**Aggregation and Grouping**

*# import module***import** pandas **as** pd  
  
*# Creating our dataset*df = pd.DataFrame([[9, 4, 8, 9],  
 [8, 10, 7, 6],  
 [7, 6, 8, 5]],  
 columns=[**'Maths'**, **'English'**,  
 **'Science'**, **'History'**])  
  
*# display dataset*print(df)  
print(df.sum())  
print(df.describe())  
print(df.agg([**'sum'**, **'min'**, **'max'**]))

print(df.groupby(by=[**'Maths'**]))  
a = df.groupby(**'Maths'**)  
print(a.first())

**Output:**

Maths English Science History

0 9 4 8 9

1 8 10 7 6

2 7 6 8 5

Maths 24

English 20

Science 23

History 20

dtype: int64

Maths English Science History

count 3.0 3.000000 3.000000 3.000000

mean 8.0 6.666667 7.666667 6.666667

std 1.0 3.055050 0.577350 2.081666

min 7.0 4.000000 7.000000 5.000000

25% 7.5 5.000000 7.500000 5.500000

50% 8.0 6.000000 8.000000 6.000000

75% 8.5 8.000000 8.000000 7.500000

max 9.0 10.000000 8.000000 9.000000

Maths English Science History

sum 24 20 23 20

min 7 4 7 5

max 9 10 8 9

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001E297DA7BE0>

English Science History

Maths

7 6 8 5

8 10 7 6

9 4 8 9

**Time Series Operations**

**Example1:**

import pandas as pd

from datetime import datetime

import numpy as np

range\_date = pd.date\_range(start ='1/1/2019', end ='1/08/2019',

                                                   freq ='Min')

print(range\_date)

**Example2:**

import pandas as pd

from datetime import datetime

import numpy as np

range\_date = pd.date\_range(start ='1/1/2019', end ='1/08/2019',

                                                   freq ='Min')

print(type(range\_date[110]))

Example3

import pandas as pd

from datetime import datetime

import numpy as np

range\_date = pd.date\_range(start ='1/1/2019', end ='1/08/2019',

                                                  freq ='Min')

df = pd.DataFrame(range\_date, columns =['date'])

df['data'] = np.random.randint(0, 100, size =(len(range\_date)))

string\_data = [str(x) for x in range\_date]

print(string\_data[1:11])

**Pivot and melt function**

**import** pandas **as** pd  
  
data = {**"City"**: [**"Pune"**, **"Satara"**, **"Solapur"**], **"ID"**: [1, 2, 3], **"Fav"**: [**"1"**, **"3"**, **"10"**]}  
  
dataf = pd.DataFrame(data)  
print(**"Before melting.."**)  
print(dataf)  
  
melt\_df = pd.melt(dataf, id\_vars=[**"ID"**], value\_vars=[**"City"**, **"Fav"**])  
print(**"After melting.."**)  
print(melt\_df)

import pandas as pd

data = {"City": ["Pune", "Satara", "Solapur"], "ID": [1, 2, 3], "Fav": ["1", "3", "10"]}

dataf = pd.DataFrame(data)

print("Before melting..")

print(dataf)

melt\_df = pd.melt(dataf, id\_vars=["ID"], value\_vars=["City","Fav"], var\_name="Expression", value\_name="Value")

print("After melting..")

print(melt\_df)

unmelt = melt\_df.pivot(index='ID', columns='Expression')

print("Post unmelting..")

print(unmelt)

**Map, Filter, Reduce and Lambda Functions with Pandas dataframe**

*1)# Create a pandas DataFrame.***import** pandas **as** pd  
**import** numpy **as** np  
technologies= {  
 **'Fee'** :[22000,25000,23000,np.NaN,26000],  
 **'Duration'**:[**'30days'**,**'50days'**,**'30days'**,**'35days'**,**'40days'**]  
 }  
df = pd.DataFrame(technologies)  
print(df)  
*# Using Lambda Function*df[**'Fee'**] = df[**'Fee'**].map(**lambda** x: x - (x\*10/100))  
print(df)

**2)import** numpy **as** np  
**import** pandas **as** pd  
df = pd.DataFrame(np.array(([2, 3, 4], [5, 6, 7])),  
 index=[**'tiger'**, **'lion'**],  
 columns=[**'one'**, **'two'**, **'three'**])  
print(df.filter(items=[**'one'**, **'three'**]))  
*# select columns by regular expression*print(df.filter(regex=**'e$'**, axis=1))

**3)import** pandas **as** pd  
data = [11, 6, 7, 3, 28, 1]  
series = pd.Series(data)  
print(series)  
*# import functools module***import** functools  
  
*# using reduce operation to apply function on the series*product = functools.reduce(**lambda** x,y : x\*y,series)  
print(**"Product:"**,product,sep=**" "**)  
**import** pandas **as** pd  
  
*# creating and initializing a list*values = [[**'Rohan'**, 455], [**'Elvish'**, 250], [**'Deepak'**, 495],  
 [**'Soni'**, 400], [**'Radhika'**, 350], [**'Vansh'**, 450]]  
  
*# creating a pandas dataframe*df = pd.DataFrame(values, columns=[**'Name'**, **'Total\_Marks'**])  
  
*# Applying lambda function to find  
# percentage of 'Total\_Marks' column  
# using df.assign()*df = df.assign(Percentage=**lambda** x: (x[**'Total\_Marks'**] / 500 \* 100))  
  
*# displaying the data frame*print(df)