120A3051
Shreya Idate
Batch: E3

EXPERIMENT NO.1

Aim : Data preparation using NumPy and Pandas.

a. Derive an index field and add it to the data set.

b. Find out the missing values.

Theory :

What is Pandas?

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Why Use Pandas?

Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

Read CSV Files

A simple way to store big data sets is to use CSV files (comma separated files). CSV files contains plain text and is a well know format that can be read by everyone including Pandas. In our examples we will be using a CSV file called 'data.csv'.

Pandas DataFrame: set_index() function

The set_index() function is used to set the DataFrame index using existing columns. Set the DataFrame index (row labels) using one or more existing columns or arrays of the correct length. The index can replace the existing index or expand on it.

Pandas DataFrame fillna() Method

The fillna() method replaces the NULL values with a specified value. The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

Parameters

The axis, method, inplace, limit, downcast parameters are keyword arguments.

Parameter	Value	Description
value	Number String Dictionary Series DataFrame	Required, Specifies the value to replace the NULL values with. This can also be values for the entire row or column.
method	'backfill' 'bfill' 'pad' 'ffill' None	Optional, default None'. Specifies the method to use when replacing
axis	0 1 'index' 'columns'	Optional, default 0. The axis to fill the NULL values along
inplace	True False	Optional, default False. If True: the replacing is done on the current DataFrame. If False: returns a copy where the replacing is done.
limit	Number None	Optional, default None. Specifies the maximum number of NULL values to fill (if method is specified)
downcast	Dictionary None	Optional, a dictionary of values to fill for specific data types

Program :-

Part a:

Reading a .csv file.

tail(): Returns the first n rows.

head():Returns the last n rows.

```
import pandas as pd

original_data = pd.read_csv(r"C:\Users\exam\Desktop\120A3051\nba-2.csv")
data=original_data.set_index("Name")
#data = pd.read_csv(r"C:\Users\exam\Desktop\120A3051\nba-2.csv",index_col="Name")
data.head()
```

[12] ✓ 0.4s

Name	Team	Number	Position	Age	Height	Weight	College	Salary
Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0

```
data.tail()
```

[4] ✓ 0.8s

Name	Team	Number	Position	Age	Height	Weight	College	Salary
Shelvin Mack	Utah Jazz	8.0	PG	26.0	6-3	203.0	Butler	2433333.0
Raul Neto	Utah Jazz	25.0	PG	24.0	6-1	179.0	NaN	900000.0
Tibor Pleiss	Utah Jazz	21.0	C	26.0	7-3	256.0	NaN	2900000.0
Jeff Withey	Utah Jazz	24.0	C	26.0	7-0	231.0	Kansas	947276.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Select multiple columns, we have to pass a list of columns in an indexing operator:

```
first = data[["Team","Number","Age"]]
#first
first.tail()
```

[5] ✓ 0.5s

Name	Team	Number	Age
Shelvin Mack	Utah Jazz	8.0	26.0
Raul Neto	Utah Jazz	25.0	24.0
Tibor Pleiss	Utah Jazz	21.0	26.0
Jeff Withey	Utah Jazz	24.0	26.0
NaN	NaN	NaN	NaN

Indexing a DataFrame using .loc[ ] :

```
df = data.loc["Kobe Bryant"]
df
```

[8] ✓ 0.2s

```
Team          Los Angeles Lakers
Number                      24.0
Position                      SF
Age                         37.0
Height                       6-6
Weight                     212.0
College                      NaN
Salary                25000000.0
Name: Kobe Bryant, dtype: object
```

Selecting multiple rows

In order to select multiple rows, we put all the row labels in a list and pass that to .loc function.

```
df = data.loc[["Arron Afflalo","Kobe Bryant"]]
df
```

[9] ✓ 0.3s

Name	Team	Number	Position	Age	Height	Weight	College	Salary
Arron Afflalo	New York Knicks	4.0	SG	30.0	6-5	210.0	UCLA	8000000.0
Kobe Bryant	Los Angeles Lakers	24.0	SF	37.0	6-6	212.0	NaN	25000000.0

