```
#importing necessary libaries
import numpy as np
import pandas as pd
```

```
df = pd.read_csv("/content/4th December to 10th December.csv")
```

```
df.head()
```

	date	timestamp	A41 - PAVILLION VENTILATION MCC/82/01 - E (Wh)	A28 -LTHW PANTROOM MCC/07/02 - E (Wh)	A16 - ATS/25/04 LIFE SAFETY BUSBAR RISER NORTH BB/08 - E (Wh)	A17 - ATS/25/05 LIFE SAFETY BUSBAR RISER SOUTH BB/09 - E (Wh)	A12 - LIFT SWITCHBOARD 1 - E (Wh)	A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)	A25 - LEVEL 25 PLANTROOM MCC/25/04 - E (Wh)	A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07 - E (Wh)	...	A63 - ATS/07/02 WET RISER PUMP 1 - E (Wh)	A64 - ATS/07/03 WET RISER PUMP 2 - E (Wh)	A56 - COMMS 2 DB/07/COMMS02 - E (Wh)	A57 - BASEMENT DB/07/LL01 (Basement Refuge Store & Sodexo Store) - E (Wh)
0	04/12/2023 00:00	1701648000	142.579663	3655.455159	274.872294	626.596649	352.532740	2366.085565	34.541659	3351.967412	...	0	0	0.0	476.450611
1	04/12/2023 00:15	1701648900	142.484616	4095.420790	513.493000	742.056273	353.147367	2475.881402	33.357095	3531.962439	...	0	0	0.0	476.350420
2	04/12/2023 00:30	1701649800	142.060966	4268.665318	357.646083	757.917378	624.765136	2382.372539	28.355709	3866.194248	...	0	0	0.0	476.896876
3	04/12/2023 00:45	1701650700	142.485653	4260.127101	236.028031	388.788575	506.089891	2365.002583	36.376148	3784.700003	...	0	0	0.0	476.344070
4	04/12/2023 01:00	1701651600	142.514888	4266.559322	235.477628	367.646619	493.515686	2363.367291	31.884987	3269.759399	...	0	0	0.0	480.973422

5 rows × 50 columns

```
#convert the date column to datetime format
df['date'] = pd.to_datetime(df['date'])
```

```
df["date"].head()
```

```
0   2023-04-12 00:00:00
1   2023-04-12 00:15:00
2   2023-04-12 00:30:00
3   2023-04-12 00:45:00
4   2023-04-12 01:00:00
Name: date, dtype: datetime64[ns]
```

```
start_date = '2023-04-12'
end_date = '2023-07-12'
```

```
# Filter rows based on the date range
filtered_df = df[(df['date'] >= start_date) & (df['date'] <= end_date)]
```

```
# Save the filtered dataframe to a new CSV file
filtered_df.to_csv('filtered_data.csv', index=False)
```

```
df1 = pd.read_csv("filtered_data.csv")
```

```
df1.head()
```

	date	timestamp	A41 - PAVILLION VENTILATION MCC/82/01 - E (Wh)	A28 -LTHW PANTROOM MCC/07/02 - E (Wh)	A16 - ATS/25/04 LIFE SAFETY BUSBAR RISER NORTH BB/08 - E (Wh)	A17 - ATS/25/05 LIFE SAFETY BUSBAR RISER SOUTH BB/09 - E (Wh)	A12 - LIFT SWITCHBOARD 1 - E (Wh)	A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)	A25 - LEVEL 25 PLANTROOM MCC/25/04 - E (Wh)	A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07 - E (Wh)	...	A63 - ATS/07/02 WET RISER PUMP 1 - E (Wh)	A64 - ATS/07/03 WET RISER PUMP 2 - E (Wh)	A56 - COMMS 2 DB/07/COMMS02 - E (Wh)	A57 - BASEMENT DB/07/LL01 (Basement Refuge Store & Sodexo Store) - E (Wh)	D S
0	2023-04-12 00:00:00	1701648000	142.579663	3655.455159	274.872294	626.596649	352.532740	2366.085565	34.541659	3351.967412	...	0	0	0.0	476.450611	
1	2023-04-12 00:15:00	1701648900	142.484616	4095.420790	513.493000	742.056273	353.147367	2475.881402	33.357095	3531.962439	...	0	0	0.0	476.350420	
2	2023-04-12 00:30:00	1701649800	142.060966	4268.665318	357.646083	757.917378	624.765136	2382.372539	28.355709	3866.194248	...	0	0	0.0	476.896876	
3	2023-04-12 00:45:00	1701650700	142.485653	4260.127101	236.028031	388.788575	506.089891	2365.002583	36.376148	3784.700003	...	0	0	0.0	476.344070	
4	2023-04-12 01:00:00	1701651600	142.514888	4266.559322	235.477628	367.646619	493.515686	2363.367291	31.884987	3269.759399	...	0	0	0.0	480.973422	

