

Lab_assignment-1

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. Dataset: https://d396qusza40orc.cloudfront.net/exdata%2Fdata%2Fhousehold_power_consumption.zip Perform the following:

Exercise 1:

1. Load the data
2. Read first 5 rows to get headers
3. Read 2900 rows that contain information on 2007-02-01 and 2007-02-02
4. Converting Date and Time variables to Date/Time format

Exercise 2: Subset the loaded data for 2007-02-01 and 2007-02-02

Exercise 3:

1. Histogram of global active power consumption
2. Global active consumption over time
3. Energy sub metering

Exercise 1:

1. Load the data:
2. Read first 5 rows to get headers

First, download the dataset from the provided link and extract it. You'll find a file named "household_power_consumption.txt." Use the following Python code to load the data into a pandas DataFrame:

```
In [1]: import pandas as pd

# Load the data into a DataFrame
data = pd.read_csv("household_power_consumption.txt", sep=";", na_values="?")

# Display the first few rows of the DataFrame
data.head()
```

Out[1]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_meteri
0	16/12/2006	17:24:00	4.216	0.418	234.84	18.4	
1	16/12/2006	17:25:00	5.360	0.436	233.63	23.0	
2	16/12/2006	17:26:00	5.374	0.498	233.29	23.0	
3	16/12/2006	17:27:00	5.388	0.502	233.74	23.0	
4	16/12/2006	17:28:00	3.666	0.528	235.68	15.8	

3. Read 2900 rows for 2007-02-01 and 2007-02-02
4. Converting Date and Time variables to Date/Time format

In [2]: `# Read 2900 rows for 2007-02-01 and 2007-02-02`

```
# Convert the 'Date' column to datetime
data['Date'] = pd.to_datetime(data['Date'])
data.head()
```

C:\Users\devje\AppData\Local\Temp\ipykernel_1364\1974775207.py:4: UserWarning: Parsing dates in %d/%m/%Y format when dayfirst=False (the default) was specified. Pass `dayfirst=True` or specify a format to silence this warning.

```
data['Date'] = pd.to_datetime(data['Date'])
```

Out[2]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering
0	2006-12-16	17:24:00	4.216	0.418	234.84	18.4	0.0
1	2006-12-16	17:25:00	5.360	0.436	233.63	23.0	0.0
2	2006-12-16	17:26:00	5.374	0.498	233.29	23.0	0.0
3	2006-12-16	17:27:00	5.388	0.502	233.74	23.0	0.0
4	2006-12-16	17:28:00	3.666	0.528	235.68	15.8	0.0

In [3]: `data1=data[(data["Date"]=="2007-02-01") | (data["Date"]=="2007-02-02")]`
`data1`

Out[3]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_meterin
66636	2007-02-01	00:00:00	0.326	0.128	243.15	1.4	
66637	2007-02-01	00:01:00	0.326	0.130	243.32	1.4	
66638	2007-02-01	00:02:00	0.324	0.132	243.51	1.4	
66639	2007-02-01	00:03:00	0.324	0.134	243.90	1.4	
66640	2007-02-01	00:04:00	0.322	0.130	243.16	1.4	
...
69511	2007-02-02	23:55:00	3.696	0.226	240.90	15.2	
69512	2007-02-02	23:56:00	3.698	0.226	241.02	15.2	
69513	2007-02-02	23:57:00	3.684	0.224	240.48	15.2	
69514	2007-02-02	23:58:00	3.658	0.220	239.61	15.2	
69515	2007-02-02	23:59:00	3.680	0.224	240.37	15.2	

2880 rows × 9 columns

Exercise 2:

Subset the loaded data for 2007-02-01 and 2007-02-02

```
In [4]: # Subset the data based on the given dates
subset_data = data[(data["Date"]=="2007-02-01") | (data["Date"]=="2007-02-02")]

# Display the subsetted data
subset_data.head()
```

Out[4]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_meterin
66636	2007-02-01	00:00:00	0.326	0.128	243.15	1.4	
66637	2007-02-01	00:01:00	0.326	0.130	243.32	1.4	
66638	2007-02-01	00:02:00	0.324	0.132	243.51	1.4	
66639	2007-02-01	00:03:00	0.324	0.134	243.90	1.4	
66640	2007-02-01	00:04:00	0.322	0.130	243.16	1.4	

