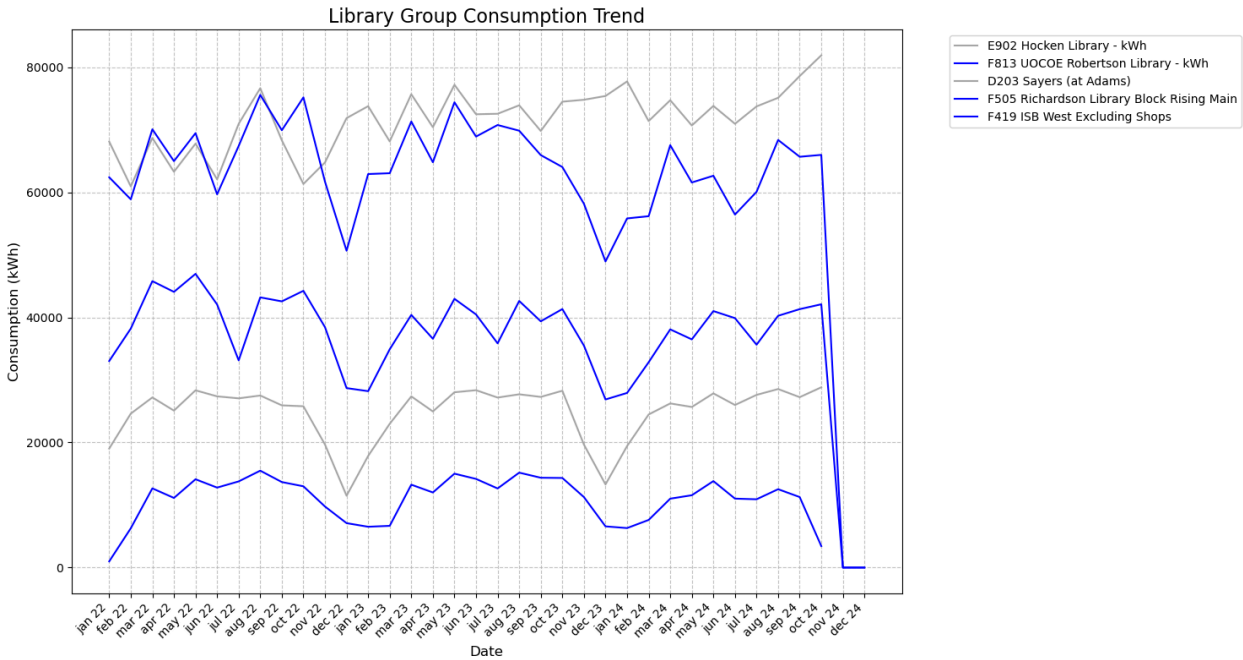
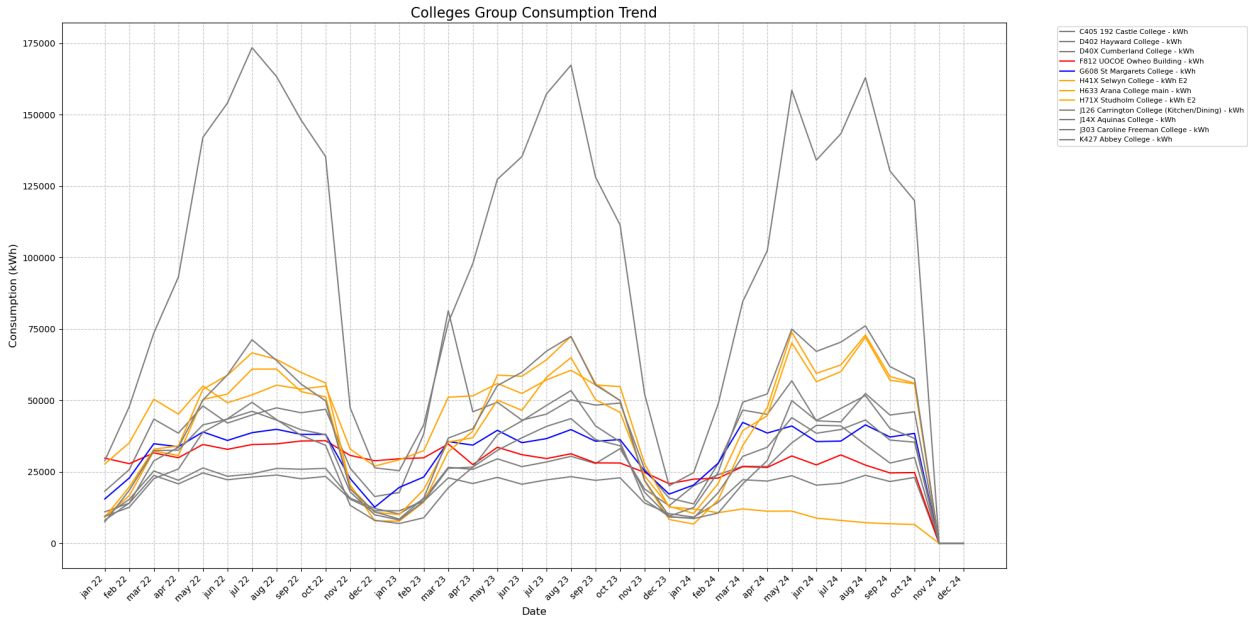
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Set the 'Meter_Location' column as the index
df.set_index('Meter_Location', inplace=True)

# Define the Libraries group
libraries = [
    "E902 Hocken Library - kWh",
    "F813 UOCOE Robertson Library - kWh",
    "D203 Sayers (at Adams)",
    "F419 ISB West Excluding Shops",
    "F505 Richardson Library Block Rising Main"
]

# Filter the dataframe to include only the Libraries group
libraries_df = df.loc[df.index.isin(libraries)]

# Transpose the dataframe for easier plotting
libraries_df_t = libraries_df.T

# Create the plot
plt.figure(figsize=(15, 8))

# Plot each library's consumption
for library in libraries_df_t.columns:
    if library.startswith('F'):
        color = 'blue'
    elif library.startswith('G'):
        color = 'green'
    elif library.startswith('H'):
        color = 'orange'
    else:
        color = 'darkgrey'

    plt.plot(libraries_df_t.index, libraries_df_t[library], label=library, color=color)

# Customize the plot
plt.title('Library Group Consumption Trend', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Consumption (kWh)', fontsize=12)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True, linestyle='--', alpha=0.7)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Adjust layout to prevent cutting off labels
plt.tight_layout()

# Show the plot
plt.show()
```



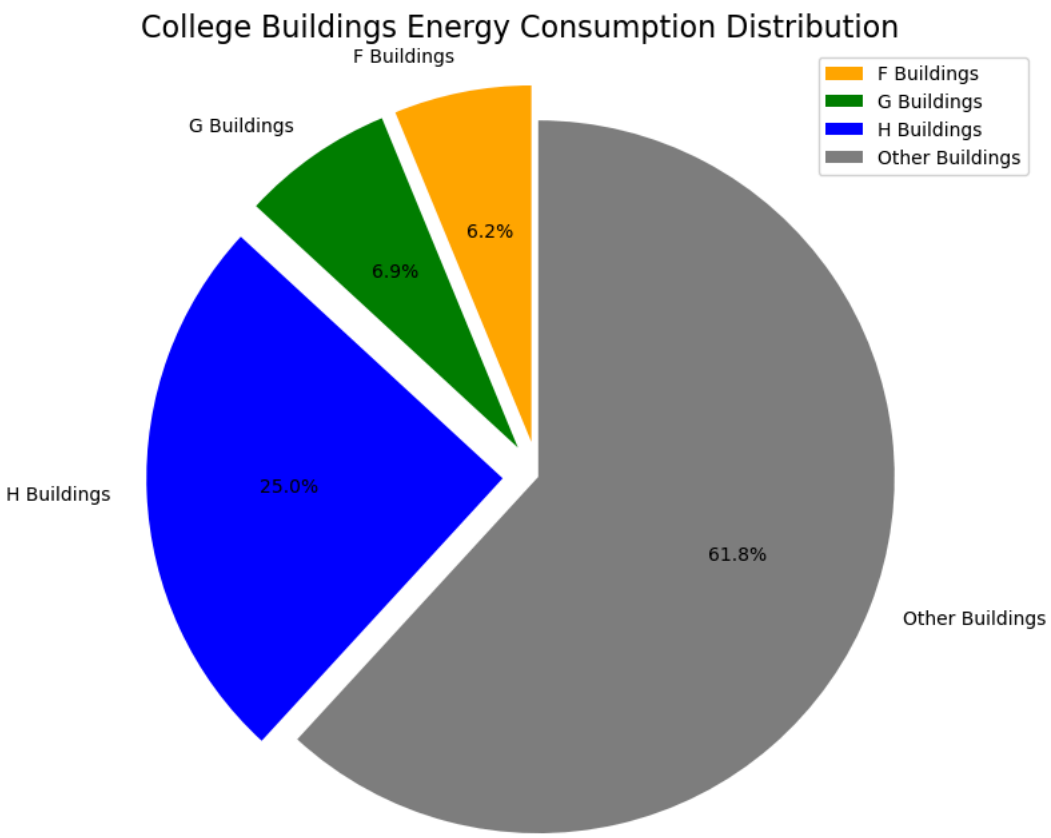
In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```python
# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Set the 'Meter_Location' column as the index
df.set_index('Meter_Location', inplace=True)

# Define the Colleges group
colleges = [
    "C405 192 Castle College - kWh",
    "D402 Hayward College - kWh",
    "D40X Cumberland College - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G608 St Margarets College - kWh",
    "H41X Selwyn College - kWh E2",
    "H633 Arana College main - kWh",
    "H71X Studholm College - kWh E2",
    "J126 Carrington College (Kitchen/Dining) - kWh",
    "J14X Aquinas College - kWh",
    "J303 Caroline Freeman College - kWh",
    "K427 Abbey College - kWh"
]

# Filter the dataframe to include only the Colleges group
colleges_df = df.loc[df.index.isin(colleges)]

# Transpose the dataframe for easier plotting
colleges_df_t = colleges_df.T

# Create the plot
plt.figure(figsize=(20, 10))

# Color mapping based on the first letter of the college code
color_map = {
    'C': 'grey',
    'D': 'grey',
    'F': 'red',
    'G': 'blue',
    'H': 'orange',
    'J': 'grey',
    'K': 'grey'
}

# Plot each college's consumption
for college in colleges_df_t.columns:
    # Determine color based on the first letter of the college code
    color = color_map.get(college[0], 'grey')

    plt.plot(colleges_df_t.index, colleges_df_t[college], label=college, color=color)

# Customize the plot
plt.title('Colleges Group Consumption Trend', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Consumption (kWh)', fontsize=12)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=8)
plt.grid(True, linestyle='--', alpha=0.7)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Adjust layout to prevent cutting off labels
plt.tight_layout()

# Show the plot
plt.show()
```



In [3]:

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

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Set the 'Meter_Location' column as the index
df.set_index('Meter_Location', inplace=True)

# Define the Colleges group
colleges = [
    "C405 192 Castle College - kWh",
    "D402 Hayward College - kWh",
    "D40X Cumberland College - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G608 St Margarets College - kWh",
    "H41X Selwyn College - kWh E2",
    "H633 Arana College main - kWh",
    "H71X Studholm College - kWh E2",
    "J126 Carrington College (Kitchen/Dining) - kWh",
    "J14X Aquinas College - kWh",
    "J303 Caroline Freeman College - kWh",
    "K427 Abbey College - kWh"
]

# Filter the dataframe to include only the Colleges group
colleges_df = df.loc[df.index.isin(colleges)]

# Calculate total consumption for F, G, and H buildings
f_buildings = colleges_df.loc[colleges_df.index.str.startswith('F'), :].sum().sum()
g_buildings = colleges_df.loc[colleges_df.index.str.startswith('G'), :].sum().sum()
h_buildings = colleges_df.loc[colleges_df.index.str.startswith('H'), :].sum().sum()

# Calculate total consumption for other buildings
other_buildings = colleges_df.loc[~colleges_df.index.str.startswith(('F', 'G', 'H')), :].sum().sum()

# Create data for the pie chart
sizes = [f_buildings, g_buildings, h_buildings, other_buildings]
labels = ['F Buildings', 'G Buildings', 'H Buildings', 'Other Buildings']
colors = ['orange', 'green', 'blue', 'gray']
explode = (0.1, 0.1, 0.1, 0)  # Explode F, G, and H slices

# Create the pie chart
plt.figure(figsize=(10, 8))
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.title('College Buildings Energy Consumption Distribution', fontsize=16)

# Add a legend
plt.legend(loc='best')

# Show the plot
plt.show()

# Print total consumption for F, G, and H buildings
print("Total Consumption:")
print(f"F Buildings: {f_buildings:.2f} kWh")
print(f"G Buildings: {g_buildings:.2f} kWh")
print(f"H Buildings: {h_buildings:.2f} kWh")
```



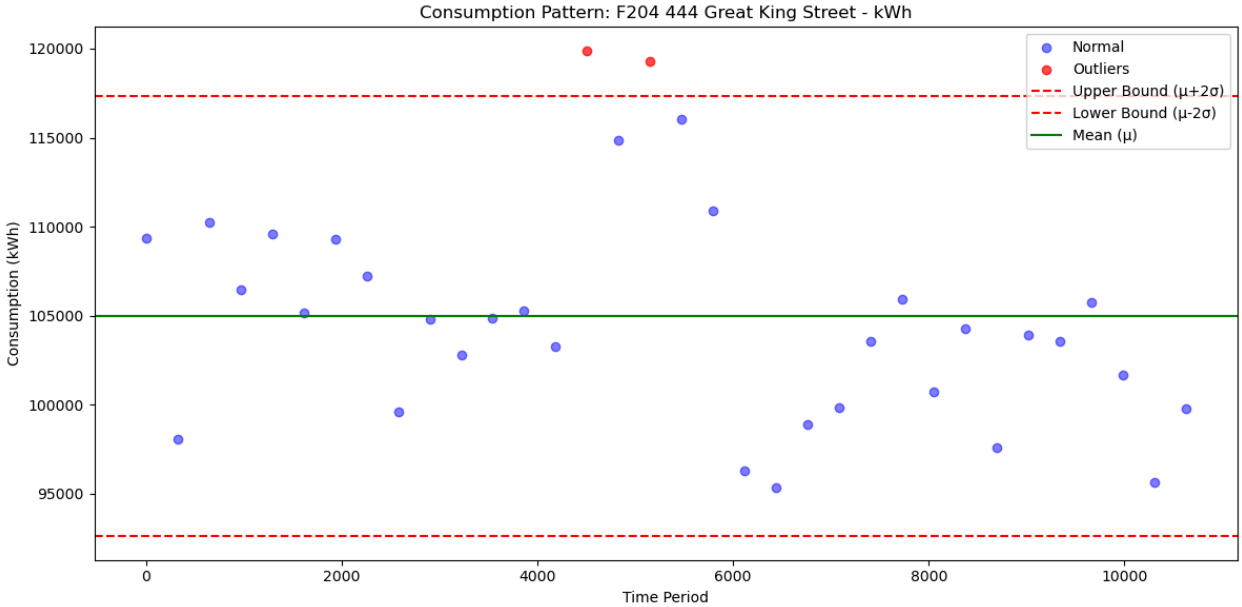

```
Total Consumption:
F Buildings: 1000603.34 kWh
G Buildings: 1108984.73 kWh
H Buildings: 4016360.03 kWh
```

In [30]:

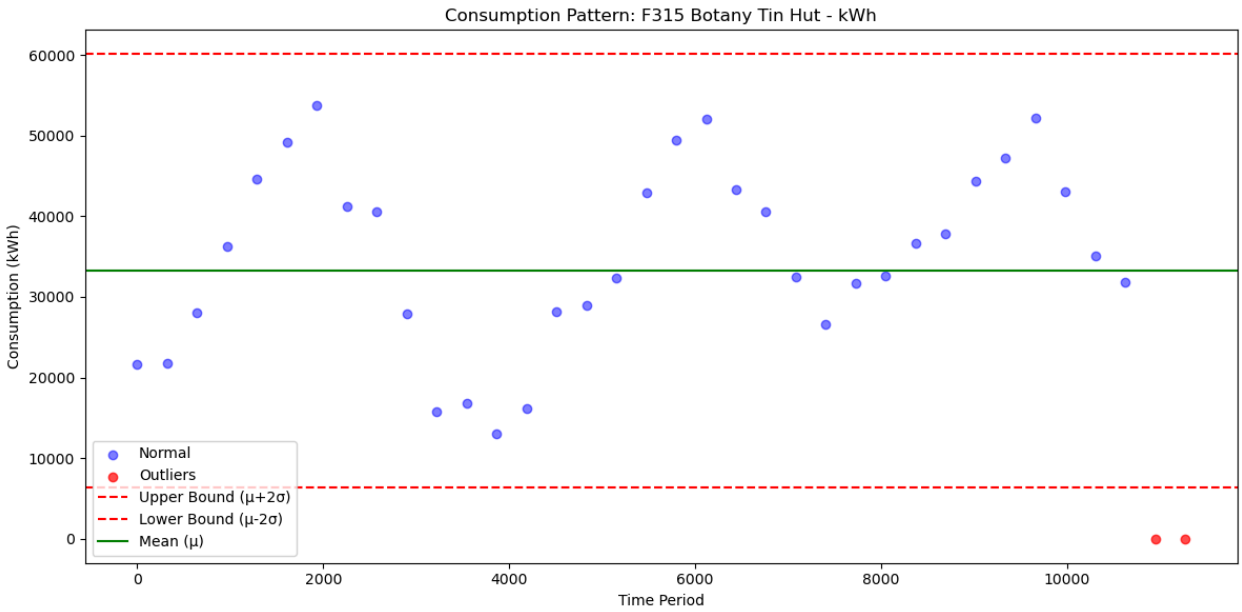
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Read the data
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Filter F, G and H meter locations
mask = df['Meter_Location'].str.contains('^[FGH]', na=False)
filtered_df = df[mask]

# Melt the dataframe
df_melted = pd.melt(filtered_df,
                    id_vars=['Meter_Location'],
                    var_name='Date',
                    value_name='Consumption')

# Function to detect outliers using mean ± 2σ method
def detect_outliers(data, meter_name):
    mean = np.mean(data)
    std = np.std(data)

    upper_bound = mean + (2 * std)
    lower_bound = mean - (2 * std)

    outliers = data[(data > upper_bound) | (data < lower_bound)]
    normal = data[(data <= upper_bound) & (data >= lower_bound)]

    return outliers, normal, upper_bound, lower_bound, mean, std

# Group by Meter_Location and analyze each building
for meter in df_melted['Meter_Location'].unique():
    building_data = df_melted[df_melted['Meter_Location'] == meter]['Consumption']

    # Skip if all values are NaN
    if building_data.isna().all():
        continue

    outliers, normal, upper, lower, mean, std = detect_outliers(building_data, meter)

    # Create subplot for each building
    plt.figure(figsize=(12, 6))
    plt.scatter(normal.index, normal, c='blue', label='Normal', alpha=0.5)
    plt.scatter(outliers.index, outliers, c='red', label='Outliers', alpha=0.7)
```

```
    plt.axhline(y=upper, color='r', linestyle='--', label='Upper Bound (μ+2σ)')
    plt.axhline(y=lower, color='r', linestyle='--', label='Lower Bound (μ-2σ)')
    plt.axhline(y=mean, color='g', linestyle='-', label='Mean (μ)')

    plt.title(f'Consumption Pattern: {meter}')
    plt.xlabel('Time Period')
    plt.ylabel('Consumption (kWh)')
    plt.legend()

    # Print statistics
    print(f"\nStatistics for {meter}:")
    print(f"Mean: {mean:.2f}")
    print(f"Standard Deviation: {std:.2f}")
    print(f"Upper Bound: {upper:.2f}")
    print(f"Lower Bound: {lower:.2f}")
    print(f"Number of outliers: {len(outliers)}")

    plt.tight_layout()
    plt.show()
```

