**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'])

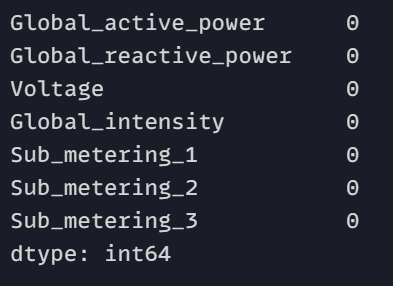
*# Data Cleaning*

df.isna().sum()



df.dropna(*inplace*=True)

df.isna().sum()



**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)

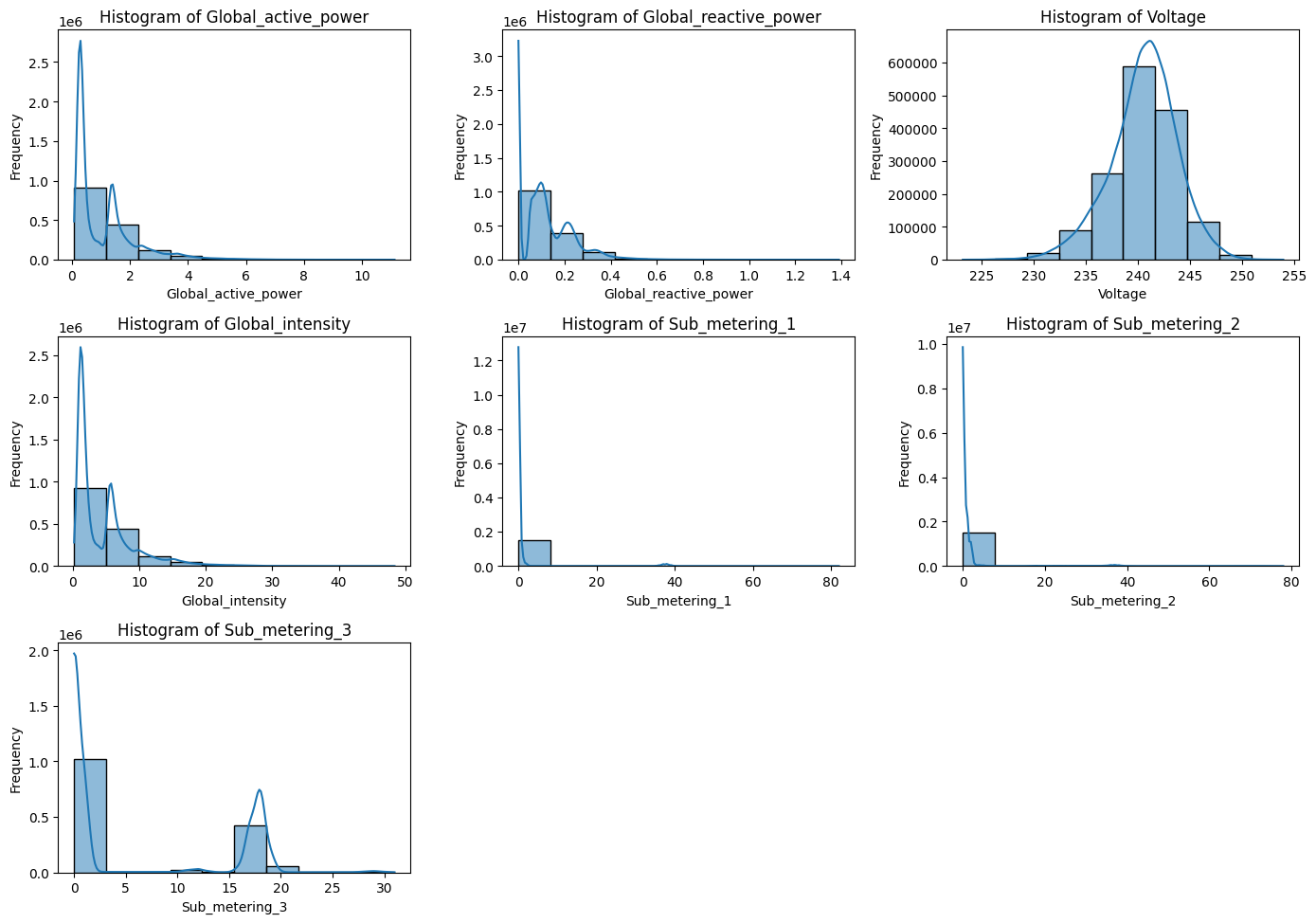
    plt.title(*f*'Histogram of {column}')

    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):

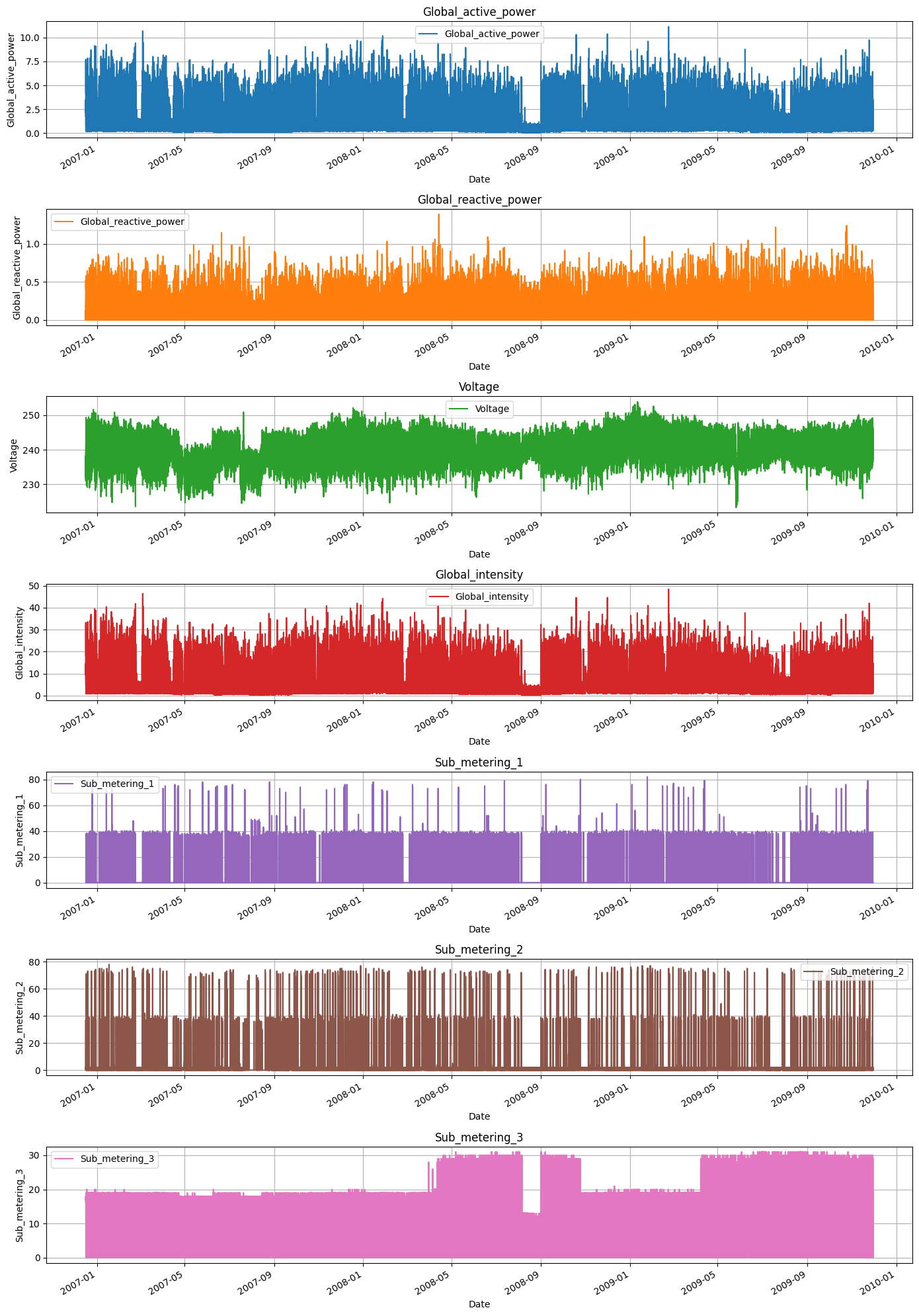
    plt.subplot(7, 1, i)

