Data Cleaning and EDA with Time Series Data

This notebook holds Assignment 2.1 for Module 2 in AAI 530, Data Analytics and the Internet of Things.

In this assignment, you will go through some basic data cleaning and exploratory analysis steps on a real IoT dataset. Much of what we'll be doing should look familiar from Module 2's lab session, but Google will be your friend on the parts that are new.

General Assignment Instructions

These instructions are included in every assignment, to remind you of the coding standards for the class. Feel free to delete this cell after reading it.

One sign of mature code is conforming to a style guide. We recommend the Google Python Style Guide. If you use a different style guide, please include a cell with a link.

Your code should be relatively easy-to-read, sensibly commented, and clean. Writing code is a messy process, so please be sure to edit your final submission. Remove any cells that are not needed or parts of cells that contain unnecessary code. Remove inessential `import` statements and make sure that all such statements are moved into the designated cell.

When you save your notebook as a pdf, make sure that all cell output is visible (even error messages) as this will aid your instructor in grading your work.

Make use of non-code cells for written commentary. These cells should be grammatical and clearly written. In some of these cells you will have questions to answer. The questions will be marked by a "Q:" and will have a corresponding "A:" spot for you. *Make sure to answer every question marked with a `Q:` for full credit.*

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
#use this cell to import additional libraries or define helper functions
```

Load and clean your data

The household electric consumption dataset can be downloaded as a zip file here along with a description of the data attributes: https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption#

First we will load this data into a pandas df and do some initial discovery

`[ ]` ↳ **25 cells hidden**

Visualizing the data

We're working with time series data, so visualizing the data over time can be helpful in identifying possible patterns or metrics that should be explored with further analysis and machine learning methods.

