Lab 4.1

Data Reading, Data Description, Data Splitting, Outlier Identification, Missing Values Identification and Handling

Objective

To understand and implement essential data preprocessing techniques including data reading, statistical description, splitting datasets, identifying and handling outliers, and managing missing values in a dataset.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```python
df=pd.read_csv(r'C:\Users\PMLS\labreports\
LAB4\1_Orignal_AEP_hourly.csv' ,parse_dates=True)
```

```python
df.head()
```

```
              Datetime   AEP_MW
0  2004-12-31 01:00:00  13478.0
1  2004-12-31 02:00:00  12865.0
2  2004-12-31 03:00:00  12577.0
3  2004-12-31 04:00:00  12517.0
4  2004-12-31 05:00:00  12670.0
```

```python
df.tail()
```

```
                   Datetime   AEP_MW
121268  2018-01-01 20:00:00  21089.0
121269  2018-01-01 21:00:00  20999.0
121270  2018-01-01 22:00:00  20820.0
121271  2018-01-01 23:00:00  20415.0
121272  2018-01-02 00:00:00  19993.0
```

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121273 entries, 0 to 121272
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
```

```
 0   Datetime  121273 non-null  object 
 1   AEP_MW    121273 non-null  float64
dtypes: float64(1), object(1)
memory usage: 1.9+ MB
```

```
df.describe()
```

```
              AEP_MW
count  121273.000000
mean    15499.513717
std      2591.399065
min      9581.000000
25%     13630.000000
50%     15310.000000
75%     17200.000000
max     25695.000000
```