5 rows × 50 columns

```
df1.shape
```

```
(289, 50)
```

Task 1

Energy Consumption Data Conversion

```
# List of columns to convert from watts to kilowatts
columns_to_convert = df1.iloc[:, 2:]
# Convert the columns from watts to kilowatts
columns_to_convert= columns_to_convert / 1000
```

Double-click (or enter) to edit

```
columns_to_convert.head()
```

	A41 - PAVILLION VENTILATION MCC/82/01 - E (Wh)	A28 -LTHW PANTROOM MCC/07/02 - E (Wh)	A16 - ATS/25/04 LIFE SAFETY BUSBAR RISER NORTH BB/08 - E (Wh)	A17 - ATS/25/05 LIFE SAFETY BUSBAR RISER SOUTH BB/09 - E (Wh)	A12 - LIFT SWITCHBOARD 1 - E (Wh)	A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)	A25 - LEVEL 25 PLANTROOM MCC/25/04 - E (Wh)	A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07 - E (Wh)	A8 - LIFT SWITCHBOARD 2 - E (Wh)	A9 - CHW PLANTROOM MCC/07/01 - E (Wh)	...	A63 - ATS/07/02 WET RISER PUMP 1 - E (Wh)	A64 - ATS/07/03 WET RISER PUMP 2 - E (Wh)	A56 - COMMS 2 DB/07/COMMS02 - E (Wh)	A57 - BASEMENT DB/07/LL01 (Basement Refuge Store & Sodexo Store) - E (Wh)	A BASE DB/07/ (Base Ten Statio E
0	0.142580	3.655455	0.274872	0.626597	0.352533	2.366086	0.034542	3.351967	0.978649	21.545979	...	0.0	0.0	0.0	0.476451	0.19
1	0.142485	4.095421	0.513493	0.742056	0.353147	2.475881	0.033357	3.531962	1.287779	21.728920	...	0.0	0.0	0.0	0.476350	0.16
2	0.142061	4.268665	0.357646	0.757917	0.624765	2.382373	0.028356	3.866194	1.048797	21.684763	...	0.0	0.0	0.0	0.476897	0.07
3	0.142486	4.260127	0.236028	0.388789	0.506090	2.365003	0.036376	3.784700	1.299497	21.601295	...	0.0	0.0	0.0	0.476344	0.07
4	0.142515	4.266559	0.235478	0.367647	0.493516	2.363367	0.031885	3.269759	1.187373	21.638796	...	0.0	0.0	0.0	0.480973	0.07

5 rows × 48 columns

```
total_columns_to_convert = columns_to_convert.sum()
```

```
#calculate the total sum of all circuits
total_columns_to_convert.sum()
```

```
15866.034793437513
```

The total energy consumed between 4th December and 7th December is approximately 15866kWh

```
Start coding or generate with AI.
```

Double-click (or enter) to edit

TASK 2

Consumption Analysis

```
#df = dataframe for the initial dataset
df2 = df.iloc[:,2:]
```

```
total_sum = df2.sum()
```

```
# Identify the top 5 consuming equipments
top_5_equipments =total_sum.nlargest(5)
```

```
top_5_equipments
```

```
A9 - CHW PLANTROOM MCC/07/01 - E (Wh)                    9.924247e+06
A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07  - E (Wh)     3.507280e+06
A14 - OFFICES AHU'S WEST MCC/70/04 - E (Wh)              2.995847e+06
A15 - OFFICES AHU'S EAST MCC/70/05 - E (Wh)              2.427284e+06
A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)     2.185489e+06
dtype: float64
```

```
#energy consumption in kilowatts for better evaluation
top_5_circuits_in_watts = top_5_circuits/1000
top_5_circuits_in_watts
```

```
A9 - CHW PLANTROOM MCC/07/01 - E (Wh)                    9924.247142
A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07  - E (Wh)     3507.280078
A14 - OFFICES AHU'S WEST MCC/70/04 - E (Wh)              2995.847161
A15 - OFFICES AHU'S EAST MCC/70/05 - E (Wh)              2427.283906
A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)     2185.489432
dtype: float64
```

