LAB ASSIGNMENT 2

Name:	Harsh Shah	**Semester:**	VII	**Division:**	6
Roll No.:	21BCP359	**Date:**	29-07-24	**Batch:**	G11
Aim:	Measurements of electric power consumption in one household with a one-minute sampling rate over a period of almost 4 years. Different electrical quantities and some sub-metering values are available.				

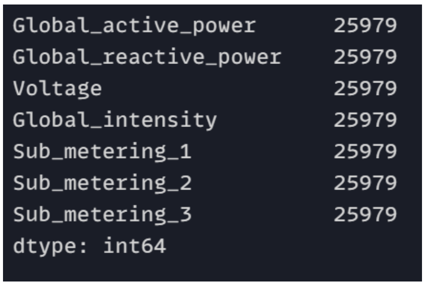
Objective

The objective of this lab assignment is to explore and analyse a dataset containing measurements of electric power consumption in a household over a period of almost 4 years. You will perform various data visualization tasks to gain insights into electrical quantities, sub-metering values, and overall trends.

Dataset: https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

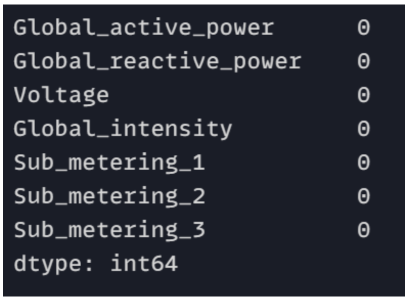
Task – 1: Load the data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv( ./data/household_power_consumption.txt', sep=';',  _values=['nan', '?'] )
df.index = pd.to_datetime(df['Date'] +' '+ df['Time'], dayfirst=True)
df.index.name = 'dt'
df = df.drop(columns = ['Date', 'Time'])

# Data Cleaning
df.isna().sum()
```



```
Global_active_power      25979
Global_reactive_power    25979
Voltage                  25979
Global_intensity         25979
Sub_metering_1           25979
Sub_metering_2           25979
Sub_metering_3           25979
dtype: int64
```

```
df.dropna(inplace=True)
df.isna().sum()
```



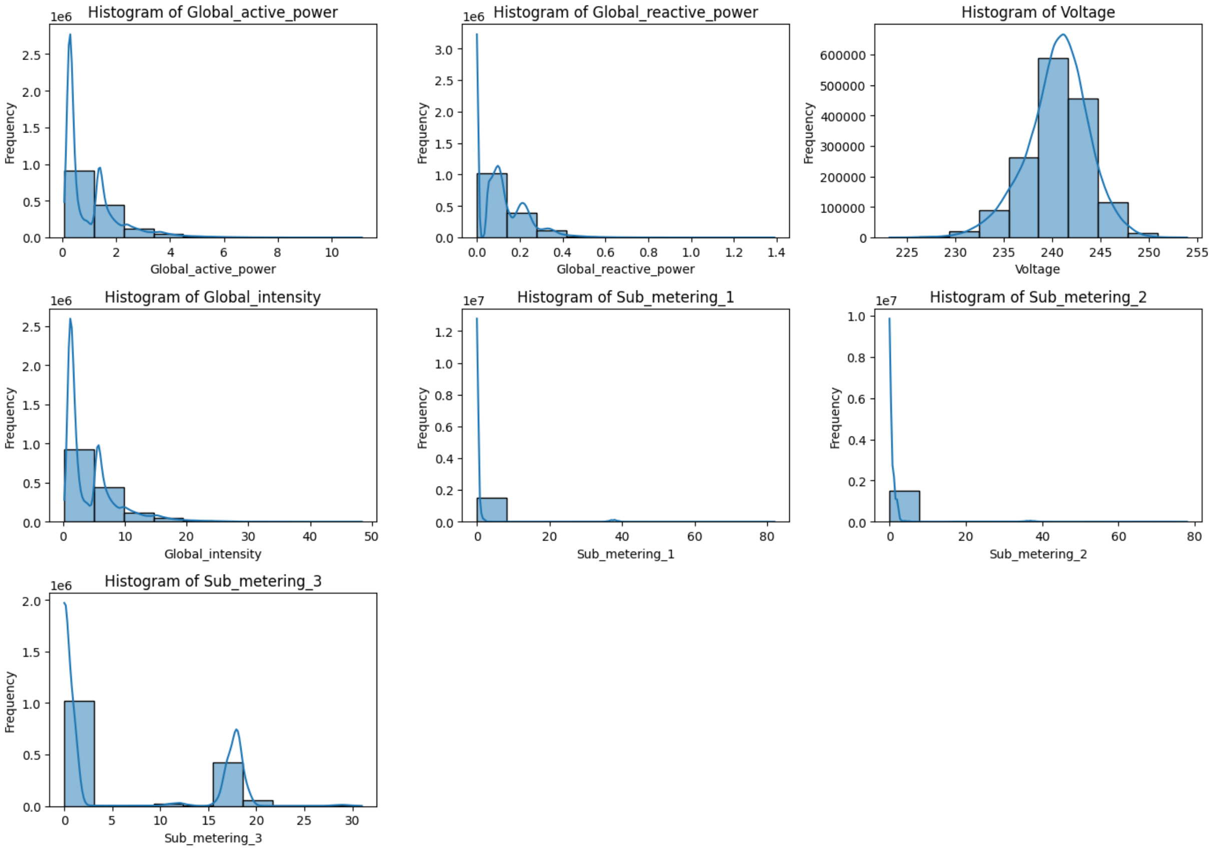
```
Global_active_power      0
Global_reactive_power    0
Voltage                  0
Global_intensity         0
Sub_metering_1           0
Sub_metering_2           0
Sub_metering_3           0
dtype: int64
```

Task – 2: Subset the data from the given dates (December 2006 and November 2009)

```
start_date = pd.Timestamp('2006-12-01')
end_date = pd.Timestamp('2009-11-30')
newdf = df.loc[start_date:end_date]
```

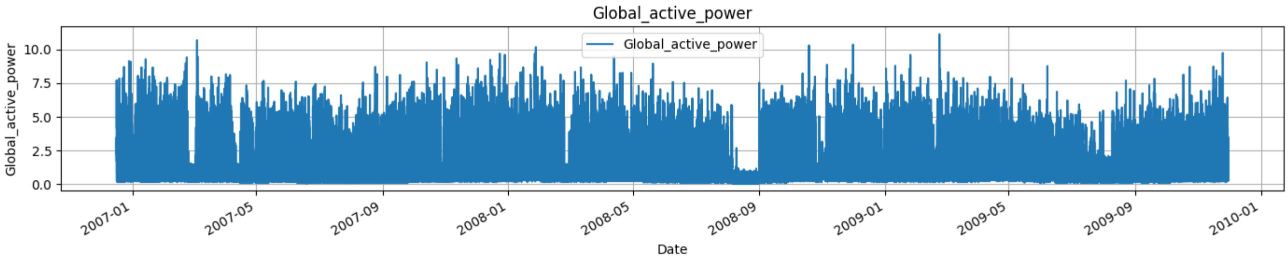
Task – 3: Create a histogram

```
plt.figure(figsize=(14, 10))
for i, column in enumerate(newdf.columns, 1):
   plt.subplot(3, 3, i)
   sns.histplot(newdf[column], kde=True, bins=10)
   plt.title(f'Histogram of {column}')
   plt.xlabel(column)
   plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

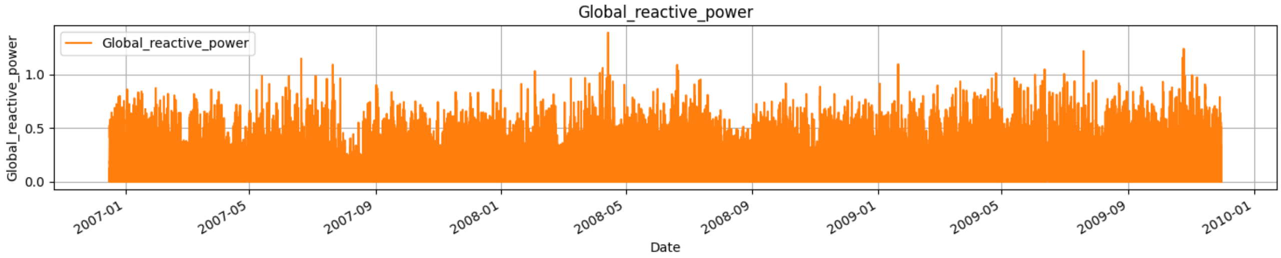


Task – 4: Create a Time series

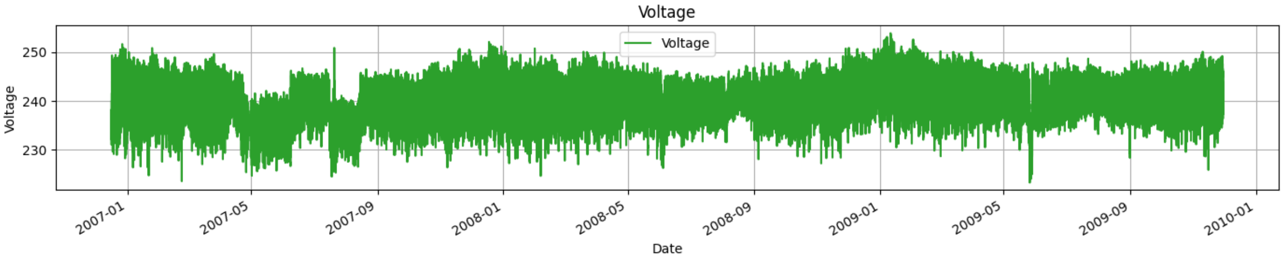
```
plt.figure(figsize=(14, 20))
for i, column in enumerate(newdf.columns, 1):
   plt.subplot(7, 1, i)
   newdf[column].plot(title=column, xlabel='Date', ylabel=column, legend=True)
   plt.grid(True)
plt.tight_layout()
plt.show()
```