    newdf[column].plot(*title*=column, *xlabel*='Date', *ylabel*=column, *legend*=True)

    plt.grid(True)

plt.tight\_layout()

plt.show()



**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')

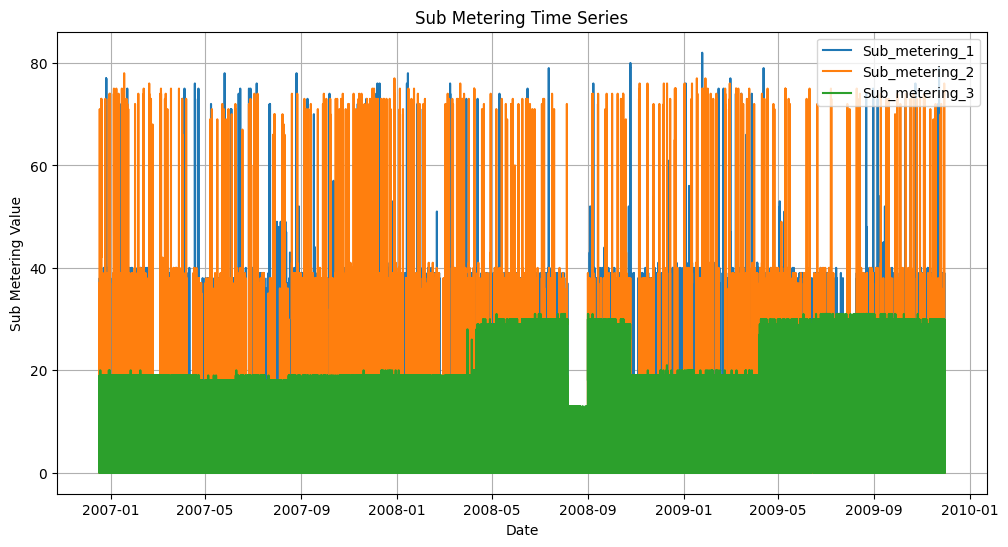
plt.xlabel('Date')

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))

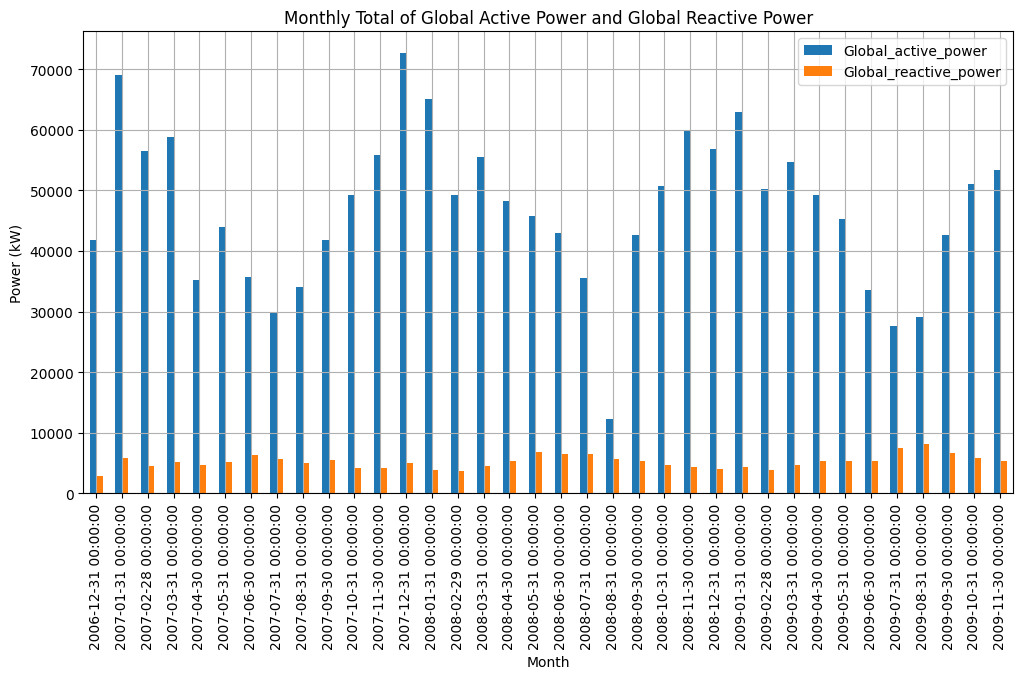
plt.title('Monthly Total of Global Active Power and Global Reactive Power')

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))

