

FUNCTION APPLICATION

PROGRAMS

#Suppose we are having dataframe of following type

```
basic da
0 5000 500
1 7500 750
```

Now want to give 50% incentive over basic and da after raise of 5%
basic and da

```
import pandas as pd
import numpy as np
import math
# own functions
def adder(adder1,adder2):
    return adder1+(adder1*(adder2/100))
def incentive(adder1,adder2):
    return adder1/adder2
#Create a Dictionary of series
d = {
'basic':pd.Series([5000,7500]),
'da':pd.Series([500,750])}
df = pd.DataFrame(d)
df1=df.pipe(adder,5).pipe(incentive,2)
print (df1)
```

OUTPUT

```
basic da
0 2625.0 262.50
1 3937.5 393.75
```

#Suppose we are having dataframe following data

```
   name  gender  age
0  Mohak  Male   31
1  Nikhil Male   32
2  Freya  Female  19
```

Now we want to display mean age on gender basis with column heading in upper case

```
import pandas as pd
import numpy as np
import math
df = pd.DataFrame()
df['name'] = ['Mohak', 'Nikhil', 'Freya']
df['gender'] = ['Male', 'Male', 'Female']
df['age'] = [31, 32, 19]
print(df)
def mean_age_by_group(dataframe, col):
    return dataframe.groupby(col).mean()
def uppercase_column_name(dataframe):
    dataframe.columns = dataframe.columns.str.upper()
    return dataframe
print(df.pipe(mean_age_by_group, col='gender')
      .pipe(uppercase_column_name)
      )
```

OUTPUT

```
      AGE
gender
Female 19.0
Male   31.5
```

#Suppose we are having dataframe following data

	store	product	revenue	quantity
0	STORE1	TV	310000	8
1	STORE2	AC	480000	10
2	STORE3	FREEZE	190000	7

Now we want to find prices per store/product

```
import pandas as pd
import numpy as np
import math
df = pd.DataFrame()
df['store'] = ['STORE1', 'STORE2', 'STORE3']
df['product'] = ['TV', 'AC', 'FREEZE']
df['revenue'] = [310000, 480000, 190000]
df['quantity']=[8,10,7]
print(df)
print(df.groupby(['store', 'product']).pipe(lambda grp:
grp.revenue.sum()/grp.quantity.sum()))
```

OUTPUT

store	product	
STORE1	TV	38750.000000
STORE2	AC	48000.000000
STORE3	FREEZE	27142.857143

A dataframe dfW is given with following data:

	Age	Wage Rate
0	20	250
1	25	350
2	30	450
3	35	550

(a) Write a program using pipe() function to add 2 to each numeric column of dataframe dfW.

(b) Find the maximum value each column using apply() function.

(c) Find the row wise maximum value using apply() function.

```
(a) # Wageplus.py
import pandas as pd
dfW = pd.DataFrame({'Age' : [20, 25, 30, 35],
'Wage Rate' : [250, 350, 450, 550]},
columns = ['Age', 'Wage Rate'])
def Add_Two(Data, aValue):
return Data + aValue
print(dfW.pipe(Add_Two, 2))
```

```
(b) dfW.loc[:, 'Age':'Wage Rate'].apply(max, axis=0)
```

```
(c) dfW.loc[:, 'Age':'Wage Rate'].apply(max, axis=1)
```

A dataframe dfB is given with following data:

Itemno	ItemName	Color	Price
--------	----------	-------	-------

1	Ball Pen	Black	16.0
---	----------	-------	------

2	Pencil	Blue	5.0
---	--------	------	-----

3	Ball Pen	Green	15.0
---	----------	-------	------

4	Gel Pen	Green	20.0
---	---------	-------	------

Answer the following questions using groupby function (assume that the dataframe name is dfB):

(a) Display Color wise item and price of each ItemName category.

(b) Find the maximum price of each ItemName.

(c) Find the minimum price of each ItemName.

(c) Count the number of items in each ItemName category.

(a) `dfX = dfB.groupby(['ItemName', 'Color'])`
`dfX.first()`

(b) `dfB.groupby('ItemName').Price.max()`

(c) `dfB.groupby('ItemName').Price.min()`

(d) `dfB.groupby('ItemName')['Color'].apply(lambda x: x.count())`

#Suppose we are having dataframe following data

	0	1	2
0	1.0	2	NaN
1	NaN	3	4.0
2	1.0	2	3.0

Now count the number of missing values in each row in Pandas dataframe?

```
import pandas as pd
import numpy as np
import math
df = pd.DataFrame([[1,2,np.nan],
[ np.nan, 3, 4],
[1,2,3]])
print(df)
df['full_count'] = df.apply(lambda x: x.count(), axis=1)
print(df)
```