Inclass - Lab (Day 3)

Table of Content

Let us import the required library

In [1]:
```
# import libraries
import numpy as np
import pandas as pd
```

Let's begin with some hands-on practice exercises

1. Pivot Table

3. Find the average sales for each store

Use the dataframe given below:

Store	Location	Sales
A	Mumbai	40000
B	Pune	45000
A	Hyderabad	50000
C	Mumbai	90000
D	Pune	89000
A	Delhi	87000
D	Hyderabad	85000
A	Pune	78000
C	Mumbai	89000

B	Pune	70000

In [35]:
```
# type your code here
df1 = pd.DataFrame(data = {'Store':['A','B','A','C','D','A','D','A','C','B'],'Loc


                           'Sales':[40000,45000,50000,90000,89000,87000,85000,780
pd.pivot_table(df1, index = 'Store',aggfunc = np.mean)
pd.pivot_table(df1, values = 'Sales', index = 'Store', columns = 'Location', aggf
```

Out[35]:

Location	Delhi	Hyderabad	Mumbai	Pune
Store				
A	87000.0	50000.0	40000.0	78000.0
B	NaN	NaN	NaN	57500.0
C	NaN	NaN	89500.0	NaN
D	NaN	85000.0	NaN	89000.0

4. Compute the average sales for each store in every city

Use the dataframe given below:

Store	Location	Sales
A	Mumbai	40000
B	Pune	45000
A	Hyderabad	50000
C	Mumbai	90000
D	Pune	89000
A	Delhi	87000
D	Hyderabad	85000
A	Pune	78000
C	Mumbai	89000
B	Pune	70000

In [36]:
```
# type your code here
#pd.pivot_table(df1, index = ['Store','Location'],aggfunc = np.mean) can use this
pd.pivot_table(df1, values = 'Sales', index = 'Store', columns = 'Location',aggfu
```

Out[36]:

Location	Delhi	Hyderabad	Mumbai	Pune
Store				
A	87000.0	50000.0	40000.0	78000.0
B	NaN	NaN	NaN	57500.0
C	NaN	NaN	89500.0	NaN
D	NaN	85000.0	NaN	89000.0

5. Compute the minimum sales for each store in every city

Use the dataframe given below:

Store	Location	Sales
A	Mumbai	40000
B	Pune	45000
A	Hyderabad	50000
C	Mumbai	90000
D	Pune	89000
A	Delhi	87000
D	Hyderabad	85000
A	Pune	78000
C	Mumbai	89000
B	Pune	70000

In [53]:
```
# type your code here
pd.pivot_table(df1,values = 'Sales',index=['Location'],columns=['Store'],aggfunc=
```

Out[53]:

Store	A	B	C	D
Location				
Delhi	87000.0	NaN	NaN	NaN
Hyderabad	50000.0	NaN	NaN	85000.0
Mumbai	40000.0	NaN	89000.0	NaN
Pune	78000.0	45000.0	NaN	89000.0

2. Duplicate

6. Find duplicate rows in the data

Use the dataframe given below:

Name	Salary	City
John	3400	Sydeny
Robert	3000	Chicago
Aadi	1600	New York
Robert	3000	Chicago
Robert	3000	Chicago
Robert	3000	Texas
Aadi	4000	London
Sachine	3000	Chicago

In [38]:
```
# type your code here
df6 = pd.DataFrame(data={'Name':['John','Robert','Aadi','Robert','Robert','Robert
                         'Salary':[3400,3000,1600,3000,3000,3000,4000,3000],
                         'City':['Sydney','Chicago','New York','Chicago','Chicago'
df6_res = df6[df6.duplicated(keep=False)]
print(df6_res)
```

```
     Name  Salary     City
1  Robert    3000  Chicago
3  Robert    3000  Chicago
4  Robert    3000  Chicago
```

7. Select duplicate rows, except the last occurrence

Use the dataframe given below:

Name	Salary	City
John	3400	Sydeny
Robert	3000	Chicago
Aadi	1600	New York
Robert	3000	Chicago
Robert	3000	Chicago
Robert	3000	Texas
Aadi	4000	London
Sachine	3000	Chicago

In [40]:
```
# type your code here
df7_res = df6[df6.duplicated(keep='last')]
print('Below is the duplicate rows except the last duplicate')
print(df7_res)
```

```
Below is the duplicate rows except the last duplicate
     Name  Salary     City
1  Robert    3000  Chicago
3  Robert    3000  Chicago
```