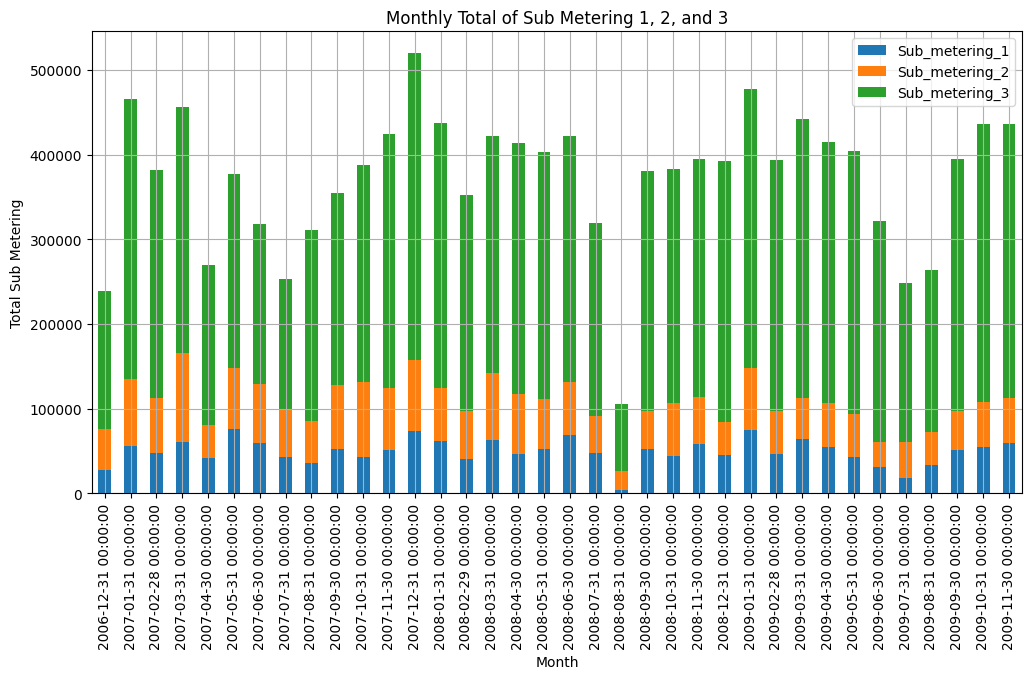
plt.title('Monthly Total of Sub Metering 1, 2, and 3')

plt.xlabel('Month')

plt.ylabel('Total Sub Metering')

plt.grid(True)

plt.show()



**Pie Chart**

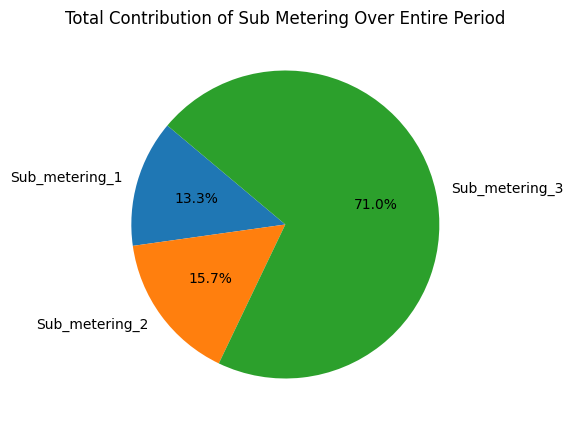
total\_sub\_metering = monthly\_sub\_metering.sum()

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):

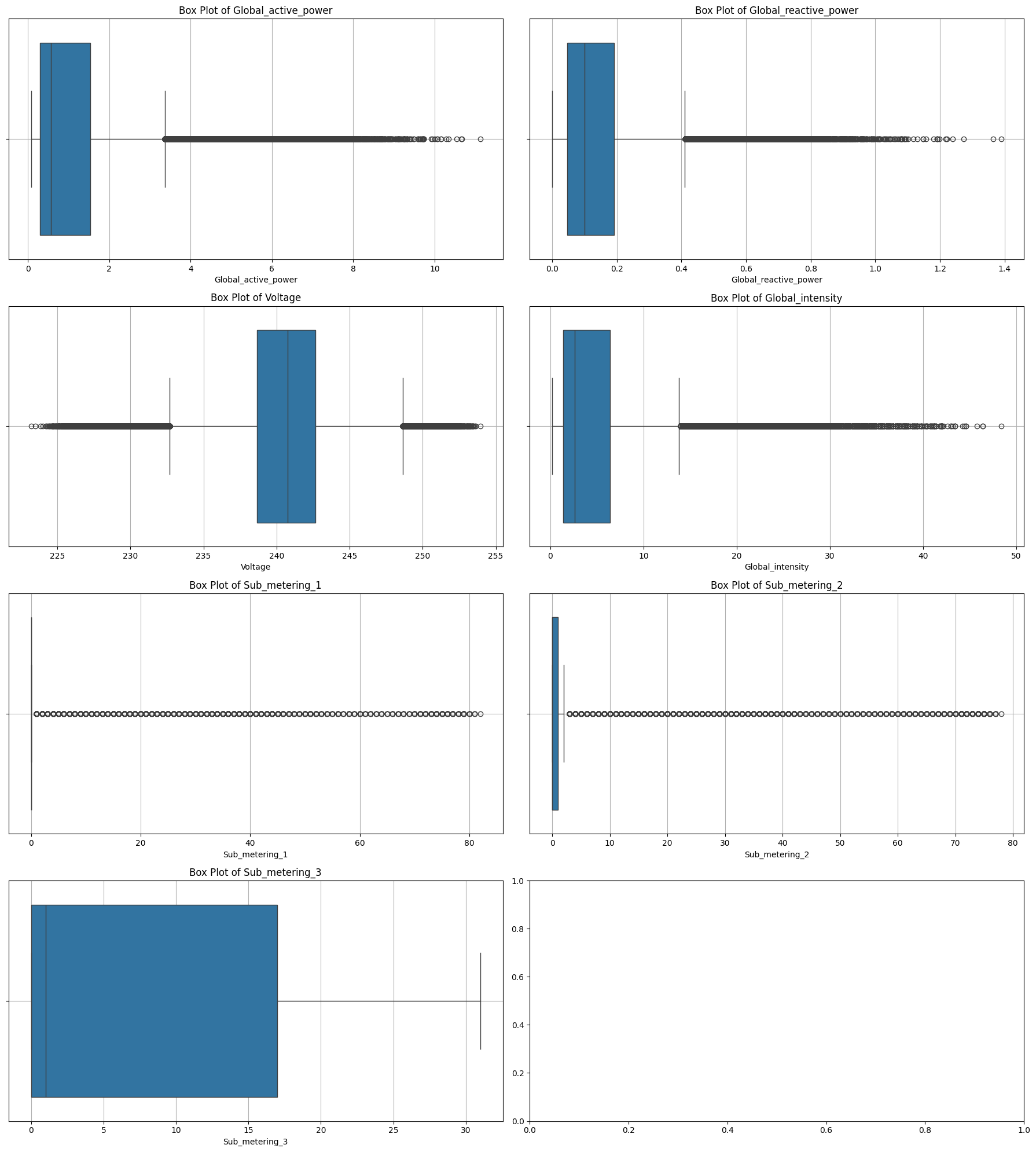
    sns.boxplot(*x*=newdf[column], *ax*=axes[i])