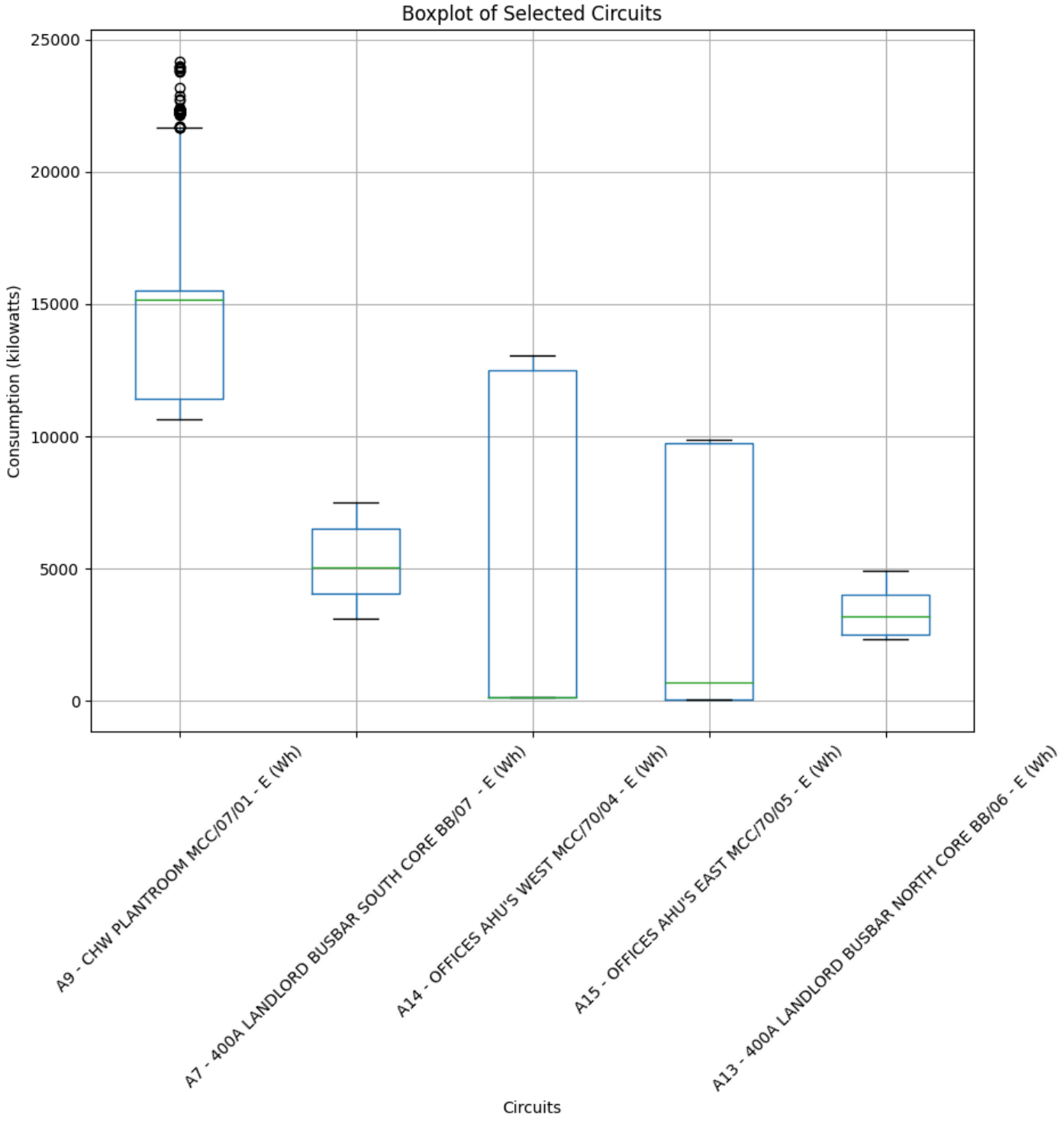
The reasons for high energy usage in these circuits are :

1) availability of high power appliances in the areas where they are situated.