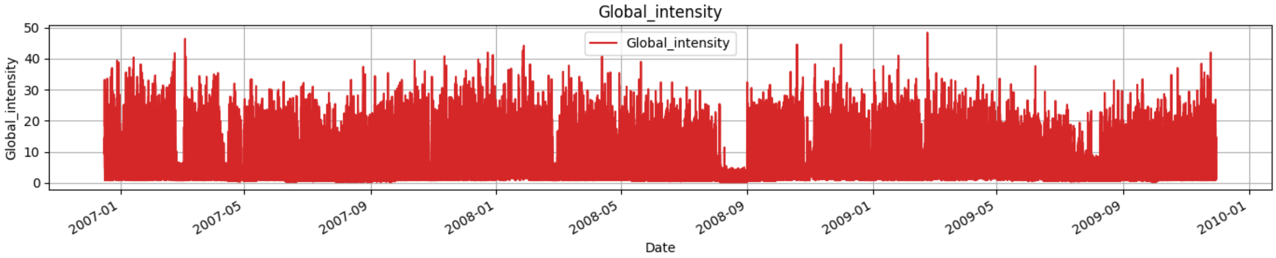
Global_active_power



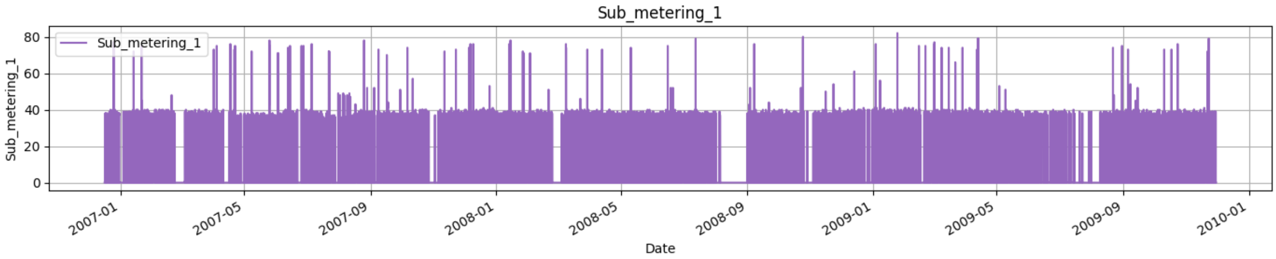
Global_reactive_power



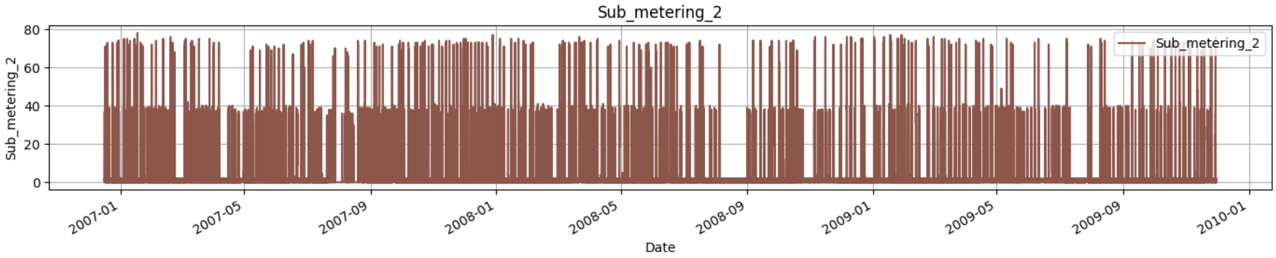
Voltage



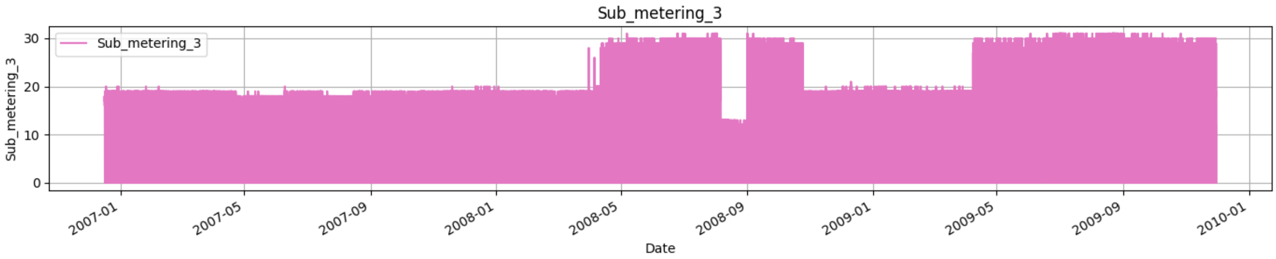
Global_intensity



Sub_metering_1



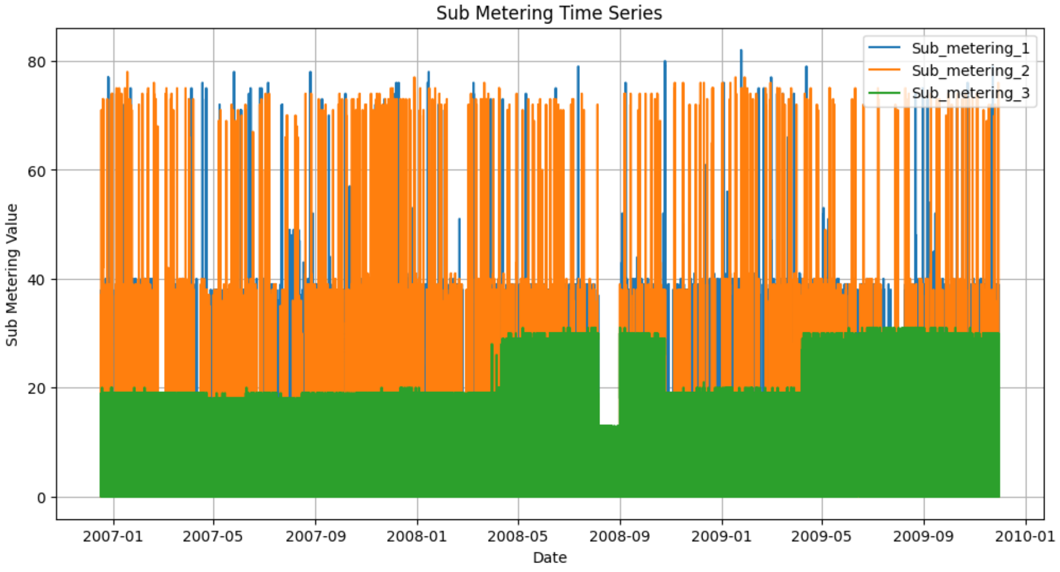
Sub_metering_2



Sub_metering_3

Task – 5: Create a plot for sub metering

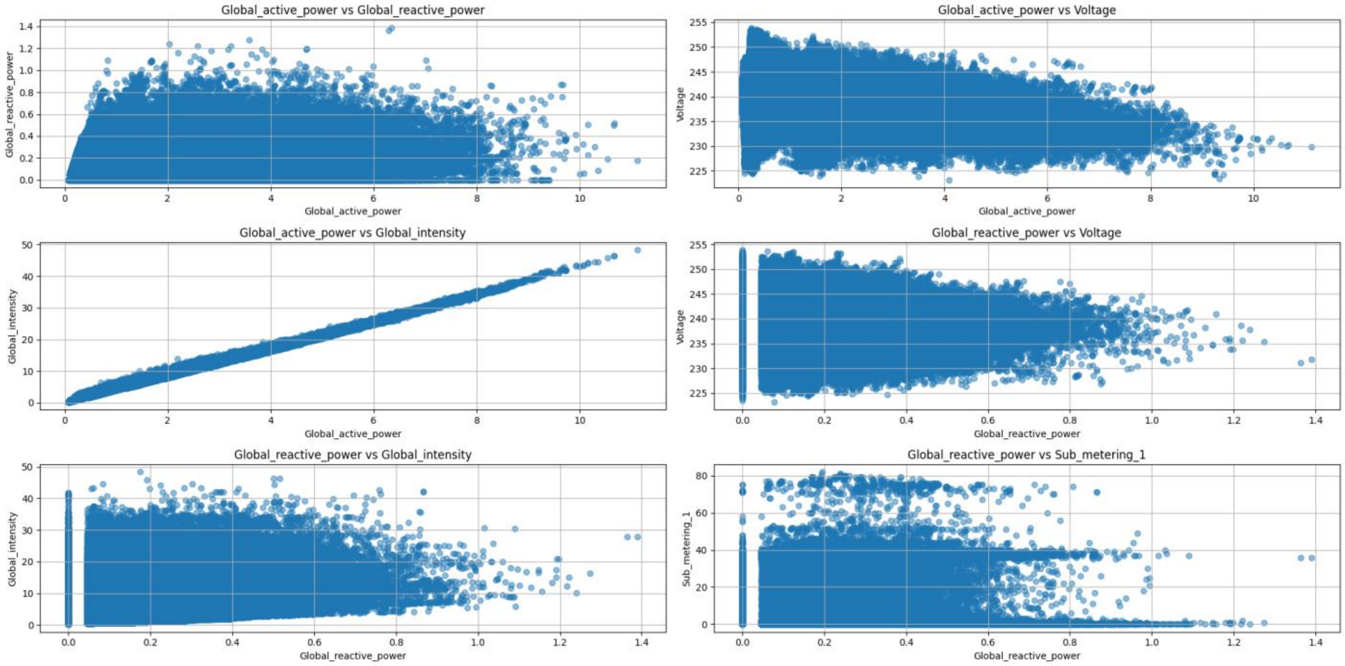
```
df_melted = newdf.reset_index().melt(id_vars='dt', value_vars=['Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3'])
plt.figure(figsize=(12, 6))
sns.lineplot(data=df_melted, x='dt', y='value', hue='variable')
plt.title('Sub Metering Time Series')
plt.xlabel('Date')
plt.ylabel('Sub Metering Value')
plt.legend(loc='upper right')
plt.grid(True)
plt.show()
```



Task – 6: Create multiple other plots.

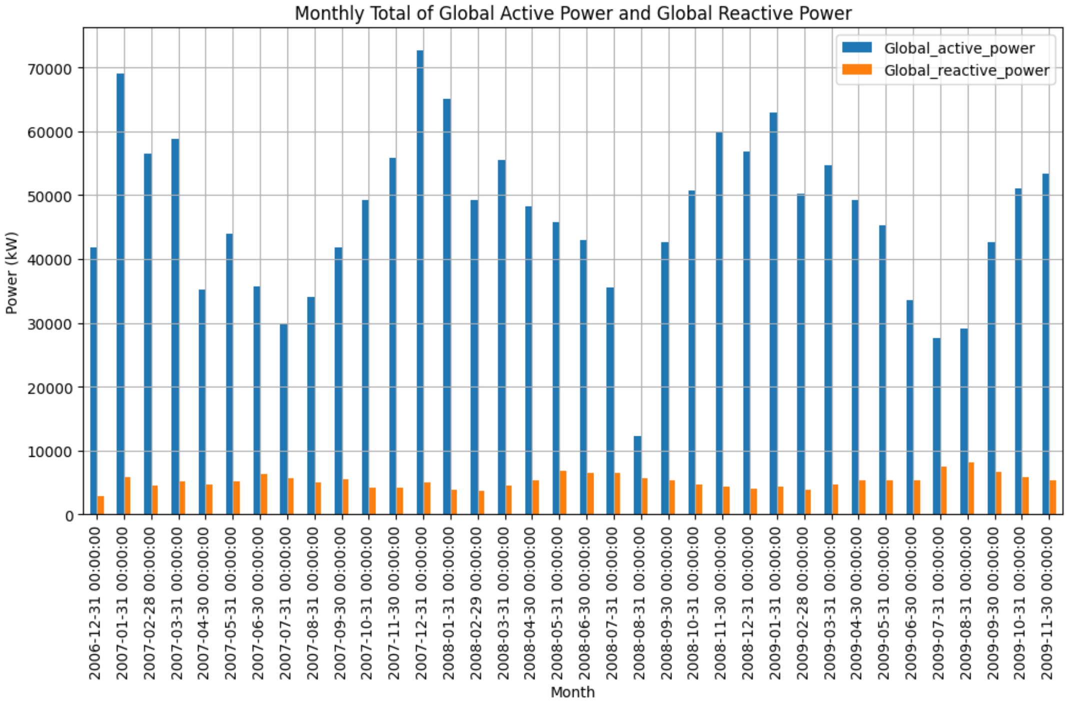
Scatter Plot

```
pairs = [
    ('Global_active_power', 'Global_reactive_power'),
    ('Global_active_power', 'Voltage'),
    ('Global_active_power', 'Global_intensity'),
    ('Global_reactive_power', 'Voltage'),
    ('Global_reactive_power', 'Global_intensity'),