    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')

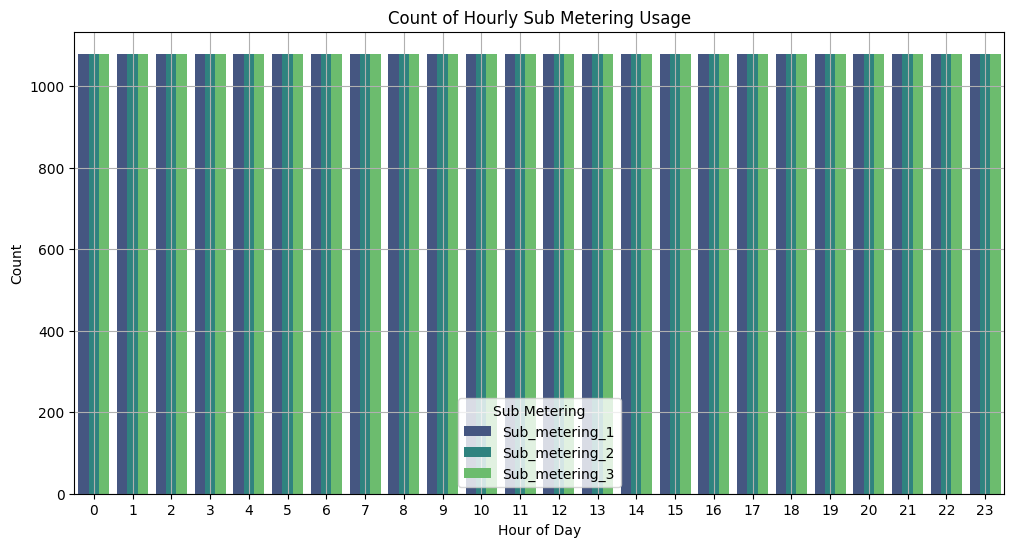
plt.xlabel('Hour of Day')

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)

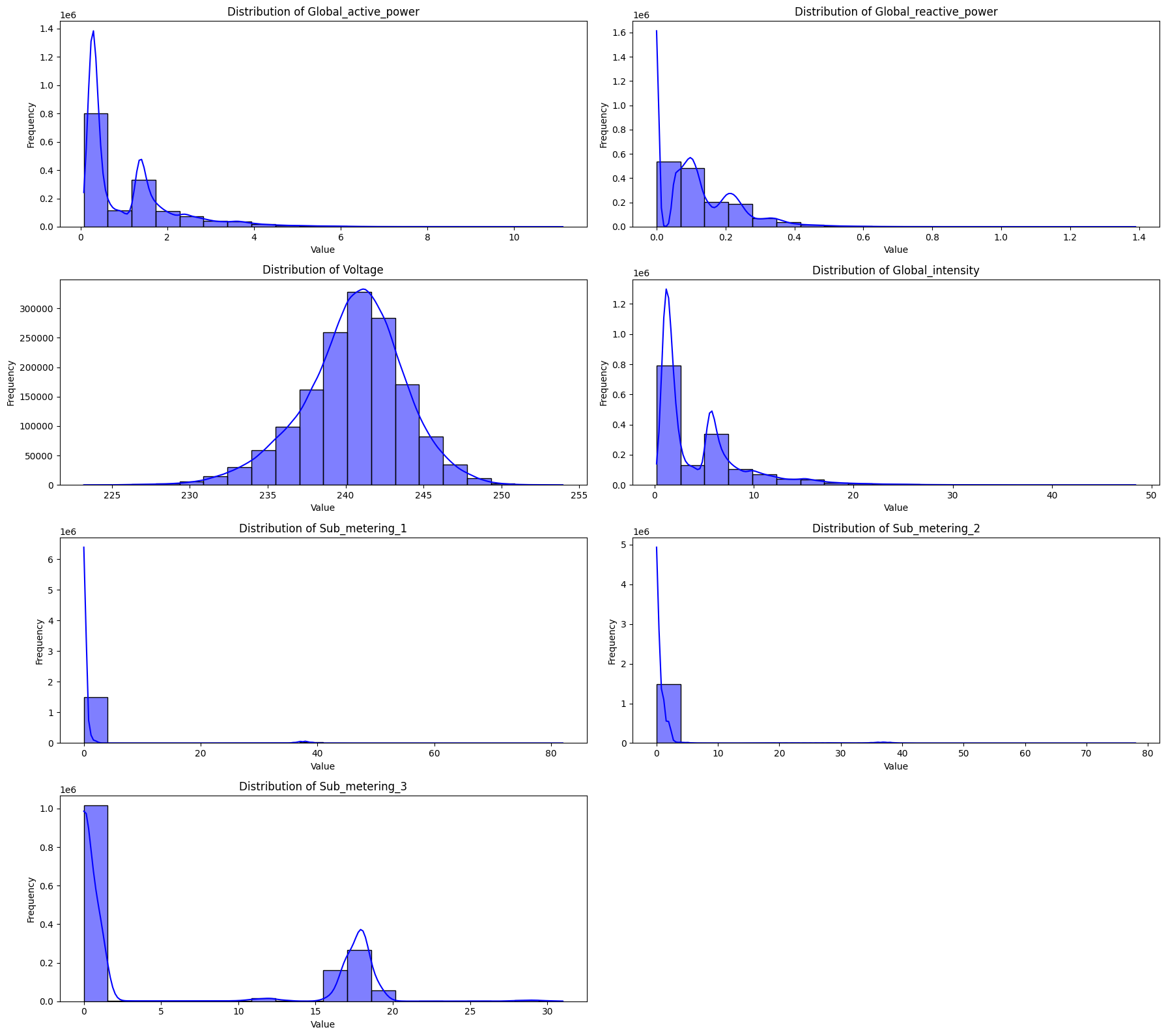
    plt.title(*f*'Distribution of {column}')

    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')