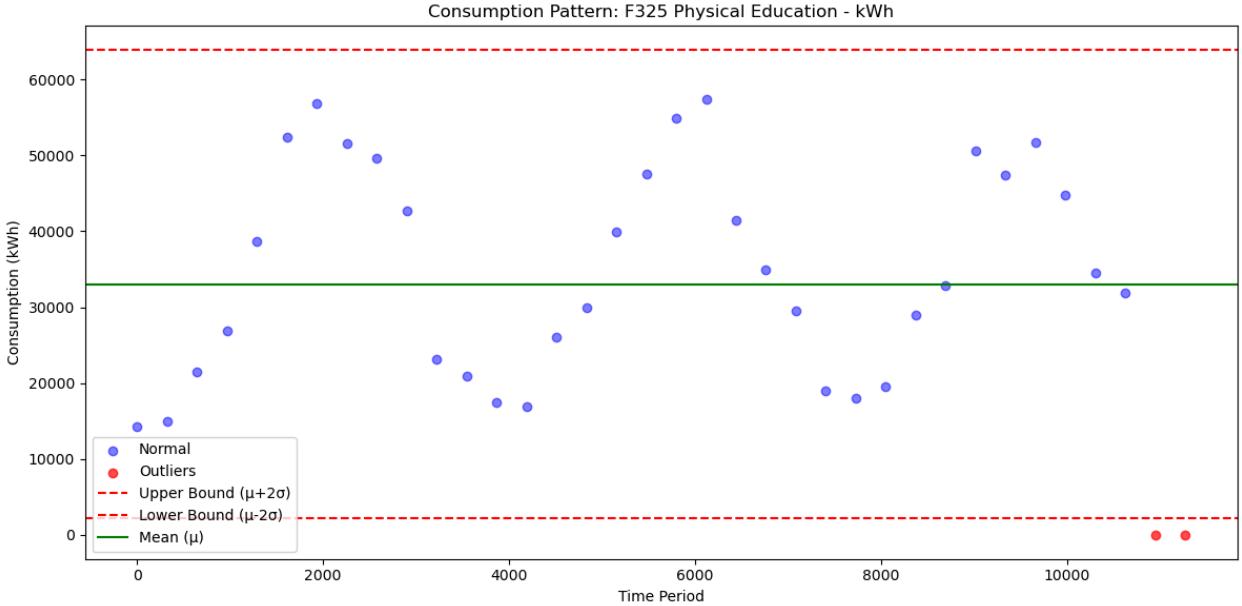
```
Statistics for F204 444 Great King Street - kWh:
Mean: 104984.12
Standard Deviation: 6175.87
Upper Bound: 117335.86
Lower Bound: 92632.37
Number of outliers: 2
```



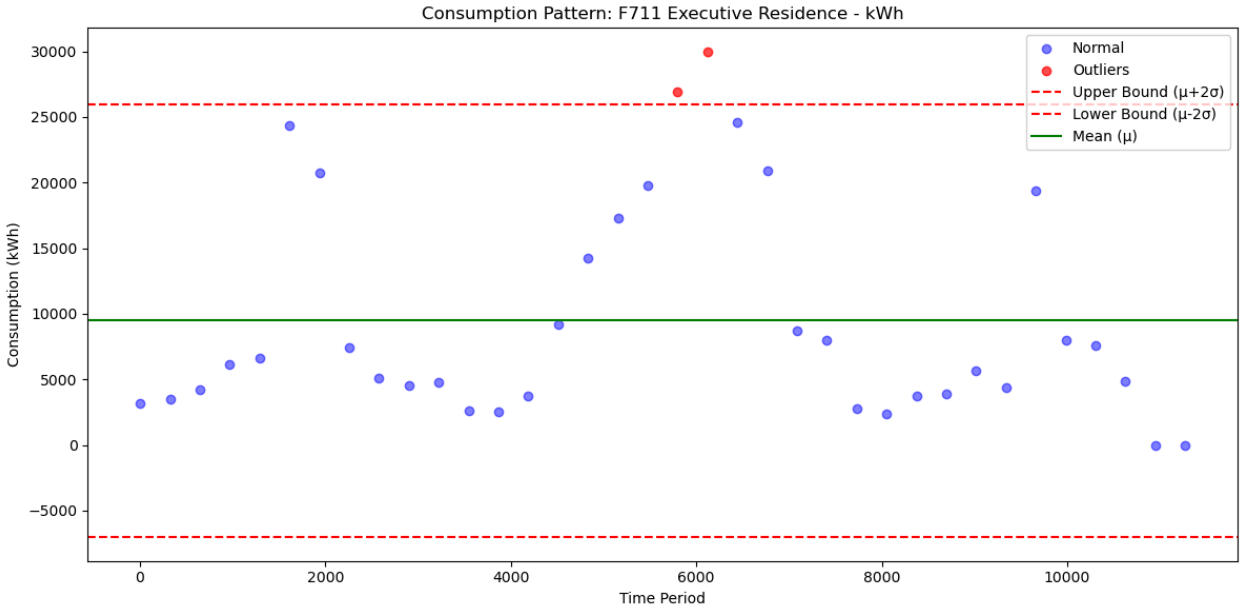

```
Statistics for F315 Botany Tin Hut - kWh:
Mean: 33226.40
Standard Deviation: 13442.96
Upper Bound: 60112.33
Lower Bound: 6340.48
Number of outliers: 2
```



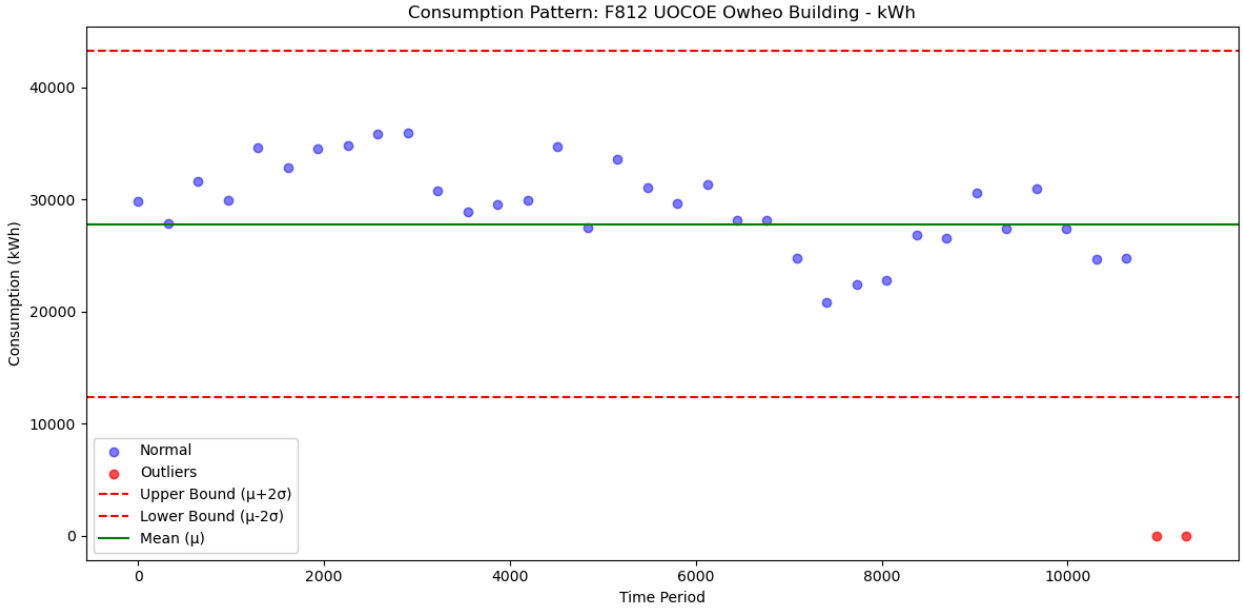
```
Statistics for F325 Physical Education - kWh:
Mean: 33025.10
Standard Deviation: 15430.65
Upper Bound: 63886.39
Lower Bound: 2163.81
Number of outliers: 2
```



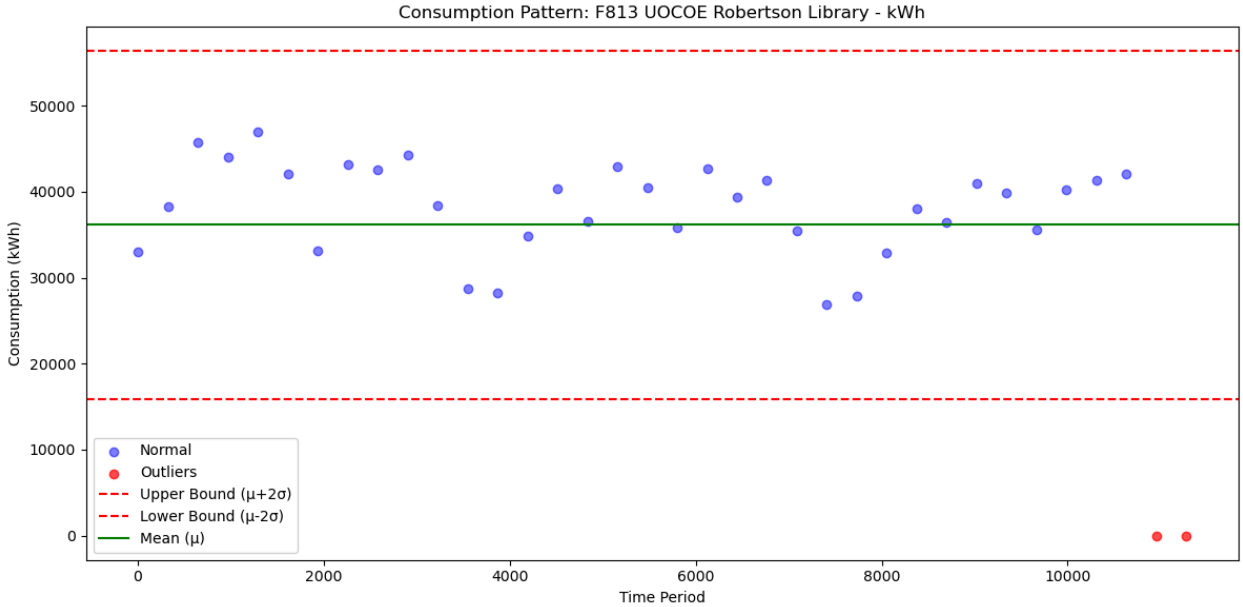

```
Statistics for F711 Executive Residence - kWh:
Mean: 9494.92
Standard Deviation: 8245.42
Upper Bound: 25985.77
Lower Bound: -6995.92
Number of outliers: 2
```



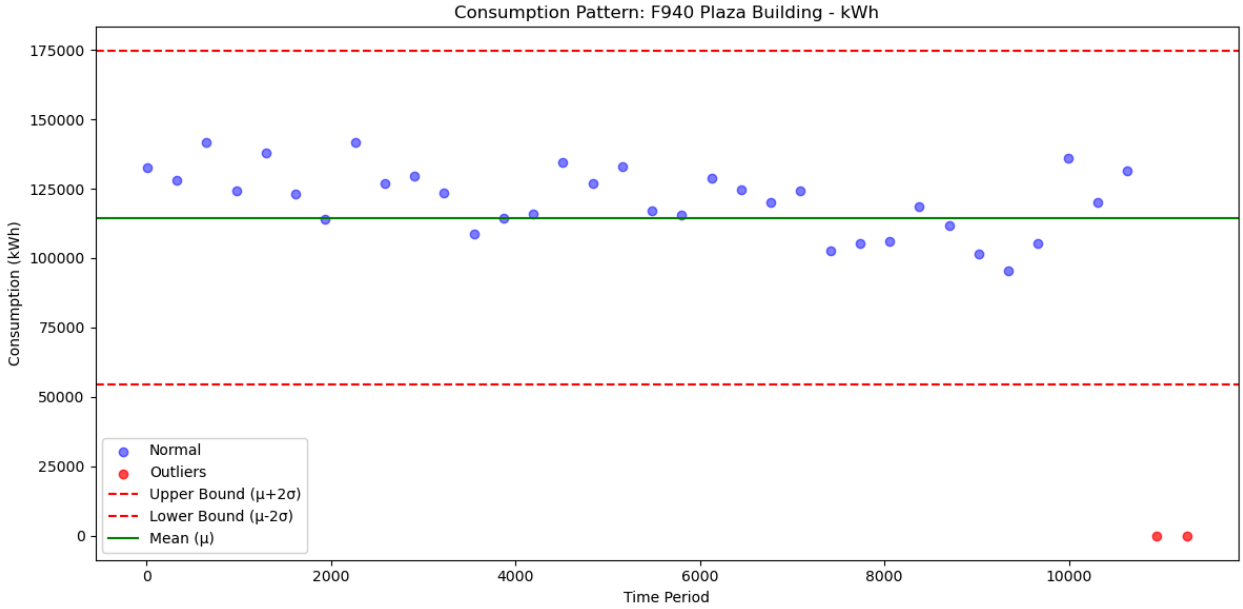

```
Statistics for F812 UOCOE Owheo Building - kWh:
Mean: 27794.54
Standard Deviation: 7709.68
Upper Bound: 43213.91
Lower Bound: 12375.17
Number of outliers: 2
```



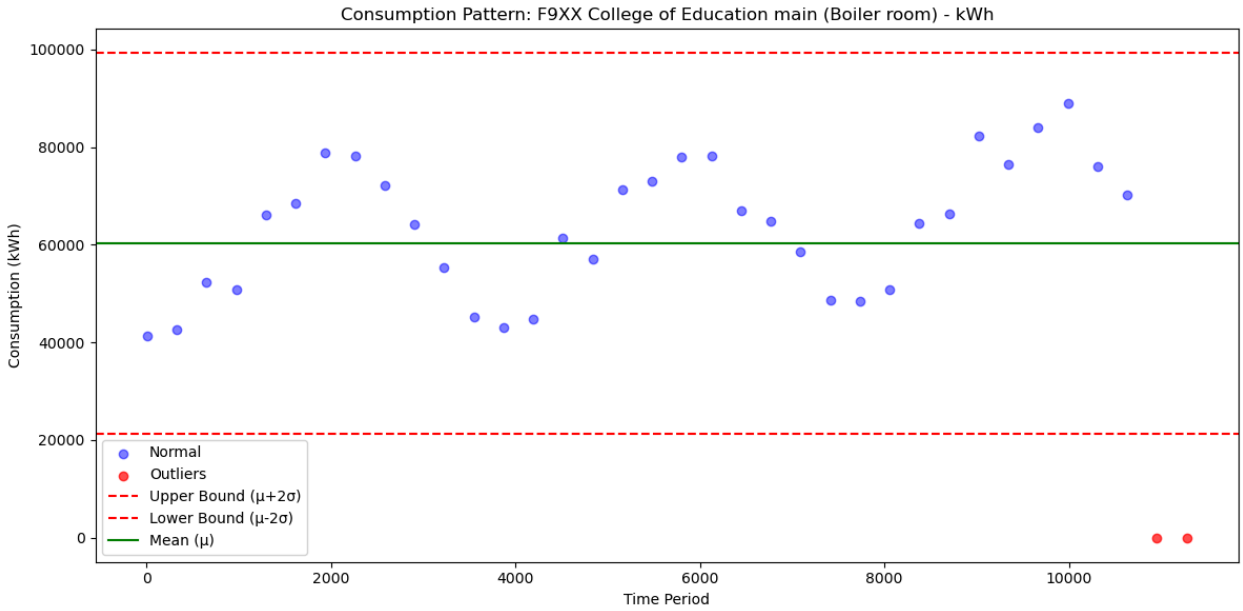

```
Statistics for F813 UOCOE Robertson Library - kWh:
Mean: 36134.23
Standard Deviation: 10102.07
Upper Bound: 56338.38
Lower Bound: 15930.08
Number of outliers: 2
```