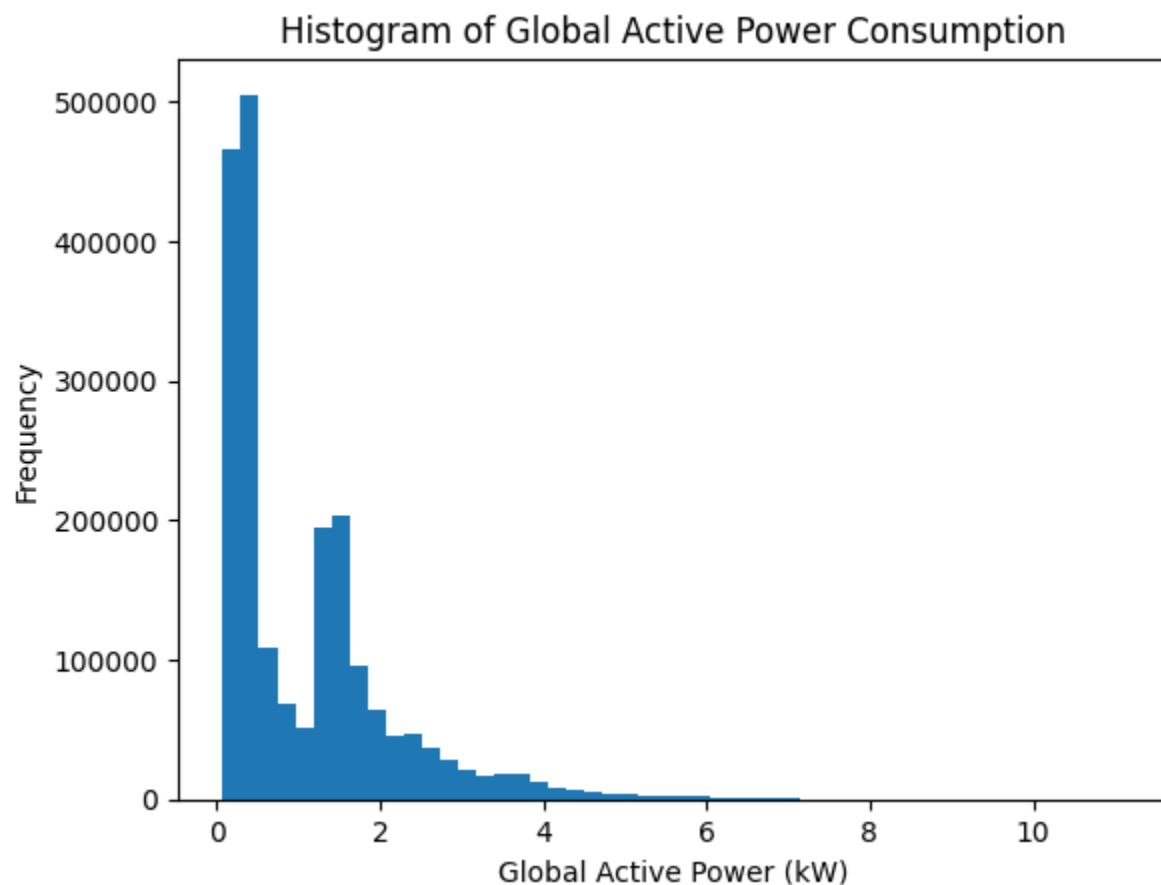
Exercise 3:

1. Histogram of global active power consumption

Create a histogram: To create a histogram of the electric power consumption, you can plot the "Global_active_power" column using matplotlib or any other plotting library:

```
In [10]: import matplotlib.pyplot as plt

# Plot a histogram of global active power consumption
plt.hist(data['Global_active_power'].dropna(), bins=50)
# plt.hist(data['Global_active_power'].dropna())
plt.xlabel('Global Active Power (kW)')
plt.ylabel('Frequency')
plt.title('Histogram of Global Active Power Consumption')
plt.figure(figsize=(12,12))
plt.show()
```