TODO: Choose four of the variables in the dataset to visualize over time and explore methods covered in our lab session to make a line chart of the cleaned data. Your charts should be separated by variable to make them more readable.

Q: Which variables did you choose and why do you think they might be interesting to compare to each other over time? Remember that data descriptions are available at the data source link at the top of the assignment.

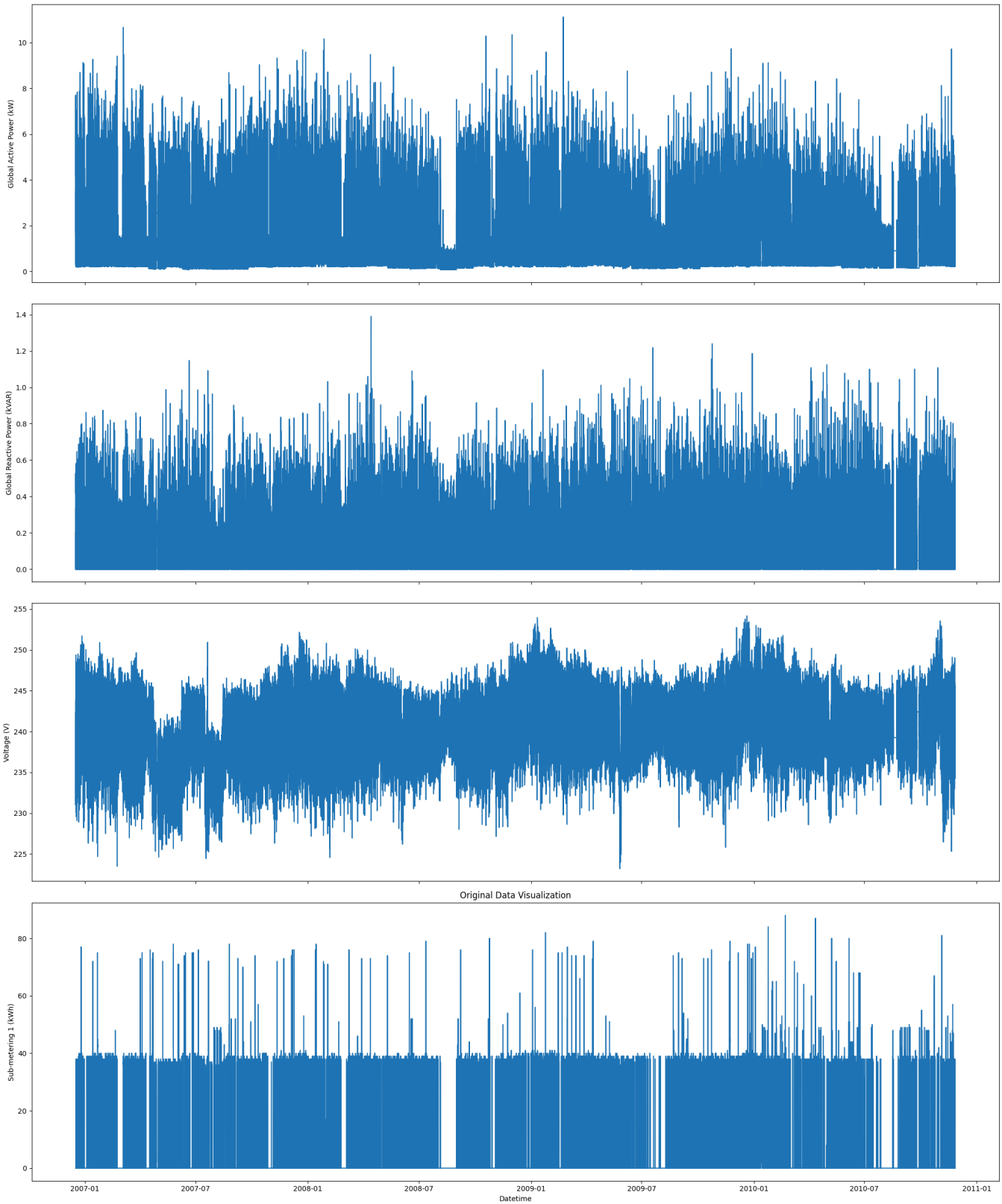
A: After going through the data descriptions are available at the data source link, it seems **Global_active_power , Global_reactive_power ,Voltage ,Sub_metering_1** are more time dependent variable and can make more sense in visualizing over time. Reasons are illustrated below :

- Overall Electricity Consumption: By visualizing **Global_active_power**, we can understand the general trend and patterns of electricity usage in the household over time.
- Reactive Power and Appliance Usage: Comparing **Global_reactive_power** with Global_active_power can reveal information about the types of appliances being used and their energy consumption characteristics. Higher reactive power often indicates the presence of inductive loads.
- Voltage Stability: Monitoring **Voltage** over time helps identify potential issues with the electricity supply, which can affect appliance performance and overall energy consumption.
- Specific Appliance Usage: By including **Sub_metering_1**, we can focus on the energy consumption of kitchen appliances, providing a more granular view of usage patterns.

```
#build your line chart here

import matplotlib.pyplot as plt
```

```
fig, axes = plt.subplots(4, 1, figsize=(20, 24), sharex=True)

# Plot Global_active_power
axes[0].plot(df['Datetime'], df['Global_active_power'])
axes[0].set_ylabel('Global Active Power (kW)')

# Plot Global_reactive_power
axes[1].plot(df['Datetime'], df['Global_reactive_power'])
axes[1].set_ylabel('Global Reactive Power (kVAR)')

# Plot Voltage
axes[2].plot(df['Datetime'], df['Voltage'])
axes[2].set_ylabel('Voltage (V)')

# Plot Sub_metering_1
axes[3].plot(df['Datetime'], df['Sub_metering_1'])
axes[3].set_ylabel('Sub-metering 1 (kWh)')
axes[3].set_xlabel('Datetime')