8. Select the duplicated rows based on the column names 'Salary' and 'City'

Use the dataframe given below:

Name	Salary	City
John	3400	Sydeny
Robert	3000	Chicago
Aadi	1600	New York
Robert	3000	Chicago
Robert	3000	Chicago
Robert	3000	Texas
Aadi	4000	London
Sachine	3000	Chicago

In [44]:
```
# type your code here
df8_res = df6[df6.duplicated(['Salary','City'])]
print('Below are the duplicates based on SALARY and CITY column')
print(df8_res)
```

```
Below are the duplicates based on SALARY and CITY column
      Name  Salary     City
3   Robert    3000  Chicago
4   Robert    3000  Chicago
7  Sachine    3000  Chicago
```

3. Replace

9. Replace 'football' with 'hockey' in the column tournament

Days	Tournament
Mon	Football
Tues	Cricket
Wed	Football

Thurs	Football
Fri	Cricket

In [50]:
```
# type your code here
tou = {'Football':'Hockey'}
df9 = pd.DataFrame(data = {'Days':['Mon','Tues','Wed','Thurs','Fri'],'Tournament'
print('Initial data frame')
print(df9)
df9.Tournament.replace(tou,inplace=True)
print("After replacing football with hockey below is the data frame")
print(df9)
```

```
Initial data frame
    Days Tournament
0    Mon   Football
1   Tues    Cricket
2    Wed   Football
3  Thurs   Football
4    Fri    Cricket
After replacing football with hockey below is the data frame
    Days Tournament
0    Mon     Hockey
1   Tues    Cricket
2    Wed     Hockey
3  Thurs     Hockey
4    Fri    Cricket
```

10. Replace all 0's with male and all 1's with female in the gender column

Use the dataframe given below:

Name	Num_Children	Gender
John	0	0
Robert	4	0
Johny	5	0
Mia	3	1

In [51]:
```
# type your code here
df10 = pd.DataFrame(data={'Name':['John','Robert','Johny','Mia'],'Num_Children':[
gen = {0:'Male',1:'Female'}
print('Initial data frame')
print(df10)
df10.Gender.replace(gen,inplace=True)
print('After replacing 0 with MALE and 1 with FEMALE')
print(df10)
```

```
Initial data frame
     Name  Num_Children  Gender
0    John             0       0
1  Robert             4       0
2   Johny             5       0
3     Mia             3       1
After replacing 0 with MALE and 1 with FEMALE
     Name  Num_Children  Gender
0    John             0    Male
1  Robert             4    Male
2   Johny             5    Male
3     Mia             3  Female
```

4. Summary Statistics

12. Find the descriptive statistics for the sales column of the given dataframe

Use the dataframe given below:

Month	Sales	Seasons
Jan	22000	Winter
Feb	27000	Winter
Mar	25000	Spring
Apr	29000	Spring
May	35000	Spring
June	67000	Summer
July	78000	Summer
Aug	67000	Summer
Sep	56000	Fall
Oct	56000	Fall
Nov	56000	Fall
Dec	60000	Winter

In [59]:
```
# type your code here
df12 = pd.DataFrame(data = {'Month':['Jan','Feb','Mar','Apr','May','June','July',
                            'Sales':[22000,27000,25000,29000,35000,67000,78000,670
                            'Seasons':['Winter','Winter','Spring','Spring','Spring
print('The data frame is:')
print(df12)
stats = df12['Sales'].describe()
print('The descriptive statistics of Sales column is: ')
print(stats)
```

```
The data frame is:
   Month  Sales Seasons
0    Jan  22000  Winter
1    Feb  27000  Winter
2    Mar  25000  Spring
3    Apr  29000  Spring
4    May  35000  Spring
5   June  67000  Summer
6   July  78000  Summer
7    Aug  67000  Summer
8    Sep  56000    Fall
9    Oct  56000    Fall
10   Nov  56000    Fall
11   Dec  60000  Winter
The descriptive statistics of Sales column is:
count       12.000000
mean     48166.666667
std      19385.249000
min      22000.000000
25%      28500.000000
50%      56000.000000
75%      61750.000000
max      78000.000000
Name: Sales, dtype: float64
```

13. Find the median of sales for each season

Use the dataframe given below:

Month	Sales	Seasons
Jan	22000	Winter
Feb	27000	Winter
Mar	25000	Spring
Apr	29000	Spring
May	35000	Spring
June	67000	Summer
July	78000	Summer
Aug	67000	Summer