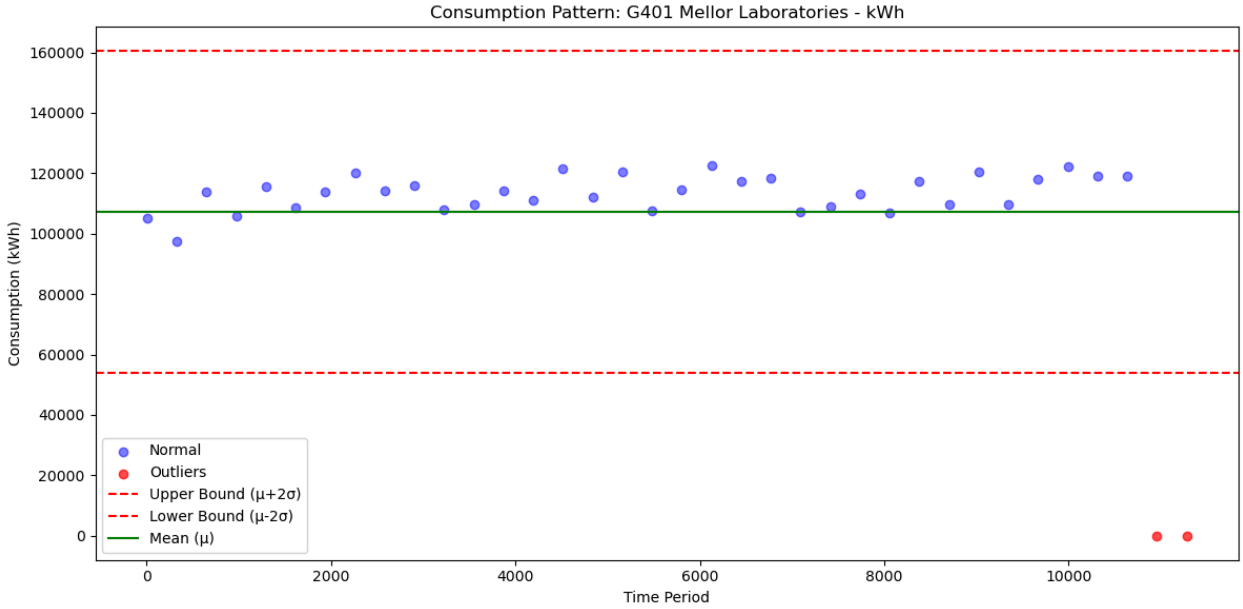

```
Statistics for F940 Plaza Building - kWh:
Mean: 114475.49
Standard Deviation: 30029.82
Upper Bound: 174535.14
Lower Bound: 54415.84
Number of outliers: 2
```



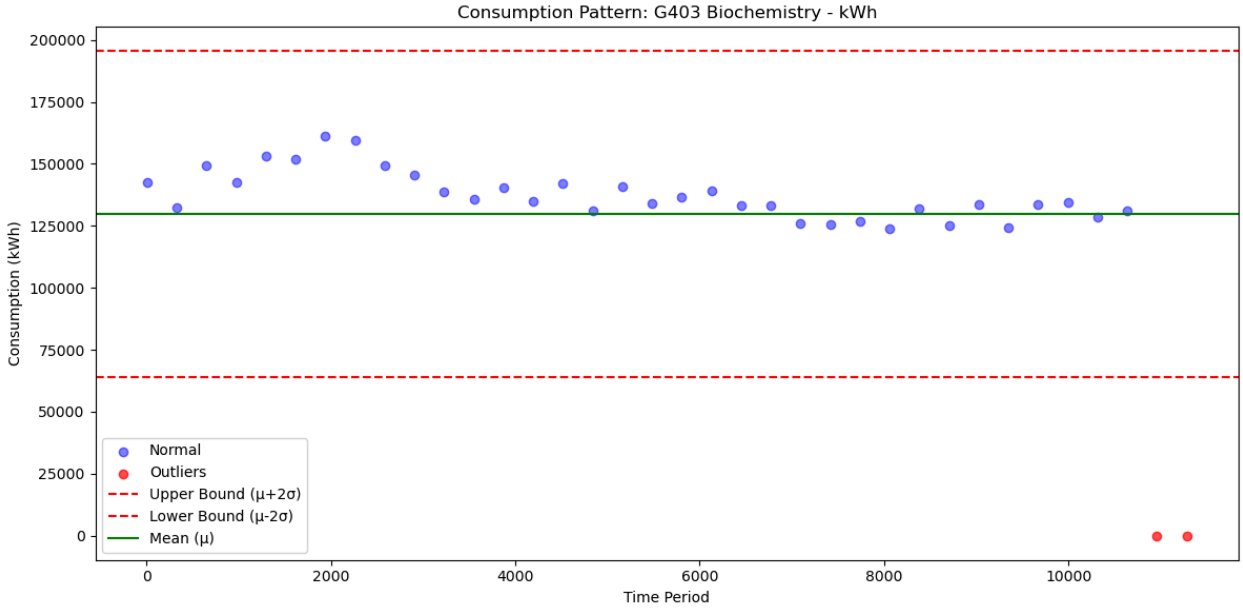

```
Statistics for F9XX College of Education main (Boiler room) - kWh:
Mean: 60263.75
Standard Deviation: 19496.11
Upper Bound: 99255.98
Lower Bound: 21271.53
Number of outliers: 2
```



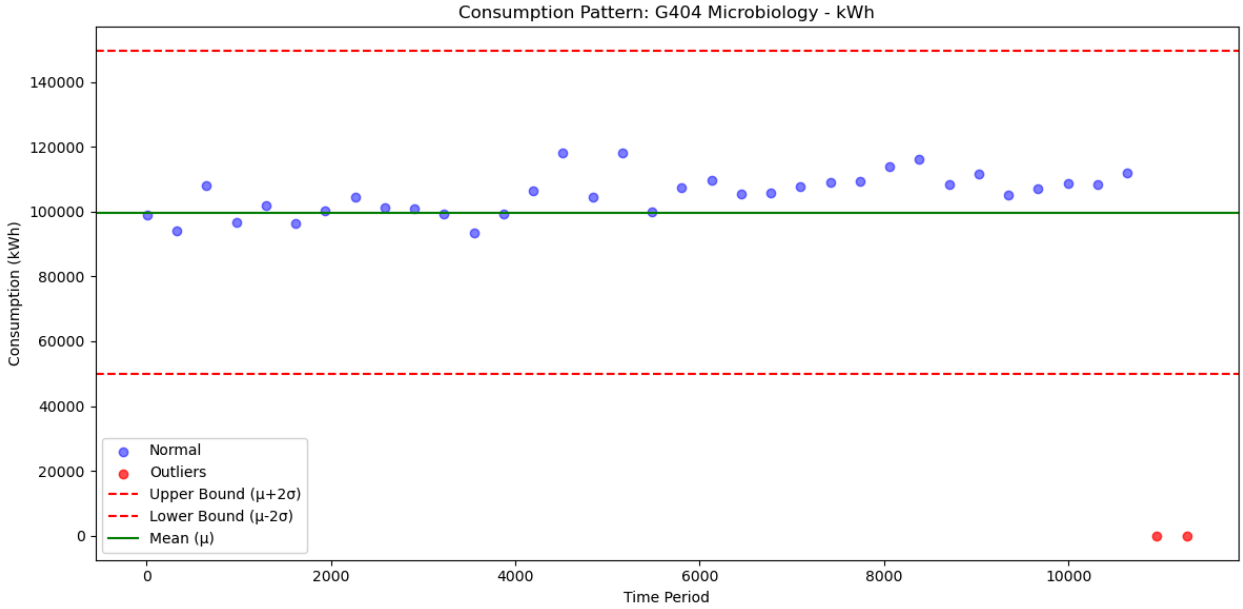

```
Statistics for G401 Mellor Laboratories - kWh:
Mean: 107214.09
Standard Deviation: 26610.74
Upper Bound: 160435.57
Lower Bound: 53992.61
Number of outliers: 2
```



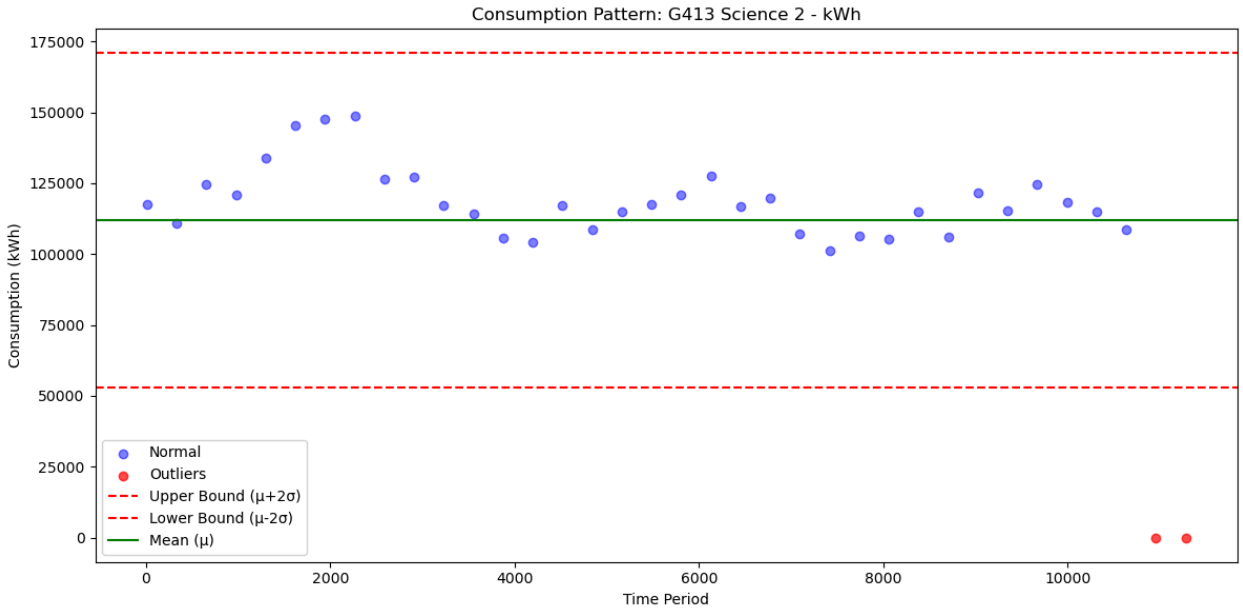

```
Statistics for G403 Biochemistry - kWh:
Mean: 129760.23
Standard Deviation: 32841.49
Upper Bound: 195443.22
Lower Bound: 64077.25
Number of outliers: 2
```



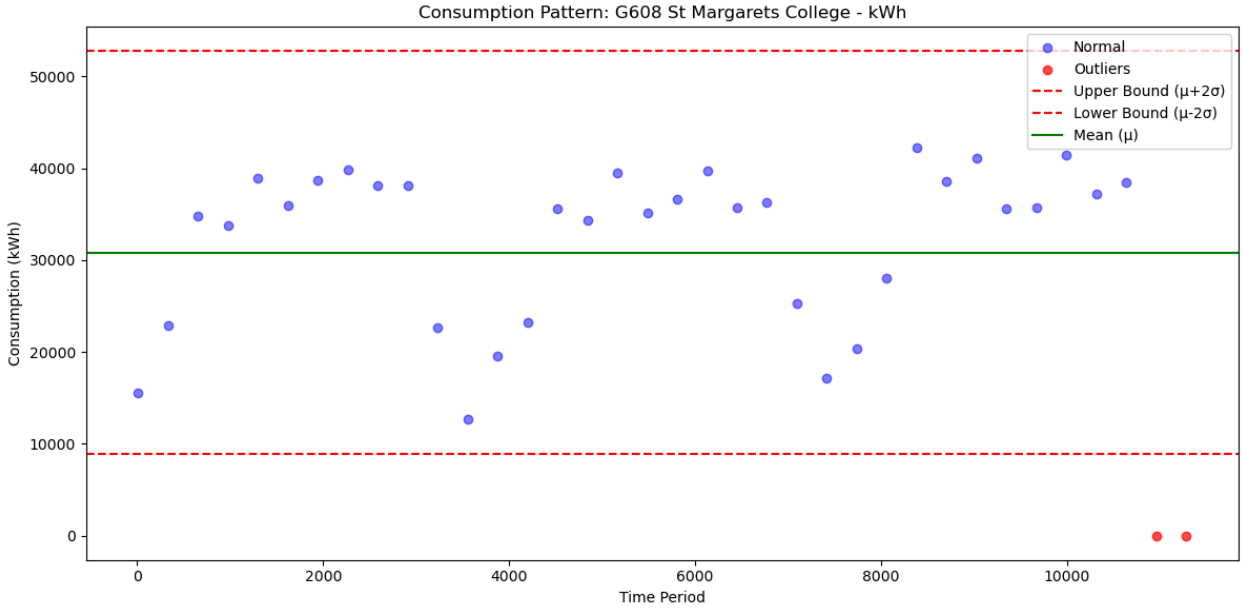

```
Statistics for G404 Microbiology - kWh:
Mean: 99665.14
Standard Deviation: 24935.07
Upper Bound: 149535.28
Lower Bound: 49795.00
Number of outliers: 2
```



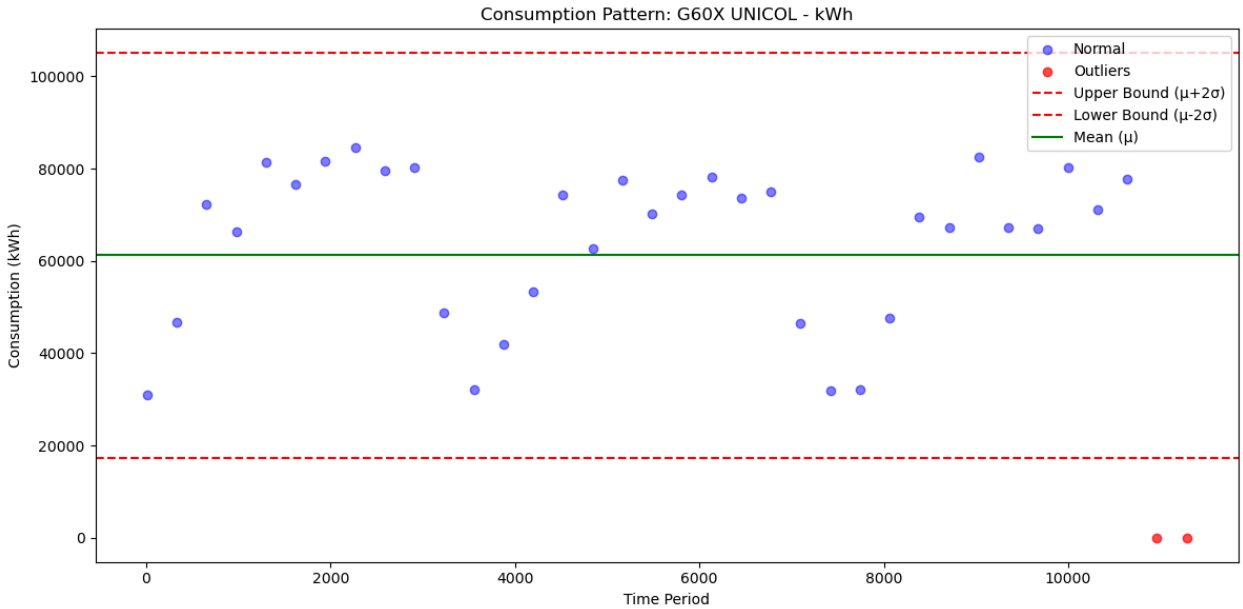

```
Statistics for G413 Science 2 - kWh:
Mean: 111979.56
Standard Deviation: 29429.36
Upper Bound: 170838.28
Lower Bound: 53120.84
Number of outliers: 2
```



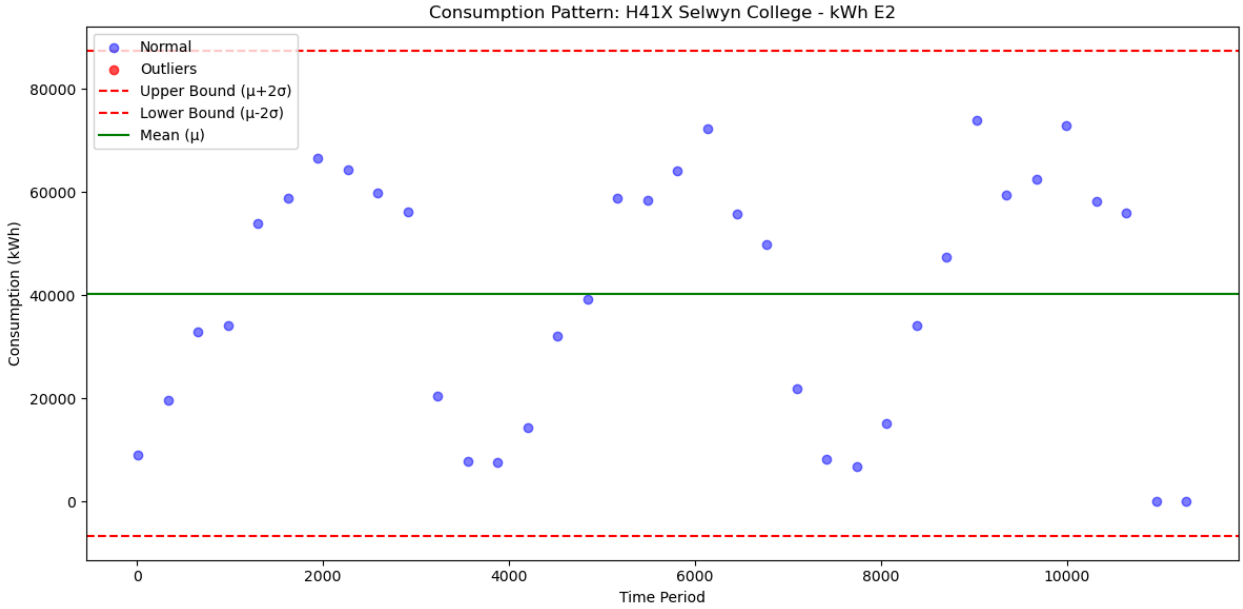

```
Statistics for G608 St Margarets College - kWh:
Mean: 30805.13
Standard Deviation: 10971.46
Upper Bound: 52748.06
Lower Bound: 8862.21
Number of outliers: 2
```



