Viviyan Richards W

205229133

Lab10. Advanced Data Wrangling in Pandas

Import necessary modules

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
excelample = pd.DataFrame({'Month': ["January", "January", "January", "January",'
```

In [3]:
```
excelample
```

Out[3]:

	Month	Category	Amount
0	January	Transportation	74.0
1	January	Grocery	235.0
2	January	Household	175.0
3	January	Entertainment	100.0
4	February	Transportation	115.0
5	February	Grocery	240.0
6	February	Household	225.0
7	February	Entertainment	125.0
8	March	Transportation	90.0
9	March	Grocery	260.0
10	March	Household	200.0
11	March	Entertainment	120.0

In [4]:
```
excelample_pivot = excelample.pivot(index="Category", columns="Month", values="Am
excelample_pivot
```

Out[4]:

Category	Month February	January	March
Entertainment	125.0	100.0	120.0
Grocery	240.0	235.0	260.0
Household	225.0	175.0	200.0
Transportation	115.0	74.0	90.0

In [5]:
```
excelample_pivot.sum(axis=1)
```

Out[5]:
```
Category
Entertainment      345.0
Grocery            735.0
Household          600.0
Transportation     279.0
dtype: float64
```

In [6]:
```
excelample_pivot.sum(axis=0)
```

Out[6]:
```
Month
February    705.0
January     584.0
March       670.0
dtype: float64
```

Pivot is just reordering your data

In [7]:
```
df = pd.DataFrame({'Fare': [7.25, 71.2833, 51.8625, 30.0708, 7.8542, 13.0],'Pclas
df
```

Out[7]:

	Fare	Pclass	Sex	Survived
0	7.2500	3	male	0
1	71.2833	1	female	1
2	51.8625	1	male	0
3	30.0708	2	female	1
4	7.8542	3	female	0
5	13.0000	2	male	1

In [8]:
```
df.pivot(index="Pclass",columns="Sex",values="Fare")
```

Out[8]:

Pclass	female	male
1	71.2833	51.8625
2	30.0708	13.0000
3	7.8542	7.2500

In [9]:
```
df.pivot(index="Pclass",columns="Sex",values="Survived")
```

Out[9]:

Pclass	female	male
1	1	0
2	1	1
3	0	0

Let's now use the full Titanic Dataset

In [10]:
```
df = sns.load_dataset('titanic')
df
```

Out[10]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 15 columns

In [11]:
```
try:
    df.pivot(index='sex', columns='pclass', values='fare')
except Exception as e:
    print("Exception!", e)
```