Dataset info

Oct 2004 to Aug 2018

```
# 2004
(24*31*2)+24*30
```

```
2208
```

```
# 2018
(24*31*4)+(24*30*2)+24*28+24*2+1
```

```
5137
```

```
total = ((24*31*2)+24*30) +
(24*31*7)*13+(24*30*4)*13+(24*28*1)*10+(24*29*1)*3  +
(24*31*4)+(24*30*2)+24*28+24*2+1
total
```

```
121297
```

```
print('Missing= ',  121297 - len(df))
```

```
Missing=  24
```

```
print('Data Points= ',  len(df))
print('Samples= ',      int(len(df)/24))
```

```
Data Points=  121273
Samples=  5053
```

```
# Drop Duplicates Except the First Occurrence
df.drop_duplicates(subset=['Datetime'], keep ='first',  inplace= True)
print('len = ',len(df))
```

```
len =  121269
```

```
df['Datetime']=pd.to_datetime(df['Datetime'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 121269 entries, 0 to 121272
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   Datetime  121269 non-null  datetime64[ns]
 1   AEP_MW    121269 non-null  float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 2.8 MB
```

```
df
```

```
                  Datetime   AEP_MW
0      2004-12-31 01:00:00  13478.0
1      2004-12-31 02:00:00  12865.0
2      2004-12-31 03:00:00  12577.0
3      2004-12-31 04:00:00  12517.0
4      2004-12-31 05:00:00  12670.0
...                    ...      ...
121268 2018-01-01 20:00:00  21089.0
121269 2018-01-01 21:00:00  20999.0
121270 2018-01-01 22:00:00  20820.0
121271 2018-01-01 23:00:00  20415.0
121272 2018-01-02 00:00:00  19993.0

[121269 rows x 2 columns]
```

```
df.set_index('Datetime', inplace=True)
df.head()
```

```
                      AEP_MW
Datetime
2004-12-31 01:00:00  13478.0
2004-12-31 02:00:00  12865.0
2004-12-31 03:00:00  12577.0
2004-12-31 04:00:00  12517.0
2004-12-31 05:00:00  12670.0
```

```
df.iloc[22:25]
```

```
                      AEP_MW
Datetime
2004-12-31 23:00:00  13478.0
2005-01-01 00:00:00  12892.0
2004-12-30 01:00:00  14097.0
```

```
df=df.sort_index(ascending=True)
df.iloc[22:25]
```

```
                      AEP_MW
Datetime                    
2004-10-01 23:00:00  14067.0
2004-10-02 00:00:00  13147.0
2004-10-02 01:00:00  12260.0
```

```
print(df.head(1))
print(df.tail(1))
```

```
                      AEP_MW
Datetime                    
2004-10-01 01:00:00  12379.0
             AEP_MW
Datetime           
2018-08-03  14809.0
```

```
missing_Timestamp=pd.date_range('2004-10-01','2018-08-
03',freq='H').difference(df.index)
print('\nNumber of Missing Timestamp= ',len(missing_Timestamp),'\n')
print(missing_Timestamp.to_list())
```

```

Number of Missing Timestamp=  28

[Timestamp('2004-10-01 00:00:00'), Timestamp('2004-10-31 02:00:00'),
Timestamp('2005-04-03 03:00:00'), Timestamp('2005-10-30 02:00:00'),
Timestamp('2006-04-02 03:00:00'), Timestamp('2006-10-29 02:00:00'),
Timestamp('2007-03-11 03:00:00'), Timestamp('2007-11-04 02:00:00'),
Timestamp('2008-03-09 03:00:00'), Timestamp('2008-11-02 02:00:00'),
Timestamp('2009-03-08 03:00:00'), Timestamp('2009-11-01 02:00:00'),
Timestamp('2010-03-14 03:00:00'), Timestamp('2010-11-07 02:00:00'),
Timestamp('2010-12-10 00:00:00'), Timestamp('2011-03-13 03:00:00'),
Timestamp('2011-11-06 02:00:00'), Timestamp('2012-03-11 03:00:00'),
Timestamp('2012-11-04 02:00:00'), Timestamp('2012-12-06 04:00:00'),
Timestamp('2013-03-10 03:00:00'), Timestamp('2013-11-03 02:00:00'),
Timestamp('2014-03-09 03:00:00'), Timestamp('2014-03-11 14:00:00'),
Timestamp('2015-03-08 03:00:00'), Timestamp('2016-03-13 03:00:00'),
Timestamp('2017-03-12 03:00:00'), Timestamp('2018-03-11 03:00:00')]
```

```
C:\Users\PMLS\AppData\Local\Temp\ipykernel_111848\4204030582.py:1:
FutureWarning: 'H' is deprecated and will be removed in a future
version, please use 'h' instead.
  missing_Timestamp=pd.date_range('2004-10-01','2018-08-
03',freq='H').difference(df.index)
```

```
df
```

```
                      AEP_MW
Datetime
2004-10-01 01:00:00  12379.0
2004-10-01 02:00:00  11935.0
2004-10-01 03:00:00  11692.0
2004-10-01 04:00:00  11597.0
2004-10-01 05:00:00  11681.0
...                      ...
2018-08-02 20:00:00  17673.0
2018-08-02 21:00:00  17303.0
2018-08-02 22:00:00  17001.0
2018-08-02 23:00:00  15964.0
2018-08-03 00:00:00  14809.0

[121269 rows x 1 columns]
```

```
df = df.resample('H').first().fillna(np.nan)  # Ensure index is in
hourly format

missing_Timestamp = pd.date_range('2004-10-01', '2018-08-03',
freq='H').difference(df.index)