2)continuous or non-stop usage of these appliances as in the usage of air conditioners in homes and offices, computers, etc

3)the usage of heavy machinery as can be found in plantrooms that has the highest record of energy usage.

```
#summary statistics of top 5 circuits
import matplotlib.pyplot as plt
columns_of_interest = ["A9 - CHW PLANTROOM MCC/07/01 - E (Wh)", "A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07  - E (Wh)", "A14 - OFFICES AHU'S WEST MCC/70/04 - E (Wh)", "A15 - OFFICE
df_selected = df[columns_of_interest]
plt.figure(figsize=(10, 8))
df_selected.boxplot()
plt.title('Boxplot of Selected Circuits')
plt.xlabel('Circuits')
plt.ylabel('Consumption (kilowatts)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```


Boxplot of Selected Circuits
Consumption (kilowatts)
25000
20000
15000
10000
5000
0
A9 - CHW PLANTROOM MCC/07/01 - E (Wh)
A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07 - E (Wh)
A14 - OFFICES AHU'S WEST MCC/70/04 - E (Wh)
A15 - OFFICES AHU'S EAST MCC/70/05 - E (Wh)
A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)
Circuits

```
df["A9 - CHW PLANTROOM MCC/07/01 - E (Wh)"].describe()
```

```
count      672.000000
mean     14768.224914
std       3443.543834
min      10641.274890
25%      11427.500858
50%      15172.281705
75%      15518.556690
max      24154.685070
Name: A9 - CHW PLANTROOM MCC/07/01 - E (Wh), dtype: float64
```

```
df["A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07  - E (Wh)"].describe()
```

```
count     672.000000
mean     5219.166783
std      1280.559553
min      3105.003989
25%      4066.361223
50%      5050.923660
75%      6516.785467
max      7498.963253
Name: A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07  - E (Wh), dtype: float64
```

```
df["A14 - OFFICES AHU'S WEST MCC/70/04 - E (Wh)"].describe()
```

```
count      672.000000
mean      4458.105895
std       5865.786541
min        122.410004
25%        124.557162
50%        125.462723
75%      12513.887960
max      13067.992220
Name: A14 - OFFICES AHU'S WEST MCC/70/04 - E (Wh), dtype: float64
```

```
df["A15 - OFFICES AHU'S EAST MCC/70/05 - E (Wh)"].describe()
```

```
count     672.000000
mean     3612.029622
std      4468.964290
min        41.820042
25%        43.670431
50%       698.475728
75%      9751.293777
max      9856.068424
Name: A15 - OFFICES AHU'S EAST MCC/70/05 - E (Wh), dtype: float64
```

```
df["A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)"].describe()
```

```
count     672.000000
mean     3252.216417
std       717.332729
min      2319.157372
25%      2509.367143
```

```
50%      3185.298637
75%      4006.603656
max      4912.509507
Name: A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh), dtype: float64
```

From the above descriptive or summary statistics, it can be inferred that:

1) A9, having the highest energy usage has both the highest mean, minimum and maximum record usage per 15 mins

2)A14, despite being the 3rd on the list, has the highest variability (Standard deviation), indicating an ununiformity in the energy usage over time, this may be due to the usage of some heavy machines at some point.

3)A9 has outliers exemplifying extremely high energy consumption as shown in the boxplot.

Double-click (or enter) to edit

Task 3

***Off-Hours Energy Usage**

```
# Filter the DataFrame to include only off hours
off_hours = ~((df['date'].dt.weekday < 5) & (df['date'].dt.hour >= 6) & (df['date'].dt.hour <= 19))
off_hours_df = df[off_hours]
```

```
filtered_off_hours_df = off_hours_df.iloc[:, 2:]
```

```
total_off_hours = filtered_off_hours_df.sum()
```

```
#total off hours energy consumption
total_off_hours.sum()
```

```
13219046.150478827
```

```
# Filter the DataFrame to include only working hours
working_hours = (df['date'].dt.weekday < 5) & (df['date'].dt.hour >= 6) & (df['date'].dt.hour <= 19)
working_hours_df = df[working_hours]
filtered_working_hours_df = working_hours_df.iloc[:, 2:]
total_working_hours = filtered_working_hours_df.sum()
total_working_hours.sum()
```

```
18631351.092450034
```

Off-hours energy consumption: **13219046.2 Wh**

Working hours energy consumption : **18631351.1 Wh**

Inference:

The off-hours energy consumption is considerably high, meaning that some machines and operations are still on-going or in service even during the off hours.

