import pandas as pd

df=pd.read_csv("Sales.csv")

df.head(5)

	Sale date	Receipt no	Ordertype name	Item name	Category name	Cost Price	Selling price	Item quantity	Item amount
0	06- 03- 2020	BL11	On Shop	Jeans - Levis	Casual Wear	900	1100	1	1100
1	06- 03- 2020	BL11	On Shop	Jeans - Levis	Casual Wear	900	1100	1	1100
2	06- 03- 2020	BL12	On Shop	Jeans - Levis	Casual Wear	900	1100	1	1100
3	06- 03- 2020	BL12	On Shop	Jeans - Levis	Casual Wear	900	1100	1	1100
Л	06- 03-	RI 12	On Shon	Jeans -	Casual	۵۸۸	1100	1	1100

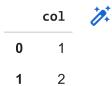
#taking Avg based on particular things
result=df.groupby(['Receipt no','Ordertype name']).agg({'Item amount':'mean'}).sort_values
result

Ttom amount



#applying condition like find the student which is having attempted 1 subject test
result_specific_col=df.groupby(['Receipt no']).agg({'Location':pd.Series.nunique})
result_specific_col=result_specific_col[result_specific_col['Location']==1].reset_index()
result_specific_col['Receipt no']

```
0
           BL126
     1
           BL137
     2
           BL140
     3
           BL169
     4
           BL176
     5
           BL207
     6
           BL208
     7
           BL216
     8
           BL222
           BL232
     9
           BL238
     10
     11
           BL243
           BL270
     12
           BL275
     13
     14
           BL286
     15
           BL295
            LN34
     16
     Name: Receipt no, dtype: object
data = {
    'col':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
}
data=pd.DataFrame(data)
data
```



#taking indexes with matching condition and replacing with some value
row_indexes=data[data['col']%3==0].index
data.loc[row_indexes,'new_1']=1
row_indexes=data[data['col']%3!=0].index
data.loc[row_indexes,'new_0']=0

data

	col	new_1	new_0	7
0	1	NaN	0.0	
1	2	NaN	0.0	
2	3	1.0	NaN	
3	4	NaN	0.0	
4	5	NaN	0.0	
5	6	1.0	NaN	
6	7	NaN	0.0	
7	8	NaN	0.0	
8	9	1.0	NaN	
9	10	NaN	0.0	
10	11	NaN	0.0	
11	12	1.0	NaN	
12	13	NaN	0.0	
13	14	NaN	0.0	
14	15	1.0	NaN	

#Combining two column after removing NAN values from table
data['col1']=data[data.columns[1:]].apply(lambda x: ' '.join(x.dropna().astype(str)),axis=
data

data

	col	new_1	new_0	col1
0	1	NaN	0.0	0.0
1	2	NaN	0.0	0.0
2	3	1.0	NaN	1.0
3	4	NaN	0.0	0.0
4	5	NaN	0.0	0.0
5	6	1.0	NaN	1.0
6	7	NaN	0.0	0.0
7	8	NaN	0.0	0.0
8	9	1.0	NaN	1.0
9	10	NaN	0.0	0.0

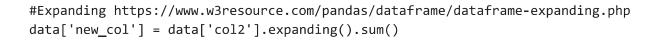
#Converting string to int
data['col1']=pd.to_numeric(data['col1'])

12 13 NaN 00 00
data['col2'] = data['col1'].apply(lambda x: 1 if x>0 else 0)

	col	new_1	new_0	col1	col2	1
0	1	NaN	0.0	0.0	0	
1	2	NaN	0.0	0.0	0	
2	3	1.0	NaN	1.0	1	
3	4	NaN	0.0	0.0	0	
4	5	NaN	0.0	0.0	0	
5	6	1.0	NaN	1.0	1	
6	7	NaN	0.0	0.0	0	
7	8	NaN	0.0	0.0	0	
8	9	1.0	NaN	1.0	1	
9	10	NaN	0.0	0.0	0	
10	11	NaN	0.0	0.0	0	
11	12	1.0	NaN	1.0	1	
12	13	NaN	0.0	0.0	0	
13	14	NaN	0.0	0.0	0	
14	15	1.0	NaN	1.0	1	

data['cumsum']=data['col2'].cumsum()
data

	col	new_1	new_0	col1	col2	cumsum
0	1	NaN	0.0	0.0	0	0
1	2	NaN	0.0	0.0	0	0
2	3	1.0	NaN	1.0	1	1
3	4	NaN	0.0	0.0	0	1
4	5	NaN	0.0	0.0	0	1
5	6	1.0	NaN	1.0	1	2
6	7	NaN	0.0	0.0	0	2
7	8	NaN	0.0	0.0	0	2
8	9	1.0	NaN	1.0	1	3
9	10	NaN	0.0	0.0	0	3
10	11	NaN	0.0	0.0	0	3
11	12	1.0	NaN	1.0	1	4
12	13	NaN	0.0	0.0	0	4
13	14	NaN	0.0	0.0	0	4
14	15	1.0	NaN	1.0	1	5



data

	col	new_1	new_0	col1	col2	cumsum	new_col
0	1	NaN	0.0	0.0	0	0	0.0
1	2	NaN	0.0	0.0	0	0	0.0
2	3	1.0	NaN	1.0	1	1	1.0
3	4	NaN	0.0	0.0	0	1	1.0
"B":	[50,	mA", "T 40, 40, , False	30, 50)],		eamC" , "	TeamA"],

cum=pd.DataFrame(cum) cum

	Α	В	С	1
0	TeamA	50	True	
1	TeamB	40	False	
2	TeamB	40	False	
3	TeamC	30	False	
4	TeamA	50	True	

s=cum.groupby('A').expanding().count()['B']==2 s=s.reset_index() S

	Α	level_1	В	1
0	TeamA	0	False	
1	TeamA	4	True	
2	TeamB	1	False	
3	TeamB	2	True	
4	TeamC	3	False	

1 TeamA

3 TeamB

Name: A, dtype: object

cum.rolling(2).sum()

	В	С	
0	NaN	NaN	
1	90.0	1.0	
2	80.0	0.0	
3	70.0	0.0	
4	80.0	1.0	

Colab paid products - Cancel contracts here

✓ 0s completed at 5:48 PM

• ×