DATA CLEANING AND PREPARATION

Ex:4a

DATE:

AIM: To do data cleaning and preparation using dataframe.

DESCRIPTION:

1) Handling missing data using pandas dataframe
2) Drop missing values using dropna()
3) Fill the missing values using fillna()
4) Replace the missing values using replace() with a scalar value. It is equivalent of fillna()
5) Through the isnull() C notnull() we can identify the NaN as Boolean result
6) Fill the missing values from forward and backward values through pad/ffill and bfill/backfill

PROGRAM:

```
#REINDEXING AND DROP
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(4,3),index=['a','b','d','f'],
        columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f'])
print("ORIGINAL       DATAFRAME       with
NaN\n",df)                    print("DROPPED
DATAFRAME\n",df.dropna()) OUTPUT:
```

```
ORIGINAL DATAFRAME with NaN
     one     two   three
a 0.332702 1.096137 0.767823
```

```
b -0.932717 1.148707 0.782676
c     NaN      NaN      NaN
d -1.401756 0.189671 0.214360
e     NaN      NaN      NaN
f -1.435522 0.430696 0.204984
DROPPED DATAFRAME
      one      two    three
a 0.332702 1.096137 0.767823
b -0.932717 1.148707 0.782676
d -1.401756 0.189671 0.214360
f -1.435522 0.430696 0.204984
```

```
#REPLACING NAN WITH FILLNA
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(4,3),index=['a','b','d','f'],
          columns=['one','two','three'])
df=df.reindex(['a','b','c'])
print("ORIGINAL DATAFRAME with NaN\n",df)
print("NaN REPLACED with 'o'")
print(df.fillna(5))
```

OUTPUT:

```
ORIGINAL DATAFRAME with NaN
      one      two    three
a 0.610345 2.468019 1.241989
b -0.315126 2.875800 0.539626
```

```
c     NaN     NaN     NaN
 NaN REPLACED with 'o'
    one     two   three
 a 0.610345 2.468019 1.241989
b -0.315126 2.875800 0.539626
 c 5.000000 5.000000 5.000000
```

```
# IS NULL FUNCTION
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(4,3),index=['a','b','d','f'],
          columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f'])
print("ORIGINAL DATAFRAME with NaN\n",df)
print("NaN WITH TRUE FILL")
print(df['one'].isnull())
```

OUTPUT:

```
ORIGINAL DATAFRAME with NaN
      one     two   three
a 1.392908 0.655801 -0.712033
      b -0.118810 -0.203114
              1.788137
c     NaN     NaN     NaN
 d 0.581012 0.192225 1.506077
e     NaN     NaN     NaN
f 0.945205 1.818632 -0.028508
NaN WITH TRUE FILL
```

```
a   False
b   False
c    True
d   False
e    True
f   False
Name: one, dtype: bool
```

```
#BACK FILL C FORWARD FILL
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(4,3),index=['a','b','d','f'],
          columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f'])
print("ORIGINAL DATAFRAME with NaN\n",df)
print("NaN FILLED WITH BACKFILL")
print(df.fillna(method='bfill'))
print("NaN FILLED WITH
forwardFILL")
print(df.fillna(method='ffill'))
```

OUTPUT:

```
ORIGINAL DATAFRAME with NaN
      one     two   three
a -0.072276 1.470502 -1.656771
b -0.787754 0.743290 -1.181253
c     NaN     NaN     NaN
d 0.103451 0.614430 0.768039
```

```
e     NaN      NaN      NaN
f 0.012438 0.127895 0.288324
NaN FILLED WITH BACKFILL
      one      two    three
a -0.072276 1.470502 -1.656771
b -0.787754 0.743290 -1.181253
c 0.103451 0.614430 0.768039
d 0.103451 0.614430 0.768039
e 0.012438 0.127895 0.288324
f 0.012438 0.127895 0.288324
NaN FILLED WITH
      forwardFILL one      two
                three
a -0.072276 1.470502 -1.656771
b -0.787754 0.743290 -1.181253
c -0.787754 0.743290 -1.181253
d 0.103451 0.614430 0.768039
e 0.103451 0.614430 0.768039
f 0.012438 0.127895 0.288324
```

```
#REPLACE
df=pd.DataFrame({'one':[10,20,80,40,50],
         'two':[60,70,80,0,10]})
print("ORIGINAL DATAFRAME\n",df)
print("PRINT DATAFRAME WITH REPLACED VALUES")
print(df.replace({10:5,80:50}))
```

OUTPUT:

```
ORIGINAL DATAFRAME
   one two
0 10   60
1 20   70
2 80   80
3 40    0
4 50   10
PRINT DATAFRAME WITH REPLACED VALUES
  one two
0  5 60
1 20   70
2 50   50
3 40    0
4 50    5
```

RESULT: Data cleaning process is done and preparation is done via dataframes.