EXPERIMENT - 3

Aim:

1. Data Cleaning - removing missing values(demonstrate removing and replacing Null values)
2. Data Cleaning - removing noisy values(Binning technique), removing outliersInterquartile Range Method,Boxplot.
3. Data Transformation - converting numerical attributes to categorical and vice versa/ one hot encoding.
4. Data Transformation - data normalization(Z- score transformation).
5. Data Reduction - reducing the number of rows by attribute-oriented induction or numerosity reduction.

Theory:

What is data cleaning?

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct.

How to Treat Outliers?

There are several ways to treat outliers in a dataset, depending on the nature of the outliers and the problem being solved. Here are some of the most common ways of treating outlier values.

● Trimming: It excludes the outlier values from our analysis. By applying this technique, our data becomes thin when more outliers are present in the dataset. Its main advantage is its fastest nature.

● Capping: In this technique, we cap our outliers data and make the limit i.e, above a particular value or less than that value, all the values will be considered as outliers, and the number of outliers in the dataset gives that capping number.

● Discretization: In this technique, by making the groups, we include the outliers in a particular group and force them to behave in the same manner as those of other points in that group. This technique is also known as Binning.

How to Detect Outliers?

● For Normal Distributions Use empirical relations of Normal distribution. The data points that fall below mean-3*(sigma) or above mean+3*(sigma) are outliers, where mean and sigma are the average value and standard deviation of a particular column.
● For Skewed Distributions Use Interquartile Range (IQR) proximity rule. The data points that fall below Q1 – 1.5 IQR or above the third quartile Q3 + 1.5 IQR are outliers, where Q1 and Q3 are the 25th and 75th percentile of the dataset, respectively. IQR represents the interquartile range and is given by Q3 – Q1.
● For Other Distributions Use a percentile-based approach. For Example, data points that are far from the 99% percentile and less than 1 percentile are considered an outlier.

Data transformation:-
Data transformation in data mining refers to the process of converting raw data into a format that is suitable for analysis and modeling. The goal of data transformation is to prepare the data for data mining so that it can be used to extract useful insights and knowledge. Data transformation typically involves several steps, including:
● Data cleaning: Removing or correcting errors, inconsistencies, and missing values in the data.
● Data integration: Combining data from multiple sources, such as databases and spreadsheets, into a single format.
● Data normalization: Scaling the data to a common range of values, such as between 0 and 1, to facilitate comparison and analysis.
● Data reduction: Reducing the dimensionality of the data by selecting a subset of relevant features or attributes.
● Data discretization: Converting continuous data into discrete categories or bins. Data aggregation: Combining data at different levels of granularity, such as by summing or averaging, to create new features or attributes. Data transformation is an important step in the data mining process as it helps to ensure that the data is in a format that is suitable for analysis and modeling, and that it is free of errors and inconsistencies

Normalization
Data normalization involves converting all data variables into a given range. Techniques that are used for normalization are:
Min-Max Normalization: ● This transforms the original data linearly. ● Suppose that: min_A is the minima and max_A is the maxima of an attribute, P ● Where v is the value you want to plot in the new range. ● v’ is the new value you get after normalizing the old value.

Z-Score Normalization

- In z-score normalization (or zero-mean normalization) the values of an attribute (A), are normalized based on the mean of A and its standard deviation
- A value, v, of attribute A is normalized to v' by computing Decimal Scaling:
- It normalizes the values of an attribute by changing the position of their decimal points
- The number of points by which the decimal point is moved can be determined by the absolute maximum value of attribute A.
- A value, v, of attribute A is normalized to v' by computing
- where j is the smallest integer such that Max(|v'|) < 1. ● Suppose: Values of an attribute P varies from -99 to 99.
- The maximum absolute value of P is 99.
- For normalizing the values we divide the numbers by 100 (i.e., j = 2) or (number of integers in the largest number) so that values come out to be as 0.98, 0.97 and so on.

Numerosity reduction

Numerosity reduction is a technique used in data mining to reduce the number of data points in a dataset while still preserving the most important information. This can be beneficial in situations where the dataset is too large to be processed efficiently, or where the dataset contains a large amount of irrelevant or redundant data points. There are several different numerosity reduction techniques that can be used in data mining, including:

● Data Sampling ● Data Aggregation: ● Data Generalization:● Data Compression

Implementation:

Step 1: Importing libraries

```
[17] import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline
```