```
Start coding or generate with AI.
```

Task 4

Peak Demand Analysis

```
df_demand = df.iloc[:, 2:]
df['total_energy'] = df_demand.sum(axis=1)
```

```
# Sort the DataFrame by total energy consumption in descending order
df_sorted = df.sort_values(by='total_energy', ascending=False)
```

```
# Identify the top N periods with the highest total energy consumption
top_n = 10
peak_periods = df_sorted.head(top_n)
```

```
# Display the peak periods
print("Top 10 Peak Demand Periods:")
print(peak_periods[['date', 'total_energy']])
```

```
Top 10 Peak Demand Periods:
                    date  total_energy
136 2023-05-12 10:00:00  77116.492740
135 2023-05-12 09:45:00  76885.790387
329 2023-07-12 10:15:00  76810.943428
228 2023-06-12 09:00:00  76755.130720
131 2023-05-12 08:45:00  76508.325686
37  2023-04-12 09:15:00  76368.447660
130 2023-05-12 08:30:00  76164.441311
233 2023-06-12 10:15:00  76090.136397
35  2023-04-12 08:45:00  76035.185786
229 2023-06-12 09:15:00  76028.265514
```

```
Start coding or generate with AI.
```

Above are the top 10 peak demand periods for energy consumption. However, energy usage and cost can be optimised by:

1) Shifting the using of heaving machines and power consuming activities to other periods

2) if 1 above can not be possible, we should minimize the activities done during these periods or restrict them to power-consuming activities alone, shifting moderately/low power-consuming activities to other periods.

3) diversification of source of energy and storing up of energy for peak periods.

```
df.head()
```

	date	timestamp	A41 - PAVILLION VENTILATION MCC/82/01 - E (Wh)	A28 -LTHW PANTROOM MCC/07/02 - E (Wh)	A16 - ATS/25/04 LIFE SAFETY BUSBAR RISER NORTH BB/08 - E (Wh)	A17 - ATS/25/05 LIFE SAFETY BUSBAR RISER SOUTH BB/09 - E (Wh)	A12 - LIFT SWITCHBOARD 1 - E (Wh)	A13 - 400A LANDLORD BUSBAR NORTH CORE BB/06 - E (Wh)	A25 - LEVEL 25 PLANTROOM MCC/25/04 - E (Wh)	A7 - 400A LANDLORD BUSBAR SOUTH CORE BB/07 - E (Wh)	...	A64 - ATS/07/03 WET RISER PUMP 2 - E (Wh)	A56 - COMMS 2 DB/07/COMMS02 - E (Wh)	A57 - BASEMENT DB/07/LL01 (Basement Refuge Store & Sodexo Store) - E (Wh)	A58 - BASEMENT DB/07/LL02 (Basement Tennant Sub Station) - E (Wh)
0	2023-04-12 00:00:00	1701648000	142.579663	3655.455159	274.872294	626.596649	352.532740	2366.085565	34.541659	3351.967412	...	0	0.0	476.450611	194.939042
1	2023-04-12 00:15:00	1701648900	142.484616	4095.420790	513.493000	742.056273	353.147367	2475.881402	33.357095	3531.962439	...	0	0.0	476.350420	167.765469
2	2023-04-12 00:30:00	1701649800	142.060966	4268.665318	357.646083	757.917378	624.765136	2382.372539	28.355709	3866.194248	...	0	0.0	476.896876	71.903968
3	2023-04-12 00:45:00	1701650700	142.485653	4260.127101	236.028031	388.788575	506.089891	2365.002583	36.376148	3784.700003	...	0	0.0	476.344070	71.944860
4	2023-04-12 01:00:00	1701651600	142.514888	4266.559322	235.477628	367.646619	493.515686	2363.367291	31.884987	3269.759399	...	0	0.0	480.973422	71.834222

5 rows × 51 columns

Task 5

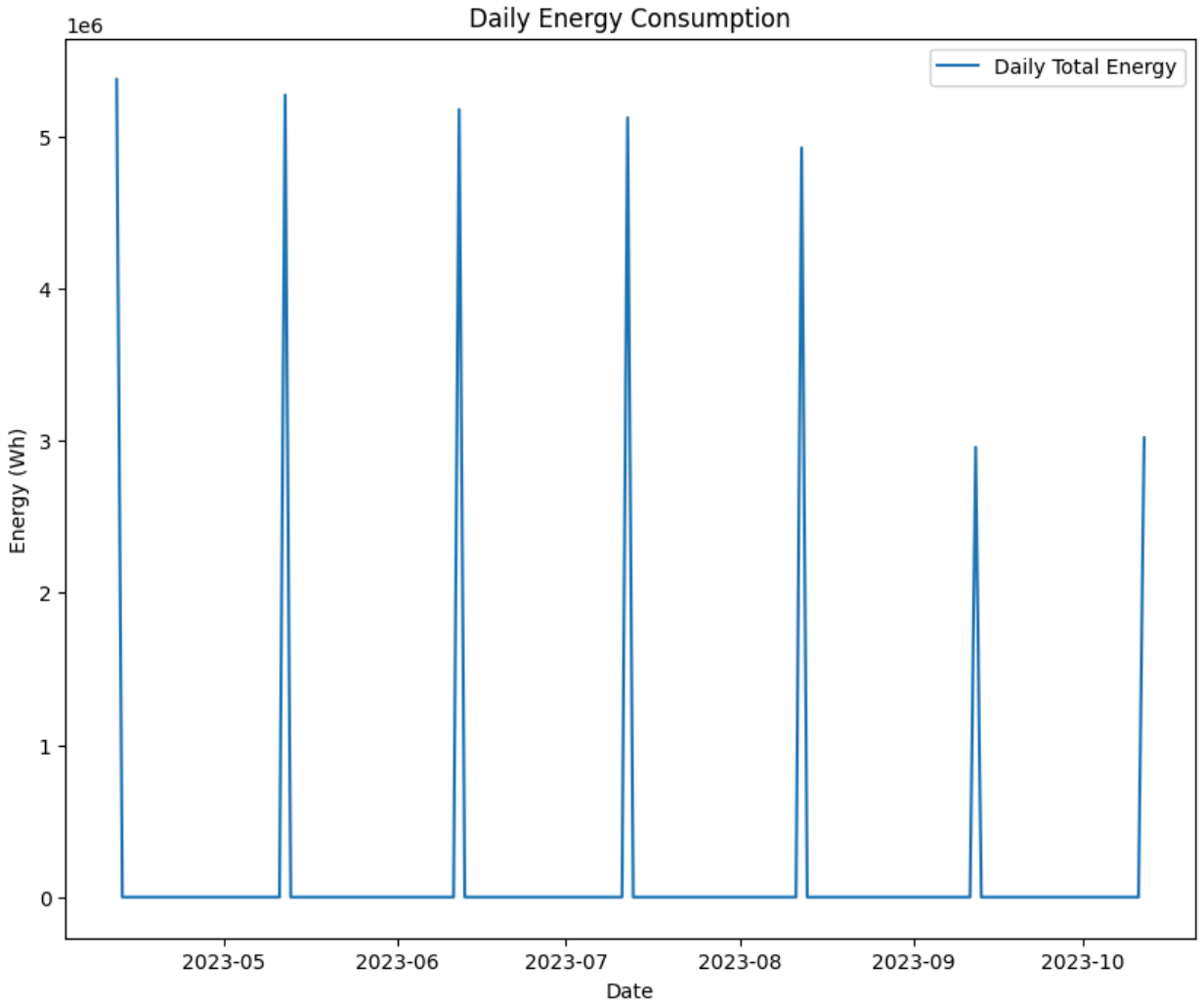
Trend Identification

