HouseLoki_1805317_Pandas_Numpy_GroupBy_Assignment

April 13, 2021

[1]:

```
# importing necessary libraries.
import pandas as pd
import numpy as np
```

[2]:

```
"""
Q1. Find those groups which have more "True" values than "False" values in the␣
 ↪below dataframe.
"""
df = pd.DataFrame({'A': ['group1', 'group1', 'group2', 'group1','group2',␣
 ↪'group1', 'group2', 'group2','group2','group1'],
                   'B': ['true', 'true', 'true', 'false', 'false', 'false',␣
 ↪'false', 'true','false','true']})
for name, group in df.groupby(['A']):
    print(name)
    print(group)
df=df.groupby(['A'])
df_group=df['B'].value_counts()
print(df_group)
t1=df_group.loc["group1","true"]
t2=df_group.loc["group2","true"]
f1=df_group.loc["group1","false"]
f2=df_group.loc["group2","false"]
if(t1>f1):
    print("\nGroup1 has more True")
else:
    print("\nGroup1 has more False")
if(t2>f2):
    print("\nGroup2 has more True")
else:
    print("\nGroup2 has more False")
```

```
group1
        A      B
0  group1   true
1  group1   true
3  group1  false
5  group1  false
9  group1   true
```

```
group2
        A      B
2  group2   true
4  group2  false
6  group2  false
7  group2   true
8  group2  false
A       B
group1  true     3
        false    2
group2  false    3
        true     2
Name: B, dtype: int64

Group1 has more True

Group2 has more False
```

[3]:

```
"""
Q2
a.Get the items not common to both series A and series B (Without using loops)
b.Get the items common to both series A and series B(Without using loops)
"""
s1 = pd.Series([1, 2, 3, 4, 5])
s2 = pd.Series([4, 5, 6, 7, 8])
value_in_a_not_in_b= s1[~s1.isin(s2)]
value_in_b_not_in_a= s2[~s2.isin(s1)]
print("Items not common to both series A and series B\n",value_in_a_not_in_b.
 ↪append(value_in_b_not_in_a))
print("Ttems common to both series A and series B\n",s1[s1.isin(s2)])
```

```
Items not common to both series A and series B
 0    1
1    2
2    3
2    6
3    7
4    8
dtype: int64
Ttems common to both series A and series B
 3    4
4    5
dtype: int64
```

[5]:

```
"""
Q3. Generate a random series of length 10 and find the positions of numbers␣
 ↪that are multiples of 3 from a series?
"""
```

```
numberSeries = pd.Series(np.random.randint(1, 10, 10))
print("Series:")
print(numberSeries)
result = np.argwhere(numberSeries % 3==0)
print("Positions of numbers that are multiples of 3:")
result
```

```
Series:
0    6
1    6
2    5
3    3
4    5
5    2
6    4
7    5
8    9
9    7
dtype: int32
Positions of numbers that are multiples of 3:
```

```
[5]: array([[0],
            [1],
            [3],
            [8]], dtype=int64)
```

```
[6]: """
     3 b) Compute the cumulative difference between the consecutive number for the␣
      ↪same series(without using loops).
     # input Series ==>[1, 3, 6, 10, 15, 21, 27, 35]
     # Desired Output # [nan, 2.0, 3.0, 4.0, 5.0, 6.0, 6.0, 8.0]
     """
     a=pd.Series([1, 3, 6, 10, 15, 21, 27, 35])
     print("Input Series:")
     print(a)
     print("\nDesired Output:")
     print(a.diff().tolist())
```

```
Input Series:
0     1
1     3
2     6
3    10
4    15
5    21
6    27
7    35
dtype: int64
```

```
Desired Output:
[nan, 2.0, 3.0, 4.0, 5.0, 6.0, 6.0, 8.0]
```

[7]:

```
# Part B
df=pd.read_csv(r'C:\Users\KIIT\Desktop\High␣
 ↪Radius\Assignments\Grouby_Assignment_Data.csv')
df.head()
```