In order to select two rows and three columns:

```
df = data.loc[["Arron Afflalo","Kobe Bryant"],["Team","Number","Position"]]
df
```

[10] ✓ 0.3s

Name	Team	Number	Position
Arron Afflalo	New York Knicks	4.0	SG
Kobe Bryant	Los Angeles Lakers	24.0	SF

Retrieving two rows and two columns by iloc method

```
df2 = original_data
df2=df2.iloc[[100,105,109],[0,1,2,3]]
df2
```

[20] ✓ 0.3s

	Name	Team	Number	Position
100	Chris Paul	Los Angeles Clippers	3.0	PG
105	C.J. Wilcox	Los Angeles Clippers	30.0	SG
109	Kobe Bryant	Los Angeles Lakers	24.0	SF

Part b:

Checking for missing values using isnull()

The isnull() method returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False.

```
import pandas as pd
import numpy as np

new_dict = {'Score 1':[100, 90, np.nan, 95],
        'Score 2': [30, 45, 56, np.nan],
        'Score 3':[np.nan, 40, 80, 98]}

# creating a dataframe from list
df = pd.DataFrame(new_dict)

# using isnull() function
df.isnull()
```

[3] ✓ 0.1s

	Score 1	Score 2	Score 3
0	False	False	True
1	False	False	False
2	True	False	False
3	False	True	False

Printing only the rows having Gender = NULL

```
data = pd.read_csv(r"C:\Users\dell\Desktop\SIES\SEM 6\AIDS Lab\DBs\employees.csv")

bool_series = pd.notnull(data["Gender"])

data[bool_series]
```

[6] ✓ 0.2s Python

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	08-06-1993	12:42 PM	97308	6.945	True	Marketing
1	Thomas	Male	3/31/1996	6:53 AM	61933	4.170	True	NaN
2	Maria	Female	4/23/1993	11:17 AM	130590	11.858	False	Finance
3	Jerry	Male	03-04-2005	1:00 PM	138705	9.340	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	101004	1.389	True	Client Services
...	...	...	...	...	...	...	...	...
994	George	Male	6/21/2013	5:47 PM	98874	4.479	True	Marketing
996	Phillip	Male	1/31/1984	6:30 AM	42392	19.675	False	Finance
997	Russell	Male	5/20/2013	12:39 PM	96914	1.421	False	Product
998	Larry	Male	4/20/2013	4:45 PM	60500	11.985	False	Business Development
999	Albert	Male	5/15/2012	6:24 PM	129949	10.169	True	Sales

855 rows × 8 columns

Filling null values with a single value

```
import pandas as pd

import numpy as np

new_dict = {'Score 1':[100, 90, np.nan, 95],
        'Score 2': [30, 45, 56, np.nan],
        'Score 3':[np.nan, 40, 80, 98]}

df = pd.DataFrame(new_dict)

# filling missing value using fillna()
df.fillna('SIES')
```

[7] ✓ 0.1s

	Score 1	Score 2	Score 3
0	100.0	30.0	SIES
1	90.0	45.0	40.0
2	SIES	56.0	80.0
3	95.0	SIES	98.0

Filling null values with the previous ones

```
import pandas as pd

import numpy as np

new_dict = {'Score 1':[100, 90, np.nan, 95],
        'Score 2': [30, 45, 56, np.nan],
        'Score 3':[np.nan, 40, 80, 98]}

df = pd.DataFrame(new_dict)

# filling a missing value with previous ones

df.fillna(method ='ffill')
```

[8] ✓ 0.1s

	Score 1	Score 2	Score 3
0	100.0	30.0	NaN
1	90.0	45.0	40.0
2	90.0	56.0	80.0
3	95.0	56.0	98.0