Sep	56000	Fall
Oct	56000	Fall
Nov	56000	Fall
Dec	60000	Winter

In [61]:
```
# type your code here
#df12.groupby('Seasons').describe()
df12.groupby('Seasons').agg(np.median)
```

Out[61]:

Seasons	Sales
Fall	56000
Spring	29000
Summer	67000
Winter	27000

14. Find the descriptive statistics for categorical data from the dataframe

Use the dataframe given below:

Month	Sales	Seasons
Jan	22000	Winter
Feb	27000	Winter
Mar	25000	Spring
Apr	29000	Spring
May	35000	Spring
June	67000	Summer
July	78000	Summer
Aug	67000	Summer
Sep	56000	Fall
Oct	56000	Fall
Nov	56000	Fall
Dec	60000	Winter

In [85]:
```
# type your code here
df12.groupby(['Seasons','Month']).describe()
```

Out[85]:

		Sales							
		count	mean	std	min	25%	50%	75%	max
Seasons	Month								
Fall	Nov	1.0	56000.0	NaN	56000.0	56000.0	56000.0	56000.0	56000.0
	Oct	1.0	56000.0	NaN	56000.0	56000.0	56000.0	56000.0	56000.0
	Sep	1.0	56000.0	NaN	56000.0	56000.0	56000.0	56000.0	56000.0
Spring	Apr	1.0	29000.0	NaN	29000.0	29000.0	29000.0	29000.0	29000.0
	Mar	1.0	25000.0	NaN	25000.0	25000.0	25000.0	25000.0	25000.0
	May	1.0	35000.0	NaN	35000.0	35000.0	35000.0	35000.0	35000.0
Summer	Aug	1.0	67000.0	NaN	67000.0	67000.0	67000.0	67000.0	67000.0
	July	1.0	78000.0	NaN	78000.0	78000.0	78000.0	78000.0	78000.0
	June	1.0	67000.0	NaN	67000.0	67000.0	67000.0	67000.0	67000.0
Winter	Dec	1.0	60000.0	NaN	60000.0	60000.0	60000.0	60000.0	60000.0
	Feb	1.0	27000.0	NaN	27000.0	27000.0	27000.0	27000.0	27000.0
	Jan	1.0	22000.0	NaN	22000.0	22000.0	22000.0	22000.0	22000.0

15. Find the kurtosis for each subject

Use the dataframe given below:

Name	Maths	Science	English
Emma	56	89	89
Mia	78	87	89
Sophia	78	78	76
James	67	89	78
John	88	78	87

In [69]:
```
# type your code here
student = [('Emma',56,89,89),
           ('Mia',78,87,89),
           ('Sophia',78,78,76),
           ('James',67,89,78),
           ('John',88,78,87)]
df15 = pd.DataFrame(student, columns = ['Name','Maths','Science','English'])
print(df15,'\n')
print('The kurtosis of Maths is:',df15.Maths.kurtosis())
print('The kurtosis of Science is:',df15.Science.kurtosis())
print('The kurtosis of English is:',df15.English.kurtosis())
```

```
     Name  Maths  Science  English
0    Emma     56       89       89
1     Mia     78       87       89
2  Sophia     78       78       76
3   James     67       89       78
4    John     88       78       87

The kurtosis of Maths is: -0.21959853904003523
The kurtosis of Science is: -3.2323317341413444
The kurtosis of English is: -2.9238812504362084
```

5. Merge, Join, Concatenate

16. Merge the given dataframes on the column 'Brand'

Use the dataframe given below:

ID	Brand	Product
101	Apple	iPhone
102	Canon	DSLR
103	Samsung	SmartPhone
104	Nikon	DSLR
105	Sony	SmartTV

ID	Brand	Quantity_sold
101	Apple	234
102	Canon	344
104	Samsung	345
103	Nikon	262
105	Sony	356