[7]:

```
   Unnamed: 0  Unnamed: 0.1 school_state teacher_prefix  \
0           0             0           in            mrs
1           1             1           fl             mr
2           2             2           az             ms
3           3             3           ky            mrs
4           4             4           tx            mrs

  project_grade_category  teacher_number_of_previously_posted_projects  \
0          grades_prek_2                                             0
1             grades_6_8                                             7
2             grades_6_8                                             1
3          grades_prek_2                                             4
4          grades_prek_2                                             1

   project_is_approved      project_subject_categories  \
0                    0                literacy_language
1                    1    history_civics_health_sports
2                    0                    health_sports
3                    1  literacy_language_math_science
4                    1                     math_science

  project_subject_subcategories   price  quantity
0                  esl_literacy  154.60        23
1  civics_government_teamsports  299.00         1
2    health_wellness_teamsports  516.85        22
3          literacy_mathematics  232.90         4
4                   mathematics   67.98         4
```

[8]:

```
"""
Q1. Find the Average price of project from each state
"""
df.groupby("school_state")["price"].mean()
```

[8]:

```
school_state
ak    337.510667
al    298.641397
ar    278.166613
az    252.355673
ca    323.282639
```

```
co    252.666940
ct    311.030415
dc    360.152267
de    234.136735
fl    297.499525
ga    308.207945
hi    365.838639
ia    284.773153
id    253.708874
il    284.538685
in    249.736221
ks    246.894763
ky    280.020031
la    358.954185
ma    328.623520
md    303.952794
me    274.640000
mi    299.793970
mn    249.289851
mo    276.094635
ms    306.512922
mt    278.000490
nc    254.037568
nd    256.985035
ne    286.515307
nh    330.994425
nj    336.891618
nm    297.836876
nv    283.005000
ny    335.973861
oh    271.301090
ok    264.163071
or    289.779098
pa    279.585613
ri    296.904035
sc    247.343586
sd    249.673433
tn    279.079419
tx    304.977799
ut    310.182139
va    267.539311
vt    289.467250
wa    283.266018
wi    301.433963
wv    258.633121
wy    307.638878
Name: price, dtype: float64
```

[9]:
```
"""
Q2. a. Find the total number of projects previously posted by all the teachers␣
 ↪belonging to each teacher prefix.
For Example all the teachers having prefix as dr have posted a total of 13␣
 ↪projects combined previously.
"""
df_grouped=df.groupby(["teacher_prefix"])
df_grouped["project_is_approved"].count()
```

[9]:
```
teacher_prefix
dr            13
mr         10645
mrs        57264
ms         38944
teacher     2360
Name: project_is_approved, dtype: int64
```

[10]:
```
"""
Q2 b) Find the prefix of the teacher who has posted the maximum of projects␣
 ↪previously.
"""
df_grouped=df.groupby(["teacher_prefix"])
s=df_grouped["teacher_number_of_previously_posted_projects"].max()
s.sort_values(ascending=False).head(1)
```

[10]:
```
teacher_prefix
mrs    451
Name: teacher_number_of_previously_posted_projects, dtype: int64
```

[11]:
```
"""
Q3. Find the number of projects approved for each project subject category␣
 ↪belonging
to the project grade category 'grades_9_12'.
"""
newdf = df[(df.project_grade_category == "grades_9_12")& (df.
 ↪project_is_approved == 1)]
newdf.reset_index(inplace=True)
newdf.groupby("project_subject_categories")["project_is_approved"].
 ↪value_counts().sort_values()
```