print('Original Data Visualization :')
plt.title('Original Data Visualization')
plt.tight_layout()
plt.show()
```

Original Data Visualization



Q: What do you notice about visualizing the raw data? Is this a useful visualization? Why or why not?

A: Following observations can be made from visualizing the raw data :

1. There appears to be a significant drop in all metrics around the beginning of 2009. Not sure why but maybe be due to a power outage or a change in data collection methods.
2. There's a clear seasonal pattern in the data, with higher values during the winter months and lower values in the summer. This is likely due to increased energy usage for heating and cooling.
3. There's a slight upward trend in Global Active Power over the time period shown. This could indicate increased electricity consumption over time.
4. There are occasional spikes and dips in the data that appear as outliers. These could indicate unusual events like power surges or temporary device malfunctions.

Although the raw data visualization provides some initial insights, I believe it's not the most effective way to analyze this type of time series data because:

- The high-frequency fluctuations and potential outliers make it difficult to identify underlying patterns and trends. The visualization appears noisy and cluttered, hindering a clear understanding of the data's behavior over time.
- The raw data represents individual measurements at a very fine granularity. This level of detail can be overwhelming and doesn't necessarily provide a meaningful overview of the overall energy consumption patterns.

.

A better approach could've been by aggregating the data over longer time intervals, like monthly averages, to smooth out fluctuations and reveal broader trends. Also applying moving averages to further reduce noise and highlight underlying patterns by calculating averages of consecutive data points.

.

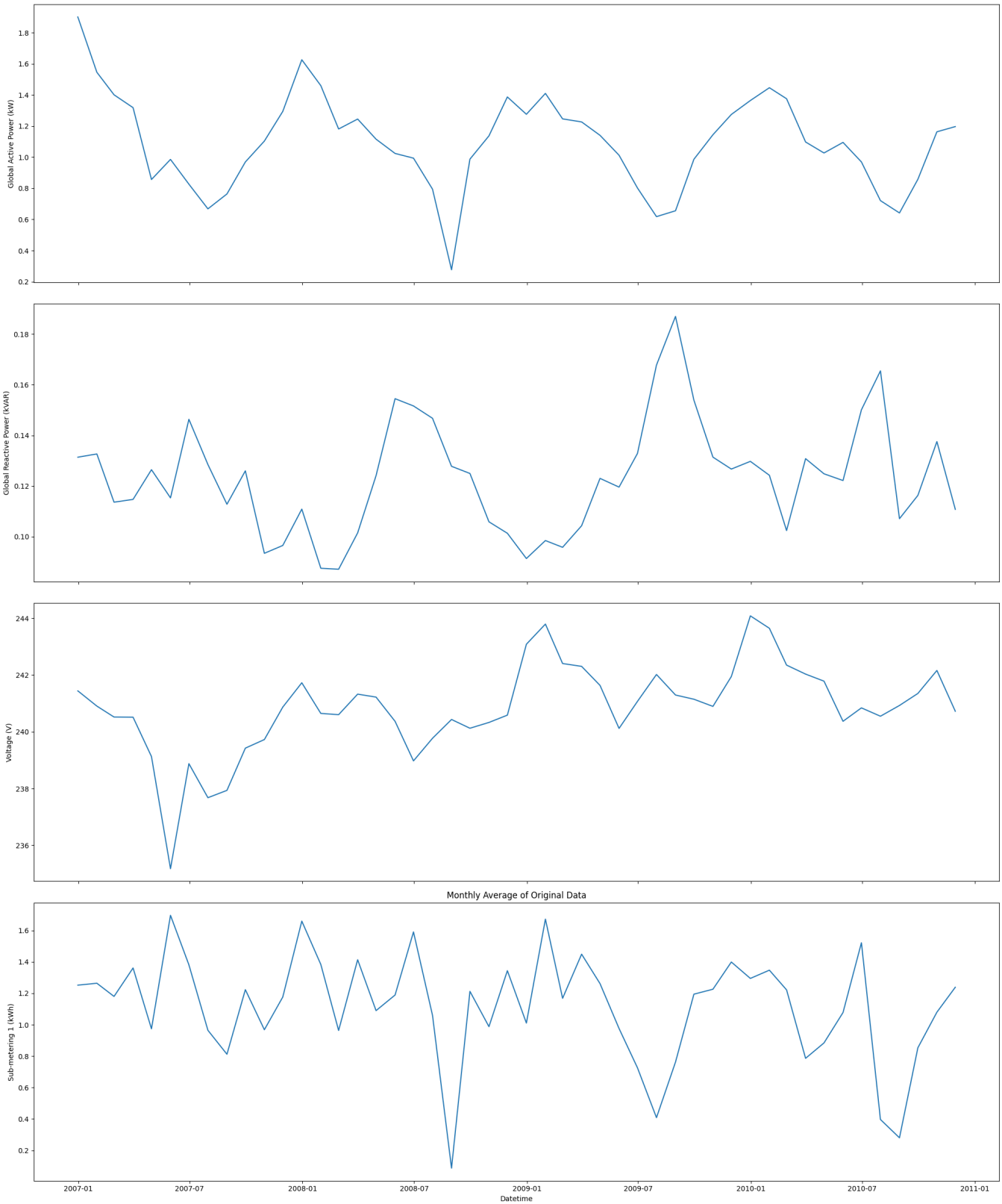
Focus on specific features of interest instead of plotting all variables together to provide a clearer picture. Identify and address potential outliers to improve the accuracy and clarity of the visualization.

TODO: Compute a monthly average for the data and plot that data in the same style as above. You should have one average per month and year (so June 2007 is separate from June 2008).