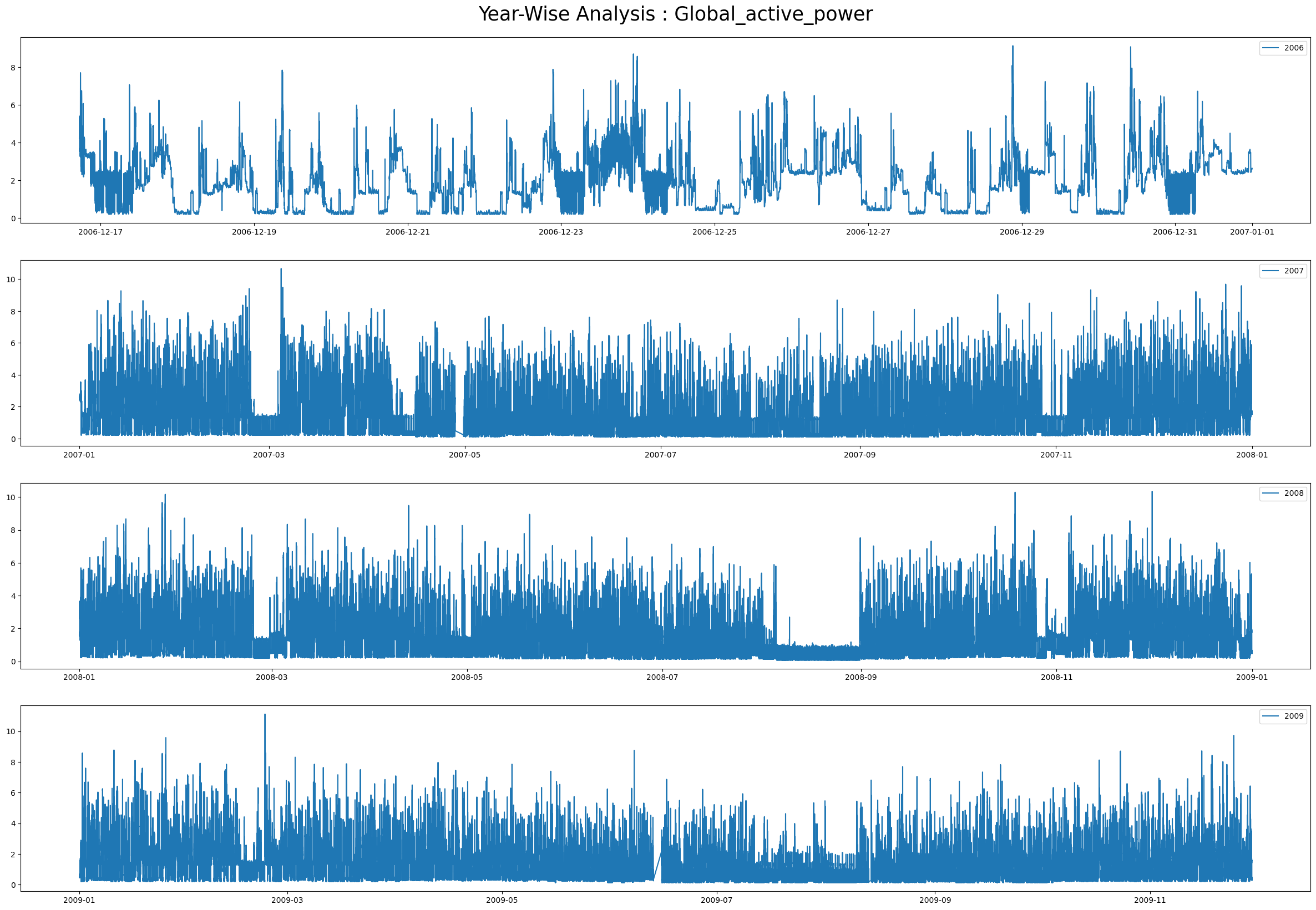
visualize\_yearly(*data*=newdf, *feat\_name*='Voltage')

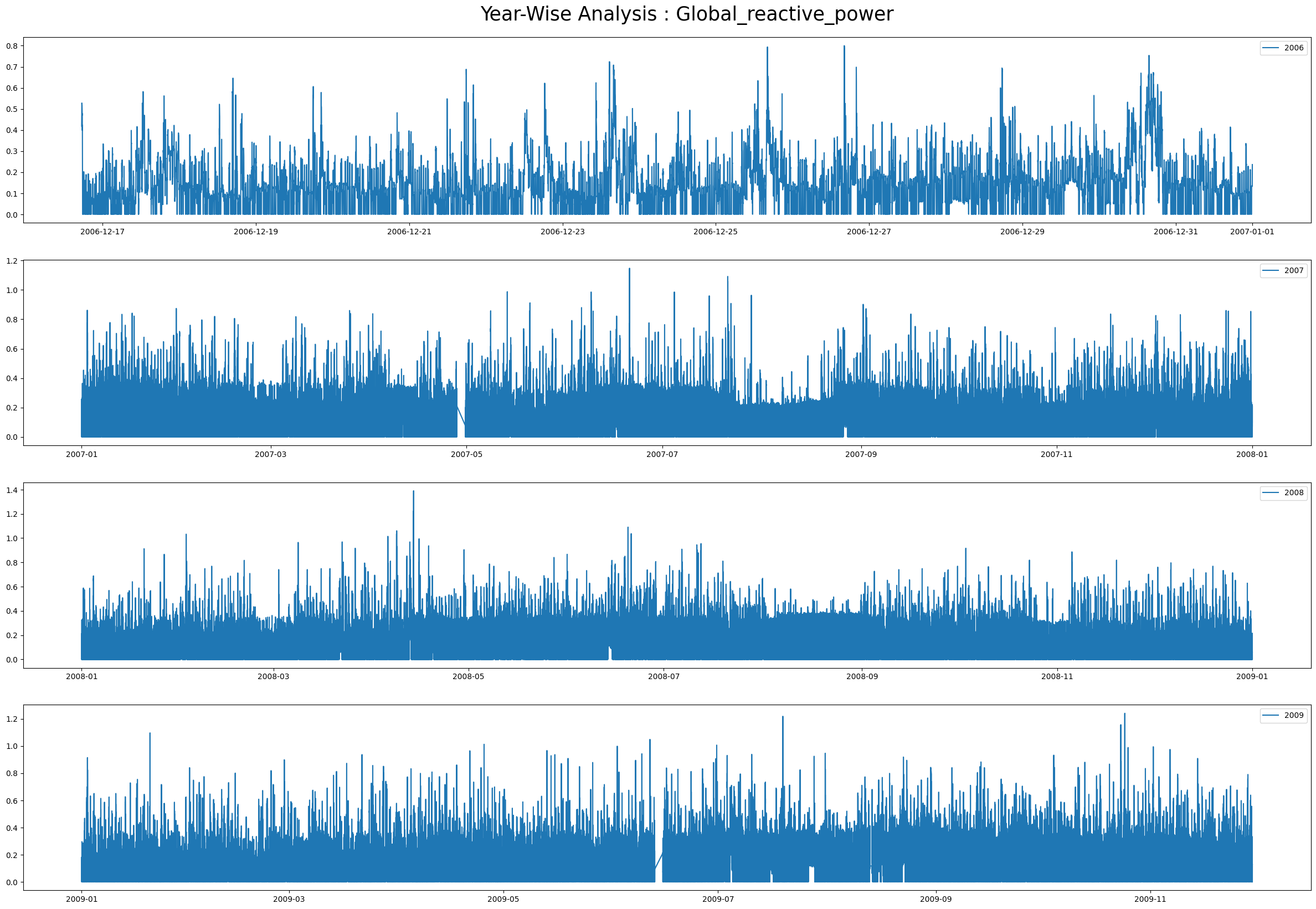
visualize\_yearly(*data*=newdf, *feat\_name*='Global\_intensity')