[11]:
```
project_subject_categories          project_is_approved
music_arts_history_civics           1                         1
music_arts_appliedlearning          1                         1
math_science_warmth_care_hunger     1                         1
appliedlearning_warmth_care_hunger  1                         1
health_sports_warmth_care_hunger    1                         2
specialneeds_warmth_care_hunger     1                         4
```

```
music_arts_health_sports            1                         4
history_civics_health_sports        1                         6
health_sports_appliedlearning       1                         6
history_civics_appliedlearning      1                         6
health_sports_history_civics        1                         7
literacy_language_health_sports     1                         8
health_sports_math_science          1                         8
health_sports_literacy_language     1                        10
music_arts_specialneeds             1                        16
specialneeds_health_sports          1                        16
history_civics_math_science         1                        16
health_sports_music_arts            1                        16
literacy_language_appliedlearning   1                        19
math_science_history_civics         1                        31
appliedlearning_health_sports       1                        34
specialneeds_music_arts             1                        38
math_science_health_sports          1                        46
literacy_language_history_civics    1                        50
history_civics_music_arts           1                        50
history_civics_specialneeds         1                        54
health_sports_specialneeds          1                        54
appliedlearning_history_civics      1                        57
math_science_literacy_language      1                        71
literacy_language_math_science      1                        72
history_civics_literacy_language    1                        92
appliedlearning_specialneeds        1                        99
math_science_music_arts             1                       139
appliedlearning_music_arts          1                       139
appliedlearning_math_science        1                       142
math_science_appliedlearning        1                       146
literacy_language_music_arts        1                       153
warmth_care_hunger                  1                       166
math_science_specialneeds           1                       177
appliedlearning_literacy_language   1                       191
literacy_language_specialneeds      1                       220
specialneeds                        1                       309
appliedlearning                     1                       384
history_civics                      1                       449
health_sports                       1                       940
music_arts                          1                       962
literacy_language                   1                      1769
math_science                        1                      1999
Name: project_is_approved, dtype: int64
```

[12]:

```
"""
Q4. Replace teacher_prefix with the average number of approved projects for␣
 ↪each teacher prefix.
```

```
"""
newdf=pd.DataFrame(df.groupby("teacher_prefix")["project_is_approved"].mean())
print(newdf)
```

```
                project_is_approved
teacher_prefix
dr                         0.692308
mr                         0.841522
mrs                        0.855546
ms                         0.843519
teacher                    0.795339
```

[13]:

```
mean_value_map={"dr":0.692308,"mr":0.841522,"mrs":0.855546,"ms":0.
 ↪843519,"teacher":0.795339}
df["teacher_prefix"]=df["teacher_prefix"].map(mean_value_map)
df.head()
```

[13]:

```
   Unnamed: 0  Unnamed: 0.1 school_state  teacher_prefix  \
0           0             0           in        0.855546
1           1             1           fl        0.841522
2           2             2           az        0.843519
3           3             3           ky        0.855546
4           4             4           tx        0.855546

  project_grade_category  teacher_number_of_previously_posted_projects  \
0          grades_prek_2                                             0
1             grades_6_8                                             7
2             grades_6_8                                             1
3          grades_prek_2                                             4
4          grades_prek_2                                             1

   project_is_approved      project_subject_categories  \
0                    0               literacy_language
1                    1    history_civics_health_sports
2                    0                   health_sports
3                    1  literacy_language_math_science
4                    1                    math_science

  project_subject_subcategories   price  quantity
0                  esl_literacy  154.60        23
1  civics_government_teamsports  299.00         1
2    health_wellness_teamsports  516.85        22
3          literacy_mathematics  232.90         4
4                   mathematics   67.98         4
```

[14]:

```
"""
Q5. Create the data frame (Train Data) as shown in the image below (Output can␣
 ↪be in fraction form or decimal form )
```

```
Image in the assignment doc.
"""
data = {'State':["A","B","C","A","A","B","A","A","C","C"],
        'Class':[0,1,1,0,1,1,0,1,1,0]}
train_data=pd.DataFrame(data)
train_data_crosstab=pd.crosstab(train_data["State"],train_data["Class"])
train_data_crosstab.columns=["Class_0","Class_1"]
print(train_data_crosstab)
encoded_train_data=pd.DataFrame()
encoded_train_data["State_0"]=train_data["State"].
 ↪map(train_data_crosstab["Class_0"]/
 ↪(train_data_crosstab["Class_0"]+train_data_crosstab["Class_1"]))
encoded_train_data["State_1"]=train_data["State"].
 ↪map(train_data_crosstab["Class_1"]/
 ↪(train_data_crosstab["Class_0"]+train_data_crosstab["Class_1"]))
encoded_train_data
```

```
       Class_0  Class_1
State
A            3        2
B            0        2
C            1        2
```

[14]:

```
    State_0   State_1
 0  0.600000  0.400000
 1  0.000000  1.000000
 2  0.333333  0.666667
 3  0.600000  0.400000
 4  0.600000  0.400000
 5  0.000000  1.000000
 6  0.600000  0.400000
 7  0.600000  0.400000
 8  0.333333  0.666667
 9  0.333333  0.666667
```