```
#compute your monthly average here
#HINT: checkout the pd.Grouper function: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Grouper.html?high

df_monthly_gap = df.groupby(pd.Grouper(key='Datetime', freq='ME'))['Global_active_power'].mean()
df_monthly_grp = df.groupby(pd.Grouper(key='Datetime', freq='ME'))['Global_reactive_power'].mean()
df_monthly_vol = df.groupby(pd.Grouper(key='Datetime', freq='ME'))['Voltage'].mean()
df_monthly_sm1 = df.groupby(pd.Grouper(key='Datetime', freq='ME'))['Sub_metering_1'].mean()
```

```
#build your linechart here
import matplotlib.pyplot as plt
fig, axes = plt.subplots(4, 1, figsize=(20, 24), sharex=True)

# Plot Global_active_power
axes[0].plot(df_monthly_gap.index, df_monthly_gap)
axes[0].set_ylabel('Global Active Power (kW)')

# Plot Global_reactive_power
axes[1].plot(df_monthly_grp.index, df_monthly_grp)
axes[1].set_ylabel('Global Reactive Power (kVAR)')

# Plot Voltage
axes[2].plot(df_monthly_vol.index, df_monthly_vol)
axes[2].set_ylabel('Voltage (V)')

# Plot Sub_metering_1
axes[3].plot(df_monthly_sm1.index, df_monthly_sm1)
axes[3].set_ylabel('Sub-metering 1 (kWh)')
axes[3].set_xlabel('Datetime')