print('\nNumber of Missing Timestamp = ', len(missing_Timestamp), '\
n')
print(missing_Timestamp.to_list())
```

```
Number of Missing Timestamp =  1

[Timestamp('2004-10-01 00:00:00')]

C:\Users\PMLS\AppData\Local\Temp\ipykernel_111848\1661212067.py:1:
FutureWarning: 'H' is deprecated and will be removed in a future
version, please use 'h' instead.
  df = df.resample('H').first().fillna(np.nan)  # Ensure index is in
hourly format
C:\Users\PMLS\AppData\Local\Temp\ipykernel_111848\1661212067.py:3:
FutureWarning: 'H' is deprecated and will be removed in a future
version, please use 'h' instead.
  missing_Timestamp = pd.date_range('2004-10-01', '2018-08-03',
freq='H').difference(df.index)
```

```
df.tail(1)
```

```
             AEP_MW
Datetime
2018-08-03  14809.0
```

```
df.reset_index(inplace=True)
len_list=df[df['AEP_MW'].isnull()].index.tolist()
df.iloc[len_list]
```

```
                  Datetime  AEP_MW
721    2004-10-31 02:00:00     NaN
4418   2005-04-03 03:00:00     NaN
9457   2005-10-30 02:00:00     NaN
13154  2006-04-02 03:00:00     NaN
18193  2006-10-29 02:00:00     NaN
21386  2007-03-11 03:00:00     NaN
27097  2007-11-04 02:00:00     NaN
30122  2008-03-09 03:00:00     NaN
35833  2008-11-02 02:00:00     NaN
38858  2009-03-08 03:00:00     NaN
44569  2009-11-01 02:00:00     NaN
47762  2010-03-14 03:00:00     NaN
53473  2010-11-07 02:00:00     NaN
54263  2010-12-10 00:00:00     NaN
56498  2011-03-13 03:00:00     NaN
62209  2011-11-06 02:00:00     NaN
65234  2012-03-11 03:00:00     NaN
70945  2012-11-04 02:00:00     NaN
71715  2012-12-06 04:00:00     NaN
73970  2013-03-10 03:00:00     NaN
79681  2013-11-03 02:00:00     NaN
82706  2014-03-09 03:00:00     NaN
82765  2014-03-11 14:00:00     NaN
91442  2015-03-08 03:00:00     NaN
100346 2016-03-13 03:00:00     NaN
109082 2017-03-12 03:00:00     NaN
117818 2018-03-11 03:00:00     NaN
```

```
print('values are missing at the following indexes:\n',len_list)
```

```
values are missing at the following indexes:
 [721, 4418, 9457, 13154, 18193, 21386, 27097, 30122, 35833, 38858,
44569, 47762, 53473, 54263, 56498, 62209, 65234, 70945, 71715, 73970,
79681, 82706, 82765, 91442, 100346, 109082, 117818]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121296 entries, 0 to 121295
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   Datetime  121296 non-null  datetime64[ns]
 1   AEP_MW    121269 non-null  float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 1.9 MB
```

```
df.head()
```

```
             Datetime   AEP_MW
0 2004-10-01 01:00:00  12379.0
1 2004-10-01 02:00:00  11935.0
2 2004-10-01 03:00:00  11692.0
3 2004-10-01 04:00:00  11597.0
4 2004-10-01 05:00:00  11681.0
```

```
df.isnull().sum()
```

```
Datetime     0
AEP_MW      27
dtype: int64
```

```
df['AEP_MW'] = df['AEP_MW'].interpolate()
```

```
df.isnull().sum()
```

```
Datetime    0
AEP_MW      0
dtype: int64
```

```
df.to_csv(r'C:\Users\PMLS\ML\
LAB4\2_Missing_Values_Filled.csv',index=False)
```

```
print('\tSummary of American Electric Power (AEP)')
print('\nStart Date: \n\t',df.head(1))
print('\nEnd Date: \n\t',  df.tail(1))
print('\nLengthBF: 121273')
print('\nLengthAF: ',len(df))
print('\nSamplesBF: 5053')
print('\nSamplesAF: ',(len(df)/24))
print('\nMissing Points: ',len(len_list))
print('\nMissing Points are at indices:\n ',len_list)
```

```
     Summary of American Electric Power (AEP)

Start Date: 
                    Datetime   AEP_MW
0 2004-10-01 01:00:00  12379.0

End Date: 
                Datetime   AEP_MW
121295 2018-08-03  14809.0

LengthBF: 121273

LengthAF:  121296

SamplesBF: 5053

SamplesAF:  5054.0
```

```
Missing Points:  27

Missing Points are at indices:
  [721, 4418, 9457, 13154, 18193, 21386, 27097, 30122, 35833, 38858,
44569, 47762, 53473, 54263, 56498, 62209, 65234, 70945, 71715, 73970,
79681, 82706, 82765, 91442, 100346, 109082, 117818]
```

```
df.iloc[721]
```

```
Datetime    2004-10-31 02:00:00
AEP_MW                  10875.5
Name: 721, dtype: object
```

```
df.isnull().sum()
```

```
Datetime    0
AEP_MW      0
dtype: int64
```

```
df=pd.read_csv('2_Missing_Values_Filled.csv')
df.describe()
```

```
              AEP_MW
count  121296.000000
mean    15499.150961
std      2591.377126
min      9581.000000
25%     13629.000000
50%     15309.000000
75%     17200.000000
max     25695.000000
```

```
15499+2591*2.8
```

```
22753.8
```