0111_pop_pandas_column

December 22, 2018

1 Using 'pop' to remove a Pandas DataFrame column and transfer to a new variable

Sometimes we may want to remove a column from a DataFrame, but at the same time transfer that column to a new variable to perform some work on it. An example is re-coding a column as shown below where we will convert a text male/female column into a number 0/1 male column.

1.1 Create a dataframe

```
In [1]: # Create pandas data frame

        import pandas as pd

        name = ['Sam', 'Bill', 'Bob', 'Ian', 'Jo', 'Anne', 'Carl', 'Toni']
        age = [22, 34, 18, 34, 76, 54, 21, 8]
        gender = ['f', 'm', 'm', 'm', 'f', 'f', 'm', 'f']
        height = [1.64, 1.85, 1.70, 1.75, 1.63, 1.79, 1.70, 1.68]

        people = pd.DataFrame()
        people['name'] = name
        people['age'] = age
        people['gender'] = gender
        people['height'] = height

        print(people)
```

```
   name  age gender  height
0   Sam   22      f    1.64
1  Bill   34      m    1.85
2   Bob   18      m    1.70
3   Ian   34      m    1.75
4    Jo   76      f    1.63
5  Anne   54      f    1.79
6  Carl   21      m    1.70
7  Toni    8      f    1.68
```

2 Pop a column (to code differently)

Rather than text male/female we'll pull out that column ad convert to 0 or 1 for male.

```
In [2]: # Pop column
        people_gender = people.pop('gender') # extracts and removes gender

        # Recode (using == gves True/False, but in Python that also has numerical values of 1/0

        male = (people_gender == 'm') * 1 # 'm' is true is converted to number

        # Put new column into DataFrame and print
        people['male'] = male
        print (people)
```

```
   name  age  height  male
0   Sam   22    1.64     0
1  Bill   34    1.85     1
2   Bob   18    1.70     1
3   Ian   34    1.75     1
4    Jo   76    1.63     0
5  Anne   54    1.79     0
6  Carl   21    1.70     1
7  Toni    8    1.68     0
```