0        0
         1      132
         2       25
         3      312
         4      217
         5      128
         6      221
         7      179
         8      261
         9      279
         10      46
         11     176
         12      63
         13       0
         14     173
         15     373
         16     295
         17     263
         18      34
         19      23
         20     167
         21     173
         22     173
         23     245
         24      31
         25     252
         26      25
         27      88
         28      37
         29     144
               ...
```

```
163    178
164     90
165    186
166    280
167     35
168     15
169    258
170    106
171      4
172     36
173     36
174    197
175     51
176     51
177     71
178     41
179     45
180    237
181    135
182    219
183     36
184    249
185    220
186    101
187     21
188    333
189    111
190      6
191     32
192     64
dtype: int64
```

```
In [22]: drinks.loc[:,'beer_servings':'wine_servings'].apply(np.argmax, axis=1)
         # which servings is largest for each row => np.argmax
```

```
Out[22]: 0        beer_servings
         1      spirit_servings
         2        beer_servings
         3        wine_servings
         4        beer_servings
         5      spirit_servings
         6        wine_servings
         7      spirit_servings
         8        beer_servings
         9        beer_servings
         10     spirit_servings
         11     spirit_servings
         12     spirit_servings
```

```
13       beer_servings
14    spirit_servings
15    spirit_servings
16       beer_servings
17       beer_servings
18       beer_servings
19       beer_servings
20       beer_servings
21    spirit_servings
22       beer_servings
23       beer_servings
24       beer_servings
25    spirit_servings
26       beer_servings
27       beer_servings
28       beer_servings
29       beer_servings
           ...
163   spirit_servings
164      beer_servings
165      wine_servings
166      wine_servings
167   spirit_servings
168   spirit_servings
169   spirit_servings
170      beer_servings
171      wine_servings
172      beer_servings
173      beer_servings
174      beer_servings
175      beer_servings
176      beer_servings
177   spirit_servings
178   spirit_servings
179      beer_servings
180   spirit_servings
181   spirit_servings
182      beer_servings
183      beer_servings
184      beer_servings
185      wine_servings
186   spirit_servings
187      beer_servings
188      beer_servings
189      beer_servings
190      beer_servings
191      beer_servings
192      beer_servings
```

```
dtype: object
```

In [ ]:
```
# maps over every element of DataFrame
#drinks.loc[:, 'beer_servings':'wine_servings'] = drinks.loc[:,'beer_servi
```