Take - home (Day 3)

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

Let's begin with some hands-on practice exercises.

Create a dataframe wherever necessary

1. Compute minimum and maximum sales for each store and location

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 [2]:
```
# write your code here
df1 = pd.DataFrame({'Store':['A','B','A','C','D','A','D','A','C','B'],
                    'Location':['Mumbai','Pune','Hyderabad','Mumbai','Pune','Delhi
                    'Sales':[40000,45000,50000,90000,89000,87000,85000,78000,8900
pd.pivot_table(df1,index = 'Location',columns='Store',aggfunc=['min','max'])
```

Out[2]:

	min				max			
	Sales				Sales			
Store	A	B	C	D	A	B	C	D
Location								
Delhi	87000.0	NaN	NaN	NaN	87000.0	NaN	NaN	NaN
Hyderabad	50000.0	NaN	NaN	85000.0	50000.0	NaN	NaN	85000.0
Mumbai	40000.0	NaN	89000.0	NaN	40000.0	NaN	90000.0	NaN
Pune	78000.0	45000.0	NaN	89000.0	78000.0	70000.0	NaN	89000.0

2. Find duplicate rows based on the column 'Name'

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
Sachin	3000	Chicago

In [4]:

```
# Write your code here
df2 = pd.DataFrame({'name':['John','Robert','Aadi','Robert','Robert','Robert','Aa
                    'Salary':[3400,3000,1600,3000,3000,3000,4000,3000],
                    'City':['Sydeny','Chicago','New York','Chicago','Chicago','Te
df2.duplicated(subset=['name'])
```

Out[4]:

```
0    False
1    False
2    False
3     True
4     True
5     True
6     True
7    False
dtype: bool
```

3. In column tournament, replace all the 'football' values with 'cricket' using numpy.where

Days	Tournament
Mon	Football
Tues	Cricket
Wed	Football
Thurs	Football
Fri	Cricket

In [5]:

```
# Write your code here
df3 = pd.DataFrame({'Days':['Mon','Tues','Wed','Thurs','Fri'],
                    'Tournament':['Football','Cricket','Football','Football','Cri

df3['Tournament'] = np.where(df3['Tournament'] == 'Football', 'Cricket', 'Cricket
df3
```

Out[5]:

	Days	Tournament
0	Mon	Cricket
1	Tues	Cricket
2	Wed	Cricket
3	Thurs	Cricket
4	Fri	Cricket

4. Get the descriptive statistics of the 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 [6]:

```
# Write your code here
df4 = pd.DataFrame({'Month':['Jan','Feb','Mar','Apr','May','June','July','Aug','S
                    'Sales':[22000,27000,25000,29000,35000,67000,78000,67000,560(
                    'Seasons':['Winter','Winter','Spring','Spring','Spring','Sumn
df4.groupby('Seasons')['Sales'].describe(include='all')
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Seasons								
Fall	3.0	56000.000000	0.000000	56000.0	56000.0	56000.0	56000.0	56000.0
Spring	3.0	29666.666667	5033.222957	25000.0	27000.0	29000.0	32000.0	35000.0
Summer	3.0	70666.666667	6350.852961	67000.0	67000.0	67000.0	72500.0	78000.0
Winter	3.0	36333.333333	20647.840888	22000.0	24500.0	27000.0	43500.0	60000.0

5. Create a new column age with values of your choice. And append it to the below dataframe.

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 [8]: # Write your code here
        df5 = pd.DataFrame({'name':['Emma','Mia','Sophia','James','John'],
                            'Maths':[56,78,78,67,88],
                            'Science':[89,87,78,89,78],
                            'English':[89,89,76,78,87]})
        En = pd.Series((20,21,22,23,24),name='Age')
        pd.concat((df5,En),axis=1)
```

Out[8]:

	name	Maths	Science	English	Age
0	Emma	56	89	89	20
1	Mia	78	87	89	21
2	Sophia	78	78	76	22
3	James	67	89	78	23
4	John	88	78	87	24

6. Perform right join to combine values based on the columns 'MA(Hons)' and 'Stud_ID' in the two dataframes

Use the dataframe given below:

Stud_ID	Name	MA(Hons)
101	Alex	History
102	Amy	English
103	Allen	Geography
104	Alice	German
105	James	History