```
Exception! Index contains duplicate entries, cannot reshape
```

In [12]:
```
df.loc[[1, 3], ["sex", 'pclass', 'fare']]
```

Out[12]:

	sex	pclass	fare
1	female	1	71.2833
3	female	1	53.1000

Pivot Tables - Aggregating while Pivoting

In [13]:
```
#Pivot Table is a multidimensional version of GroupBy aggregation.
```

In [14]:
```
df.pivot_table(index='sex', columns='pclass', values='fare')
```

Out[14]:

pclass	1	2	3
sex			
female	106.125798	21.970121	16.118810
male	67.226127	19.741782	12.661633

Create a Pivot table with maximum 'fare' values for 'sex' vs 'pclass' columns

In [15]:
```
df.pivot_table(index='sex', columns='pclass',
values='fare', aggfunc='max')
```

Out[15]:

pclass	1	2	3
sex			
female	512.3292	65.0	69.55
male	512.3292	73.5	69.55

Create a Pivot table with the count of 'fare' values for 'sex' vs 'pclass' columns

In [16]:
```
df.pivot_table(index='sex', columns='pclass',values='fare', aggfunc='count')
```

Out[16]:

pclass	1	2	3
sex			
female	94	76	144
male	122	108	347

In [17]:
```
pd.crosstab(index=df['sex'], columns=df['pclass'])
```

Out[17]:

pclass	1	2	3
sex			
female	94	76	144
male	122	108	347

Exercise: Make a pivot table with the mean survival rates for pclass vs sex

In [18]:
```
df.pivot_table(index='sex', columns='pclass',values='survived', aggfunc='mean')
```

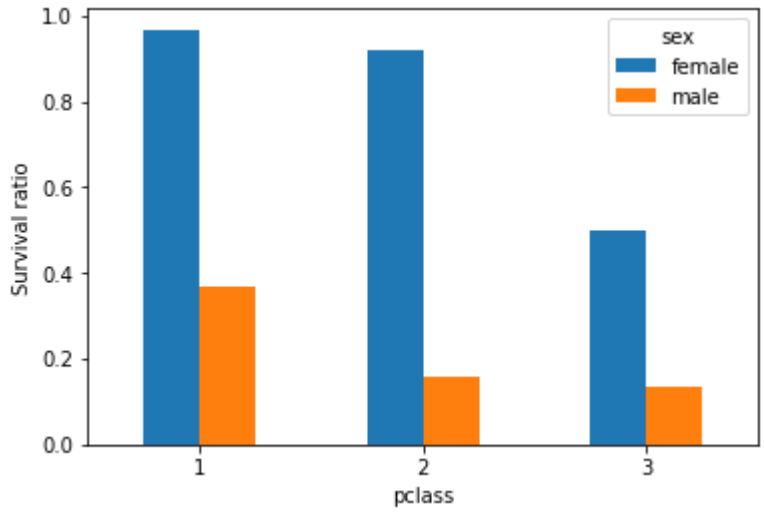
Out[18]:

pclass	1	2	3
sex			
female	0.968085	0.921053	0.500000
male	0.368852	0.157407	0.135447

Plot Bar Chart for Survival ratio

In [19]:
```
fig, ax1 = plt.subplots()
df.pivot_table(index='pclass', columns='sex',values='survived', aggfunc='mean').p
ax1.set_ylabel('Survival ratio')
```

Out[19]: Text(0, 0.5, 'Survival ratio')



Make a pivot table of the median Fare payed by aged vs sex

In [20]:
```
median_age_table=df.pivot_table(index='age', columns='sex',values='fare', aggfunc
```

In [21]:
```
median_age_table.head()
```

Out[21]:

sex	female	male
age		
0.42	NaN	8.5167
0.67	NaN	14.5000
0.75	19.2583	NaN
0.83	NaN	23.8750
0.92	NaN	151.5500

Exercise: Make a pivot table of the median Fare payed by 'underaged' vs 'sex'

In [22]:
```
df['underaged'] =df["age"]<=18
```

In [23]:
```
df
```

Out[23]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 16 columns

In [24]:
```
# Now, make the pivot table for underaged
median_age_table=df.pivot_table(index='underaged', columns='sex',values='fare', a
```

In [25]:
```
median_age_table
```

Out[25]:

sex	female	male
underaged		
False	24.1500	10.3354
True	20.2875	20.2500

Grouping Pivot table

In [26]:
```
age = pd.cut(df['age'], [0, 18, 80])
df.pivot_table('survived', ['sex', age], 'class')
```

Out[26]:

	class	First	Second	Third
sex	**age**			
female	**(0, 18]**	0.909091	1.000000	0.511628
	(18, 80]	0.972973	0.900000	0.423729
male	**(0, 18]**	0.800000	0.600000	0.215686
	(18, 80]	0.375000	0.071429	0.133663

In [27]:
```
fare = pd.qcut(df['fare'], 2)
df.pivot_table('survived', ['sex', age], [fare, 'class'])
```

Out[27]:

	fare	(-0.001, 14.454]			(14.454, 512.329]		
	class	First	Second	Third	First	Second	Third
sex	age						
female	(0, 18]	NaN	1.000000	0.714286	0.909091	1.000000	0.318182
	(18, 80]	NaN	0.880000	0.444444	0.972973	0.914286	0.391304
male	(0, 18]	NaN	0.000000	0.260870	0.800000	0.818182	0.178571
	(18, 80]	0.0	0.098039	0.125000	0.391304	0.030303	0.192308

Multiple Aggregate Functions

In [28]:
```
df.pivot_table(index='sex', columns='class',aggfunc={'survived':sum, 'fare':'mean
```

Out[28]:

	fare			survived		
class	First	Second	Third	First	Second	Third
sex						
female	106.125798	21.970121	16.118810	91	70	72
male	67.226127	19.741782	12.661633	45	17	47

Melt - from Pivot Table to long or tidy format

In [29]:
```
pivoted = df.pivot_table(index='sex', columns='pclass', values='fare').reset_inde
```

In [30]:
```
pivoted
```

Out[30]:

pclass	sex	1	2	3
0	female	106.125798	21.970121	16.118810
1	male	67.226127	19.741782	12.661633

In [31]:
```
pd.melt(pivoted)
```

Out[31]:

	pclass	value
0	sex	female
1	sex	male
2	1	106.125798
3	1	67.226127
4	2	21.970121
5	2	19.741782
6	3	16.11881
7	3	12.661633

In [32]:
```
pd.melt(pivoted, id_vars=['sex'] )#, var_name='pclass', value_name='fare')
```

Out[32]:

	sex	pclass	value
0	female	1	106.125798
1	male	1	67.226127
2	female	2	21.970121
3	male	2	19.741782
4	female	3	16.118810
5	male	3	12.661633

Reshaping with stack and unstack

In [33]:
```
df2 = pd.DataFrame({'A':['one', 'one', 'two', 'two'],'B':['a', 'b', 'a', 'b'],'C'
df2
```

Out[33]:

	A	B	C
0	one	a	0
1	one	b	1
2	two	a	2
3	two	b	3

In [34]:
```
df2 = df2.set_index(['A', 'B']) # Indeed, you can combine two indices
df2
```

Out[34]:

		C
A	**B**	
one	**a**	0
	b	1
two	**a**	2
	b	3

In [35]:
```
result = df2['C'].unstack()
result
```

Out[35]:

B	**a**	**b**
A		
one	0	1
two	2	3

In [36]:
```
df2 = result.stack().reset_index(name='C')
df2
```

Out[36]:

	A	**B**	**C**
0	one	a	0
1	one	b	1
2	two	a	2
3	two	b	3

Mimick Pivot Table

In [37]:
```
df
```

Out[37]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 16 columns

In [38]:
```
df.pivot_table(index='pclass', columns='sex',values='survived', aggfunc='mean')
```

Out[38]:

sex	female	male
pclass		
1	0.968085	0.368852
2	0.921053	0.157407
3	0.500000	0.135447

Exercise:

- Get the same result as above based on a combination of `groupby` and `unstack` </l i>
- First use `groupby` to calculate the survival ratio for all groups `unstack`
- Then, use `unstack` to reshape the output of the groupby operation

In [39]:
```
temp=df.groupby(['pclass','sex'])['survived'].agg('mean')
temp.unstack()
```

Out[39]:

sex	female	male
pclass		
1	0.968085	0.368852
2	0.921053	0.157407
3	0.500000	0.135447

In [ ]:
```

```