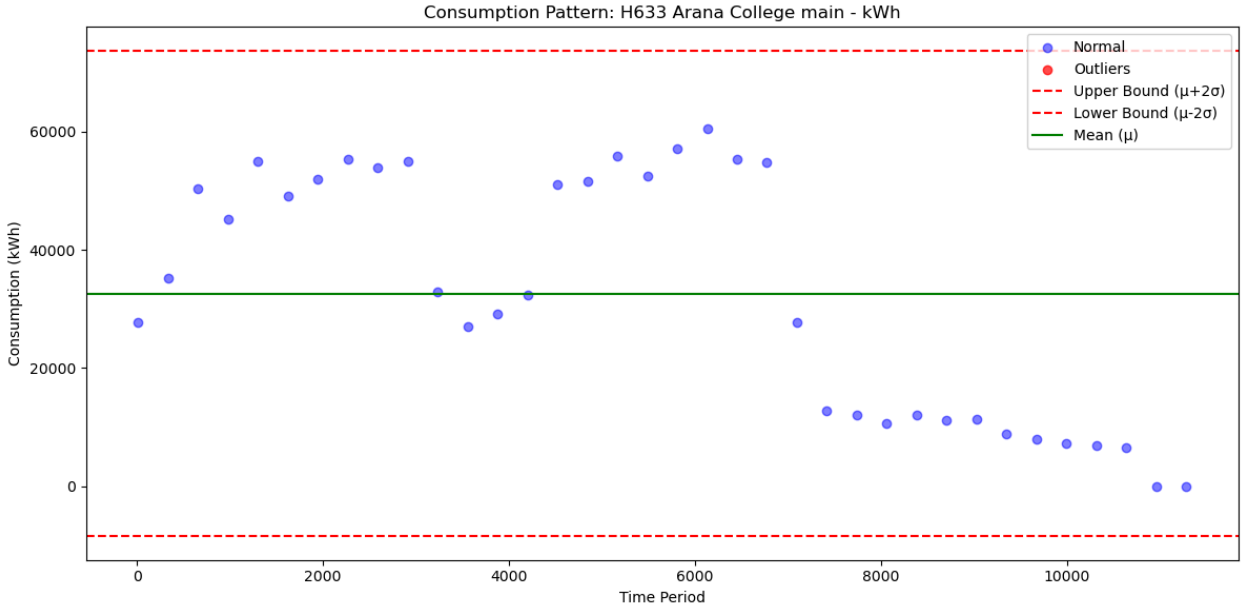

```
Statistics for G60X UNICOL - kWh:
Mean: 61188.88
Standard Deviation: 21915.54
Upper Bound: 105019.96
Lower Bound: 17357.80
Number of outliers: 2
```



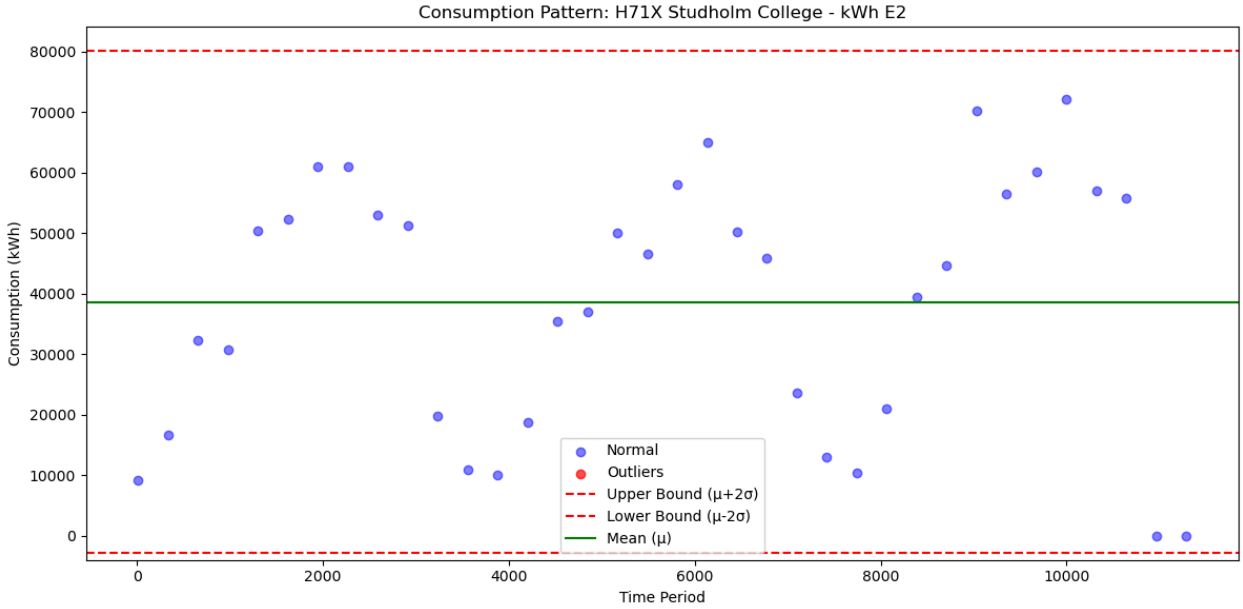

```
Statistics for H41X Selwyn College - kWh E2:
Mean: 40371.04
Standard Deviation: 23471.56
Upper Bound: 87314.16
Lower Bound: -6572.09
Number of outliers: 0
```



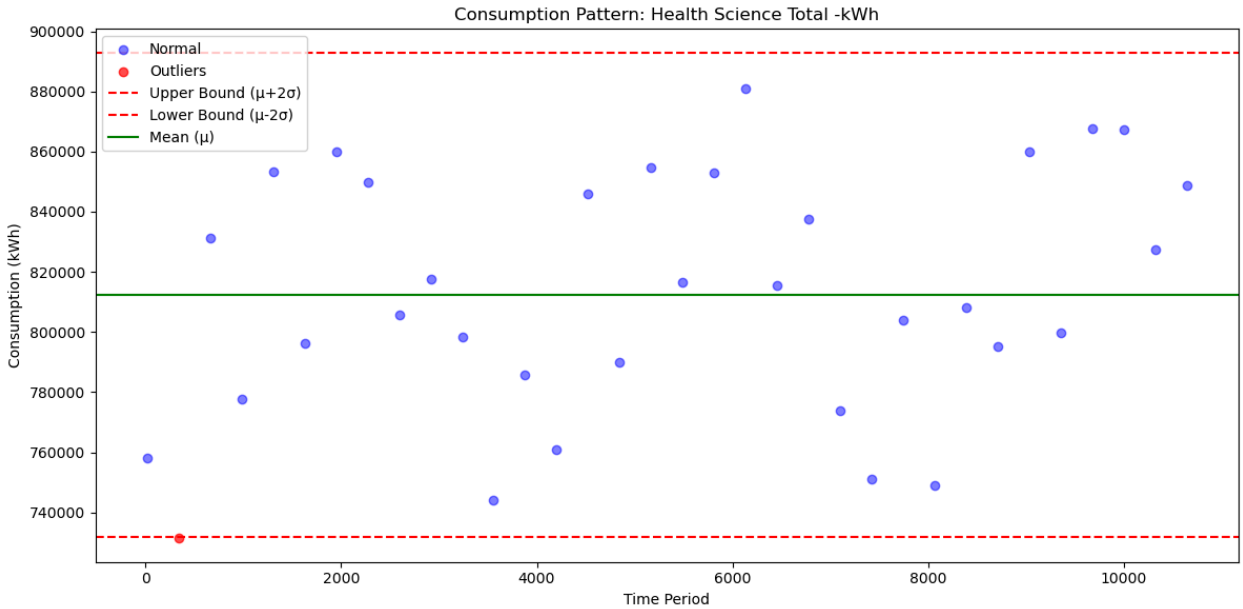

```
Statistics for H633 Arana College main - kWh:
Mean: 32624.58
Standard Deviation: 20514.81
Upper Bound: 73654.21
Lower Bound: -8405.05
Number of outliers: 0
```



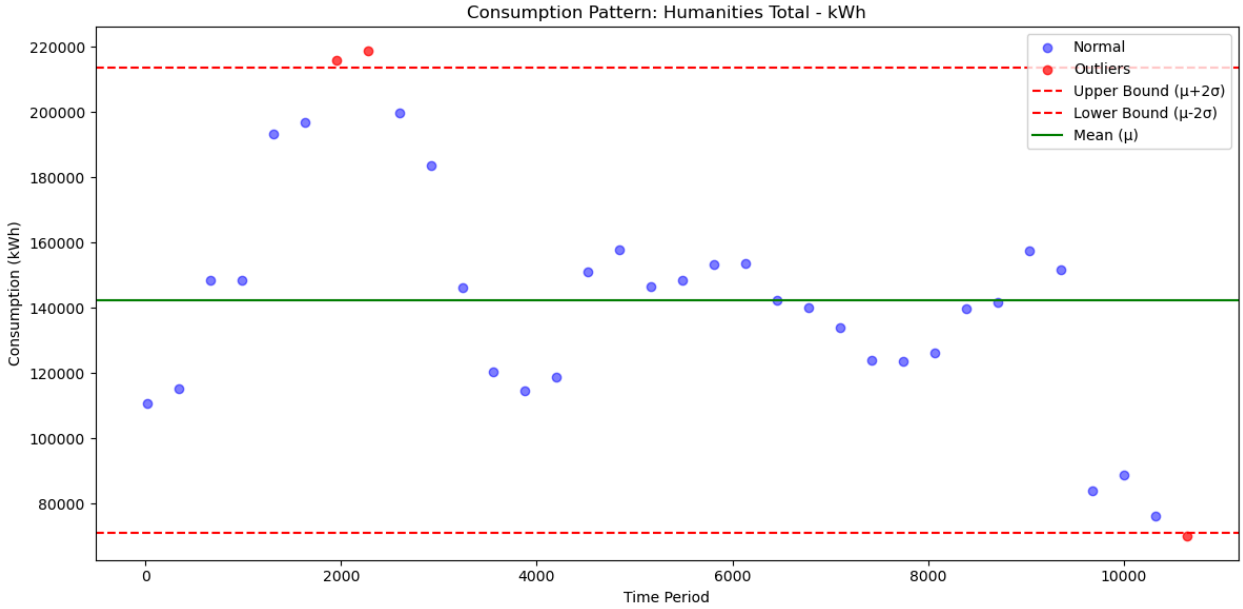

```
Statistics for H71X Studholm College - kWh E2:
Mean: 38569.94
Standard Deviation: 20740.17
Upper Bound: 80050.29
Lower Bound: -2910.40
Number of outliers: 0
```



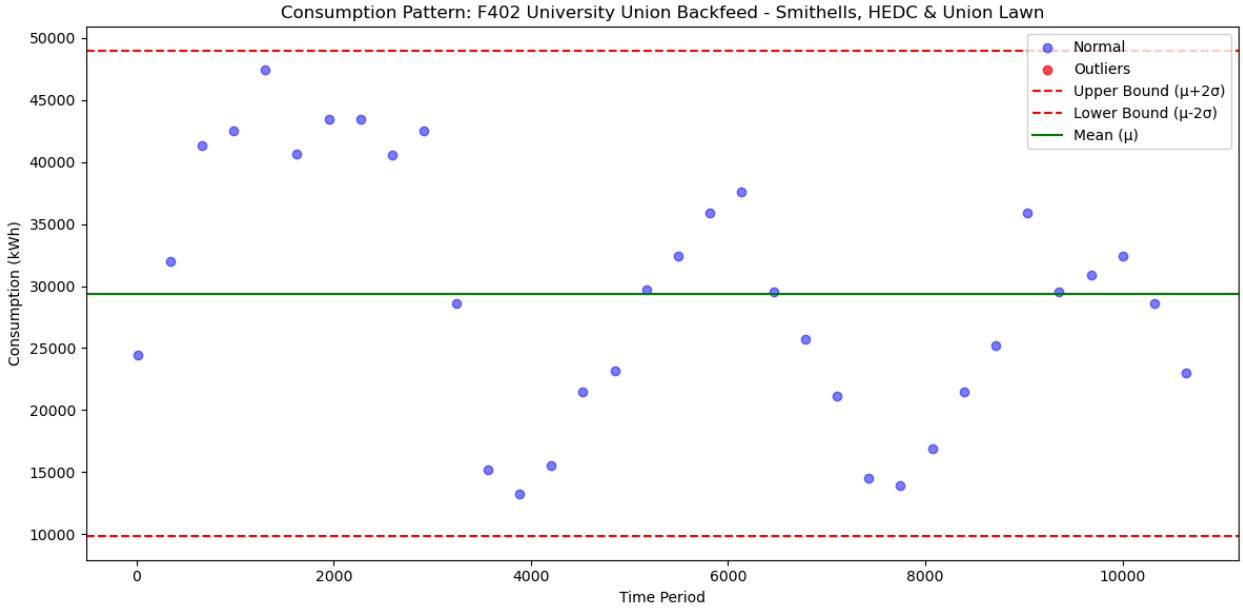

```
Statistics for Health Science Total -kWh:
Mean: 812209.99
Standard Deviation: 40253.49
Upper Bound: 892716.96
Lower Bound: 731703.01
Number of outliers: 1
```



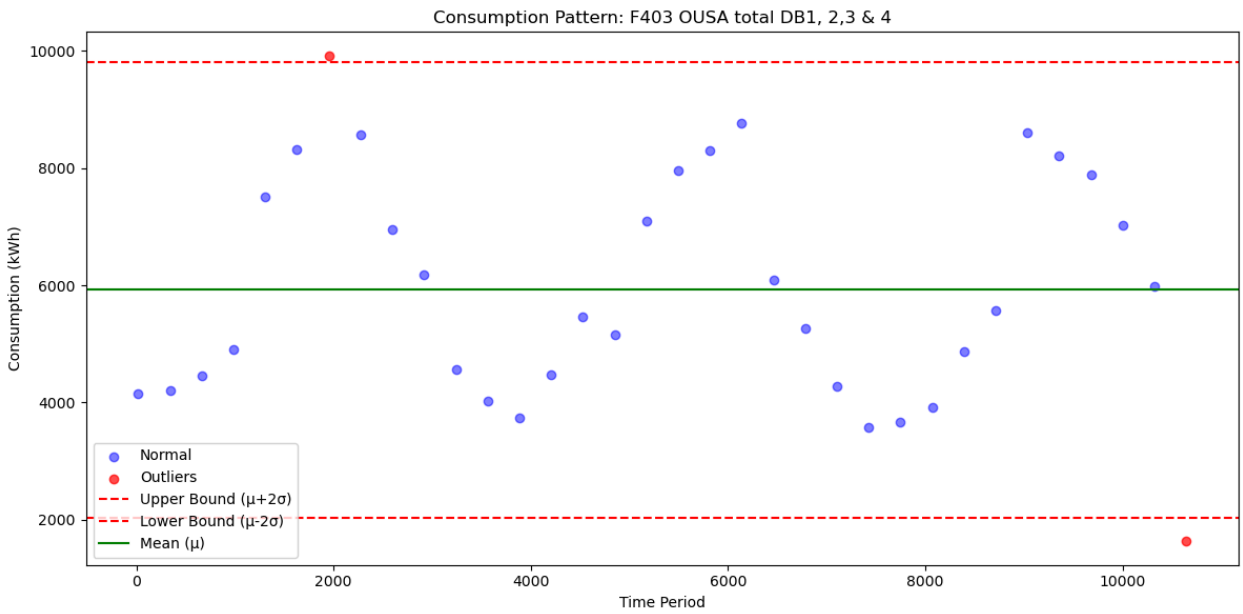

```
Statistics for Humanities Total - kWh:
Mean: 142420.15
Standard Deviation: 35665.14
Upper Bound: 213750.43
Lower Bound: 71089.87
Number of outliers: 3
```



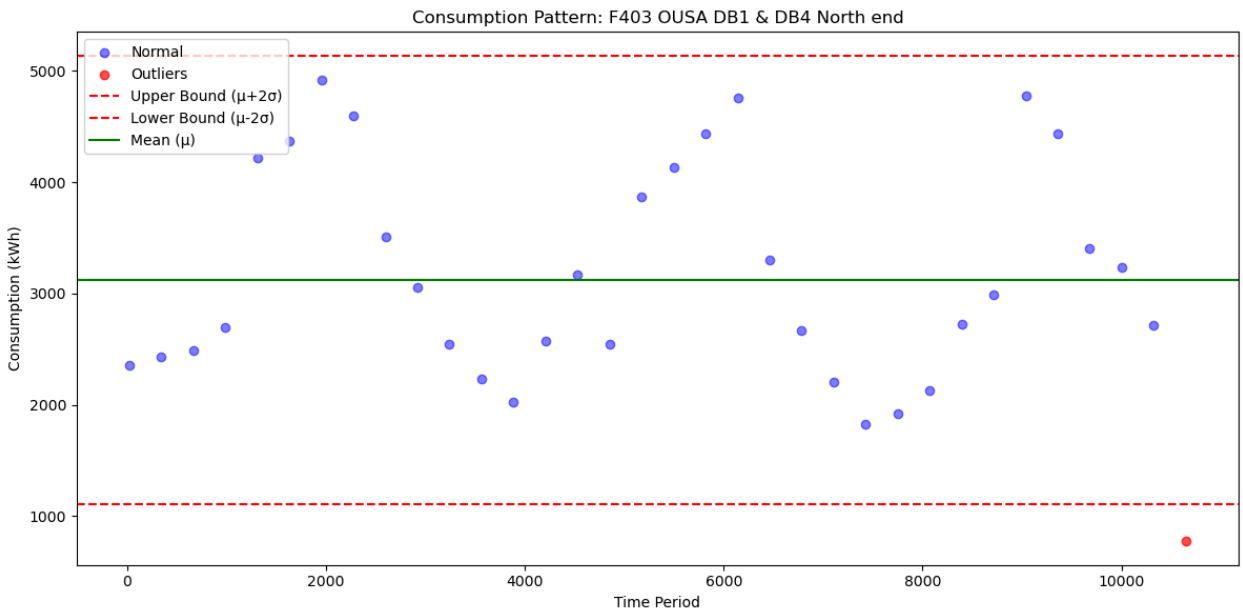

```
Statistics for F402 University Union Backfeed - Smithells, HEDC & Union Lawn:
Mean: 29404.46
Standard Deviation: 9759.65
Upper Bound: 48923.76
Lower Bound: 9885.17
Number of outliers: 0
```