print('Monthly Average of Original Data :')
plt.title('Monthly Average of Original Data')
plt.tight_layout()
plt.show()
```

Monthly Average of Original Data :



Q: What patterns do you see in the monthly data? Do any of the variables seem to move together?

A: Again there seems to be a clear seasonal pattern in all four graphs, with peaks and troughs occurring at regular intervals. This suggests that there is a cyclical component to the data, which could be related to factors such as weather, holidays, or other recurring events.

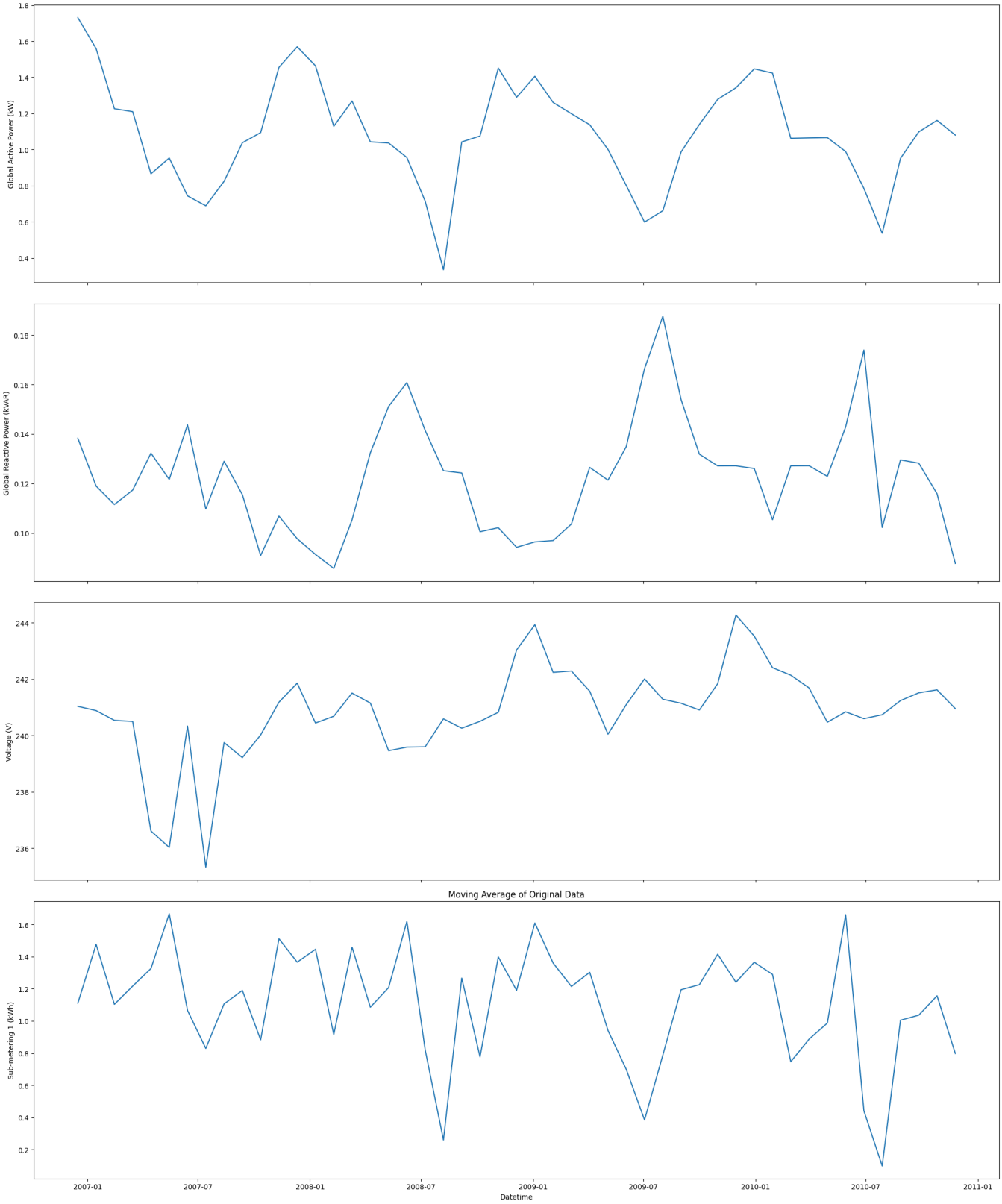
Yes the variables seem to move together, the four graphs seem to be correlated to some degree, meaning that they move in similar directions at the same time. This suggests that there might be a common underlying factor influencing all four variables.

TODO: Now compute a 30-day moving average on the original data and visualize it in the same style as above. Hint: If you use the rolling() function, be sure to consider the resolution of our data.

