Name: Mayuri Shridatta Yerande
Class: D15B
Roll No: 70

EXPERIMENT - 1

AIM: Data preparation using NumPy and Pandas

TO DO:

- Load data in Pandas.
- Description of the dataset.
- Drop columns that aren't useful
- Drop rows with maximum missing values.
- Take care of missing data.
- Create dummy variables.
- Find out outliers (manually)
- standardization and normalization of columns

ABOUT DATASET:

Link of our dataset:
https://www.stats.govt.nz/information-releases/electronic-card-transactions-november-2022/

Our Dataset - The electronic card transactions (ECT) series covers all debit, credit, and charge card transactions with New Zealand-based merchants. It can be used to indicate changes in consumer spending and economic activity.

The columns in our dataset incluse:
Series_reference,Period,Data_value,Suppressed,STATUS,UNITS,Magnitude,subject,group,series_title_1, series_title_2,series_title_3, series_title_4, series_title_5

THEORY:

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

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.

Advantages:-
Fast and efficient for manipulating and analyzing data.
Data from different file objects can be loaded.
Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
Data set merging and joining.
Flexible reshaping and pivoting of data sets
Provides time-series functionality.
Powerful group by functionality for performing split-apply-combine operations on data sets.

Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.

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

Pandas is usually imported under the **pd** alias

```
import pandas as pd
```

$$z_i = \frac{x_i - \bar{x}}{\sigma}$$

Standardization:- In statistics and machine learning, data standardization is a process of converting data to z-score values based on the mean and standard deviation of the data. The resulting standardized value shows the number of standard deviations the raw value is away from the mean. Basically each value of a given feature of a dataset will be converted to a representative number of standard deviations that it's away from the mean of the feature.

Data Normalization: Data Normalization could also be a typical practice in machine learning which consists of transforming numeric columns to a standard scale. In machine learning, some feature values differ from others multiple times. The features with higher values will dominate the learning process.

Steps Needed for Normalization:-

Here, we will apply some techniques to normalize the data and discuss these with the help of examples. For this, let's understand the steps needed for data normalization with Pandas.

1. Import Library (Pandas)
2. Import / Load / Create data.
3. Use the technique to normalize the data.

SCREENSHOTS:

1. Load data in Pandas.

```
#Loading the Dataset
import pandas as pd
data = pd.read_csv("dataset.csv")
data
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Seri
0	ECTA.S19A1	2001.03	2462.5	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	
1	ECTA.S19A1	2002.03	17177.2	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	
2	ECTA.S19A1	2003.03	22530.5	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	
3	ECTA.S19A1	2004.03	28005.1	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06)	Total values - Electronic card transactions	Actual	RTS total industries	NaN	

```
[ ] #Printing Number of Columns and Rows
    rows = len(data.axes[0])
    cols = len(data.axes[1])
    print(rows)
    print(cols)
```

```
19129
14
```

2. Description of the dataset.

```
data.describe()
```

	Period	Data_value	Magnitude	Series_title_4	Series_title_5	col	Dummy
count	19129.000000	1.780800e+04	19129.00000	0.0	0.0	17075.000000	17075.000000
mean	2011.693308	1.554829e+07	4.21057	NaN	NaN	0.815637	0.815637
std	6.225121	8.558495e+07	2.74498	NaN	NaN	0.387791	0.387791
min	2000.010000	-5.130000e+01	0.00000	NaN	NaN	0.000000	0.000000
25%	2006.110000	1.861750e+02	0.00000	NaN	NaN	1.000000	1.000000
50%	2012.020000	1.218700e+03	6.00000	NaN	NaN	1.000000	1.000000
75%	2017.060000	4.335650e+03	6.00000	NaN	NaN	1.000000	1.000000
max	2022.110000	1.874441e+09	6.00000	NaN	NaN	1.000000	1.000000

3. Drop columns and rows that aren't useful

```
#Dropping the columns that are not useful
#We dropped the "STATUS" column
data.drop(['STATUS'], axis=1)
```

	Series_reference	Period	Data_value	Suppressed	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Series_title
0	ECTA.S19A1	2001.03	2462.5	NaN	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	N
1	ECTA.S19A1	2002.03	17177.2	NaN	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	N
2	ECTA.S19A1	2003.03	22530.5	NaN	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	N

```
#Dropping the rows that are not useful
update_data = data.dropna(subset=['Data_value']).copy()
update_data
```

Initially the number of rows was 19129. Now there are 17808 rows.

0	ECTA.S19A1	2001.03	2462.5	1	F	Dollars	6	Transactions (ANZSIC06) - ECT	Electronic card transactions A/...	Actual	RTS total industries	NaN	
1	ECTA.S19A1	2002.03	17177.2	2	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN
2	ECTA.S19A1	2003.03	22530.5	3	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN
3	ECTA.S19A1	2004.03	28005.1	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN
4	ECTA.S19A1	2005.03	30629.6	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19124	ECTQ.S4AXP	2021.09	34.8	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19125	ECTQ.S4AXP	2021.12	33.3	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19126	ECTQ.S4AXP	2022.03	33.7	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19127	ECTQ.S4AXP	2022.06	33.5	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19128	ECTQ.S4AXP	2022.09	33.2	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN

17808 rows × 14 columns

✓ 0s completed at 9:21 PM

4. Drop rows with maximum missing values.

The maximum number of missing values is 4 Thus we will drop the rows which have more than 4 missing values and keep only the ones which have missing values less than 4.