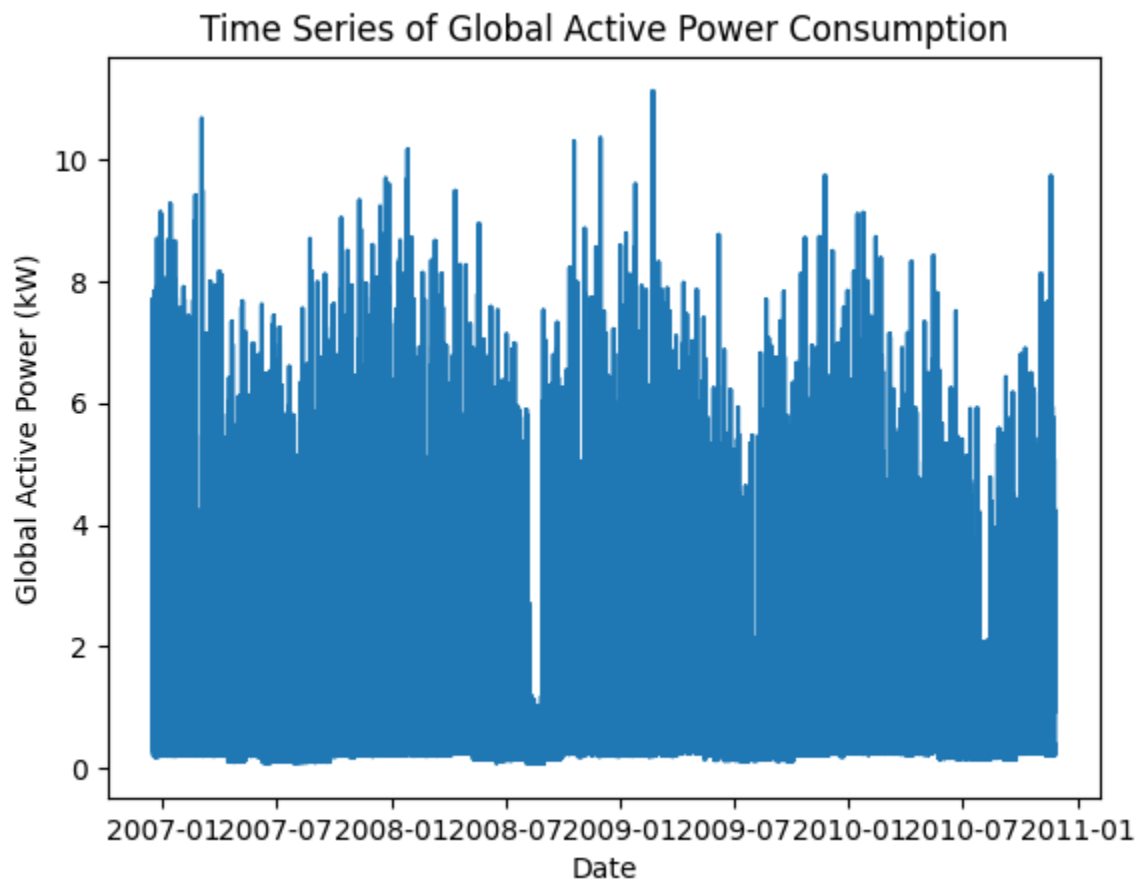


<Figure size 1200x1200 with 0 Axes>

2. Global active consumption over time

Create a time series: To create a time series plot of the electric power consumption over time, you can plot the "Global_active_power" column against the "Date" column:

```
In [11]: # Create a time series plot of global active power consumption
plt.plot(data['Date'], data['Global_active_power'])
plt.xlabel('Date')
plt.ylabel('Global Active Power (kW)')
plt.title('Time Series of Global Active Power Consumption')
plt.figure(figsize=(12,10))
plt.show()
```