    ('Global_reactive_power', 'Sub_metering_1'),
]
nrows, ncols = 3, 2
fig, axes = plt.subplots(nrows, ncols, figsize=(20, 10))
axes = axes.flatten()
for ax, (x_col, y_col) in zip(axes, pairs):
    ax.scatter(newdf[x_col], newdf[y_col], marker='o', alpha=0.5)
    ax.set_title(f'{x_col} vs {y_col}')
    ax.set_xlabel(x_col)
    ax.set_ylabel(y_col)
    ax.grid(True)
plt.tight_layout()
plt.show()
```

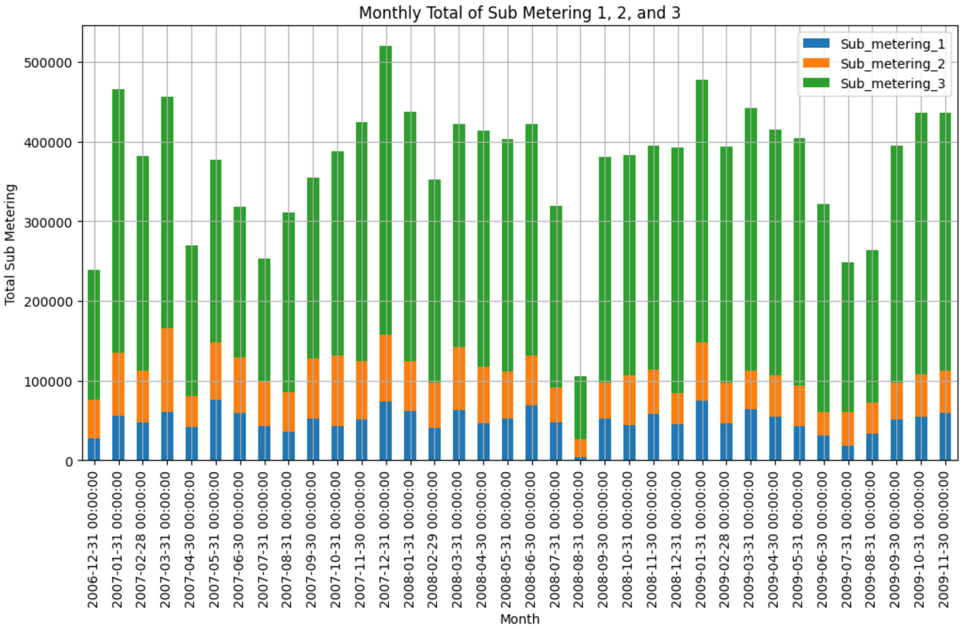


Bar Chart

```
monthly_data = newdf.resample('ME').sum()
monthly_data[['Global_active_power', 'Global_reactive_power']].plot(kind='bar', figsize=(12, 6))
plt.title('Monthly Total of Global Active Power and Global Reactive Power')
plt.xlabel('Month')
plt.ylabel('Power (kW)')
plt.grid(True)
plt.show()
```

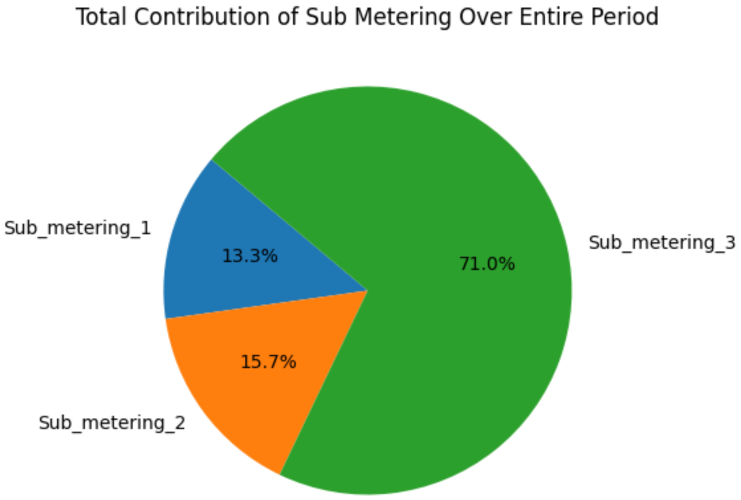


```
monthly_sub_metering = newdf[['Sub_metering_1', 'Sub_metering_2',
'Sub_metering_3']].resample('ME').sum()

monthly_sub_metering.plot(kind='bar', stacked=True, figsize=(12, 6))

plt.title('Monthly Total of Sub Metering 1, 2, and 3')

plt.xlabel('Month')

plt.ylabel('Total Sub Metering')

plt.grid(True)

plt.show()
```



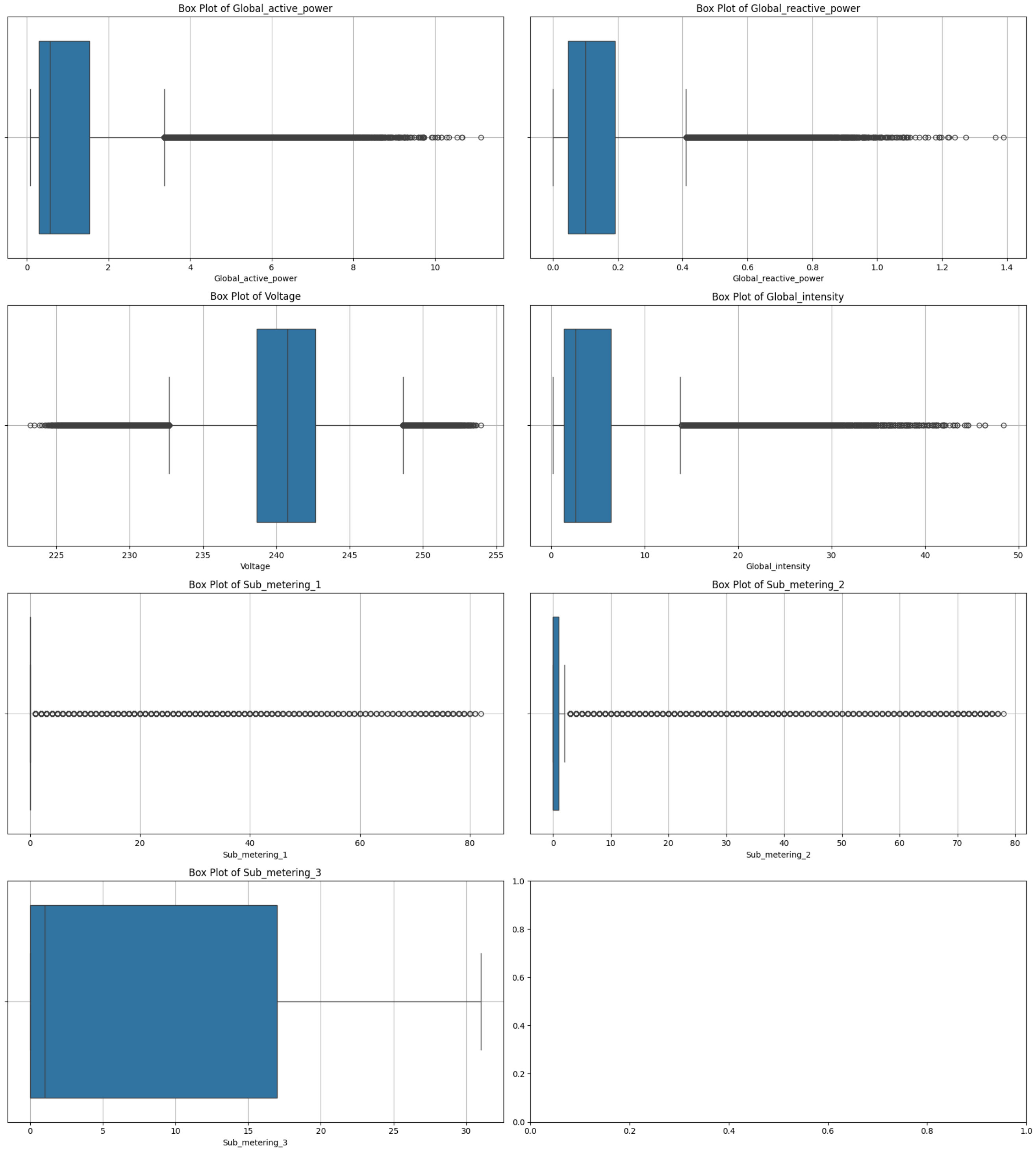
Pie Chart

```
total_sub_metering = monthly_sub_metering.sum()
plt.figure(figsize=(5, 5))
plt.pie(total_sub_metering, labels=total_sub_metering.index, autopct='%1.1f%%', startangle=140)
plt.title('Total Contribution of Sub Metering Over Entire Period')
plt.show()
```



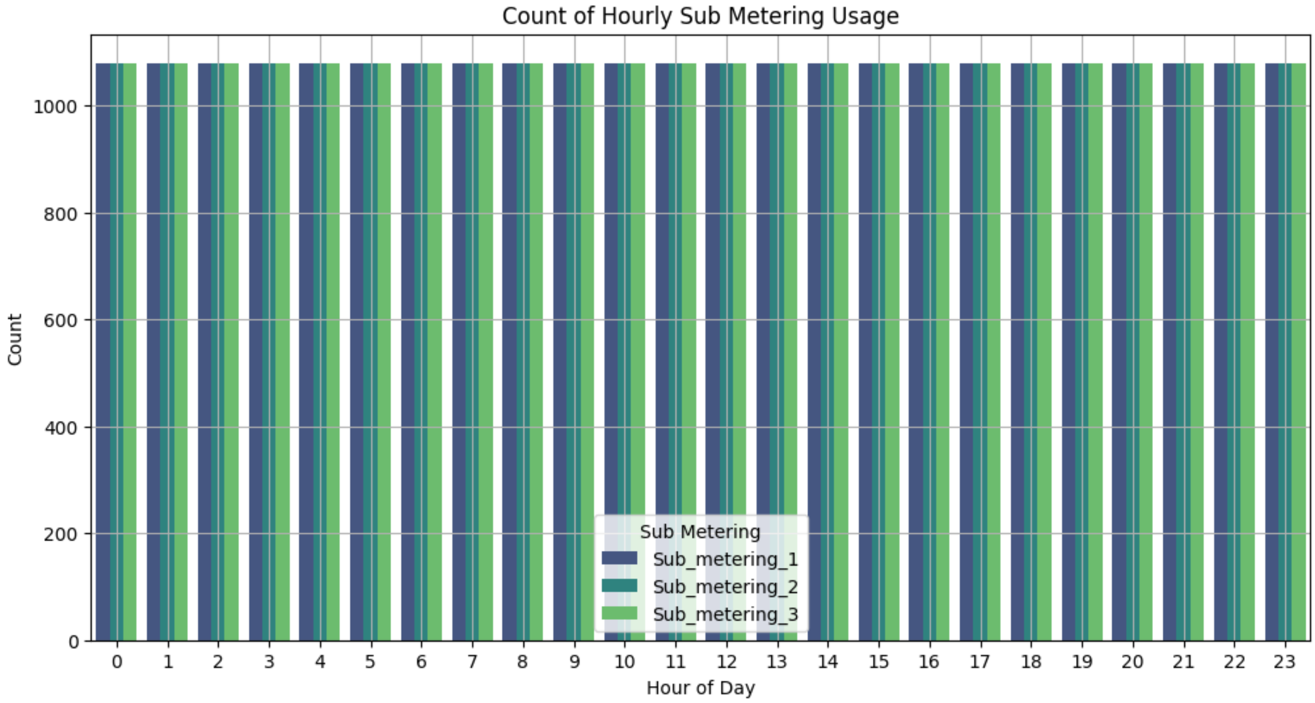
Box Plot

```
columns_to_plot = ['Global_active_power', 'Global_reactive_power', 'Voltage',
                   'Global_intensity', 'Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3']
fig, axes = plt.subplots(nrows=4, ncols=2, figsize=(18, 20))
axes = axes.flatten()
for i, column in enumerate(columns_to_plot):
    sns.boxplot(x=newdf[column], ax=axes[i])
    axes[i].set_title(f'Box Plot of {column}')
    axes[i].grid(True)
plt.tight_layout()
plt.show()
```



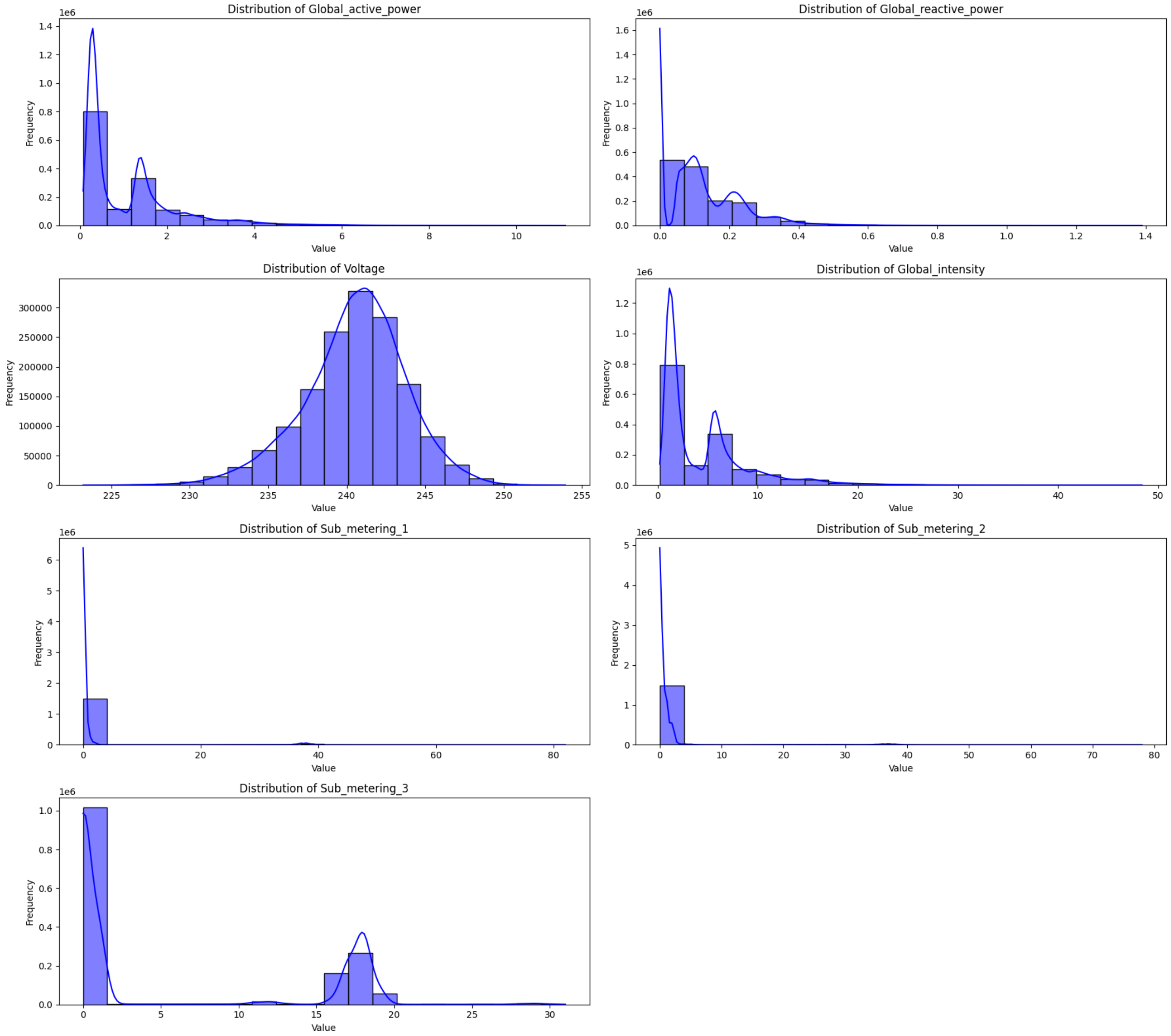
Count Plot

```
hourly_data = newdf[['Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3']].resample('h').sum()
hourly_data['Hour'] = hourly_data.index.hour
hourly_data_melted = hourly_data.melt(id_vars='Hour', var_name='Sub_metering', value_name='Total')

plt.figure(figsize=(12, 6))
sns.countplot(x='Hour', data=hourly_data_melted, hue='Sub_metering', palette='viridis')
plt.title('Count of Hourly Sub Metering Usage')
plt.xlabel('Hour of Day')
plt.ylabel('Count')
plt.legend(title='Sub Metering')
plt.grid(True)
plt.show()
```



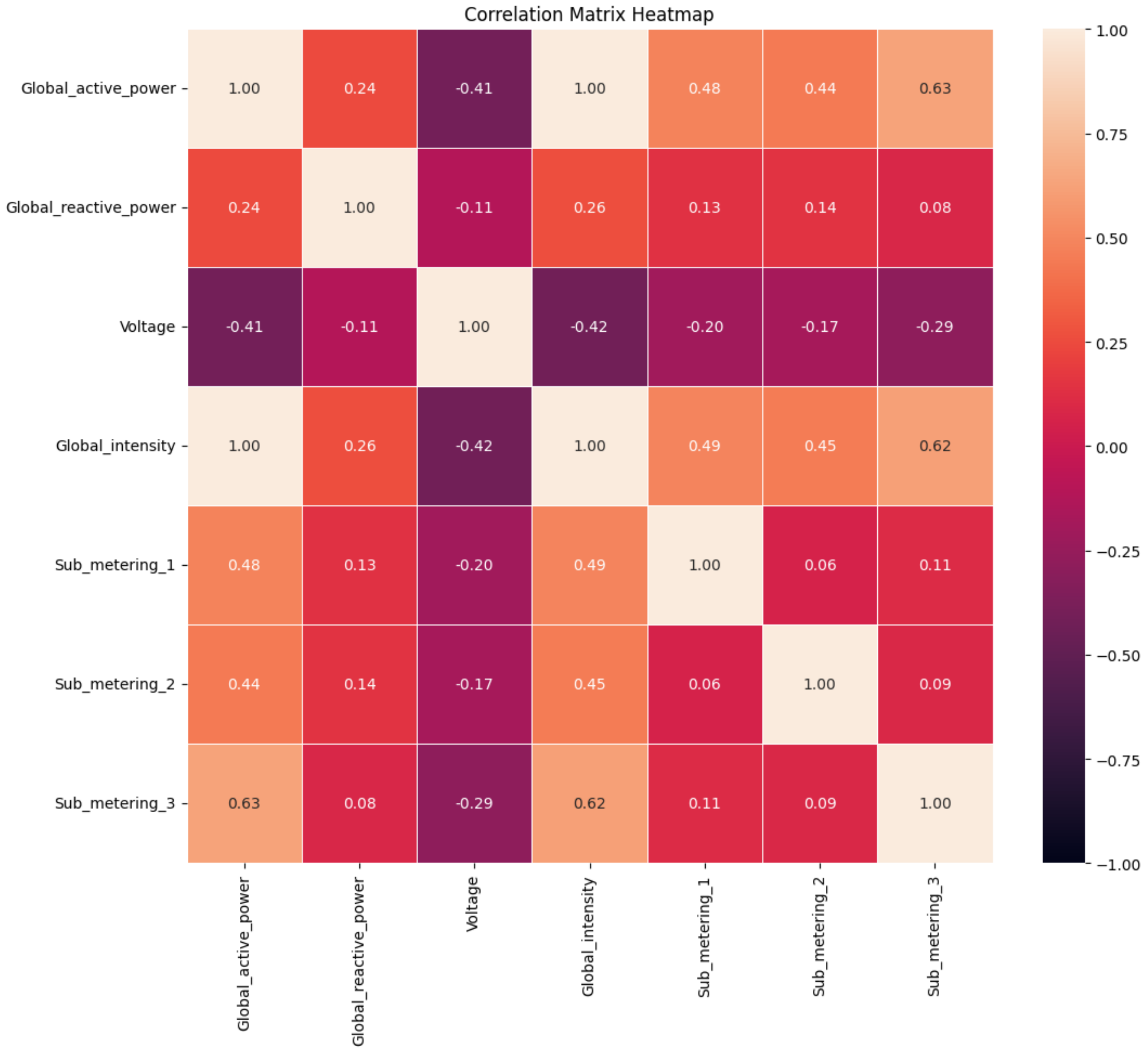
Distplot

```
columns_to_plot = ['Global_active_power', 'Global_reactive_power', 'Voltage',