Filling null value with the next ones

```
import pandas as pd

import numpy as np

new_dict = {'Score 1':[100, 90, np.nan, 95],
        'Score 2': [30, 45, 56, np.nan],
        'Score 3':[np.nan, 40, 80, 98]}

df = pd.DataFrame(new_dict)

# filling null value using fillna() function

df1=df.fillna(method ='bfill')
df1.fillna(method ='ffill')
```

[9] ✓ 0.1s

	Score 1	Score 2	Score 3
0	100.0	30.0	40.0
1	90.0	45.0	40.0
2	95.0	56.0	80.0
3	95.0	56.0	98.0

Filling null values in CSV File

```
data = pd.read_csv(r"C:\Users\dell\Desktop\SIES\SEM 6\AIDS Lab\DBs\employees.csv")

data[10:25]
```

[10] ✓ 0.1s

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
10	Louise	Female	08-12-1980	9:01 AM	63241	15.132	True	NaN
11	Julie	Female	10/26/1997	3:19 PM	102508	12.637	True	Legal
12	Brandon	Male	12-01-1980	1:08 AM	112807	17.492	True	Human Resources
13	Gary	Male	1/27/2008	11:40 PM	109831	5.831	False	Sales
14	Kimberly	Female	1/14/1999	7:13 AM	41426	14.543	True	Finance
15	Lillian	Female	06-05-2016	6:09 AM	59414	1.256	False	Product
16	Jeremy	Male	9/21/2010	5:56 AM	90370	7.369	False	Human Resources
17	Shawn	Male	12-07-1986	7:45 PM	111737	6.414	False	Product
18	Diana	Female	10/23/1981	10:27 AM	132940	19.082	False	Client Services
19	Donna	Female	7/22/2010	3:48 AM	81014	1.894	False	Product
20	Lois	NaN	4/22/1995	7:18 PM	64714	4.934	True	Legal
21	Matthew	Male	09-05-1995	2:12 AM	100612	13.645	False	Marketing
22	Joshua	NaN	03-08-2012	1:58 AM	90816	18.816	True	Client Services
23	NaN	Male	6/14/2012	4:19 PM	125792	5.042	NaN	NaN
24	John	Male	07-01-1992	10:08 PM	97950	13.873	False	Client Services

```
data = pd.read_csv(r"C:\Users\dell\Desktop\SIES\SEM 6\AIDS Lab\DBs\employees.csv")

data["Gender"].fillna("No Gender", inplace = True)

data[10:25]
```

[11] ✓ 0.2s

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
10	Louise	Female	08-12-1980	9:01 AM	63241	15.132	True	NaN
11	Julie	Female	10/26/1997	3:19 PM	102508	12.637	True	Legal
12	Brandon	Male	12-01-1980	1:08 AM	112807	17.492	True	Human Resources
13	Gary	Male	1/27/2008	11:40 PM	109831	5.831	False	Sales
14	Kimberly	Female	1/14/1999	7:13 AM	41426	14.543	True	Finance
15	Lillian	Female	06-05-2016	6:09 AM	59414	1.256	False	Product
16	Jeremy	Male	9/21/2010	5:56 AM	90370	7.369	False	Human Resources
17	Shawn	Male	12-07-1986	7:45 PM	111737	6.414	False	Product
18	Diana	Female	10/23/1981	10:27 AM	132940	19.082	False	Client Services
19	Donna	Female	7/22/2010	3:48 AM	81014	1.894	False	Product
20	Lois	No Gender	4/22/1995	7:18 PM	64714	4.934	True	Legal
21	Matthew	Male	09-05-1995	2:12 AM	100612	13.645	False	Marketing
22	Joshua	No Gender	03-08-2012	1:58 AM	90816	18.816	True	Client Services
23	NaN	Male	6/14/2012	4:19 PM	125792	5.042	NaN	NaN
24	John	Male	07-01-1992	10:08 PM	97950	13.873	False	Client Services