<Figure size 1200x1000 with 0 Axes>

3. Energy sub metering

Create a plot for sub-metering: To create a plot for sub-metering values, you can plot the relevant columns from the dataset. Assuming the sub-metering values are stored in the columns "Sub_metering_1," "Sub_metering_2," and "Sub_metering_3," you can plot them as follows:

```
In [12]: # Create a plot for sub-metering
plt.plot(data['Date'], data['Sub_metering_1'], label='Sub_metering_1')
plt.plot(data['Date'], data['Sub_metering_2'], label='Sub_metering_2')
plt.plot(data['Date'], data['Sub_metering_3'], label='Sub_metering_3')
plt.xlabel('Date')
plt.ylabel('Sub-metering Values')
plt.title('Sub-metering Values Over Time')
plt.legend()

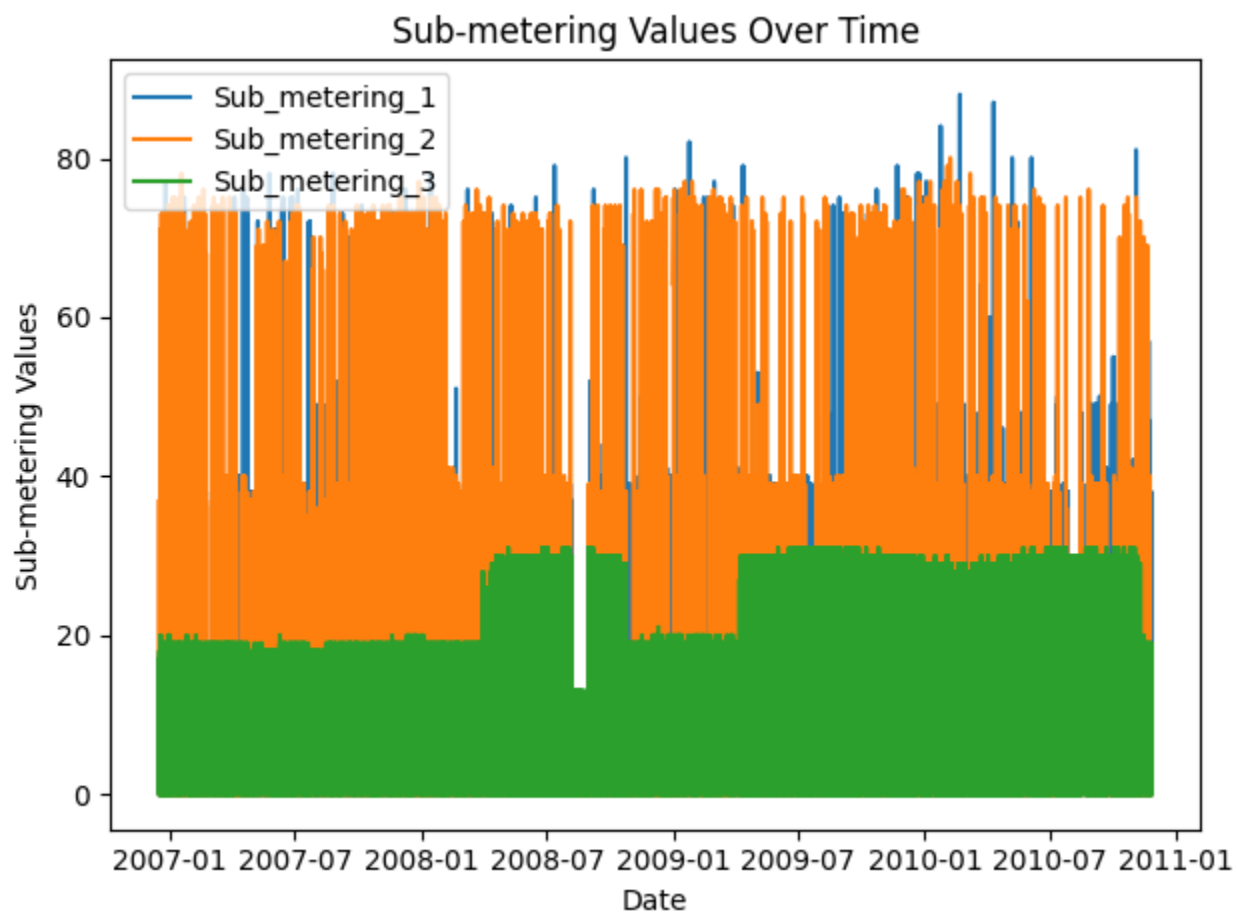
plt.tight_layout()
plt.figure(figsize=(12,14))
plt.show()
```

C:\Users\devje\AppData\Local\Temp\ipykernel_1364\4225496983.py:10: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.

```
plt.tight_layout()
```

D:\Python311\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.

```
fig.canvas.print_figure(bytes_io, **kw)
```