                   'Global_intensity', 'Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3']
plt.figure(figsize=(18, 16))
for i, column in enumerate(columns_to_plot):
    plt.subplot(4, 2, i + 1)
    sns.histplot(newdf[column], kde=True, color='blue', bins=20)
    plt.title(f'Distribution of {column}')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



Heatmap

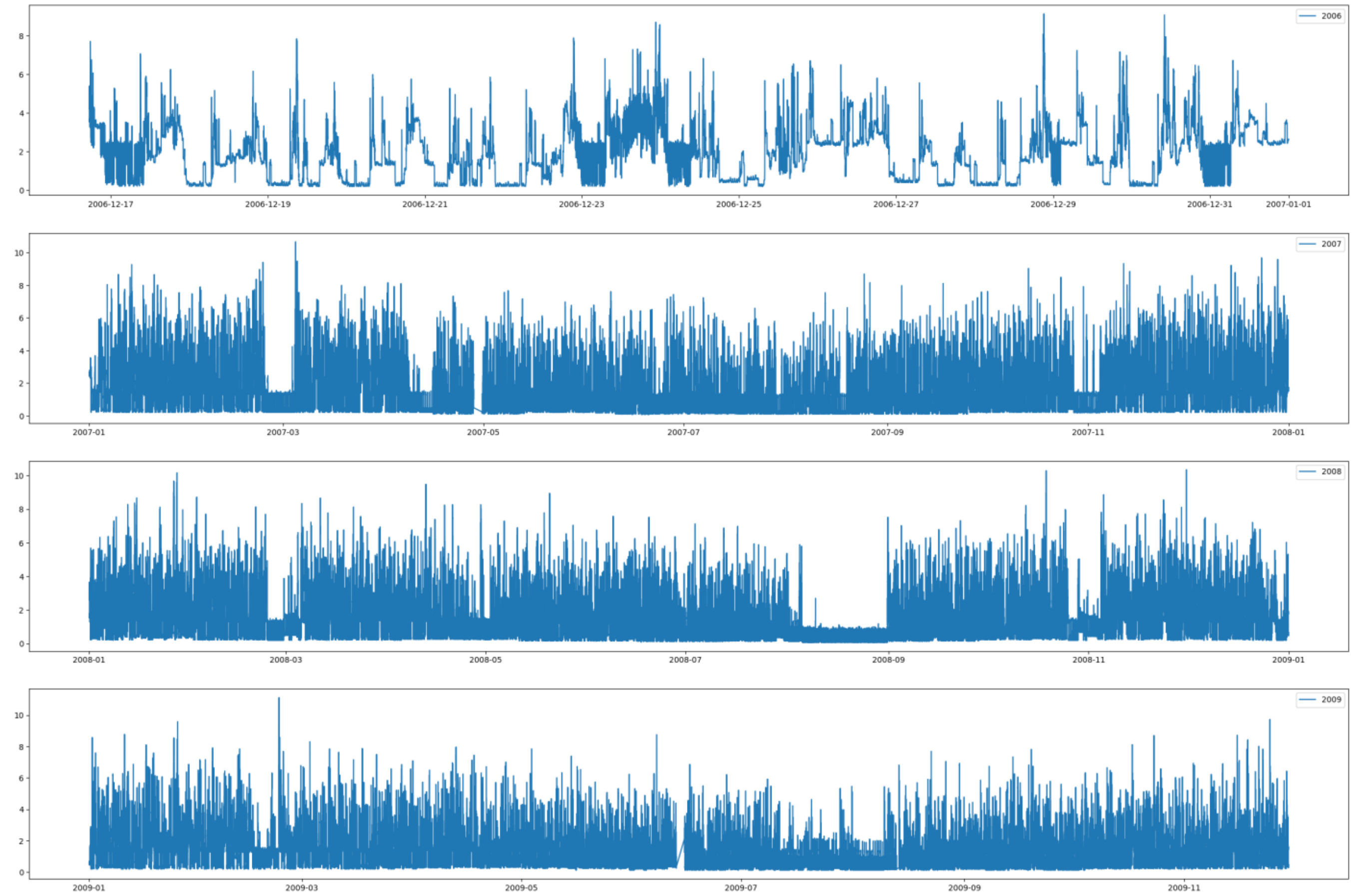
```
correlation_matrix = newdf.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', linewidths=0.5, vmin=-1, vmax=1)
plt.title('Correlation Matrix Heatmap')
plt.show()
```



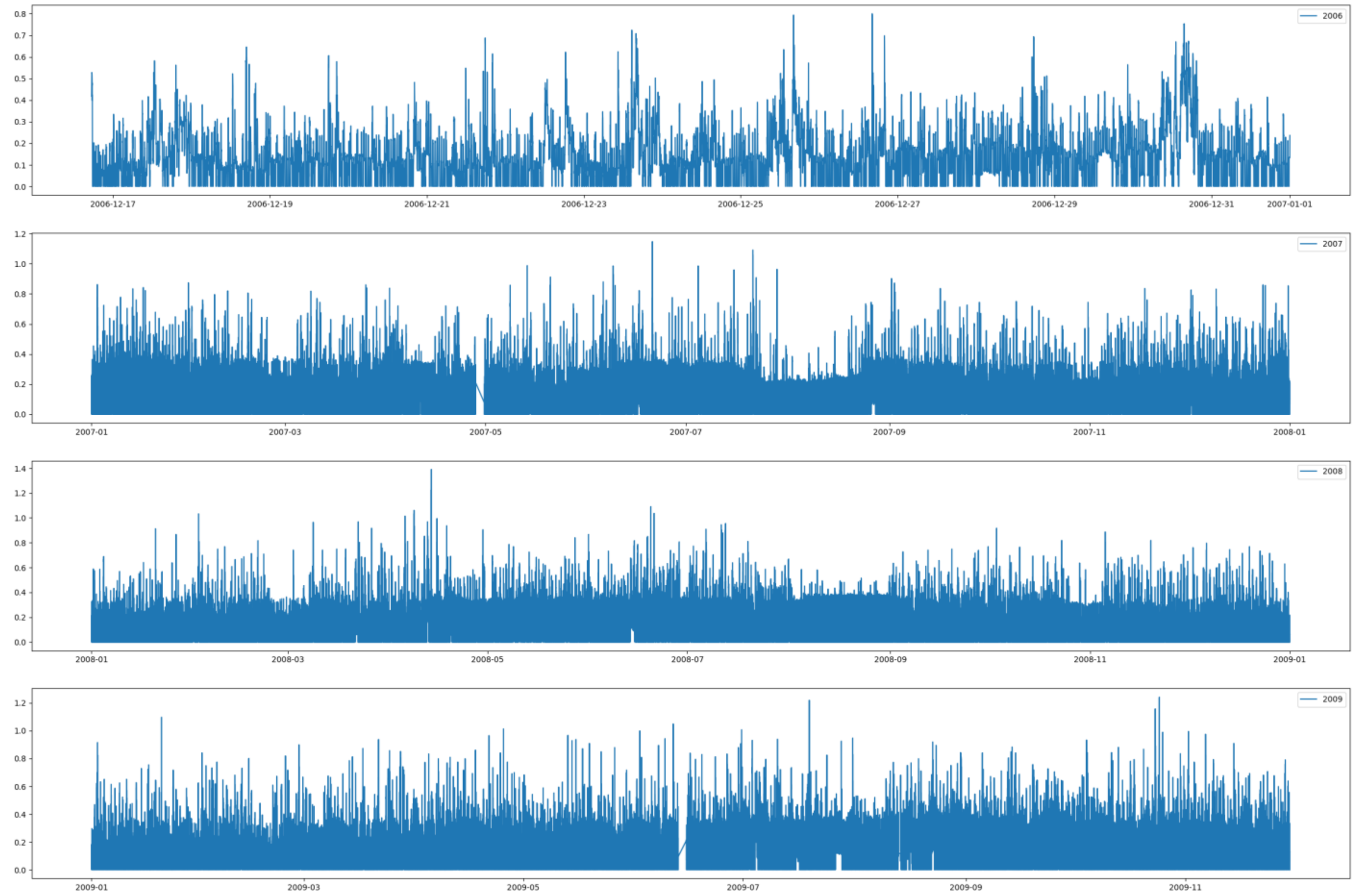
Visualise Each Parameter Early