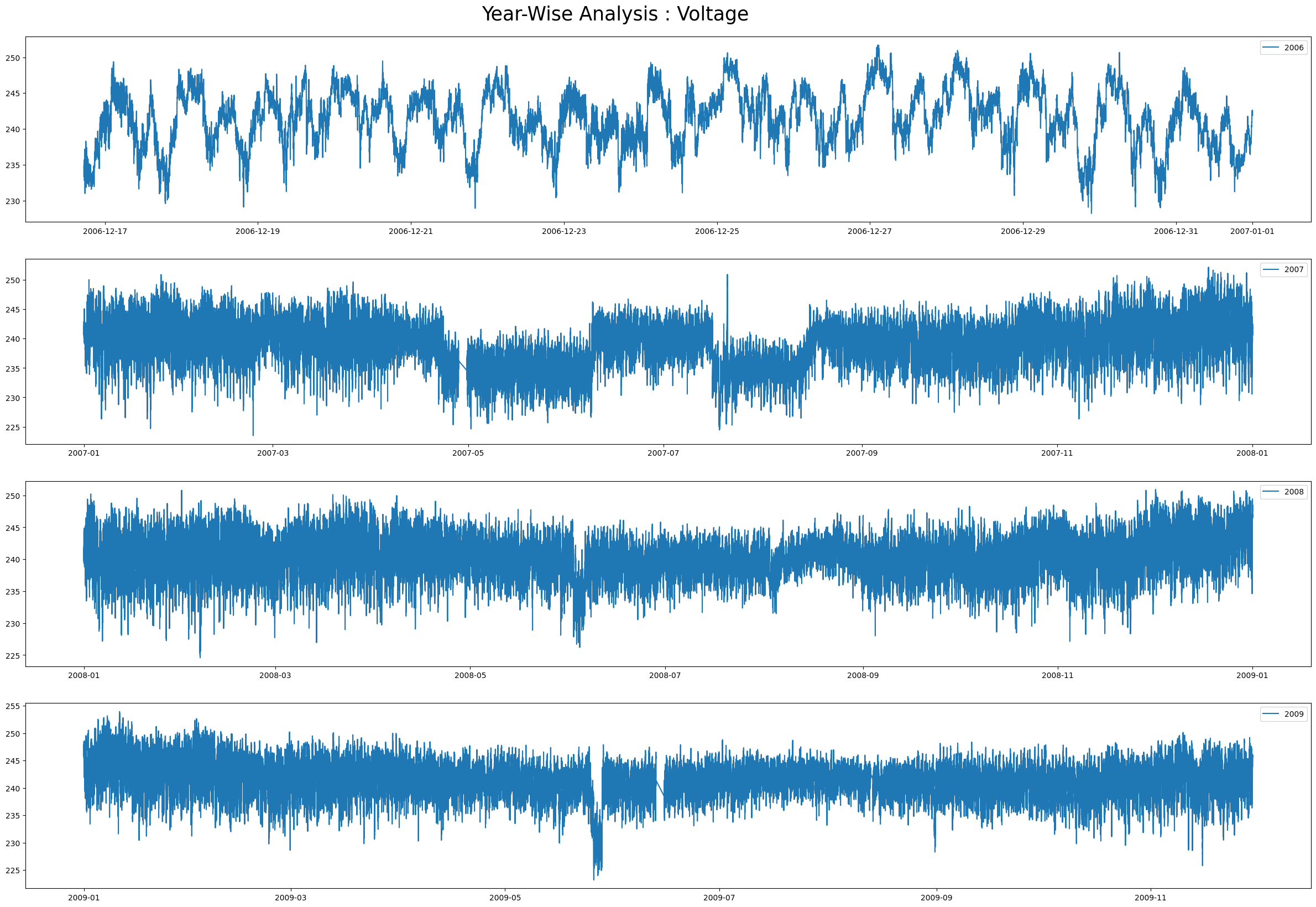
visualize\_yearly(*data*=newdf, *feat\_name*='Sub\_metering\_1')

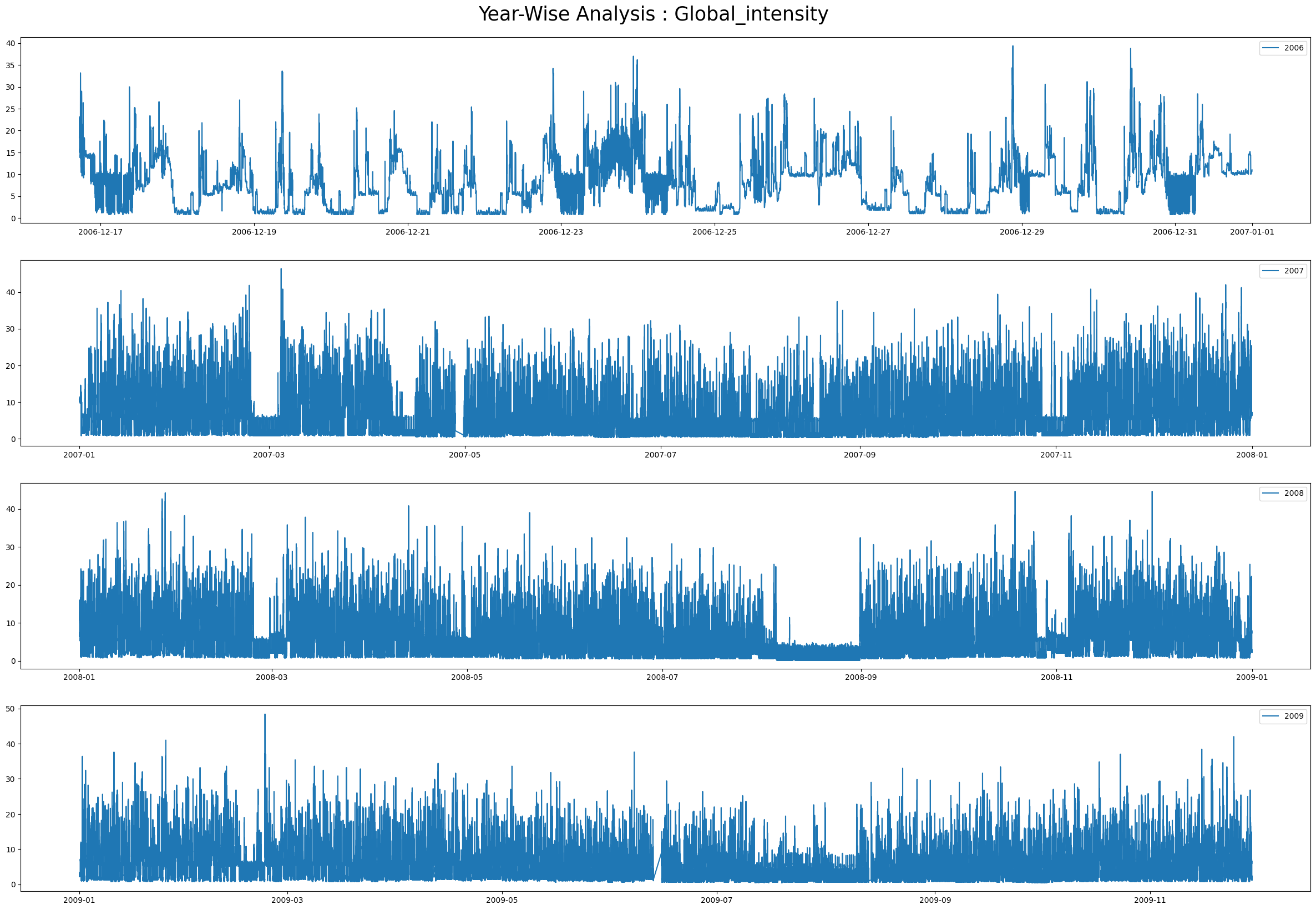
visualize\_yearly(*data*=newdf, *feat\_name*='Sub\_metering\_2')