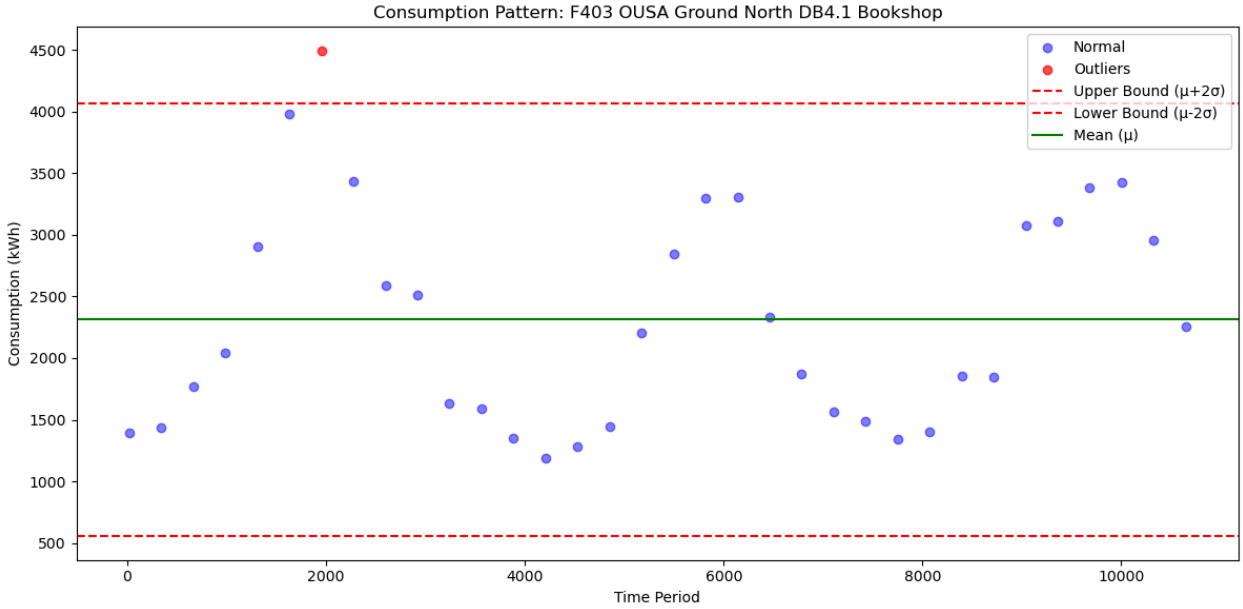

```
Statistics for F403 OUSA total DB1, 2,3 & 4:
Mean: 5921.08
Standard Deviation: 1943.82
Upper Bound: 9808.72
Lower Bound: 2033.44
Number of outliers: 2
```



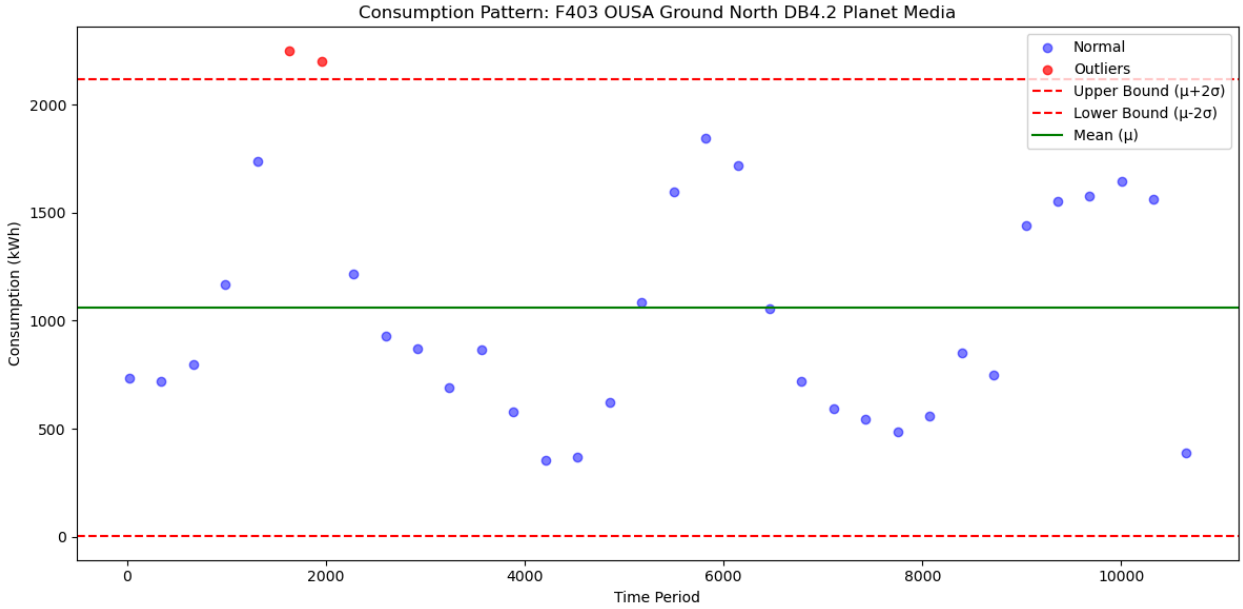

```
Statistics for F403 OUSA DB1 & DB4 North end:
Mean: 3118.66
Standard Deviation: 1006.74
Upper Bound: 5132.14
Lower Bound: 1105.17
Number of outliers: 1
```



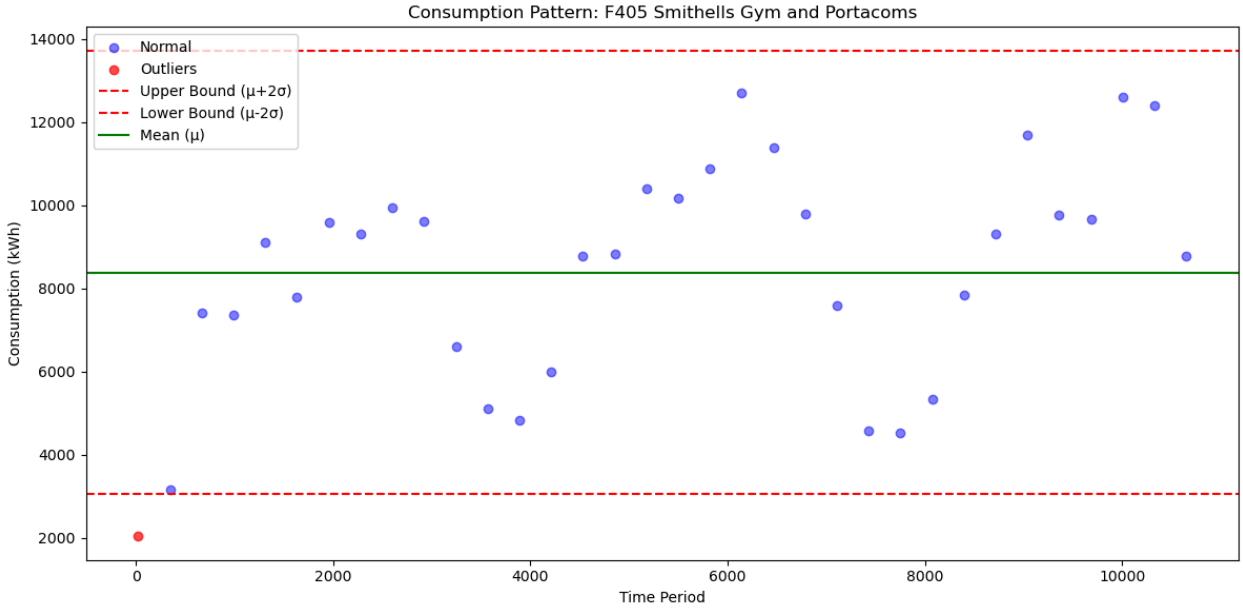

```
Statistics for F403 OUSA Ground North DB4.1 Bookshop:
Mean: 2311.84
Standard Deviation: 876.53
Upper Bound: 4064.90
Lower Bound: 558.78
Number of outliers: 1
```



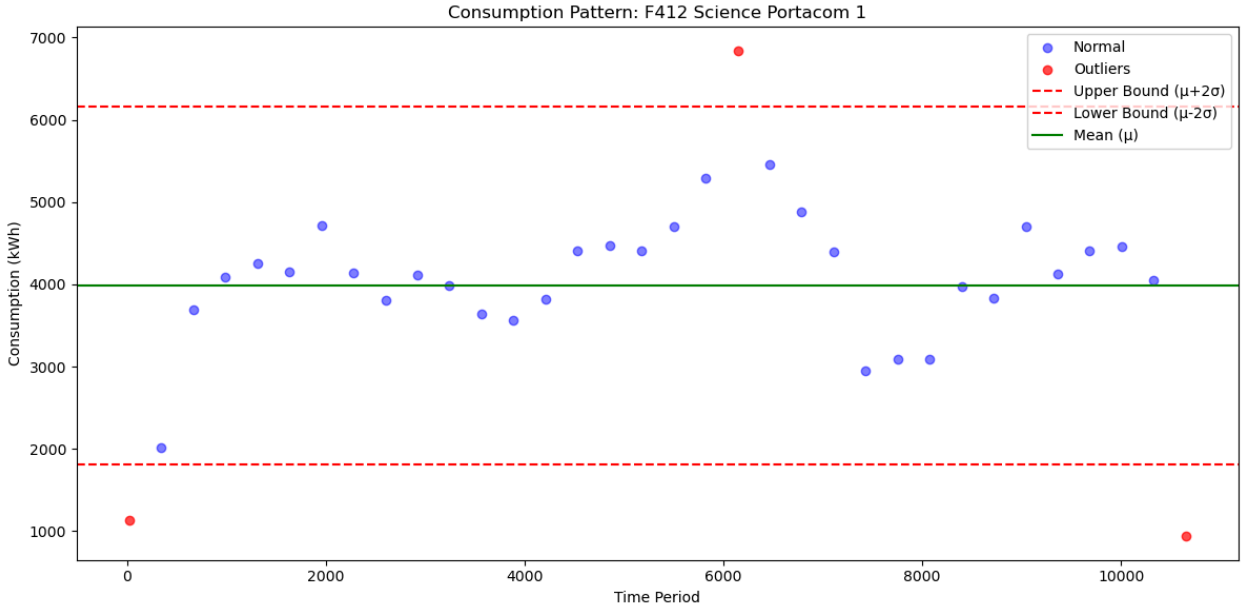

```
Statistics for F403 OUSA Ground North DB4.2 Planet Media:
Mean: 1061.02
Standard Deviation: 527.87
Upper Bound: 2116.77
Lower Bound: 5.27
Number of outliers: 2
```



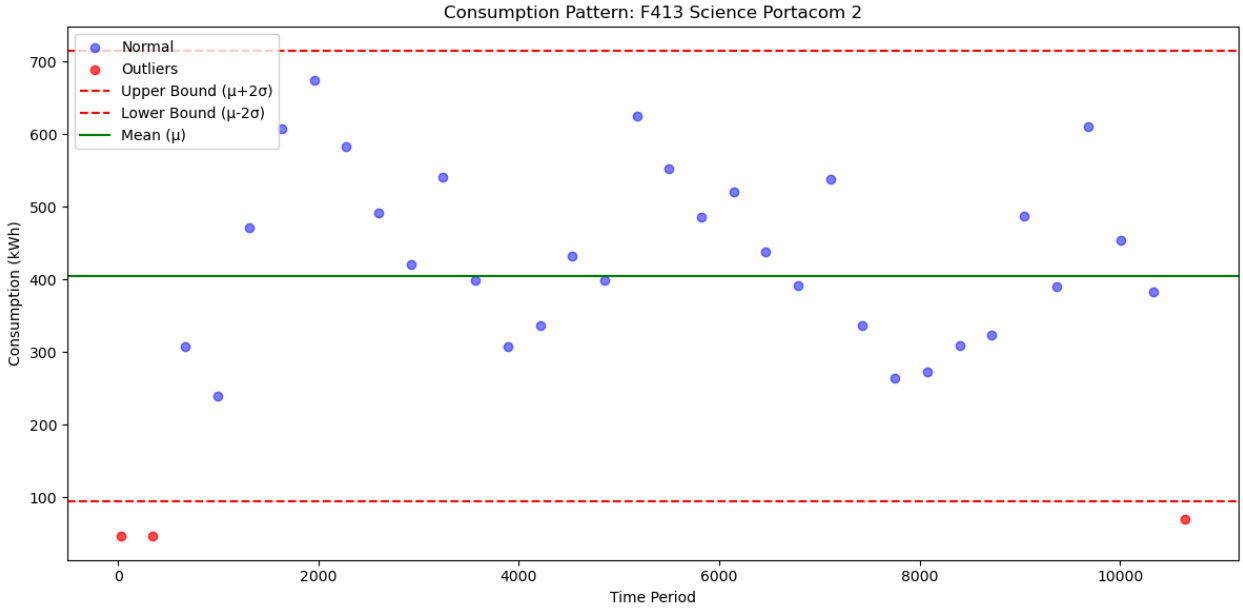

```
Statistics for F405 Smithells Gym and Portacoms:
Mean: 8377.08
Standard Deviation: 2665.06
Upper Bound: 13707.19
Lower Bound: 3046.97
Number of outliers: 1
```



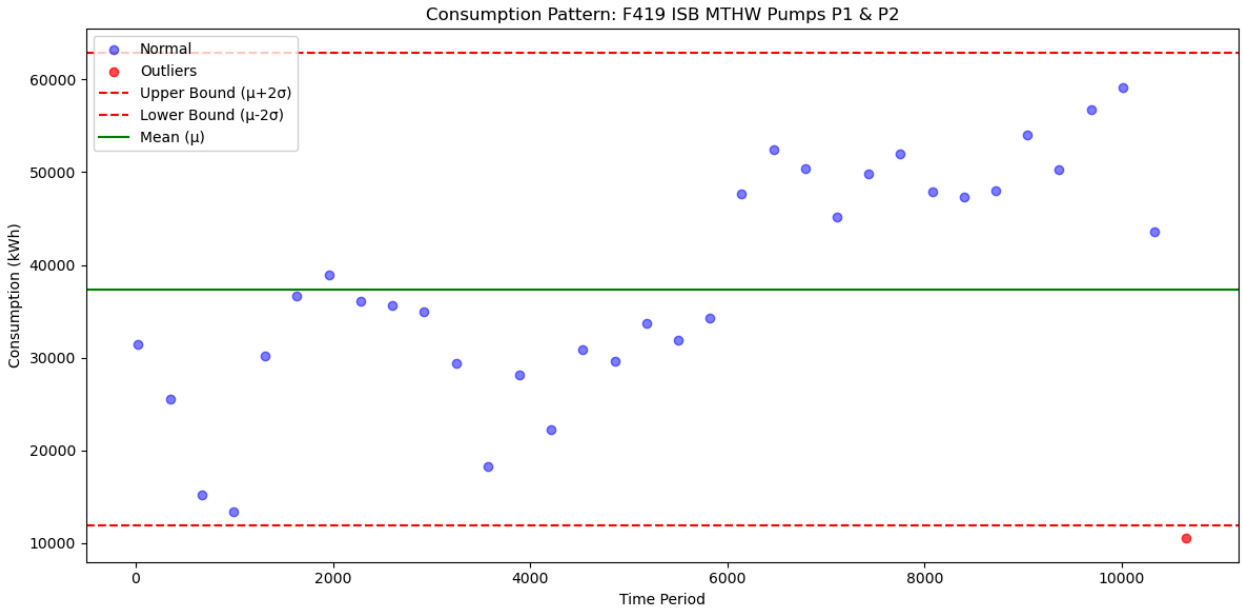

```
Statistics for F412 Science Portacom 1:
Mean: 3988.40
Standard Deviation: 1085.56
Upper Bound: 6159.53
Lower Bound: 1817.28
Number of outliers: 3
```



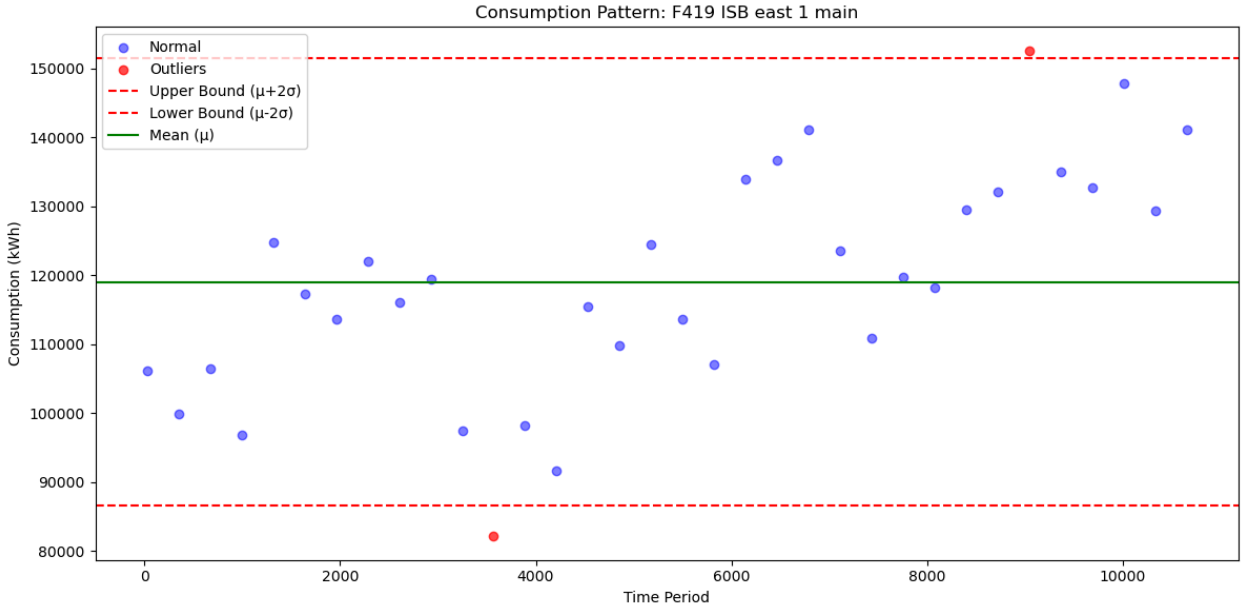

```
Statistics for F413 Science Portacom 2:
Mean: 404.49
Standard Deviation: 154.78
Upper Bound: 714.06
Lower Bound: 94.93
Number of outliers: 3
```