```
def visualize_yearly(data, feat_name):
    fig, axis = plt.subplots(4, 1, figsize=(30, 20))
    for i, d in enumerate(zip(axis, list(data[feat_name].groupby(data.index.year)))):
        d[0].plot(pd.DataFrame(d[1][1]), label=d[1][0])
        d[0].legend(loc='upper right')

    fig.text(0.40, 0.9, 'Year-Wise Analysis : %s ' % feat_name, va='center', fontdict={'fontsize': 25})
    plt.show()

visualize_yearly(data=newdf, feat_name='Global_active_power')
visualize_yearly(data=newdf, feat_name='Global_reactive_power')
visualize_yearly(data=newdf, feat_name='Voltage')
visualize_yearly(data=newdf, feat_name='Global_intensity')
visualize_yearly(data=newdf, feat_name='Sub_metering_1')
visualize_yearly(data=newdf, feat_name='Sub_metering_2')
visualize_yearly(data=newdf, feat_name='Sub_metering_3')
```

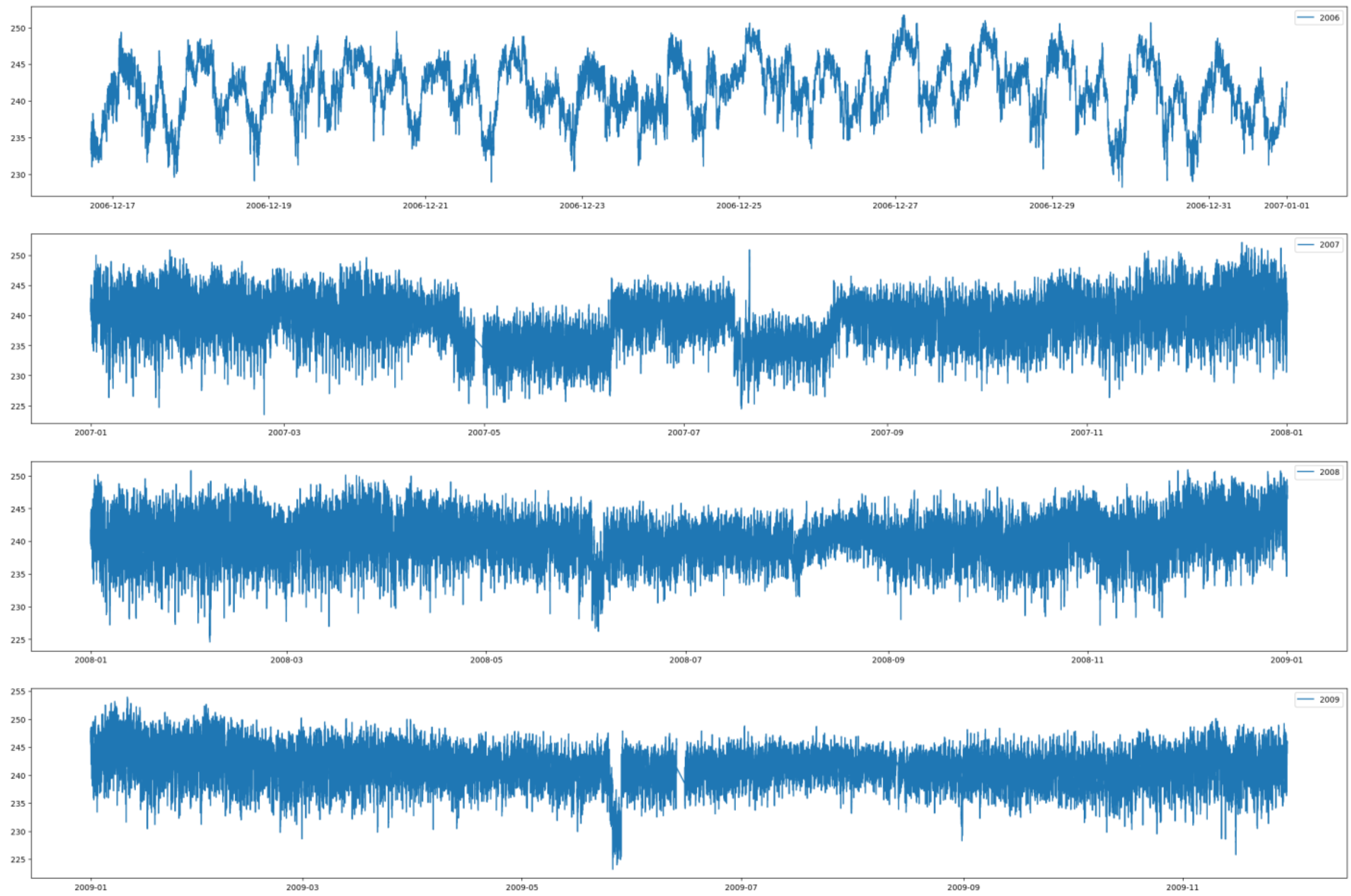
Year-Wise Analysis : Global_active_power



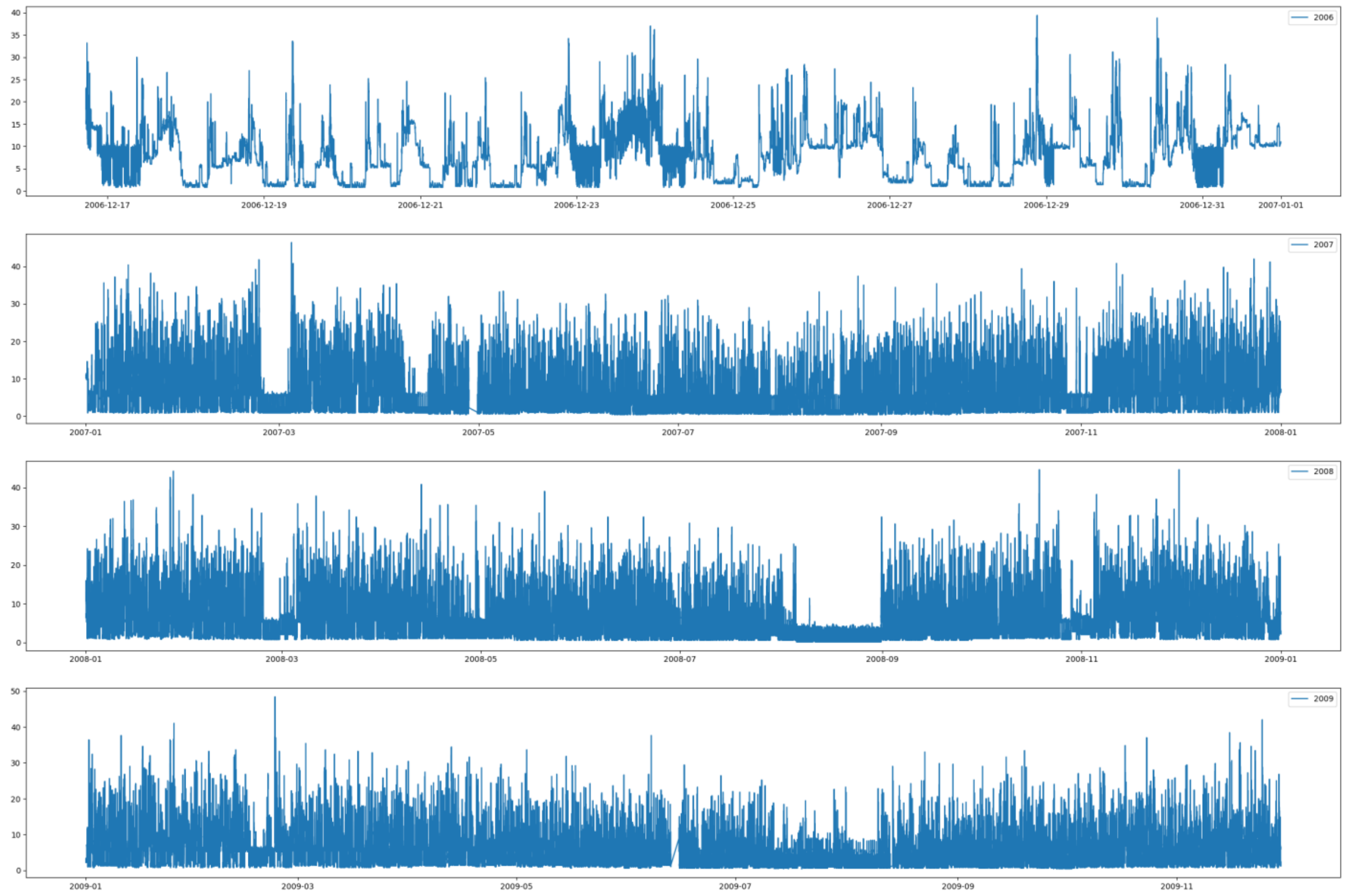
Year-Wise Analysis : Global_reactive_power



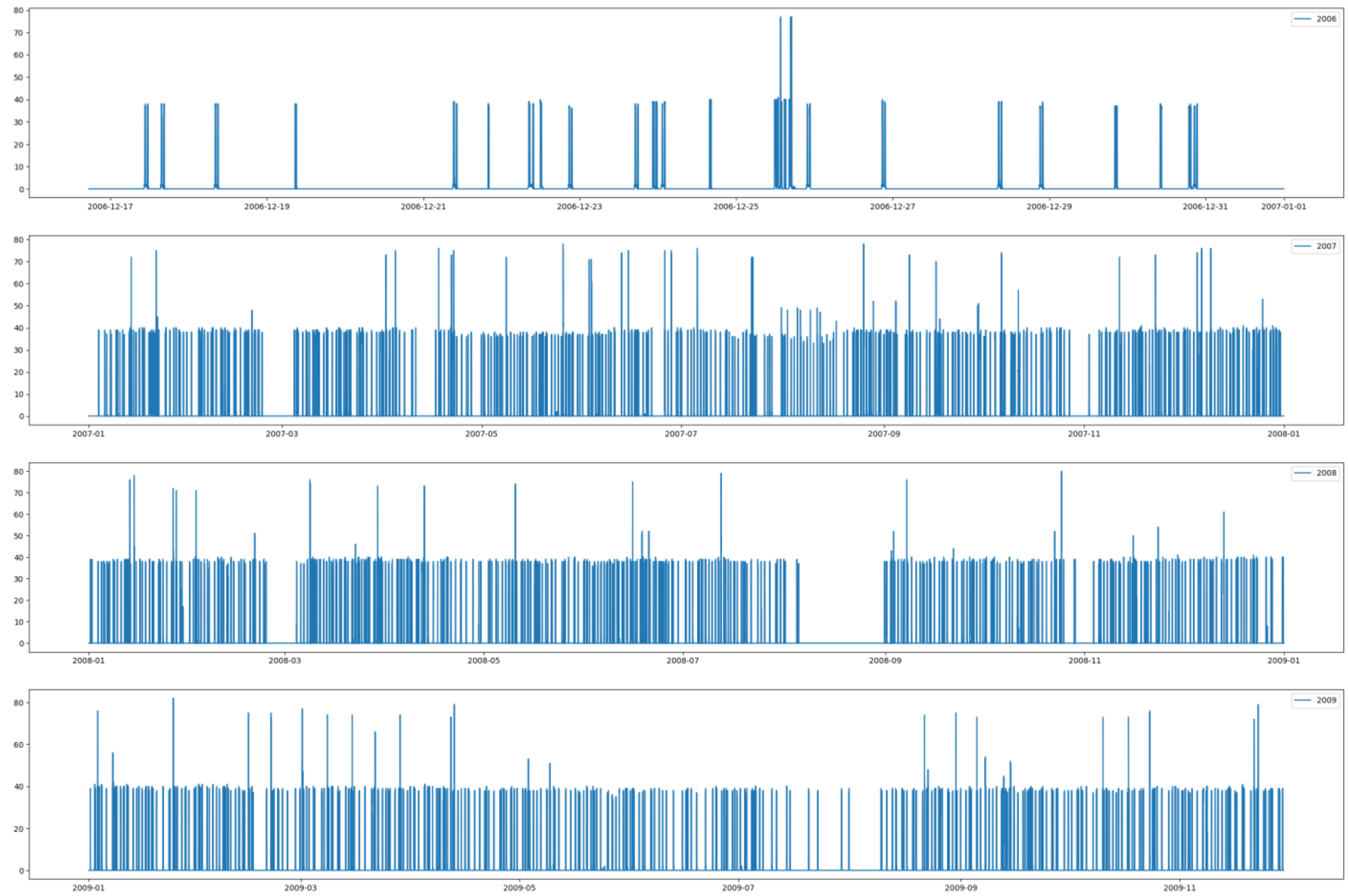
Year-Wise Analysis : Voltage



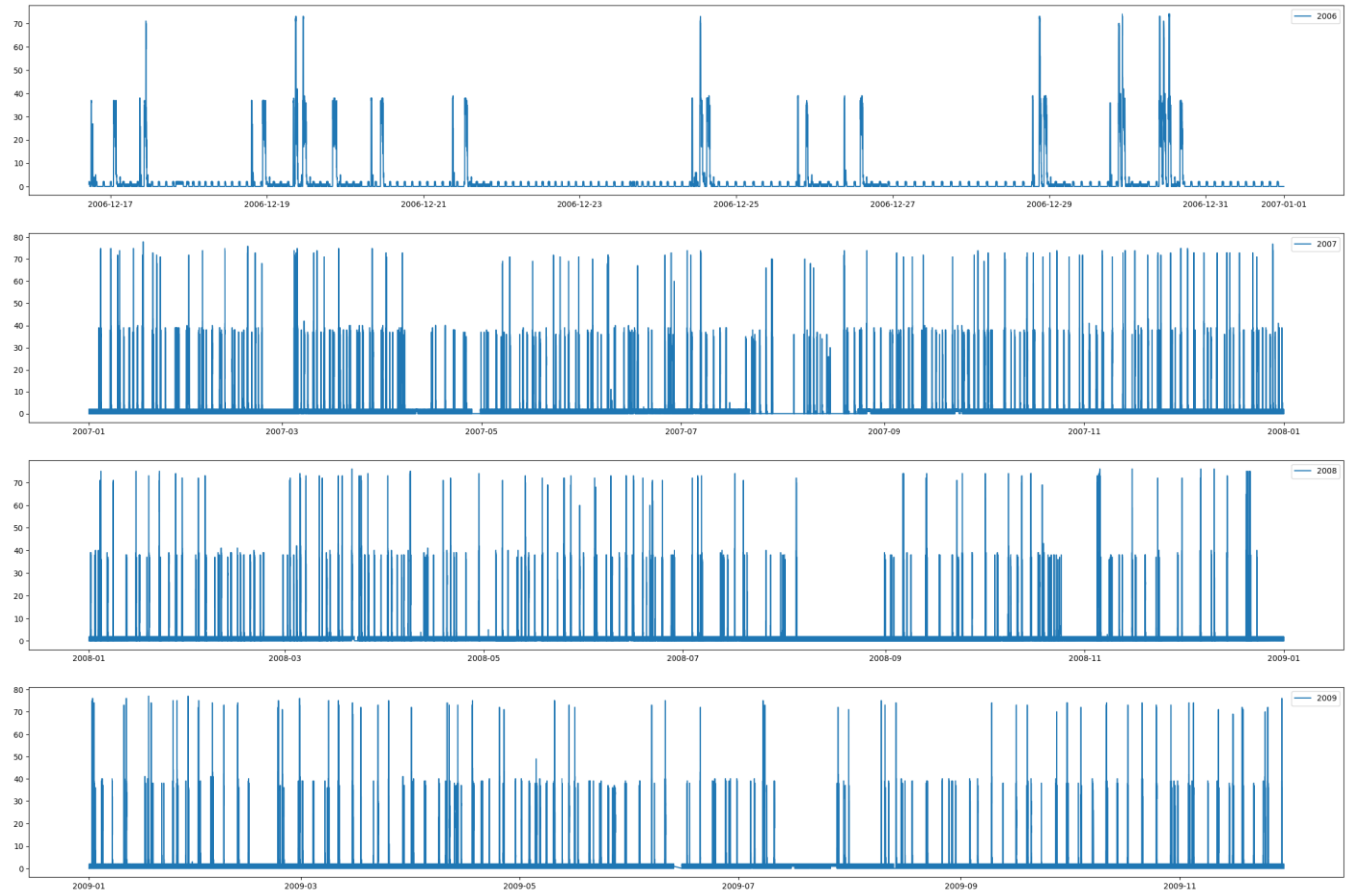