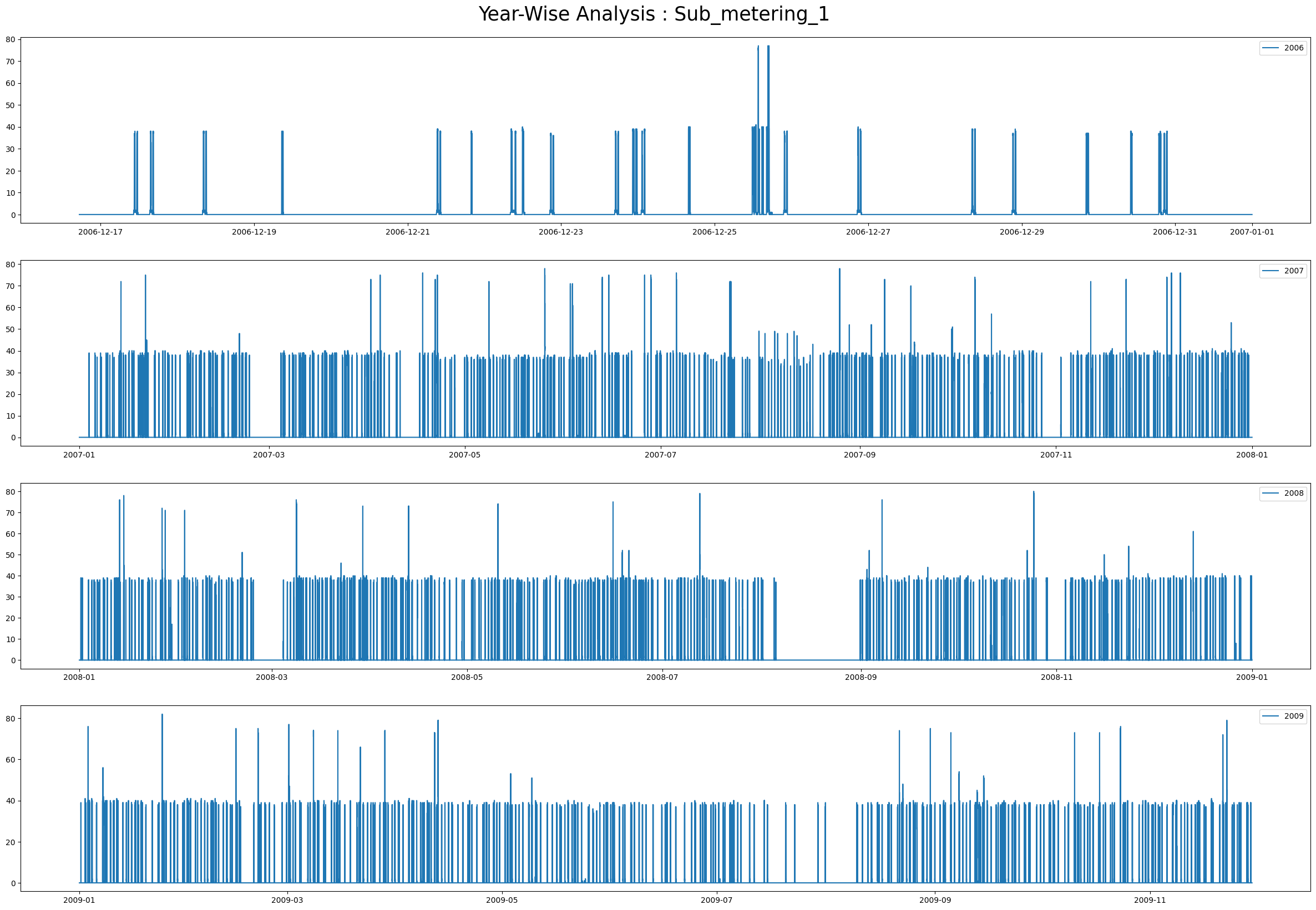
visualize\_yearly(*data*=newdf, *feat\_name*='Sub\_metering\_3')