Stud_ID	Res_City	MA(Hons)
101	Delhi	English
102	Mumbai	History
103	Delhi	Fine Arts
104	Chennai	German
105	Hyderabad	History

In [9]:

```
# Write your code here
df6a = pd.DataFrame({'Stud_ID':[101,102,103,104,105],
                     'Name':['Alex','Amy','Allen','Alice','James'],
                     'MA(Hons)':['History','English','Geography','German','Histor
df6b = pd.DataFrame({'Stud_ID':[101,102,103,104,105],
                     'Res_City':['Delhi','Mumbai','Delhi','Chennai','Hyderabad'],
                    'MA(Hons)':['History','English','Geography','German','History
df6a.set_index(['Stud_ID', 'MA(Hons)'], inplace= True)
df6b.set_index(['Stud_ID','MA(Hons)'], inplace= True)

df6a.join(df6b, how='right', rsuffix='df6b')
```

Out[9]:

```
                            Name   Res_City
Stud_ID   MA(Hons)
     101    History         Alex      Delhi
     102    English          Amy     Mumbai
     103  Geography        Allen      Delhi
     104     German        Alice    Chennai
     105    History        James  Hyderabad
```

7. Using the dataframes created in question 6, perform inner join to combine values based on the columns 'MA(Hons)' and 'Stud_ID' in the two dataframes

In [10]:

```
# Write your code here
df6a.join(df6b,how='inner')
```

Out[10]:

```
                            Name   Res_City
Stud_ID   MA(Hons)
     101    History         Alex      Delhi
     102    English          Amy     Mumbai
     103  Geography        Allen      Delhi
     104     German        Alice    Chennai
     105    History        James  Hyderabad
```

8. Concatenate two dataframes along the columns

Use the dataframe given below:

Stud_ID	Name
101	Alex
102	Amy
103	Allen
104	Alice
105	James

Res_City	MA(Hons)
Delhi	English
Mumbai	History
Delhi	Fine Arts
Chennai	German
Hyderabad	History

In [11]:
```
# Write your code here
df8a = pd.DataFrame({'Stud_ID':[101,102,103,104,104],
                     'Name':['Alex','Amy','Allen','Alice','James']})
df8b = pd.DataFrame({'Res_City':['Delhi','Mumbai','Delhi','Chennai','Hyderabad'],
                    'MA(Hons)':['History','English','Geography','German','History
pd.concat([df8a,df8b],axis=1)
```

Out[11]:

	Stud_ID	Name	Res_City	MA(Hons)
0	101	Alex	Delhi	History
1	102	Amy	Mumbai	English
2	103	Allen	Delhi	Geography
3	104	Alice	Chennai	German
4	104	James	Hyderabad	History

9. Calculate minimum, maximum and average sales for each season

Use the dataframe given below:

ID	Name	Subject
101	Alex	Maths
102	Amy	English
103	Allen	Science
104	Alice	German
105	Ayoung	History

ID	Name	Subject
101	Billy	English
102	Brian	Science
103	Bran	Social Science
104	Bryce	German
105	Betty	History

In [9]:

```
# Write your code here
```

10. Find all the duplicate entries based on the columns X and Y.

Use the dataframe given below:

X	Y	Z
1	2	5
2	2	6
3	1	2
1	2	6
2	2	1
3	4	6
2	2	2
2	2	8

In [12]:

```
df = pd.DataFrame({'X':[1,2,3,1,2,3,2,2],
                   'Y':[2,2,1,2,2,4,2,2],
                   'Z':[5,6,2,6,1,6,2,8]})
df['X'].duplicated()
```

Out[12]:

```
0    False
1    False
2    False
3     True
4     True
5     True
6     True
7     True
Name: X, dtype: bool
```

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 [13]:

```
# Write your code here
f= df4['Sales'].groupby(df4['Seasons'])
print(f.min())
print(f.max())
print(f.mean())
```

```
Seasons
Fall      56000
Spring    25000
Summer    67000
Winter    22000
Name: Sales, dtype: int64
Seasons
Fall      56000
Spring    35000
Summer    78000
Winter    60000
Name: Sales, dtype: int64
Seasons
Fall      56000.000000
Spring    29666.666667
Summer    70666.666667
Winter    36333.333333
Name: Sales, dtype: float64
```

In [ ]: