**Implement Programs For Time Series Data Cleaning, Loading, And Handling Time Series Data And Pre-Processing Techniques**

**EX.No:1 DATE: 25/01/2**

**AIM:**

To clean, preprocess, and visualize electricity production data, focusing on trend analysis and handling missing values.

**ALGORITHM:**

1. Import pandas and matplotlib.pyplot.
2. Load data from "Plant\_1\_Generation\_Data.csv" into DataFrame df.
3. Explore df's structure, missing values, statistics, and correlations.
4. Check and handle missing values (if any). Dataset is complete in this case.
5. Convert 'DATE\_TIME' to datetime, set as index, and sort df.
6. Resample df to daily frequency, aggregating relevant columns.
7. Create a line plot visualizing 'DC\_POWER' over time.
8. Create a histogram visualizing the 'DAILY\_YIELD' distribution.

**CODE:**

import pandas as pd

df = pd.read\_csv('Plant\_1\_Generation\_Data.csv')

print("DataFrame shape:", df.shape)

print("\nDataFrame info:")

print(df.info())

print("\nMissing values per column:")

print(df.isnull().sum())

print("\nDC Power and AC Power descriptive statistics:")

print(df[['DC\_POWER', 'AC\_POWER']].describe())

df['DATE\_TIME'] = pd.to\_datetime(df['DATE\_TIME'], infer\_datetime\_format=True)

print("\nCorrelation matrix for numerical variables:")

print(df.corr(numeric\_only=True))

missing\_values = df.isnull().sum()

print("Missing values in each column:\n", missing\_values)

if missing\_values.sum() > 0:

pass

else:

print("\nThe DataFrame is complete. No missing values found.")

if not pd.api.types.is\_datetime64\_any\_dtype(df['DATE\_TIME']):

df['DATE\_TIME'] = pd.to\_datetime(df['DATE\_TIME'])

df.set\_index('DATE\_TIME', inplace=True)

df.sort\_index(inplace=True)

df\_daily = df.resample('D').agg({'DC\_POWER': 'sum', 'AC\_POWER': 'sum', 'DAILY\_YIELD': 'sum', 'TOTAL\_YIELD': 'sum'})

from matplotlib import pyplot as plt

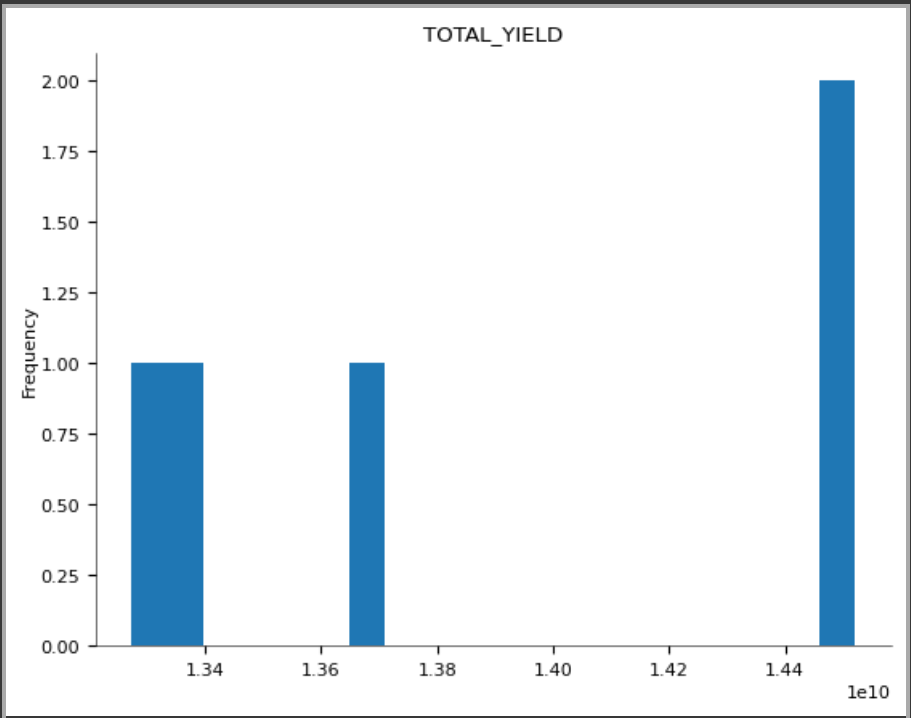
\_df\_7['DC\_POWER'].plot(kind='line', figsize=(8, 4), title='DC\_POWER')

plt.gca().spines[['top', 'right']].set\_visible(False)

\_df\_2['DAILY\_YIELD'].plot(kind='hist', bins=20, title='DAILY\_YIELD')

plt.gca().spines[['top', 'right',]].set\_visible(False)

**OUTPUT:**

****

**RESULT:**

Thus the program has been completed and verified successfully.