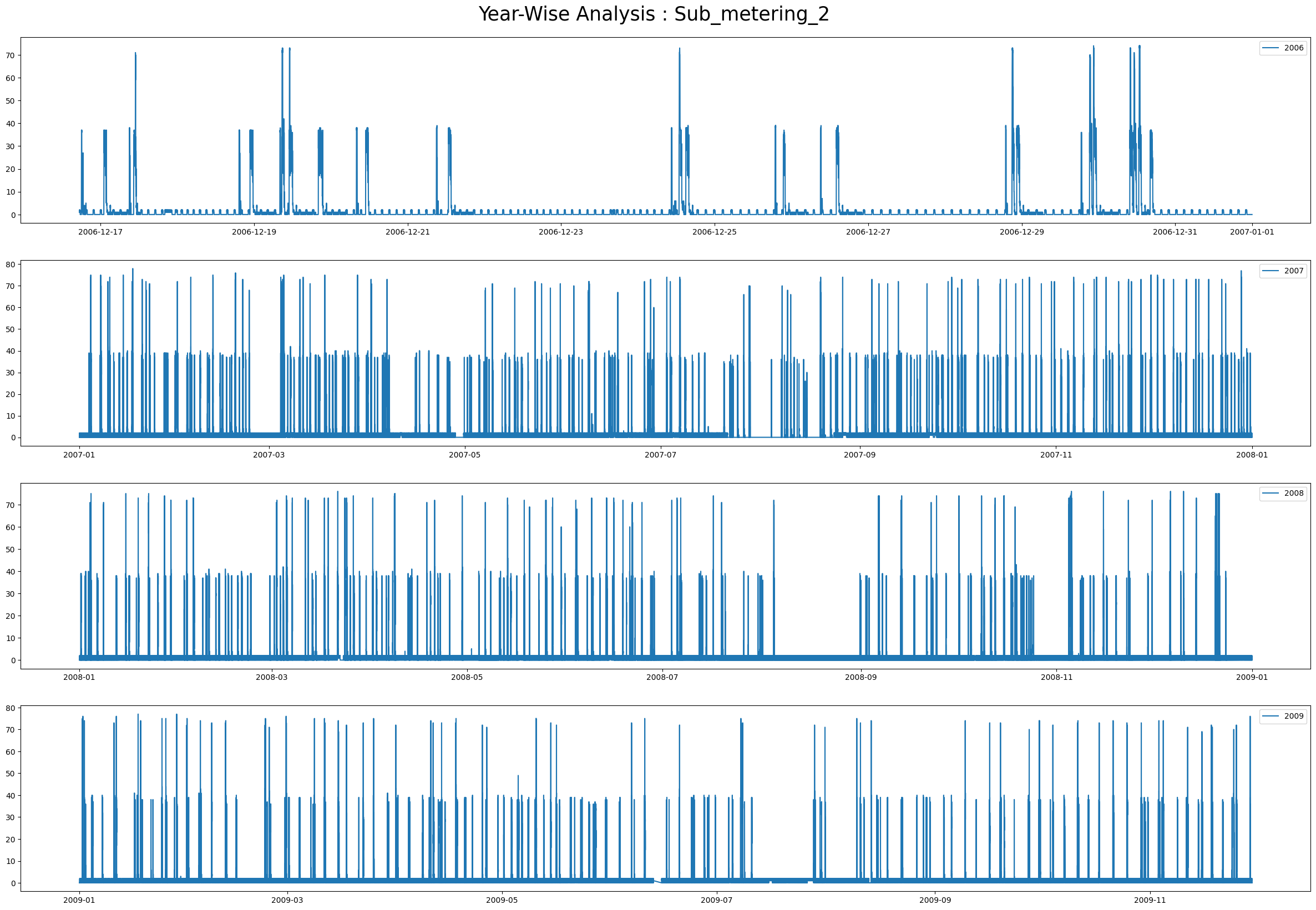


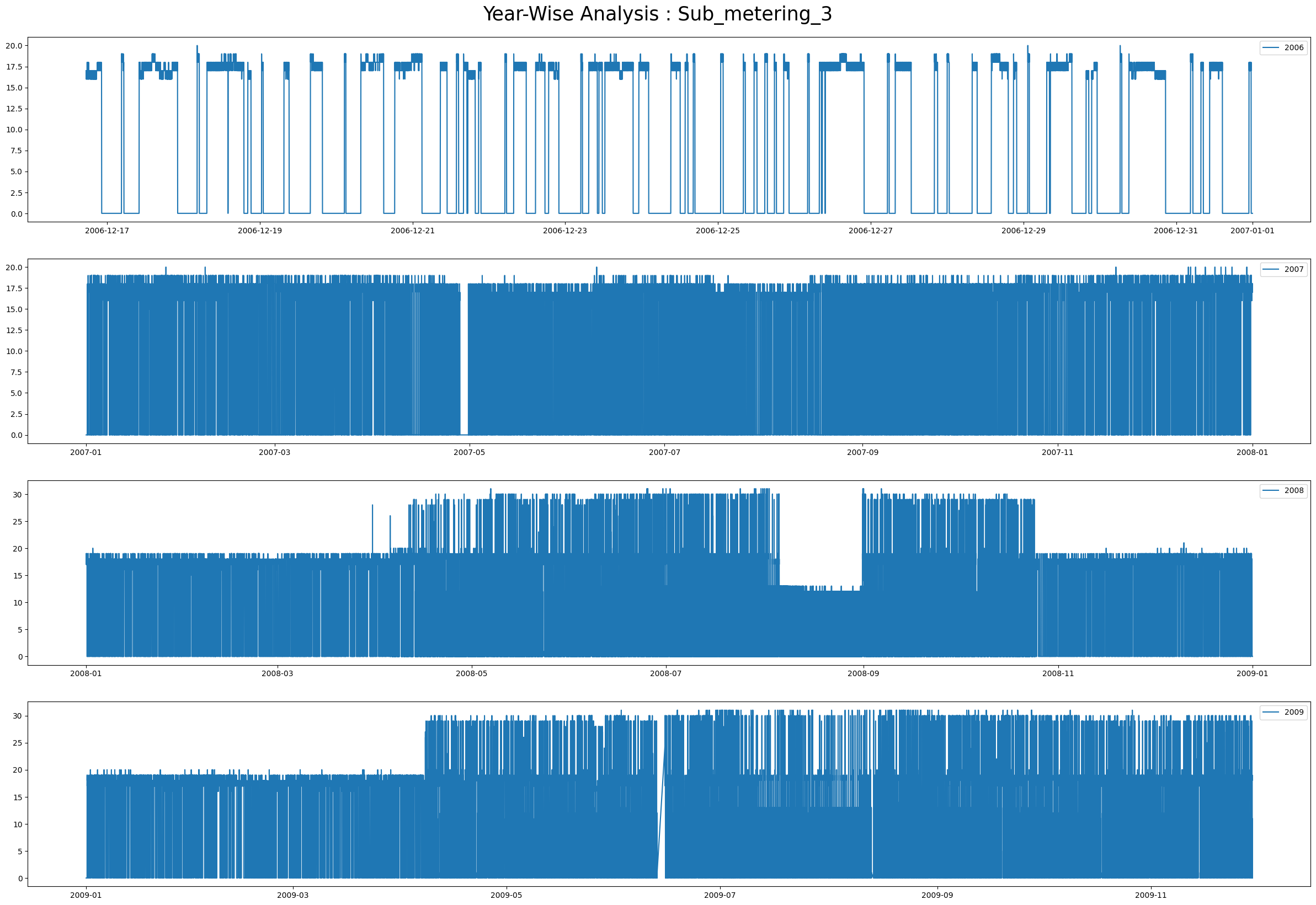












**Use of each type of plot**

* **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.