Remove duplicates and replace values

September 25, 2019

0.1 Environment

```
In [4]: import numpy as np
        import pandas as pd
        PREVIOUS_MAX_ROWS = pd.options.display.max_rows
        pd.options.display.max_rows = 20
        np.random.seed(12345)
        import matplotlib.pyplot as plt
        plt.rc('figure', figsize=(10, 6))
        np.set_printoptions(precision=4, suppress=True)
```

0.2 Data cleaning

Remove duplicates

```
In [5]: data=pd.DataFrame({'k1':['one','two']*3+['two'],
                          'k2':[1,1,2,3,3,4,4]})
        data
```

```
Out[5]:     k1  k2
        0  one   1
        1  two   1
        2  one   2
        3  two   3
        4  one   3
        5  two   4
        6  two   4
```

```
In [7]: data.duplicated() #both k1 and k2 match the previous cells
```

```
Out[7]: 0    False
        1    False
        2    False
        3    False
        4    False
        5    False
        6     True
        dtype: bool
```

```
In [8]: data.drop_duplicates()

Out[8]:     k1  k2
        0  one   1
        1  two   1
        2  one   2
        3  two   3
        4  one   3
        5  two   4
```

```
In [12]: data['v1']=range(7) #create another column from 0 to 6
         data

Out[12]:     k1  k2  v1
         0  one   1   0
         1  two   1   1
         2  one   2   2
         3  two   3   3
         4  one   3   4
         5  two   4   5
         6  two   4   6
```

```
In [13]: data.drop_duplicates(['k1']) #remove duplicates of k1 keep the first

Out[13]:     k1  k2  v1
         0  one   1   0
         1  two   1   1
```

```
In [14]: data.drop_duplicates(['k1','k2'],keep='last') #remove duplicates of k1,k2 jointly but

Out[14]:     k1  k2  v1
         0  one   1   0
         1  two   1   1
         2  one   2   2
         3  two   3   3
         4  one   3   4
         6  two   4   6
```

Transform data using a function

```
In [16]: data = pd.DataFrame({'food': ['bacon', 'pulled pork', 'bacon',
                                       'Pastrami', 'corned beef', 'Bacon',
                                       'pastrami', 'honey ham', 'nova lox'],
                              'ounces': [4, 3, 12, 6, 7.5, 8, 3, 5, 6]}) #generate data
         data

Out[16]:           food  ounces
         0        bacon     4.0
         1  pulled pork     3.0
```

```
2        bacon    12.0
3     Pastrami     6.0
4  corned beef     7.5
5        Bacon     8.0
6     pastrami     3.0
7    honey ham     5.0
8     nova lox     6.0
```

```
In [18]: meat_to_animal = {
           'bacon': 'pig',
           'pulled pork': 'pig',
           'pastrami': 'cow',
           'corned beef': 'cow',
           'honey ham': 'pig',
           'nova lox': 'salmon'
         }
         meat_to_animal #labels I want to assign to the food
```

```
Out[18]: {'bacon': 'pig',
          'pulled pork': 'pig',
          'pastrami': 'cow',
          'corned beef': 'cow',
          'honey ham': 'pig',
          'nova lox': 'salmon'}
```

```
In [19]: lowercased = data['food'].str.lower() #change food to lower case
         lowercased
```

```
Out[19]: 0          bacon
         1    pulled pork
         2          bacon
         3       pastrami
         4    corned beef
         5          bacon
         6       pastrami
         7      honey ham
         8       nova lox
         Name: food, dtype: object
```

```
In [20]: data['animal'] = lowercased.map(meat_to_animal) #label assigned
         data
```

```
Out[20]:           food  ounces  animal
         0        bacon     4.0     pig
         1  pulled pork     3.0     pig
         2        bacon    12.0     pig
         3     Pastrami     6.0     cow
         4  corned beef     7.5     cow
         5        Bacon     8.0     pig
```

```
6     pastrami     3.0     cow
7    honey ham     5.0     pig
8     nova lox     6.0  salmon
```

```
In [21]: data['food'].map(lambda x: meat_to_animal[x.lower()]) #all at once
```

```
Out[21]: 0       pig
         1       pig
         2       pig
         3       cow
         4       cow
         5       pig
         6       cow
         7       pig
         8    salmon
         Name: food, dtype: object
```

Replace values

```
In [22]: data = pd.Series([1., -999., 2., -999., -1000., 3.])
         data
```

```
Out[22]: 0       1.0
         1    -999.0
         2       2.0
         3    -999.0
         4   -1000.0
         5       3.0
         dtype: float64
```

```
In [23]: data.replace(-999, np.nan) #replace -999 with missing
```

```
Out[23]: 0       1.0
         1       NaN
         2       2.0
         3       NaN
         4   -1000.0
         5       3.0
         dtype: float64
```

```
In [25]: data.replace([-999, -1000], np.nan)
```

```
Out[25]: 0    1.0
         1    NaN
         2    2.0
         3    NaN
         4    NaN
         5    3.0
         dtype: float64
```

```
In [26]: data.replace([-999, -1000], [np.nan, 0]) #replace -1000 with zero

Out[26]: 0    1.0
         1    NaN
         2    2.0
         3    NaN
         4    0.0
         5    3.0
         dtype: float64

In [28]: data.replace({-999: np.nan, -1000: 0}) #a short cut

Out[28]: 0    1.0
         1    NaN
         2    2.0
         3    NaN
         4    0.0
         5    3.0
         dtype: float64

In [ ]:
```