ADVANCED PYTHON PROGRAMMING LAB

PANDAS (ADDITIONAL EXPERIMENTS)

(COVERS EXERCISES 8,9 & 10)

Creating Pandas data frame from random array values and reindexing data frame.

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f','h'],columns=['one', 'two', 'three'])
print("Resultant data frame")
print("----------------------")
print(df)
print("Resultant data frame after reindexing")
print("----------------------")
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print(df)
```

#0UTPUT

```
Resultant data frame
----------------------
      one      two    three
a  1.044886 -1.303467  0.532721
c -0.149762  1.336974 -0.897988
e -0.665634  1.020039 -0.104622
f  0.246851 -0.050827  0.188627
h -0.101682 -0.015673 -0.579404
Resultant data frame after reindexing
----------------------
      one      two    three
a  1.044886 -1.303467  0.532721
b      NaN      NaN      NaN
c -0.149762  1.336974 -0.897988
d      NaN      NaN      NaN
e -0.665634  1.020039 -0.104622
f  0.246851 -0.050827  0.188627
g      NaN      NaN      NaN
h -0.101682 -0.015673 -0.579404
```

Checking for Missing Values and Data Cleansing/ Filling Missing data

a) Creating Pandas data frame from random array values, re indexing data frame and to check NaN values from respective column.

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f','h'],columns=['one', 'two', 'three'])
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print("Test NaN values for Column 'one'")
print(df['one'].isnull())
```

#0UTPUT

```
Test NaN values for Column 'one'
a    False
b     True
c    False
d     True
e    False
f    False
g     True
h    False
Name: one, dtype: bool
```

b) Creating Pandas data frame from random array values,re indexing data frame and **replacing** NaN values with scalar value 0.

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(3, 3), index=['a', 'c', 'e'],columns=['one','two', 'three'])
df = df.reindex(['a', 'b', 'c'])
print("Before Replacing NaN")
print("-----------------------")
print(df)
print ("After NaN replaced with '0':")
print("------------------------------")
print(df.fillna(0))
```

#0UTPUT

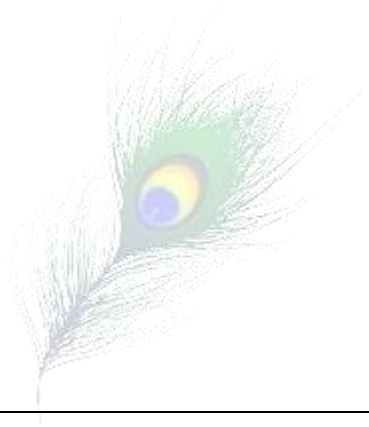
```
Before Replacing NaN
------------------------
      one      two    three
a  1.435278 -0.012853 -0.237361
b      NaN      NaN      NaN
c -0.474834 -0.530156 -1.608475
After NaN replaced with '0':
------------------------------
      one      two    three
a  1.435278 -0.012853 -0.237361
b  0.000000  0.000000  0.000000
c -0.474834 -0.530156 -1.608475
```

c) Creating Pandas data frame from random array values, re indexing data frame and **replacing** NaN values with Forward filling.

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f','h'],columns=['one', 'two', 'three'])
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
```

```
print("Before filling")
print("-------------------------")
print(df)
print("After Filling")
print("----------------------------")
print(df.fillna(method='pad'))
```

```
#0UTPUT
Before filling
-------------------------
      one      two    three
a  2.379086  1.677030  0.239215
b       NaN       NaN       NaN
c  0.250916  1.184965  1.589383
d       NaN       NaN       NaN
e -0.199118  1.256023  0.791248
f  1.402298  0.398270  1.267459
g       NaN       NaN       NaN
h  0.928289 -0.510150 -0.516183
After Filling
----------------------------
      one      two    three
a  2.379086  1.677030  0.239215
b  2.379086  1.677030  0.239215
c  0.250916  1.184965  1.589383
d  0.250916  1.184965  1.589383
e -0.199118  1.256023  0.791248
f  1.402298  0.398270  1.267459
g  1.402298  0.398270  1.267459
h  0.928289 -0.510150 -0.516183
```



d) Creating Pandas data frame from random array values, re indexing data frame and **replacing** NaN values with backward filling.