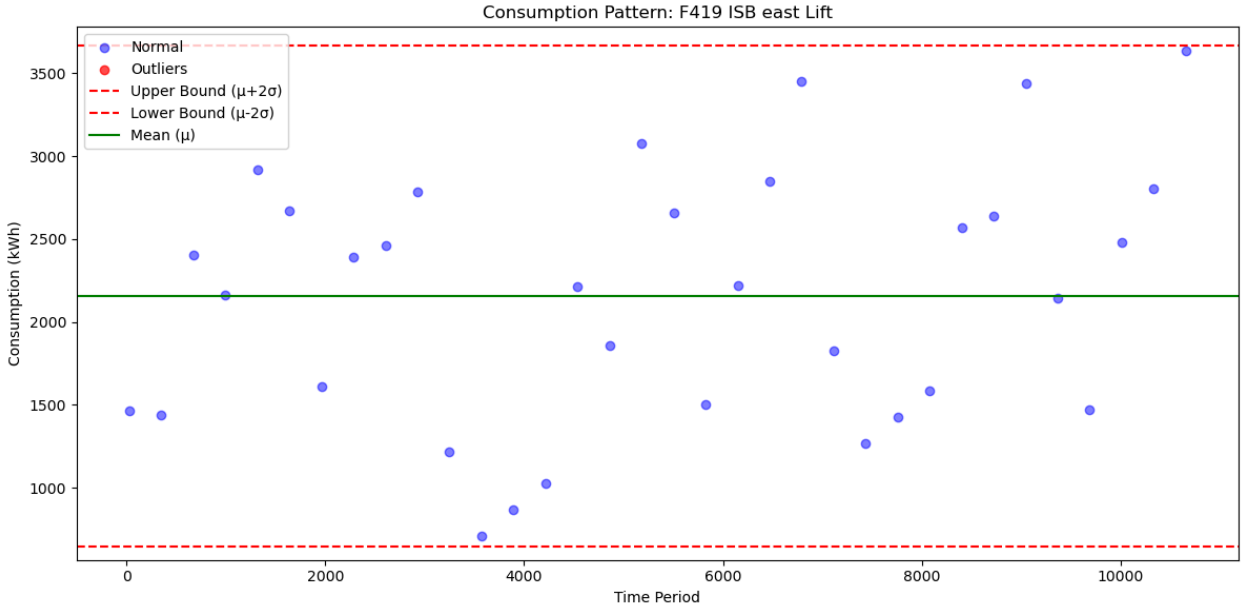

```
Statistics for F419 ISB MTHW Pumps P1 & P2:
Mean: 37385.27
Standard Deviation: 12720.53
Upper Bound: 62826.34
Lower Bound: 11944.20
Number of outliers: 1
```



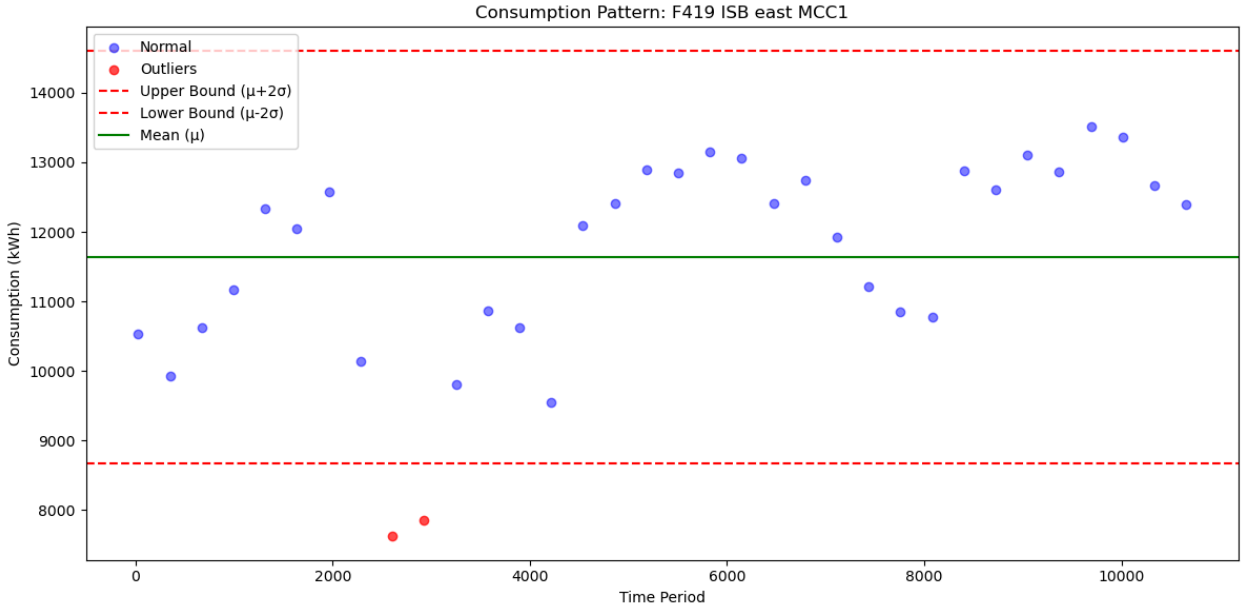

```
Statistics for F419 ISB east 1 main:
Mean: 119018.35
Standard Deviation: 16213.43
Upper Bound: 151445.21
Lower Bound: 86591.49
Number of outliers: 2
```



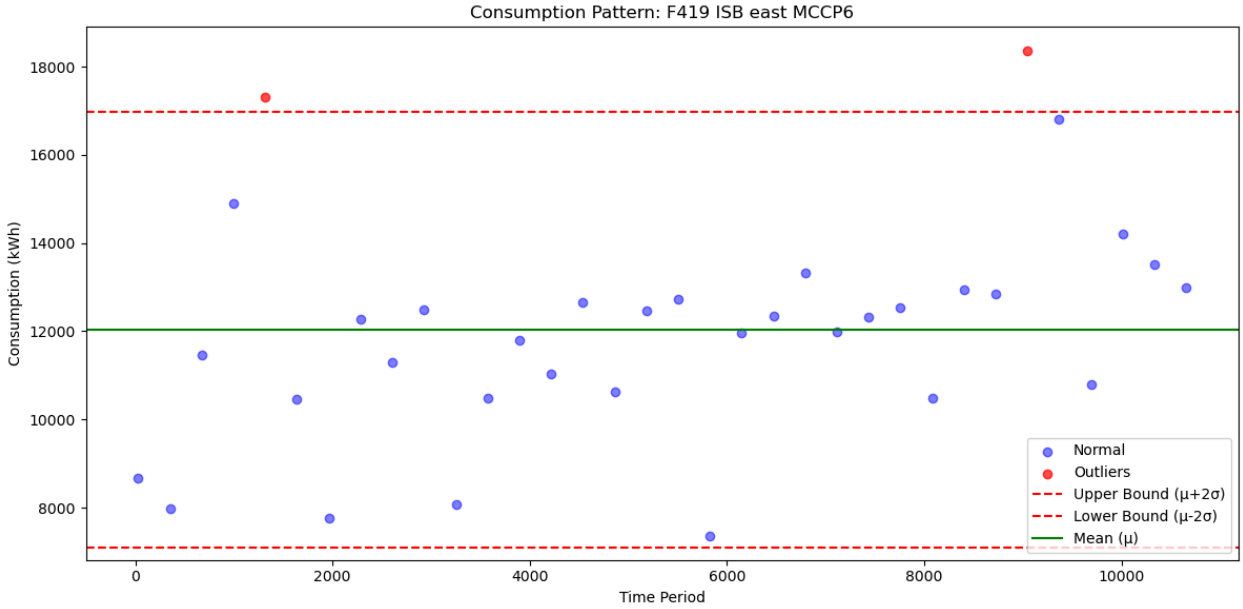

```
Statistics for F419 ISB east Lift:
Mean: 2153.96
Standard Deviation: 755.51
Upper Bound: 3664.98
Lower Bound: 642.95
Number of outliers: 0
```



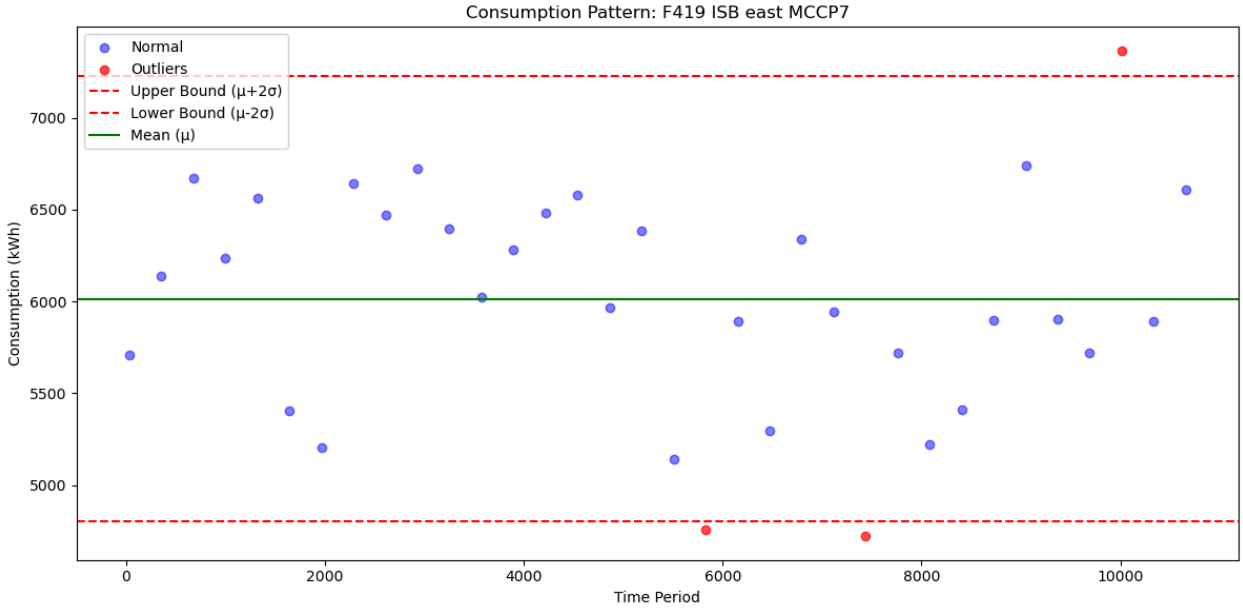

```
Statistics for F419 ISB east MCC1:
Mean: 11631.90
Standard Deviation: 1482.71
Upper Bound: 14597.32
Lower Bound: 8666.49
Number of outliers: 2
```



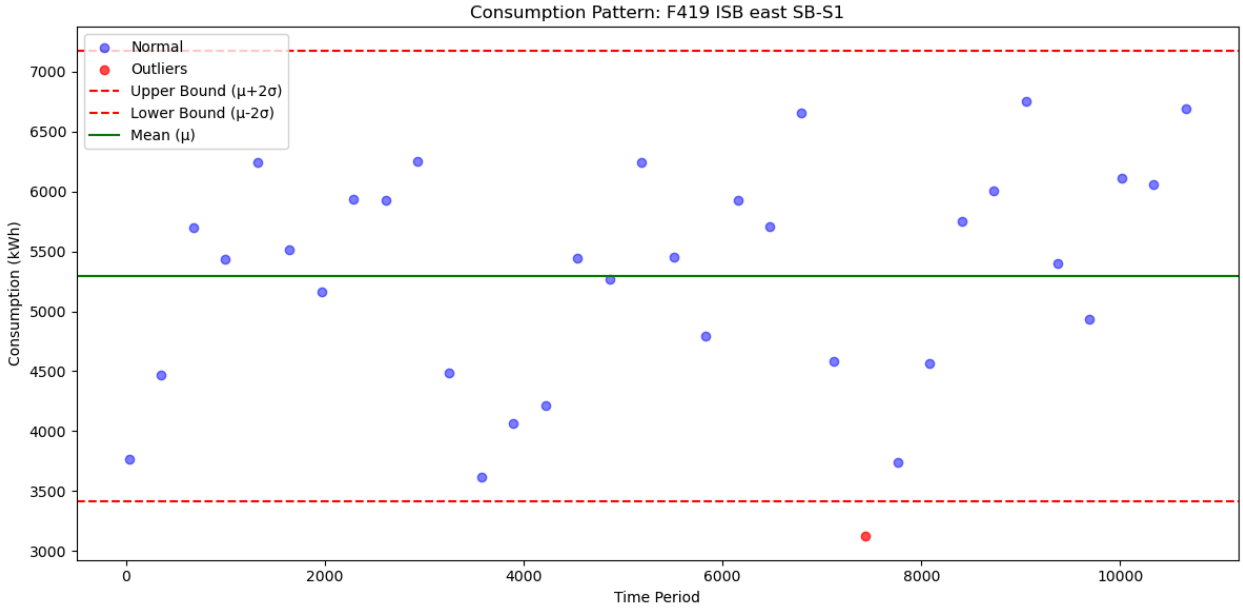

```
Statistics for F419 ISB east MCCP6:
Mean: 12035.29
Standard Deviation: 2473.72
Upper Bound: 16982.74
Lower Bound: 7087.85
Number of outliers: 2
```



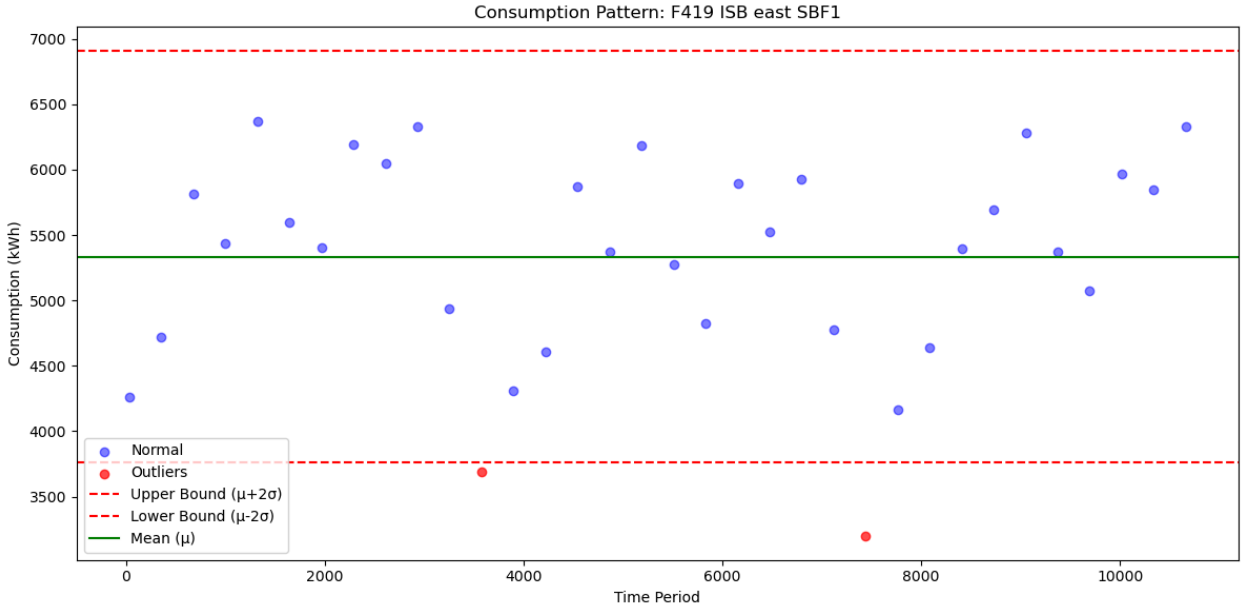

```
Statistics for F419 ISB east MCCP7:
Mean: 6013.10
Standard Deviation: 605.30
Upper Bound: 7223.70
Lower Bound: 4802.50
Number of outliers: 3
```