```
#Dropping rows with maximum missing values
data3 =data[data.isnull().sum(axis=1)<4]
data3
```

							(ANZSIC06) - ECT	transactions A/...					
1	ECTA.S19A1	2002.03	17177.2	2	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN
2	ECTA.S19A1	2003.03	22530.5	3	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN
400	ECTA.S4A1V	2004.03	53.8	NaN	F	Dollars	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Mean transaction value	Value ($)	NaN	NaN
401	ECTA.S4A1V	2005.03	54.0	NaN	F	Dollars	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Mean transaction value	Value ($)	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19124	ECTQ.S4AXP	2021.09	34.8	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19125	ECTQ.S4AXP	2021.12	33.3	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19126	ECTQ.S4AXP	2022.03	33.7	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19127	ECTQ.S4AXP	2022.06	33.5	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN
19128	ECTQ.S4AXP	2022.09	33.2	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Electronic card transactions by mean and propo...	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN

3498 rows × 14 columns

5. Take care of missing data.

+ Code + Text

```
#Check missing Values
data.isnull()
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Series_title_4
0	False	False	False	True	False	False	False	False	False	False	False	True	True
1	False	False	False	True	False	False	False	False	False	False	False	True	True
2	False	False	False	True	False	False	False	False	False	False	False	True	True
3	False	False	False	True	False	False	False	False	False	False	False	True	True
4	False	False	False	True	False	False	False	False	False	False	False	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
19124	False	False	False	True	False	False	False	False	False	False	False	False	True
19125	False	False	False	True	False	False	False	False	False	False	False	False	True
19126	False	False	False	True	False	False	False	False	False	False	False	False	True
19127	False	False	False	True	False	False	False	False	False	False	False	False	True

```
#Total count of null values in each column
data.isnull().sum()
```

```
Series_reference        0
Period                  0
Data_value              0
Suppressed          18929
STATUS                  0
UNITS                   0
Magnitude               0
Subject                 0
Group                   0
Series_title_1          0
Series_title_2          0
Series_title_3      15364
Series_title_4      19129
Series_title_5      19129
dtype: int64
```

```
[49] #Total count of null values in each column
row = data.isnull().sum(axis=1)
row
```

```
0        4
1        4
2        4
3        4
4        4
       ..
19124    3
19125    3
19126    3
19127    3
19128    3
Length: 19129, dtype: int64
```

```
#Total count of null values
data.isnull().sum().sum()
```

```
72551
```

```
[51] #Replacing null values with 0
     data2 = data.fillna(value = 0)
     data2
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Seri
0	ECTA.S19A1	2001.03	2462.5	0	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	0	
1	ECTA.S19A1	2002.03	17177.2	0	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	0	
2	ECTA.S19A1	2003.03	22530.5	0	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	0	
3	ECTA.S19A1	2004.03	28005.1	0	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	0	

- **Replacing Null values with previous values in the column**

Before:

```
data.head(30)
```

								- ECT	A/...			
20	ECTA.S19A1	2021.03	68203.1	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN
21	ECTA.S19A1	2022.03	71690.6	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN
22	ECTA.S19A2	2001.03	NaN	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries	NaN
23	ECTA.S19A2	2002.03	NaN	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06)	Total values - Electronic card transactions	Actual	RTS core industries	NaN

After:

```
[31] #Filling null values with previous values
     data4 = data.fillna(method = 'pad')
     data4.head(30)
     #edit dataset
```

									- ECT	A/...			
21	ECTA.S19A1	2022.03	71690.6	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN
22	ECTA.S19A2	2001.03	71690.6	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries	NaN
23	ECTA.S19A2	2002.03	71690.6	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries	NaN
24	ECTA.S19A2	2003.03	71690.6	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06)	Total values - Electronic card transactions	Actual	RTS core industries	NaN

- **Replacing Null values with values of previous column**

```
[18] data5 = data.fillna(method = 'pad', axis=1)
     data5
     #Suppressed column took the values of Data_Value column ie Previous column
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Seri
0	ECTA.S19A1	2001.03	2462.5	2462.5	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	RTS total industries	
1	ECTA.S19A1	2002.03	17177.2	17177.2	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	RTS total industries	
2	ECTA.S19A1	2003.03	22530.5	22530.5	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	RTS total industries	
3	ECTA.S19A1	2004.03	28005.1	28005.1	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	RTS total industries	

- **Replacing Null values with values of next column**

```
data5 = data.fillna(method = 'bfill', axis=1)
data5
#Suppressed column took the values of STATUS column ie. Next column
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Seri
0	ECTA.S19A1	2001.03	2462.5	F	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	
1	ECTA.S19A1	2002.03	17177.2	F	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	
2	ECTA.S19A1	2003.03	22530.5	F	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	
3	ECTA.S19A1	2004.03	28005.1	F	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	

- **Interpolate function:-**

Before:-