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f','h'],columns=['one', 'two', 'three'])
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print("Before filling")
print("-------------------------")
print(df)
print("After Filling")
print("----------------------------")
print(df.fillna(method='bfill'))
```

#0UTPUT

Before filling

```
-------------------------
      one      two    three
a -1.835436  2.400621 -0.624968
b      NaN      NaN      NaN
c -1.795986  2.152084 -0.040556
d      NaN      NaN      NaN
e  2.015179 -1.540970 -2.223884
f  0.446519 -0.457721  1.444836
g      NaN      NaN      NaN
h -1.148582  0.949568  0.786975
After Filling
-----------------------------
      one      two    three
a -1.835436  2.400621 -0.624968
b -1.795986  2.152084 -0.040556
c -1.795986  2.152084 -0.040556
d  2.015179 -1.540970 -2.223884
e  2.015179 -1.540970 -2.223884
f  0.446519 -0.457721  1.444836
g -1.148582  0.949568  0.786975
h -1.148582  0.949568  0.786975
```

Dropping Missing values

a) Creating Pandas data frame from random array values, re indexing data frame and **dropping** NaN values (Missing values).

```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(5, 3), index=['a', 'c', 'e', 'f','h'],columns=['one', 'two', 'three'])
df = df.reindex(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'])
print("Before Dropping")
print("-----------------------------")
print(df)
print("After Dropping")
print("-----------------------------")
print(df.dropna())
```

#0UTPUT

```
Before Dropping
-----------------------------
      one      two    three
a -1.201111  0.971163 -1.033865
b      NaN      NaN      NaN
c -1.020736  1.468244 -0.114225
d      NaN      NaN      NaN
e -1.241028 -0.476123 -0.480864
f  0.278258 -1.851143 -0.056096
g      NaN      NaN      NaN
h  0.009966  0.245584  0.731419
After Dropping
-----------------------------
```

```
      one      two    three
a -1.201111  0.971163 -1.033865
c -1.020736  1.468244 -0.114225
e -1.241028 -0.476123 -0.480864
f  0.278258 -1.851143 -0.056096
h  0.009966  0.245584  0.731419
```

Demonstrate CSV files data to demonstrate data processing using Pandas

Note: i) Download any .csv files from dataset repositories.
ii)Upload into Jupyter IDE.

a) Reading csv files

```
import pandas as pd
df = pd.read_csv("reserap.csv")
print(df)
```

#0UTPUT
All records from dataset displayed as output

To do the following experiment download data set from

https://data.gov.in/search?title=soil%20data%20set

Data set Name: Daily data of Soil Moisture during April 2023

b) Reading specific rows from csv files
c) Reading specific rows
d) Reading specific columns
e) Reading specific rows and columns

```
import pandas as pd
df = pd.read_csv("soilapril.csv")
print(df)
print("After Dropping NA Coulmn values")
print("----------------------------------------")
df1=df.dropna()
print(df1)
print("Reading Specific Rows")
print("-------------------------------------------")
df2=df1.loc[df['Avg_smlvl_at15cm'] >= 20]
print(df2)
print("Reading Specific row")
```

```
print("-----------------------------------------")
df3=df1.iloc[2]
print(df3)
print("Reading Selected Columns")
print("-----------------------------------------------------------")
df4=pd.read_csv("soilapril.csv",usecols=['State','District','Avg_smlvl_at15cm'])
df5=df4.dropna()
print(df5)
print("Reading Selected Rows and Columns")
print("-------------------------------------------------------------")
df6=pd.read_csv("soilapril.csv",usecols=['District','Avg_smlvl_at15cm'])
df7=df6.dropna()
df8=df7.loc[df['State'] == 'Andhra Pradesh']
print(df8)
```

#0UTPUT

Respective records from dataset displayed as output.