HARSH KASLIWAL

2003085

C21

EXPERIMENT 12

AIM: Programs Based on Pandas in Python

THEORY:

PANDAS

Pandas is a Python library for **data analysis**. Started by Wes McKinneyin 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries.

Dataframes

Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). AData frame is a two-dimensional data structure, i.e., data is aligned in a tabularfashion in rows and columns. Pandas DataFrame consists of three principal components, the **data**, **rows**, and **columns**.

Series

A series in Python is a kind of one-dimensional array of any data type that we specified in the pandas module. The only difference you can find was, each value in a Python pandas series is associated with the index. The default indexvalue of the Python pandas Series is from 0 to number – 1, or you can specifyyour own index values.

Task 1.

Write a pandas program to:
i) add, subtract, multiple and divide two pandas series
ii) compare the elements of the two Pandas Series.
iii) convert a dictionary to a Pandas series.
iv) convert a NumPy array to a Pandas series.

Code:

```
# arithmetic operations on panda seriesimport numpy as
np
import pandas as pd

ls1 = [1, 2, 3, 4, 5, 6]
ls2 = [5, 2, 6, 4, 30, 12]
ser1 = pd.Series(ls1)
ser2 = pd.Series(ls2)

print("ser1:\n", ser1)
print("ser2:\n", ser2)

# addition print("\nAddition: ")
ser = ser1 + ser2 print(ser)

# subtraction print("\nSubtraction:
")serSub = ser1 - ser2

# multiplication print("\nMultiplication:
")serMul = ser1 * ser2 print(serMul)

# Division print("\nDivision: ")
serDiv = ser1 / ser2
print(serDiv)

# Comparing both series print("\nComparing both
series: ")print(ser1 == ser2)

# Panda series from dictionary
print("\nConverting a Dictionary to Panda series: ")dictTem = {
        'a': 11,
```

```
        'b': 22,
        'c': 33,
        'd': 44,
        'e': 55,
        'f': 66,
        'g': 77,
}
serTem = pd.Series(dictTem)
print("Dict: ",dictTem) print("Series: ")
print(serTem)

# Converting a numpy array to pandas series print("\nConverting a numpy
array to pandas series:")
arr = np.array([1,2,3,5,6,9])print("Numpy
Array: ",arr) serAr = pd.Series(arr)
print("Series: ") print(serAr)
```

OUTPUT

```
ser1:
 0    1
1    2
2    3
3    4
4    5
5    6
dtype: int64
ser2:
 0     5
1     2
2     6
3     4
4    30
5    12
dtype: int64

Addition:
0     6
1     4
2     9
3     8
4    35
5    18
dtype: int64
```

```
Subtraction:
0    -4
1     0
2    -3
3     0
4   -25
5    -6
dtype: int64

Multiplication:
0      5
1      4
2     18
3     16
4    150
5     72
dtype: int64

Division:
0    0.200000
1    1.000000
2    0.500000
3    1.000000
4    0.166667
5    0.500000
dtype: float64
```

```
Comparing both series:
0    False
1     True
2    False
3     True
4    False
5    False
dtype: bool

Converting a Dictionary to Panda series:
Dict:  {'a': 11, 'b': 22, 'c': 33, 'd': 44, 'e': 55, 'f': 66, 'g': 77}
Series:
a    11
b    22
c    33
d    44
e    55
f    66
g    77
dtype: int64

Converting a numpy array to pandas series:
Numpy Array:  [1 2 3 5 6 9]
Series:
0    1
1    2
2    3
3    5
4    6
5    9
dtype: int32
```

Task 2:

Write a program to read csv file in a dataframe, replace missing values with any value, drop the row if all values are missing or contain null values.

CODE:

```
import pandas as pd
import numpy as np

# creating dataframes as df1 and df2
df1 = pd.DataFrame({'ID': [1, 2, 3, 5, 7, 8],
                    'Name': ['Sam', 'John', 'Bridge',
                             'Edge', 'Joe', 'Hope']})

df2 = pd.DataFrame({'ID': [1, 2, 4, 5, 6, 8, 9],
                    'Marks': [67, 92, 75, 83, 69, 56, 81]})

# merging df1 and df2 by ID
# i.e. the rows with common ID's get
# Operations on CSV files
import numpy as np
import pandas as pd

flights = pd.read_csv("flights.csv")
print(flights)

# Checking for missing values
print("\n",flights.notnull())
```

```
# filling missing values
# flights["flight Num"].fillna("NoNumber",inplace=True)
# flights.replace(to_replace=np.nan,value="NoValue")
flights = flights.fillna("NoValue")

# dropping the row if there is a NULL Value
flights = flights.dropna()

print("\n",flights)
```

Csv File:

	A	B	C	D	E	F	G
1	flight Num	flight Nam	origin				
2		Lufthansa	Berlin				
3	LF896	Vistara	Singapore				
4	VS692		Mumbai				
5	AI9648	Qatar Airways					
6							
7							
8							
9							

OutPut:

Replacing Null Values with random values

```
  flight Num    flight Name     origin
0        NaN      Lufthansa     Berlin
1      LF896        Vistara  Singapore
2      VS692            NaN     Mumbai
3     AI9648  Qatar Airways        NaN

    flight Num  flight Name  origin
0        False         True    True
1         True         True    True
2         True        False    True
3         True         True   False

   flight Num    flight Name     origin
0    NoValue      Lufthansa     Berlin
1      LF896        Vistara  Singapore
2      VS692        NoValue     Mumbai
3     AI9648  Qatar Airways    NoValue
```

Droping Row if it will have null values

```
  flight Num    flight Name     origin
0        NaN      Lufthansa     Berlin
1      LF896        Vistara  Singapore
2      VS692            NaN     Mumbai
3     AI9648  Qatar Airways        NaN

    flight Num  flight Name  origin
0        False         True    True
1         True         True    True
2         True        False    True
3         True         True   False

   flight Num flight Name     origin
1      LF896     Vistara  Singapore
```

Task 3:

Write a program to demonstrate merging of Frames:

i) on the basis of id
ii) using how

Code:

```
# merged i.e. {1,2,5,8}
print("Merging based on ID")
df = pd.merge(df1, df2, on="ID")
print(df)
# merging df1 and df2 by ID
# i.e. the rows with common ID's get merged
# with all the ID's of left dataframe i.e. df1
# and NaN for columns of df2 where ID do not match
print("Merging with how")
df = pd.merge(df1, df2, on="ID", how="left")
print("\n",df)
```

OUTPUT:

```
Merging based on ID
   ID  Name  Marks
0   1   Sam     67
1   2  John     92
2   5  Edge     83
3   8  Hope     56
Merging with how

    ID    Name  Marks
0   1     Sam   67.0
1   2    John   92.0
2   3  Bridge    NaN
3   5    Edge   83.0
4   7     Joe    NaN
5   8    Hope   56.0
```