Year-Wise Analysis : Global_intensity

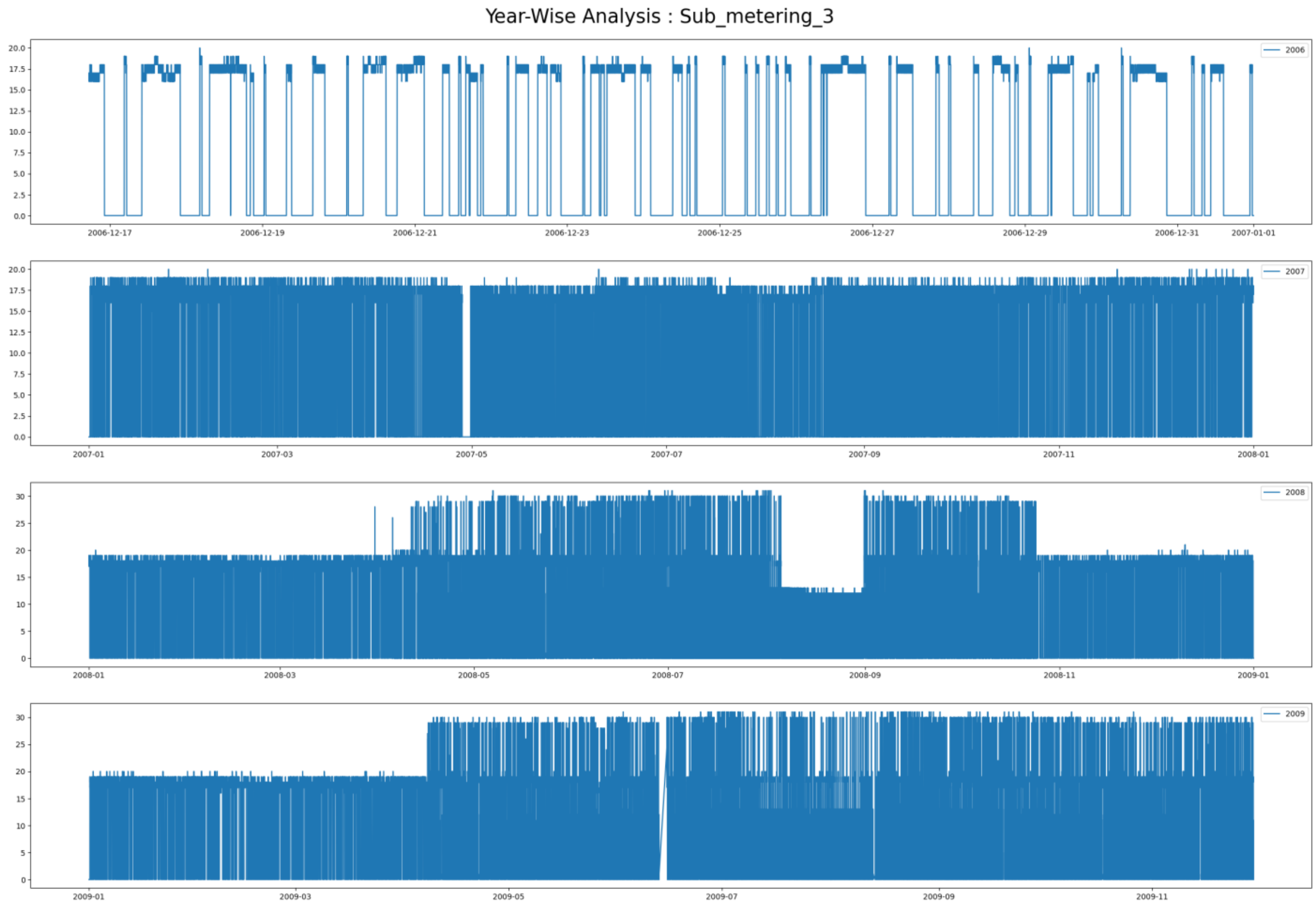


Year-Wise Analysis : Sub_metering_1



Year-Wise Analysis : Sub_metering_2





Use of each type of plot

- **Histogram**: Ideal for understanding the distribution of power consumption levels.
- **Time Series Plot**: Best for analysing trends and patterns over the 4-year period.
- **Plot for Sub Metering**: Key for comparing the contributions of different sub-meters over time.
- **Scatterplot**: Useful for examining relationships between two continuous variables.
- **Bar Chart**: Great for comparing quantities across different time periods.
- **Pie Chart**: Shows proportions of sub-metering contributions to total consumption.
- **Count Plot**: Displays the frequency of different power consumption categories.
- **Boxplot**: Summarizes distributions and highlights outliers in consumption data.
- **Heatmap**: Effective for identifying correlations and patterns across variables and time.
- **Distplot**: Combines a histogram and KDE to show a detailed distribution.