```
# Set the datetime column as the index
df.set_index('date', inplace=True)
daily_data = df.resample('D').sum()
```

```
plt.figure(figsize=(10, 8))

# Plot daily data
df['total_energy'] = df_demand.sum(axis=1)
plt.subplot()
plt.plot(daily_data.index, daily_data['total_energy'], label='Daily Total Energy')
plt.title('Daily Energy Consumption')
plt.xlabel('Date')
plt.ylabel('Energy (Wh)')
plt.legend()
```

<matplotlib.legend.Legend at 0x78f670231030>



Start coding or generate with AI.

The observed trend indicates an almost uniform decrease in energy consumption between 2023-05 and 2023-08 and a substantial decrease towards 2023-09 and 2023-10. Generally, the energy consumption decreased across the days which may be due to reduction in the use of heavy machines and increased use of man power, or increased manual operation and less dependence on machinery operations.

Double-click (or enter) to edit

Task 5

Environmental Impact Estimation

The carbon footprint is calculated by multiplying the input value in kWh by 0.85 (emission factor)

```
#filter initial dataframe
df_final = df.iloc[:, 2:]
df_sum = df_final.sum()
```

```
df_sum.sum()
```

```
31753640.625240564
```

```
#calculate the carbon footprint
carbon_footprint = df_sum.sum()* 0.85
carbon_footprint
```

```
26990594.531454477
```

The carbon footprint = 26990595 kg CO2 per Wh

```
Start coding or generate with AI.
```