<Figure size 1200x1400 with 0 Axes>

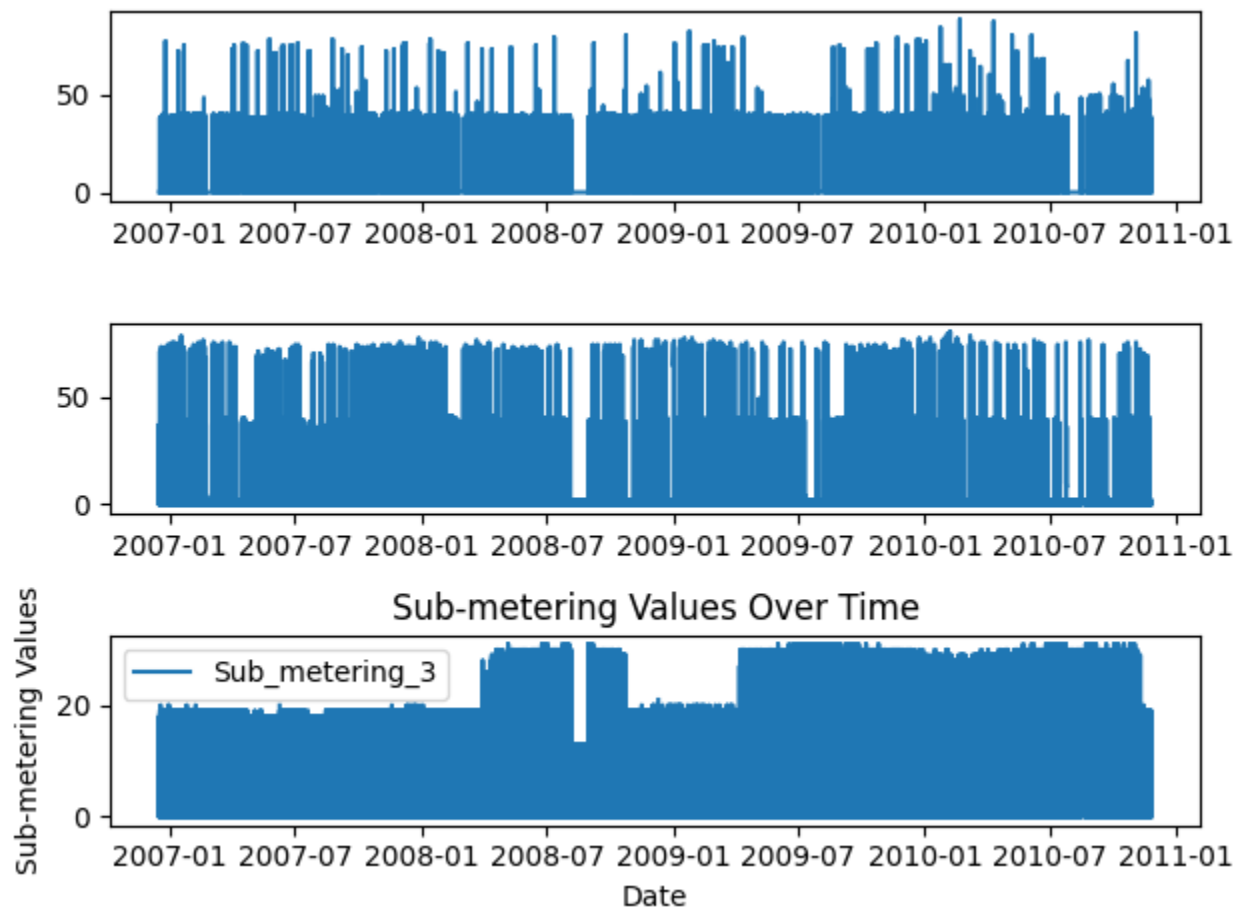
```
In [13]: # subplotting
plt.subplot(3, 1, 1)
plt.plot(data['Date'], data['Sub_metering_1'], label='Sub_metering_1')
plt.subplot(3, 1, 2)

plt.plot(data['Date'], data['Sub_metering_2'], label='Sub_metering_2')

plt.subplot(3, 1, 3)
plt.plot(data['Date'], data['Sub_metering_3'], label='Sub_metering_3')

plt.xlabel('Date')
plt.ylabel('Sub-metering Values')
plt.title('Sub-metering Values Over Time')
plt.legend()

plt.tight_layout()
plt.figure(figsize=(12,14))
plt.show()
```



<Figure size 1200x1400 with 0 Axes>

In []: