EXP No:5

DATA CLEANING AND PREPARATION

[ ]:
```
import pandas as pd
import numpy as np
a=pd.DataFrame(np.random.
  ↪randn(5,3),index=['a','c','e','f','h'],columns=['One','Two','Three'])
print(a)
a=a.reindex(['a','b','c','d','e','f','g','h'])
print(a)
```

```
        One       Two     Three
a  0.599844  1.445202 -1.223061
c  0.292735  0.865780 -0.471783
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
h  0.110259 -1.922232  0.253248
        One       Two     Three
a  0.599844  1.445202 -1.223061
b       NaN       NaN       NaN
c  0.292735  0.865780 -0.471783
d       NaN       NaN       NaN
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
g       NaN       NaN       NaN
h  0.110259 -1.922232  0.253248
```


[ ]:
```
a1=a
print(a.dropna())
```

```
        One       Two     Three
a  0.599844  1.445202 -1.223061
c  0.292735  0.865780 -0.471783
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
h  0.110259 -1.922232  0.253248
```

[ ]:
```
a2=a1
print(a1.fillna(0))
```

```
        One       Two     Three
a  0.599844  1.445202 -1.223061
b  0.000000  0.000000  0.000000
c  0.292735  0.865780 -0.471783
d  0.000000  0.000000  0.000000
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
g  0.000000  0.000000  0.000000
h  0.110259 -1.922232  0.253248
```

[ ]:
```
a3=a2
print(a2.fillna(method='pad'))
```

```
        One       Two     Three
a  0.599844  1.445202 -1.223061
b  0.599844  1.445202 -1.223061
c  0.292735  0.865780 -0.471783
d  0.292735  0.865780 -0.471783
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
g -1.123648  1.540607 -0.347031
h  0.110259 -1.922232  0.253248
```

```
<ipython-input-5-b927352de9fc>:2: FutureWarning: DataFrame.fillna with 'method'
is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill()
instead.
  print(a2.fillna(method='pad'))
```

[ ]:
```
a4=a3
print(a3.fillna(method='bfill'))
```

```
        One       Two     Three
a  0.599844  1.445202 -1.223061
b  0.292735  0.865780 -0.471783
c  0.292735  0.865780 -0.471783
```



```
d -0.014980  1.944982  1.774031
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
g  0.110259 -1.922232  0.253248
h  0.110259 -1.922232  0.253248
```

```
<ipython-input-6-1f554bed7946>:2: FutureWarning: DataFrame.fillna with 'method'
is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill()
instead.
  print(a3.fillna(method='bfill'))
```

[ ]:
```
a5=a4
print(a4.bfill())
```

```
        One       Two     Three
a  0.599844  1.445202 -1.223061
b  0.292735  0.865780 -0.471783
c  0.292735  0.865780 -0.471783
d -0.014980  1.944982  1.774031
e -0.014980  1.944982  1.774031
f -1.123648  1.540607 -0.347031
g  0.110259 -1.922232  0.253248
h  0.110259 -1.922232  0.253248
```

[ ]:
```
print(a['One'].isnull())
print(a['One'].notnull())
```

```
a    False
b     True
c    False
d     True
e    False
f    False
g     True
h    False
Name: One, dtype: bool
a     True
b    False
c     True
d    False
e     True
f     True
g    False
h     True
Name: One, dtype: bool
```

[ ]:
```
b=pd.
  ↪DataFrame([[11,'a'],[12,'b'],[13,'c'],[14,'d'],[15,'e'],[103,'f'],[101,'g'],[18,'h']],colu
```



```
print(b)
```

```
   Age Name
0   11    a
1   12    b
2   13    c
3   14    d
4   15    e
5  103    f
6  101    g
7   18    h
```

[ ]:
```
print(b.replace({103:16,101:17}))
```

```
   Age Name
0   11    a
1   12    b
2   13    c
3   14    d
4   15    e
5   16    f
6   17    g
7   18    h
```

EXP NO:6

DATA WRANGLING

[ ]:
```
import pandas as pd
import numpy as np
d1={"name":["salini","Mary","Johncy"],"age":[40,60,38]}
d2={"Qualified":[True,False,True]}
df1=pd.DataFrame(d1)
df2=pd.DataFrame(d2)
nd=df1.join(df2)
print(nd)
```

```
     name  age  Qualified
0  salini   40       True
1    Mary   60      False
2  Johncy   38       True
```

[ ]:
```
df=pd.DataFrame({"team":['A','B','C','D'],"points":[88,89,99,98],"assist":
 ↪[17,14,16,12],"rebounds":[22,21,25,38]})
print(df)
df1=pd.melt(df,id_vars=['team'],value_vars=['points','assist','rebounds'])
print(df1)
```

```
  team  points  assist  rebounds
```



```
0    A      88      17        22
1    B      89      14        21
2    C      99      16        25
3    D      98      12        38
   team  variable  value
0     A    points     88
1     B    points     89
2     C    points     99
3     D    points     98
4     A    assist     17
5     B    assist     14
6     C    assist     16
7     D    assist     12
8     A  rebounds     22
9     B  rebounds     21
10    C  rebounds     25
11    D  rebounds     38
```

Data Grouping Function

[ ]:
```
b=pd.
 ↪DataFrame([['Hen',80],['Hen',100],['Parrot',40],['Parrot',30],['Finges',10],['Finges',15]],
print(b)
```

```
     Name  Speed
0     Hen     80
1     Hen    100
2  Parrot     40
3  Parrot     30
4  Finges     10
5  Finges     15
```

[ ]:
```
b.groupby(['Name']).mean()
```

[ ]:
```
         Speed
Name
Finges   12.5
Hen      90.0
Parrot   35.0
```

[ ]:
```
b.groupby(['Name']).sum()
```

[ ]:
```
         Speed
Name
Finges      25
Hen        180
Parrot      70
```


[ ]:
```
b.groupby(['Name']).count()
```

[ ]:
```
         Speed
Name
Finges       2
Hen          2
Parrot       2
```

[ ]:
```
b.groupby(['Name']).first()
```

[ ]:
```
         Speed
Name
Finges      10
Hen         80
Parrot      40
```

[ ]:
```
.groupby(['Name']).last()
```

[ ]:
```
         Speed
Name
Finges      15
Hen        100
Parrot      30
```

[ ]:
```
b.groupby(['Name']).size()
```

[ ]:
```
Name
Finges    2
Hen       2
Parrot    2
dtype: int64
```