```
data.head(30)
```

								- ECT	A/...			
20	ECTA.S19A1	2021.03	68203.1	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN
21	ECTA.S19A1	2022.03	71690.6	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN
22	ECTA.S19A2	2001.03	NaN	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries	NaN
23	ECTA.S19A2	2002.03	NaN	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06)	Total values - Electronic card transactions	Actual	RTS core industries	NaN

After:-

```
data['Data_value'] = data['Data_value'].interpolate(method='linear')
data.head(30)
```

									- ECT	A/...		
21	ECTA.S19A1	2022.03	71690.600	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries
22	ECTA.S19A2	2001.03	59928.375	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries
23	ECTA.S19A2	2002.03	48166.150	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries
24	ECTA.S19A2	2003.03	36403.925	NaN	C	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS core industries
25	ECTA.S19A2	2004.03	24641.700	NaN	F	Dollars	6	Electronic Card Transactions	Total values - Electronic card	Actual	RTS core

6. Create dummy variables.

```
#Creating dummy variables
data['Dummy'] = data.UNITS.map({'Dollars': 1,'Percent': 0})
data
#If UNITS is Dollars then Dummy column prints 1 and if its Percent then it prints 0
```

_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Series_title_4	Series_title_5	col	Dummy
2462.5	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN	1.0	1.0
17177.2	NaN	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	NaN	NaN	1.0	1.0
22530.5	NaN	F	Dollars	6	Electronic Card Transactions	Total values - Electronic card	Actual	RTS total industries	NaN	NaN	NaN	1.0	1.0

7. Find out outliers (manually)

```
#Finding outliers
min_thresold,max_thresold = data.Data_value.quantile([0.001,0.999])
max_thresold,min_thresold
```

```
(1313897520.391054, -12.473700000000001)
```

```
[ ] #These are the outliers
    data[data.Data_value<min_thresold]
```

	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Seri
1030	ECTM.S19A1AC	2020.04	-47.5	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Totals - Electronic card transactions by divis...	Actual	RTS total industries	Percentage change from same period previous year	
1047	ECTM.S19A1AC	2021.09	-14.9	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Totals - Electronic card transactions by divis...	Actual	RTS total industries	Percentage change from same period previous year	
1535	ECTM.S19A2AC	2020.04	-43.7	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Totals - Electronic card transactions by divis...	Actual	RTS core industries	Percentage change from same period previous year	
1552	ECTM.S19A2AC	2021.09	-14.2	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06) - ECT	Totals - Electronic card transactions by divis...	Actual	RTS core industries	Percentage change from same period previous year	

8. standardization and normalization of columns

Standardization:

```
[19] #Standardization
     from sklearn.preprocessing import StandardScaler
     scaler = StandardScaler()
     standardized_data = scaler.fit_transform(data[['Period','Data_value']])
     print(standardized_data)
```

```
[[-1.71299269 -0.18164714]
 [-1.55234905 -0.1814752 ]
 [-1.3917054  -0.18141265]
 ...
 [ 1.66052389 -0.18167552]
 [ 1.6653432  -0.18167552]
 [ 1.67016251 -0.18167552]]
```

```
[23] #coverting above array output to a dataframe
     standardized_df = pd.DataFrame(standardized_data, columns=data[['Period','Data_value']].columns)
     standardized_df
```

	Period	Data_value
0	-1.712993	-0.181647
1	-1.552349	-0.181475
2	-1.391705	-0.181413
3	-1.231062	-0.181349
4	-1.070418	-0.181318
...	...	...
19124	1.509519	-0.181676
19125	1.514338	-0.181676
19126	1.660524	-0.181676
19127	1.665343	-0.181676
19128	1.670163	-0.181676

19129 rows × 2 columns

Normalization:

```
[44] #Normalization
     X_Data = data[['Period','Data_value']]
     #Y_Data = data['Data_value']
```

```
from sklearn.preprocessing import MinMaxScaler
X_Data.head(10)
```

	Period	Data_value
0	2001.03	2462.5
1	2002.03	17177.2
2	2003.03	22530.5
3	2004.03	28005.1
4	2005.03	30629.6
5	2006.03	33317.4
6	2007.03	36422.0
7	2008.03	39198.0
8	2009.03	40629.4
9	2010.03	41815.4

```
[46] scalar = MinMaxScaler()
     scalar.fit(X_Data)
     new_data = scalar.transform(X_Data)
     new_data
```

```
array([[4.61538462e-02, 1.34109297e-06],
       [9.14027149e-02, 9.19127226e-06],
       [1.36651584e-01, 1.20472166e-05],
       ...,
       [9.96380090e-01, 4.53468463e-08],
       [9.97737557e-01, 4.52401479e-08],
       [9.99095023e-01, 4.50801002e-08]])
```

CONCLUSION: In this practical, We learnt how to deal with dataset in different aspects. We learnt how to take care of the missing data such as replacing it with 0,or other column/row. Sometimes, We need an extra column in our dataset thus we learnt to create dummy data. We learnt about standardization and normalization which makes our data precise. Thus we successfully performed all the necessary operations on our dataset.