```
#compute your moving average here
df_moving_monthly_gap = df.groupby(pd.Grouper(key='Datetime', freq='30D'))['Global_active_power'].mean()
df_moving_monthly_grp = df.groupby(pd.Grouper(key='Datetime', freq='30D'))['Global_reactive_power'].mean()
df_moving_monthly_vol = df.groupby(pd.Grouper(key='Datetime', freq='30D'))['Voltage'].mean()
df_moving_monthly_sm1 = df.groupby(pd.Grouper(key='Datetime', freq='30D'))['Sub_metering_1'].mean()
```

```
#build your line chart on the moving average here
import matplotlib.pyplot as plt
fig, axes = plt.subplots(4, 1, figsize=(20, 24), sharex=True)

# Plot Global_active_power
axes[0].plot(df_moving_monthly_gap.index, df_moving_monthly_gap)
axes[0].set_ylabel('Global Active Power (kW)')

# Plot Global_reactive_power
axes[1].plot(df_moving_monthly_grp.index, df_moving_monthly_grp)
axes[1].set_ylabel('Global Reactive Power (kVAR)')

# Plot Voltage
axes[2].plot(df_moving_monthly_vol.index, df_moving_monthly_vol)
axes[2].set_ylabel('Voltage (V)')

# Plot Sub_metering_1
axes[3].plot(df_moving_monthly_sm1.index, df_moving_monthly_sm1)
axes[3].set_ylabel('Sub-metering 1 (kWh)')
axes[3].set_xlabel('Datetime')

print('Moving Average of Original Data :')
plt.title('Moving Average of Original Data')
plt.tight_layout()
plt.show()
```

Moving Average of Original Data :



Q: How does the moving average compare to the monthly average? Which is a more effective way to visualize this data and why?

A: The Moving average in comparision to the Monthly average is more effective when trying to capture long-term trends and remove noise. The Monthly average would have been more effective for understanding seasonal patterns but if the goal is to analyze the overall trend, the moving

average is better as it provides a continuous view of data without sudden jumps.

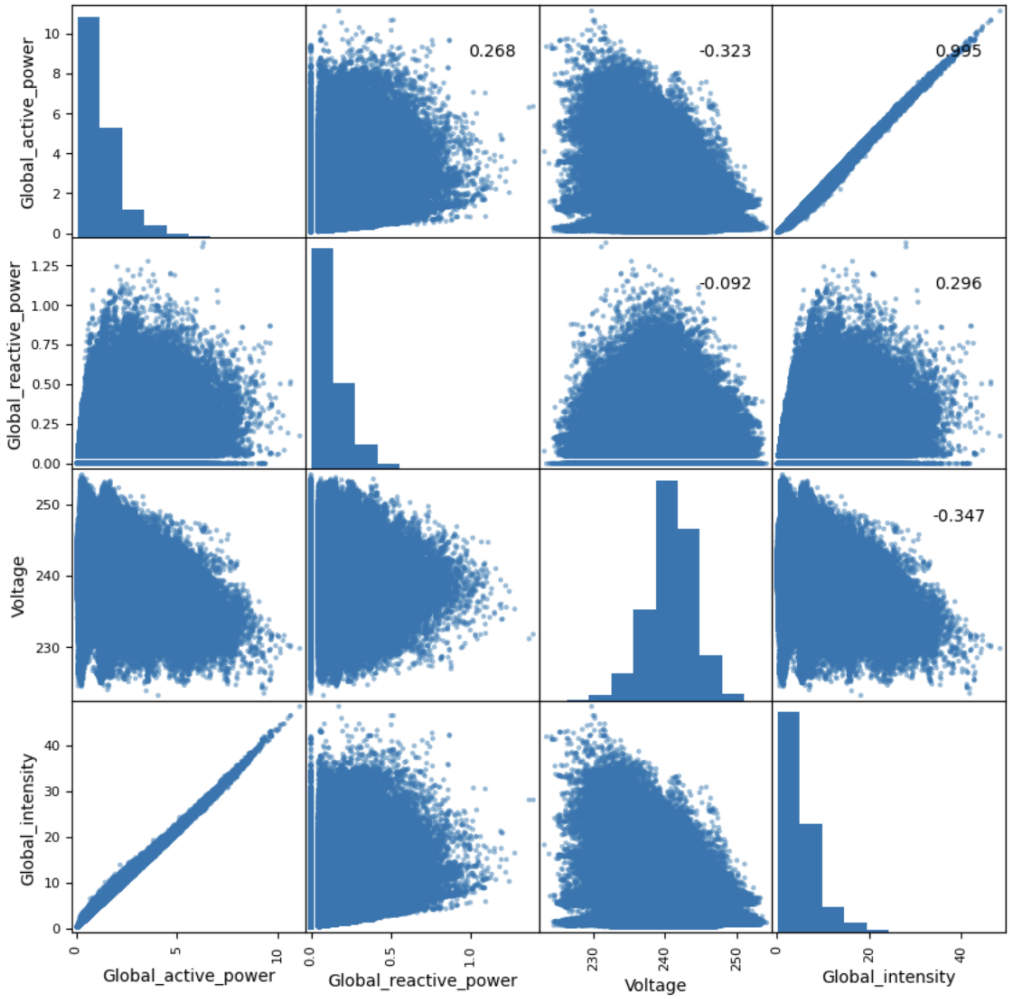
Since we are dealing with time series data where oure goal is to uncover patterns, trends, and seasonality that can inform further analysis and machine learning models. The monthly average would more effective because :

- Moving average smooths short-term fluctuations, making long-term trends clearer.
- Monthly average aggregates data per month, which may obscure short-term variations.
- Moving average is better for detecting trends and preparing data for machine learning models.
- Monthly average is useful for identifying seasonality but may miss intra-month patterns.
- Machine learning models benefit from moving average as it helps extract trend-related features.
- Moving average reduces noise, making anomalies and deviations easier to detect.

Data Covariance and Correlation

Let's take a look at the Correlation Matrix for the four global power variables in the dataset.