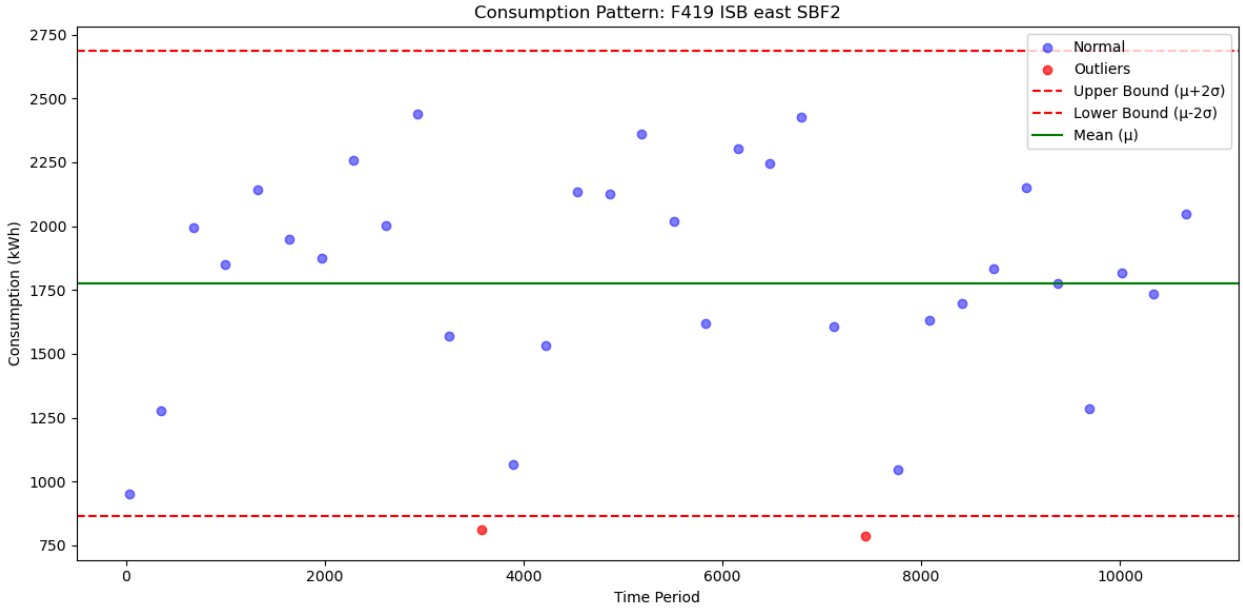

```
Statistics for F419 ISB east SB-S1:
Mean: 5293.64
Standard Deviation: 938.96
Upper Bound: 7171.57
Lower Bound: 3415.71
Number of outliers: 1
```



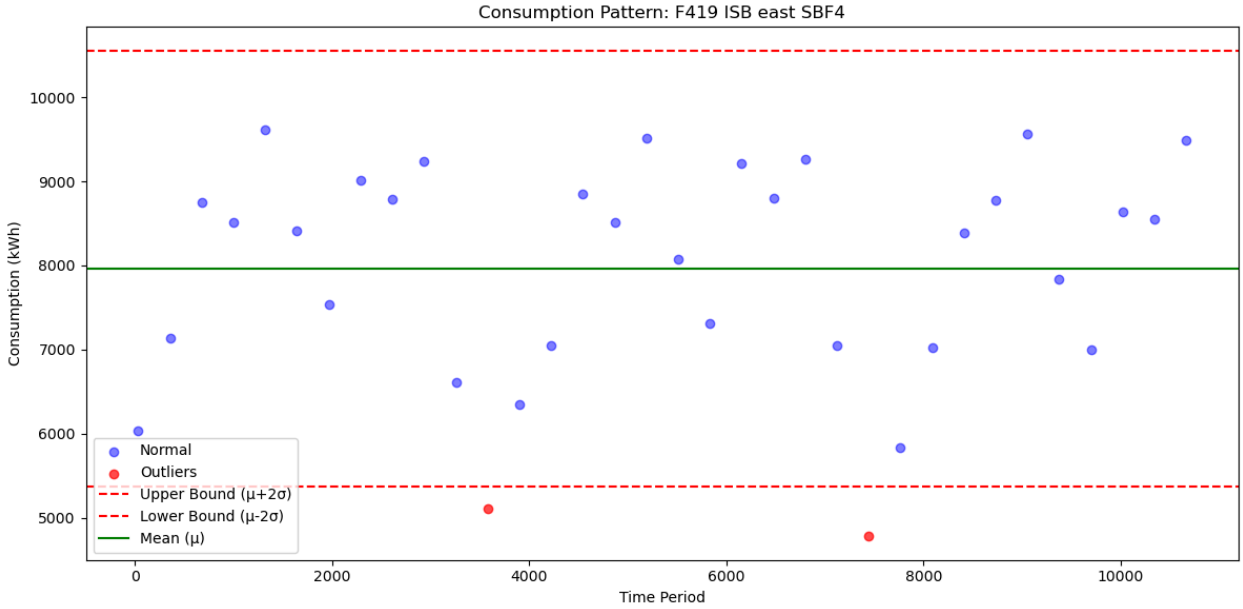

```
Statistics for F419 ISB east SBF1:
Mean: 5333.73
Standard Deviation: 786.55
Upper Bound: 6906.83
Lower Bound: 3760.63
Number of outliers: 2
```



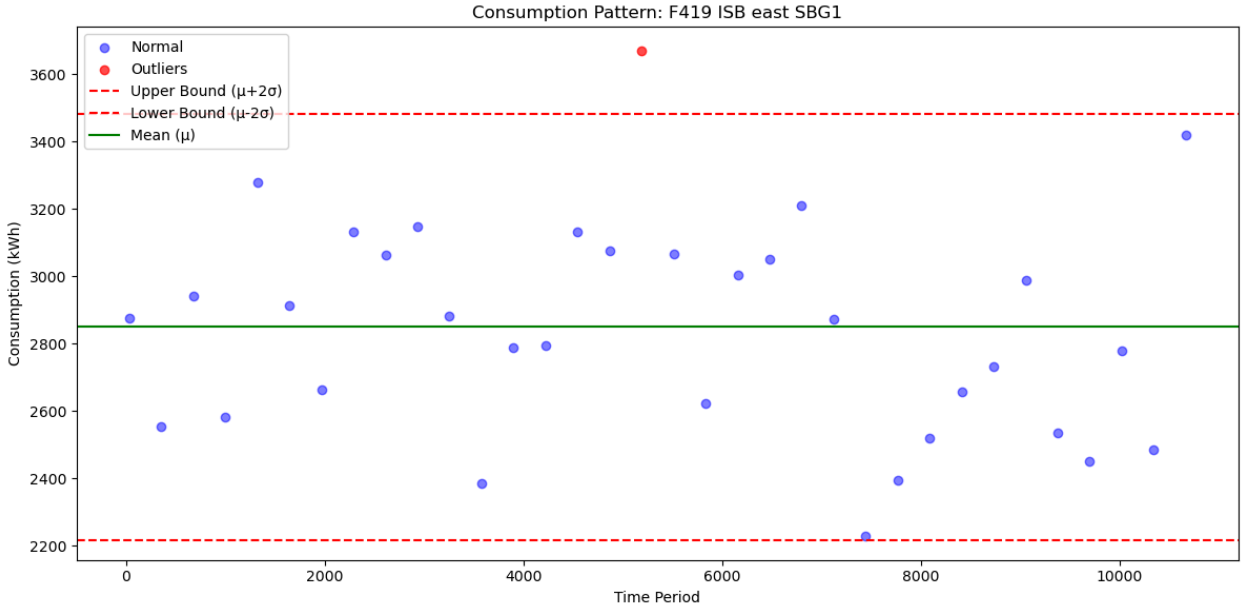

```
Statistics for F419 ISB east SBF2:
Mean: 1775.46
Standard Deviation: 455.07
Upper Bound: 2685.60
Lower Bound: 865.31
Number of outliers: 2
```



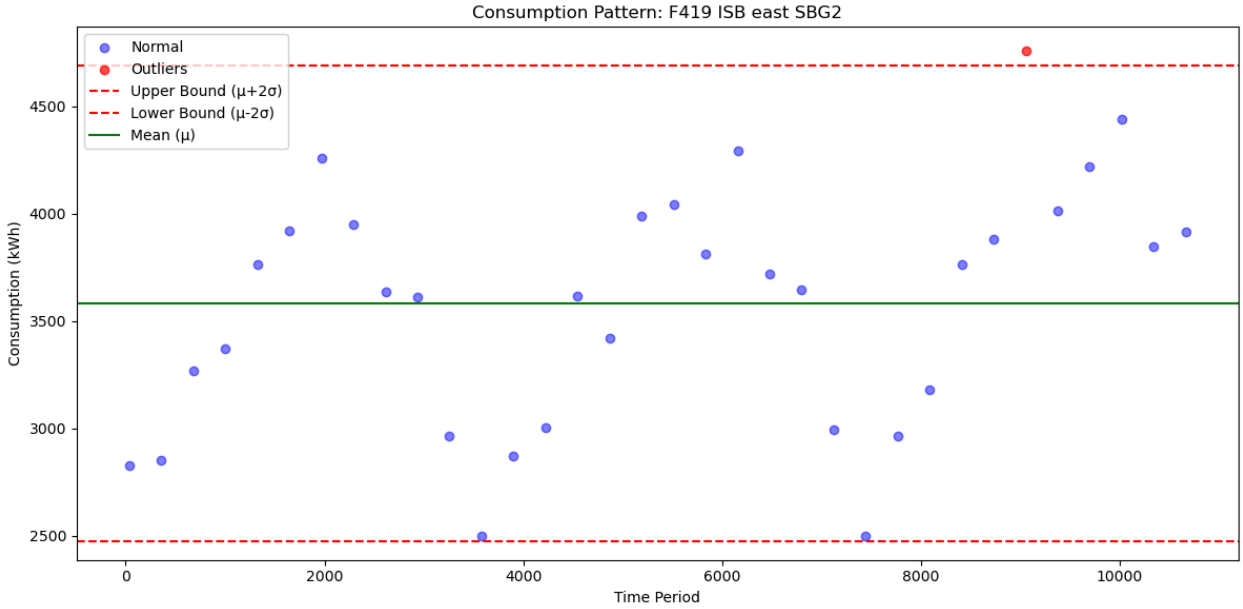

```
Statistics for F419 ISB east SBF4:
Mean: 7960.18
Standard Deviation: 1294.11
Upper Bound: 10548.40
Lower Bound: 5371.96
Number of outliers: 2
```



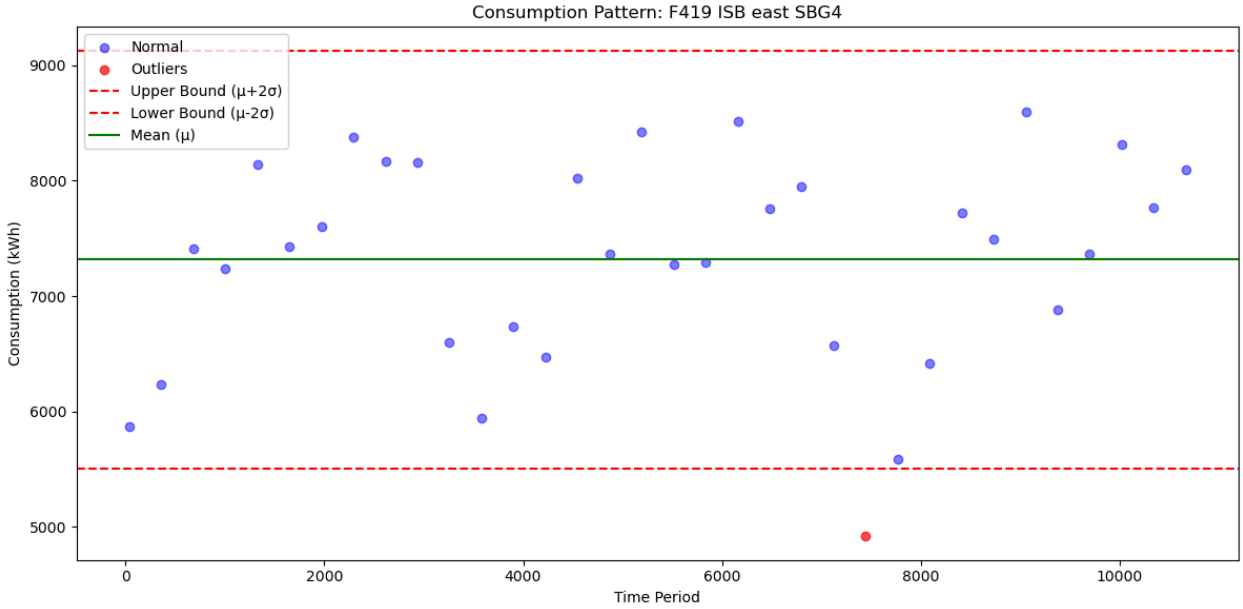

```
Statistics for F419 ISB east SBG1:
Mean: 2849.74
Standard Deviation: 316.91
Upper Bound: 3483.56
Lower Bound: 2215.92
Number of outliers: 1
```



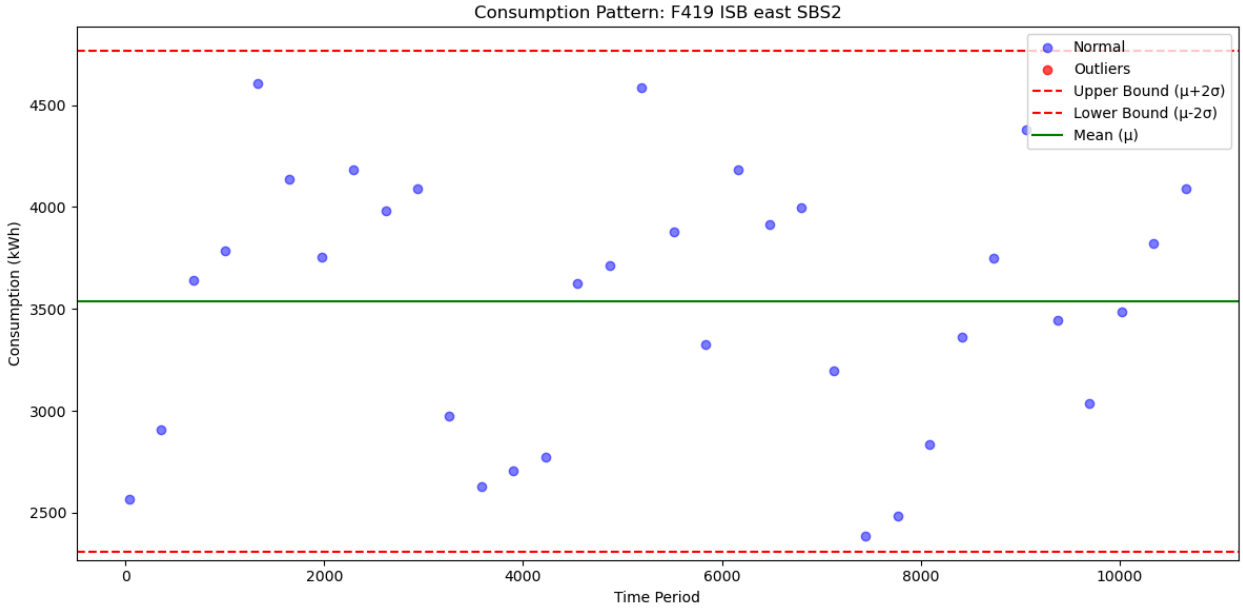

```
Statistics for F419 ISB east SBG2:
Mean: 3583.43
Standard Deviation: 553.50
Upper Bound: 4690.42
Lower Bound: 2476.43
Number of outliers: 1
```



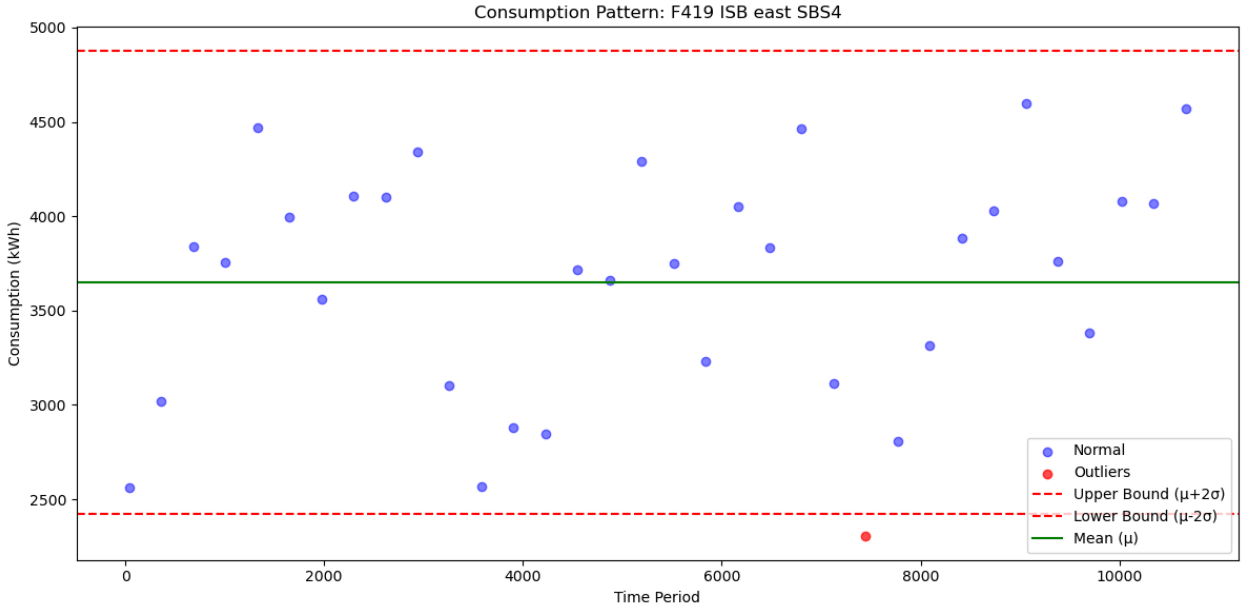

```
Statistics for F419 ISB east SBG4:
Mean: 7315.81
Standard Deviation: 903.62
Upper Bound: 9123.05
Lower Bound: 5508.58
Number of outliers: 1
```