Step 2: Read the dataset

```
#Loading the Dataset
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	Series_title_4	Series_
0	ECTA.S19A1	2001.03	2462.5	1	F	Dollars	6	Electronic Card Transactions (ANZSIC06) - ECT	Total values - Electronic card transactions A/...	Actual	RTS total industries	NaN	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	
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	
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	
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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
19124	ECTQ.S4AXP	2021.09	34.8	NaN	F	Percent	0	Electronic Card Transactions (ANZSIC06)	Electronic card transactions by mean	Actual	Debit card usage as a proportion of total ECT ...	Proportion (%)	NaN	

Step 3: Describe the dataset and check the datatype

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19129 entries, 0 to 19128
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Series_reference  19129 non-null  object
 1   Period            19129 non-null  float64
 2   Data_value        17808 non-null  float64
 3   Suppressed        203 non-null    object
 4   STATUS            19129 non-null  object
 5   UNITS             19129 non-null  object
 6   Magnitude         19129 non-null  int64
 7   Subject           19129 non-null  object
 8   Group             19129 non-null  object
 9   Series_title_1    19129 non-null  object
 10  Series_title_2    19129 non-null  object
 11  Series_title_3    3765 non-null   object
 12  Series_title_4    0 non-null      float64
 13  Series_title_5    0 non-null      float64
dtypes: float64(4), int64(1), object(9)
memory usage: 2.0+ MB
```

```
data.describe()
```

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

Step 4: Finding the null values

```
data.isnull().sum()
```

```
Series_reference        0
Period                  0
Data_value           1321
Suppressed          18926
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
```

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

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

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

```
73869
```

Step 5: Replacing null values with its mode

```
[24] print(data['Data_value'].mode())
```

```
0    0.8
Name: Data_value, dtype: float64
```

```
[25] data['Data_value'].fillna(data['Data_value'].mode()[0], inplace=True)
```

```
data['Data_value'].isnull().sum()
```

```
0
```

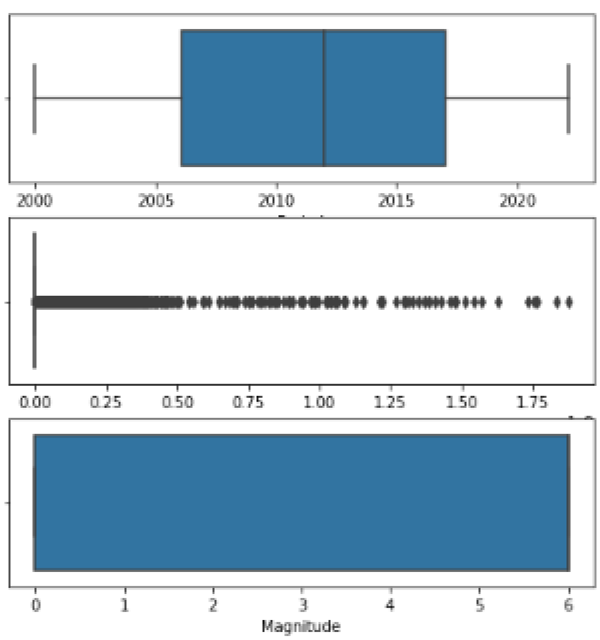
Step 6: Dropping Null values

```
data.dropna(inplace=True)
data.isnull().sum()
```

```
Series_reference    0.0
Period              0.0
Data_value          0.0
Suppressed          0.0
STATUS              0.0
UNITS               0.0
Magnitude           0.0
Subject             0.0
Group               0.0
Series_title_1      0.0
Series_title_2      0.0
Series_title_3      0.0
Series_title_4      0.0
Series_title_5      0.0
Dummy               0.0
dtype: float64
```

Step 7: We plot boxplot to see the outliers and the distribution of the data.

```
cols = ['Period','Data_value','Magnitude']
fig, ax = plt.subplots (3,1, figsize = (7, 7))
for i in range(len(cols)):
  sns.boxplot(data, x=cols[i], ax = ax[i])
plt.show()
```



Step 8: Convert categorical data to numerical data for UNITS column by creating a new column named “Dummy” which shows the converted numeric values of the UNITS column.

```
[30] 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
```

eries_title_2	Series_title_3	Series_title_4	Series_title_5	Dummy
RTS total industries	NaN	NaN	NaN	1.0
RTS total industries	NaN	NaN	NaN	1.0
RTS total industries	NaN	NaN	NaN	1.0
RTS total industries	NaN	NaN	NaN	1.0
RTS total industries	NaN	NaN	NaN	1.0
...	...	...	...	...
)ebit card usage as a proportion of total ECT ...	Proportion (%)	NaN	NaN	0.0

Step 9: We performed data normalization on the “magnitude” column and brought it under 0 to 1 range.

```
[31] data['Magnitude'] = ((data['Magnitude'] -data['Magnitude'].mean())/data['Magnitude'].std())
     data['Magnitude'].describe

<bound method NDFrame.describe of 0        0.651892
1        0.651892
2        0.651892
3        0.651892
4        0.651892
           ...
19124   -1.533917
19125   -1.533917
19126   -1.533917
19127   -1.533917
19128   -1.533917
Name: Magnitude, Length: 19129, dtype: float64>
```

<u>**Conclusion:**</u> We successfully implemented data cleaning, took care of the null values, data transformation and data reduction using Z score transformation on our dataset