Filling a null values using replace() method: replace the all Nan value in the data frame with specific value.

```
data = pd.read_csv(r"C:\Users\dell\Desktop\SIES\SEM 6\AIDS Lab\DBs\employees.csv")

data.replace(to_replace = np.nan, value = -99)
```

[12] ✓ 0.1s

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	08-06-1993	12:42 PM	97308	6.945	True	Marketing
1	Thomas	Male	3/31/1996	6:53 AM	61933	4.170	True	-99
2	Maria	Female	4/23/1993	11:17 AM	130590	11.858	False	Finance
3	Jerry	Male	03-04-2005	1:00 PM	138705	9.340	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	101004	1.389	True	Client Services
...	...	...	...	...	...	...	...	...
995	Henry	-99	11/23/2014	6:09 AM	132483	16.655	False	Distribution
996	Phillip	Male	1/31/1984	6:30 AM	42392	19.675	False	Finance
997	Russell	Male	5/20/2013	12:39 PM	96914	1.421	False	Product
998	Larry	Male	4/20/2013	4:45 PM	60500	11.985	False	Business Development
999	Albert	Male	5/15/2012	6:24 PM	129949	10.169	True	Sales

Using interpolate() function to fill the missing values using linear method.

```
import pandas as pd

import numpy as np

new_dict = pd.DataFrame({"A":[12, 4, 5, None, 1],
                         "B":[None, 2, 54, 3, None],
                         "C":[20, 16, None, 3, 8],
                         "D":[14, 3, None, None, 6]})


df = pd.DataFrame(new_dict)

# to interpolate the missing values
df.interpolate(method ='linear', limit_direction ='backward')
```

[14] ✓ 0.1s

	A	B	C	D
0	12.0	2.0	20.0	14.0
1	4.0	2.0	16.0	3.0
2	5.0	54.0	9.5	4.0
3	3.0	3.0	3.0	5.0
4	1.0	NaN	8.0	6.0

Drop rows with at least one Nan value (Null value) using dropna() function.

The dropna() method removes the rows that contains NULL values. The dropna() method returns a new DataFrame object unless the inplace parameter is set to True, in that case the dropna() method does the removing in the original DataFrame instead.

```
import pandas as pd

import numpy as np

new_dict = {'Score 1':[100, 90, np.nan, 95],
        'Score 2': [30, np.nan, 45, 56],
        'Score 3':[52, 40, 80, 98],
        'Score 4':[np.nan, np.nan, np.nan, 65]}

df = pd.DataFrame(new_dict)


# using dropna() function
df.dropna()
```

[23] ✓ 0.1s

	Score 1	Score 2	Score 3	Score 4
3	95.0	56.0	98	65.0

Dropping Rows with at least 1 null value in CSV file

```
data = pd.read_csv(r"C:\Users\dell\Desktop\SIES\SEM 6\AIDS Lab\DBs\employees.csv")

new_data = data.dropna(axis = 0, how ='any')

new_data
```

[24] ✓ 0.1s

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	08-06-1993	12:42 PM	97308	6.945	True	Marketing
2	Maria	Female	4/23/1993	11:17 AM	130590	11.858	False	Finance
3	Jerry	Male	03-04-2005	1:00 PM	138705	9.340	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	101004	1.389	True	Client Services
5	Dennis	Male	4/18/1987	1:35 AM	115163	10.125	False	Legal
...	...	...	...	...	...	...	...	...
994	George	Male	6/21/2013	5:47 PM	98874	4.479	True	Marketing
996	Phillip	Male	1/31/1984	6:30 AM	42392	19.675	False	Finance
997	Russell	Male	5/20/2013	12:39 PM	96914	1.421	False	Product
998	Larry	Male	4/20/2013	4:45 PM	60500	11.985	False	Business Development
999	Albert	Male	5/15/2012	6:24 PM	129949	10.169	True	Sales

764 rows × 8 columns

Conclusion : Successfully learned about data preparation using NumPy and Pandas