In [72]:
```
# type your code here
df16a = pd.DataFrame(data = {'ID':[101,102,103,104,105],
                             'Brand':['Apple','Canon','Samsung','Nikon','Sony'],
                             'Product':['iPhone','DSLR','SmartPhone','DSLR','Smart
df16b = pd.DataFrame(data = {'ID':[101,102,104,103,105],
                             'Brand':['Apple','Canon','Samsung','Nikon','Sony'],
                             'Quantity_sold':[234,344,345,262,356]})
print('The first data frame is:\n',df16a)
print('The second data frame is:\n',df16b)
print('After Merging the 2 data frames on key Brand:\n')
print(pd.merge(df16a, df16b, how ='inner', on ='Brand'))
```

```
The first data frame is:
     ID    Brand     Product
0  101    Apple      iPhone
1  102    Canon        DSLR
2  103  Samsung  SmartPhone
3  104    Nikon        DSLR
4  105     Sony     SmartTV
The second data frame is:
     ID    Brand  Quantity_sold
0  101    Apple            234
1  102    Canon            344
2  104  Samsung            345
3  103    Nikon            262
4  105     Sony            356
After Merging the 2 data frames on key Brand:

   ID_x    Brand     Product  ID_y  Quantity_sold
0   101    Apple      iPhone   101            234
1   102    Canon        DSLR   102            344
2   103  Samsung  SmartPhone   104            345
3   104    Nikon        DSLR   103            262
4   105     Sony     SmartTV   105            356
```

17. Using the dataframes created in question 16, merge the given dataframes by 'ID' and 'Brand'

In [73]:
```
# type your code here
print('After Merging the 2 data frames on keys ID and Brand:\n')
print(pd.merge(df16a, df16b, how ='inner', on =['ID','Brand']))
```

```
After Merging the 2 data frames on keys ID and Brand:

    ID  Brand  Product  Quantity_sold
0  101  Apple   iPhone            234
1  102  Canon     DSLR            344
2  105   Sony  SmartTV            356
```

18. Using the dataframes created in question 16, perform left join to combine values in the columns 'Brand' and 'ID'

In [77]:
```
# type your code here
pd.merge(df16a,df16b,on=['Brand','ID'],how='left')
```

Out[77]:

	ID	Brand	Product	Quantity_sold
0	101	Apple	iPhone	234.0
1	102	Canon	DSLR	344.0
2	103	Samsung	SmartPhone	NaN
3	104	Nikon	DSLR	NaN
4	105	Sony	SmartTV	356.0

19. Using the dataframes created in question 16, perform outer join to combine values in the columns 'Brand' and 'ID'

In [78]:
```
# type your code here
pd.merge(df16a,df16b,on=['Brand','ID'],how='outer')
```

Out[78]:

	ID	Brand	Product	Quantity_sold
0	101	Apple	iPhone	234.0
1	102	Canon	DSLR	344.0
2	103	Samsung	SmartPhone	NaN
3	104	Nikon	DSLR	NaN
4	105	Sony	SmartTV	356.0
5	104	Samsung	NaN	345.0
6	103	Nikon	NaN	262.0

20. Concatenate rows of the given dataframes and assign the store as its key

Products in Store_A:

ID	Brand	Product
101	Apple	iPhone
102	Canon	DSLR
103	Samsung	SmartPhone
104	Nikon	DSLR
105	Sony	SmartTV

In [88]:
```
# type your code here
df20_a = pd.DataFrame(data={'ID':[101,102,103,104,105],
                            'Brand':['Apple','Canon','samsung','Nikon','Sony'],
                            'Product':['iPhone','DSLR','SmartPhone','DSLR','SmartT
print('The 1st Data frame is:\n',df20_a)
df20_b = pd.DataFrame(data={'ID':[103,104,104,107,101],
                            'Brand':['Apple','Apple','Samsung','Canon','Sony'],
                            'Product':['iPhone','iPod','SmartPhone','DSLR','SmartT
print('The 2nd Data frame is:\n',df20_b)
res_df = pd.concat((df20_a,df20_b),axis=0,keys=['Store_A','Store_B'])
print('The resulting data frame after concatination is:\n',res_df)
```

```
The 1st Data frame is:
     ID    Brand     Product
0  101    Apple      iPhone
1  102    Canon        DSLR
2  103  samsung  SmartPhone
3  104    Nikon        DSLR
4  105     Sony     SmartTV
The 2nd Data frame is:
     ID    Brand     Product
0  103    Apple      iPhone
1  104    Apple        iPod
2  104  Samsung  SmartPhone
3  107    Canon        DSLR
4  101     Sony     SmartTV
The resulting data frame after concatination is:
              ID    Brand     Product
Store_A 0  101    Apple      iPhone
        1  102    Canon        DSLR
        2  103  samsung  SmartPhone
        3  104    Nikon        DSLR
        4  105     Sony     SmartTV
Store_B 0  103    Apple      iPhone
        1  104    Apple        iPod
        2  104  Samsung  SmartPhone
        3  107    Canon        DSLR
        4  101     Sony     SmartTV
```