```
axes = pd.plotting.scatter_matrix(df[['Global_active_power', 'Global_reactive_power', 'Voltage', 'Global_intensity']], alpha=0.5,figsize = [10,10])
corr = df[['Global_active_power', 'Global_reactive_power', 'Voltage', 'Global_intensity']].corr(method = 'spearman').to_numpy() #nonlinear
for i, j in zip(*plt.np.triu_indices_from(axes, k=1)):
    axes[i, j].annotate("%.3f" %corr[i,j], (0.8, 0.8), xycoords='axes fraction', ha='center', va='center')
plt.show()
```



Q: Describe any patterns and correlations that you see in the data. What effect does this have on how we use this data in downstream tasks?

A:

Patterns and Correlations noticed :

- **Strong +ve Correlation:** Global Active Power and Global Intensity have a very strong positive correlation (correlation coefficient of 0.995). This means that as one increases, the other increases almost proportionally.
- **Weak +ve Correlation:** Global Active Power and Global Reactive Power exhibit a weak positive correlation (0.268).
- **Weak -ve Correlation:** Voltage and Global Intensity show a weak negative correlation (-0.347).
- **Almost no Correlation:** Global Reactive Power and Voltage have almost no correlation (-0.092).

Effect on downstream tasks and how it can help:

- **Feature Selection:** The strong correlation between Global Active Power and Global Intensity suggests that one of them could potentially be dropped during feature selection without losing much information. *This can help in reducing the dimensionality of the data and simplifying the model.*
- **Multicollinearity:** The strong positive correlation might also lead to multicollinearity issues in regression models. Multicollinearity occurs when two or more independent variables are highly correlated, *which can destabilize the model and make it difficult to interpret the coefficients.*
- **Model Choice:** The correlations can influence the choice of model. For example, if we are building a model to predict Global Active Power, *we might consider using Global Intensity as a strong predictor.*
- **Data Transformation:** In some cases, it might be beneficial to apply transformations (like logarithmic) to the data *to reduce the impact of strong correlations and improve model performance.*