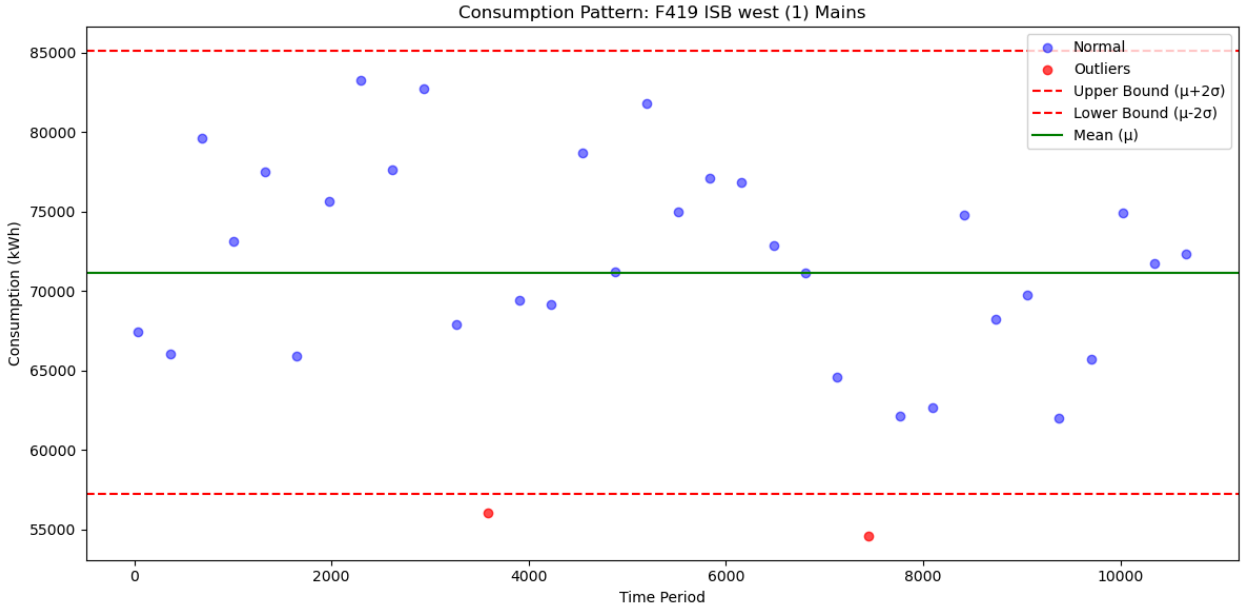

```
Statistics for F419 ISB east SBS2:
Mean: 3535.94
Standard Deviation: 614.15
Upper Bound: 4764.25
Lower Bound: 2307.64
Number of outliers: 0
```



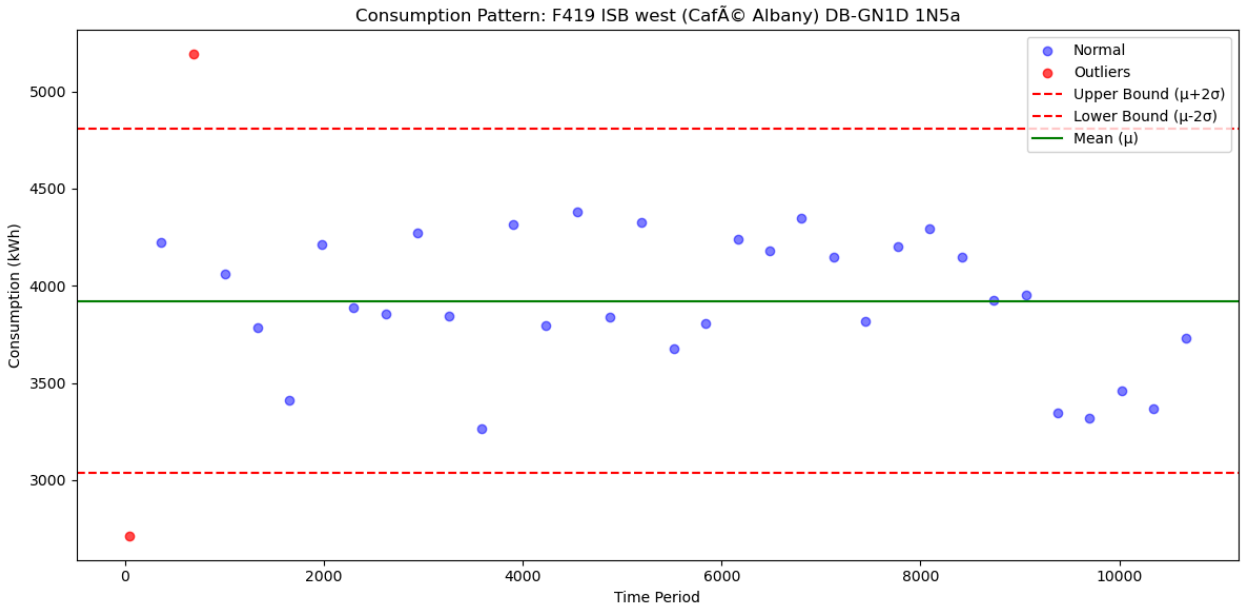

```
Statistics for F419 ISB east SBS4:
Mean: 3648.88
Standard Deviation: 612.91
Upper Bound: 4874.71
Lower Bound: 2423.06
Number of outliers: 1
```



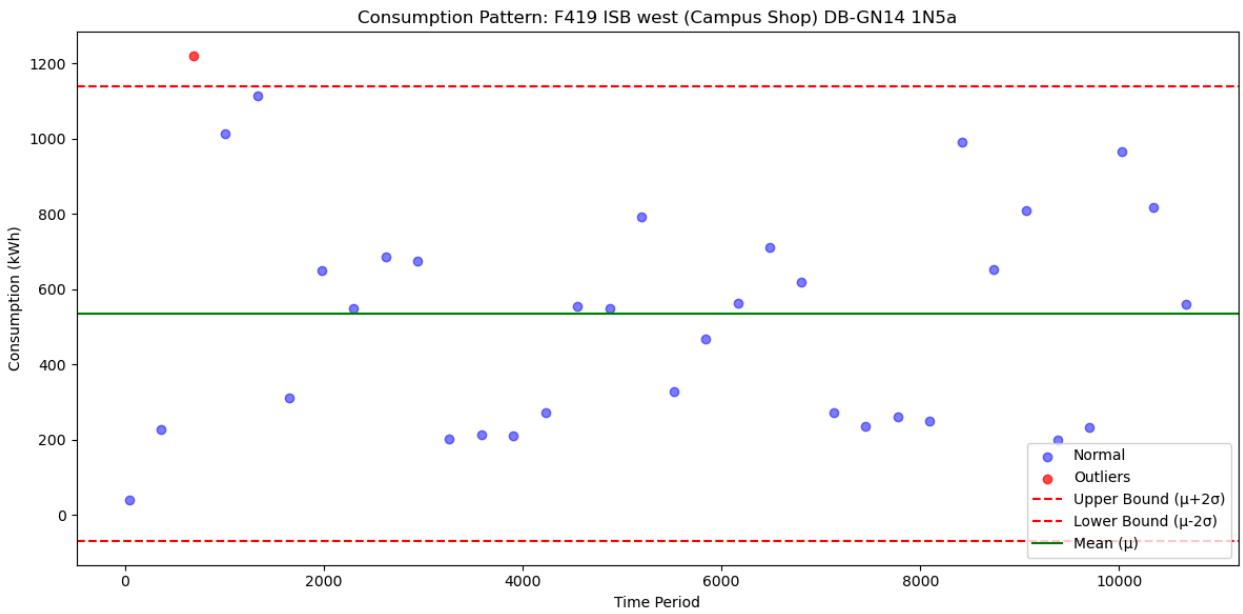

```
Statistics for F419 ISB west (1) Mains:
Mean: 71164.24
Standard Deviation: 6967.94
Upper Bound: 85100.13
Lower Bound: 57228.36
Number of outliers: 2
```



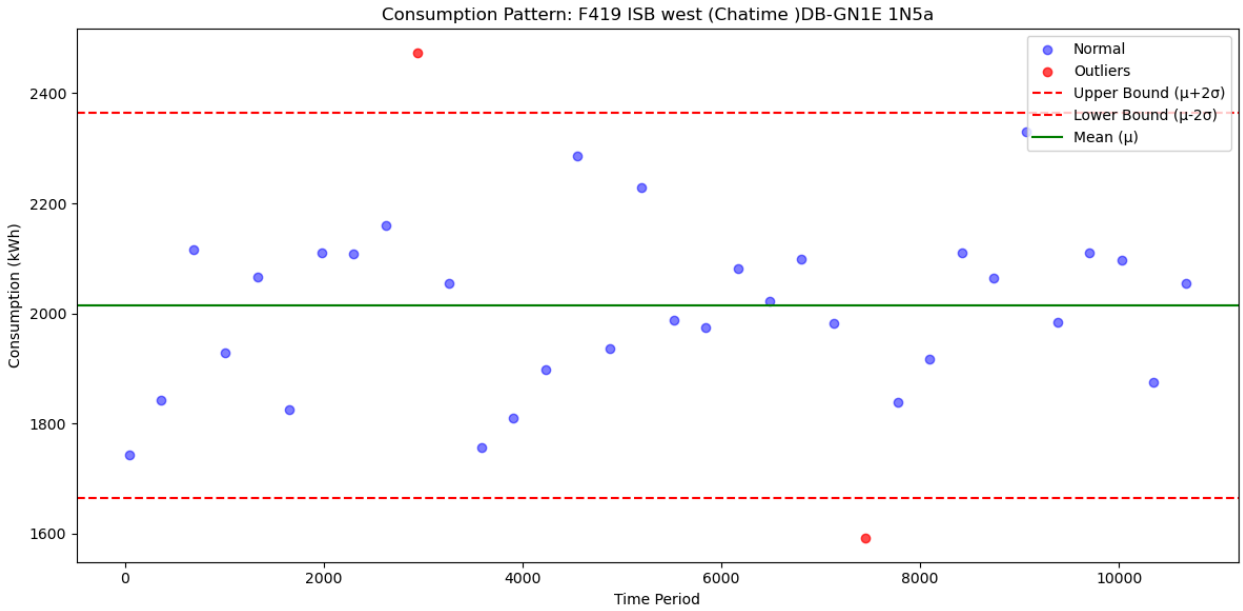

```
Statistics for F419 ISB west (CafÃ© Albany) DB-GN1D 1N5a:
Mean: 3921.52
Standard Deviation: 443.06
Upper Bound: 4807.64
Lower Bound: 3035.41
Number of outliers: 2
```



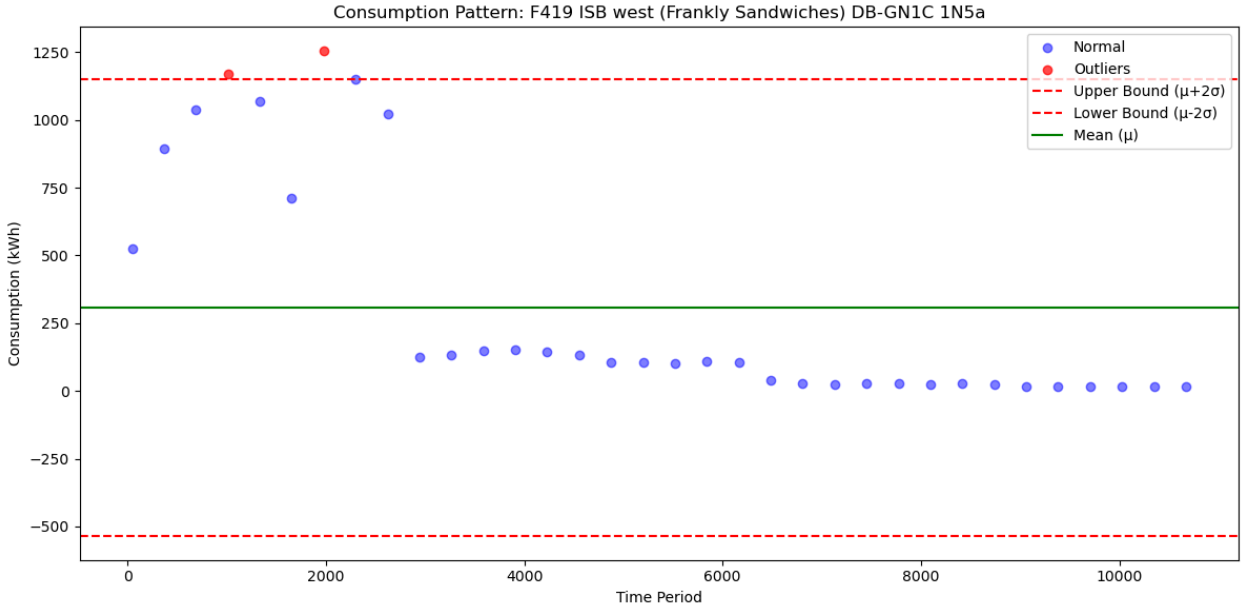

```
Statistics for F419 ISB west (Campus Shop) DB-GN14 1N5a:
Mean: 536.19
Standard Deviation: 302.35
Upper Bound: 1140.90
Lower Bound: -68.52
Number of outliers: 1
```



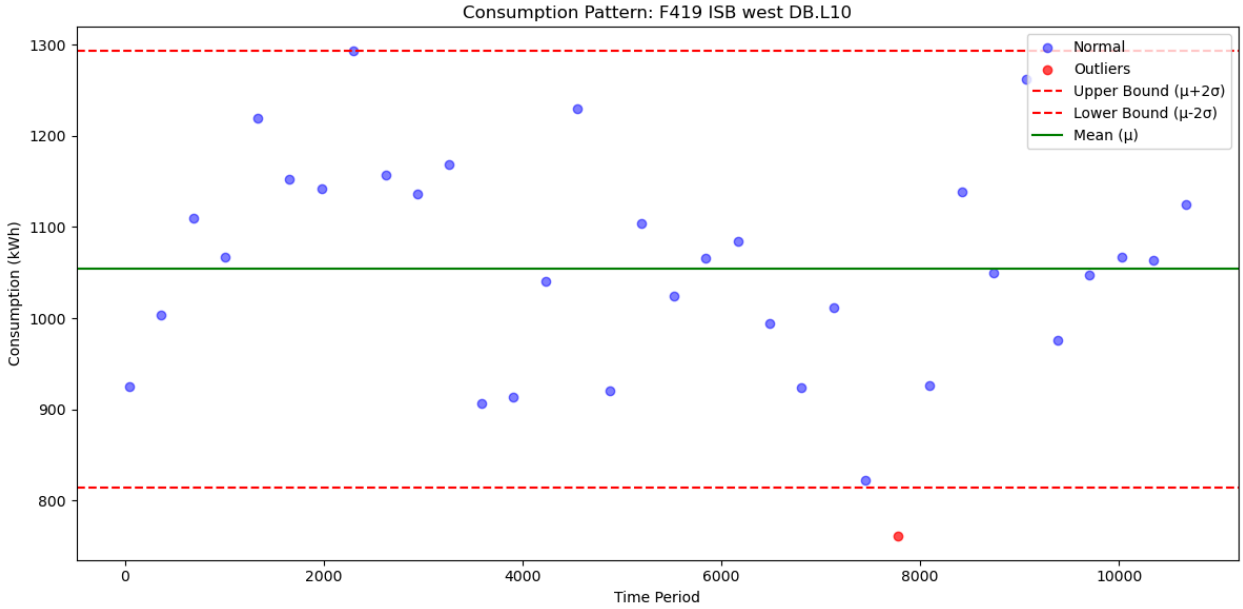

```
Statistics for F419 ISB west (Chatime )DB-GN1E 1N5a:
Mean: 2013.54
Standard Deviation: 175.10
Upper Bound: 2363.74
Lower Bound: 1663.34
Number of outliers: 2
```



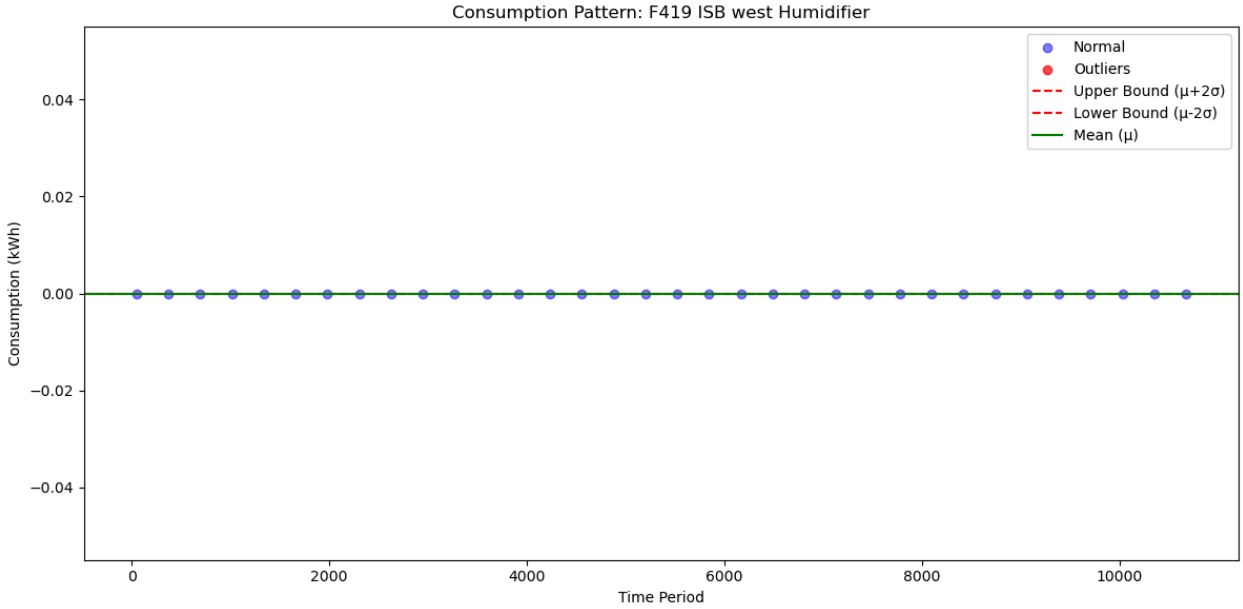

```
Statistics for F419 ISB west (Frankly Sandwiches) DB-GN1C 1N5a:
Mean: 309.14
Standard Deviation: 421.56
Upper Bound: 1152.27
Lower Bound: -533.98
Number of outliers: 2
```



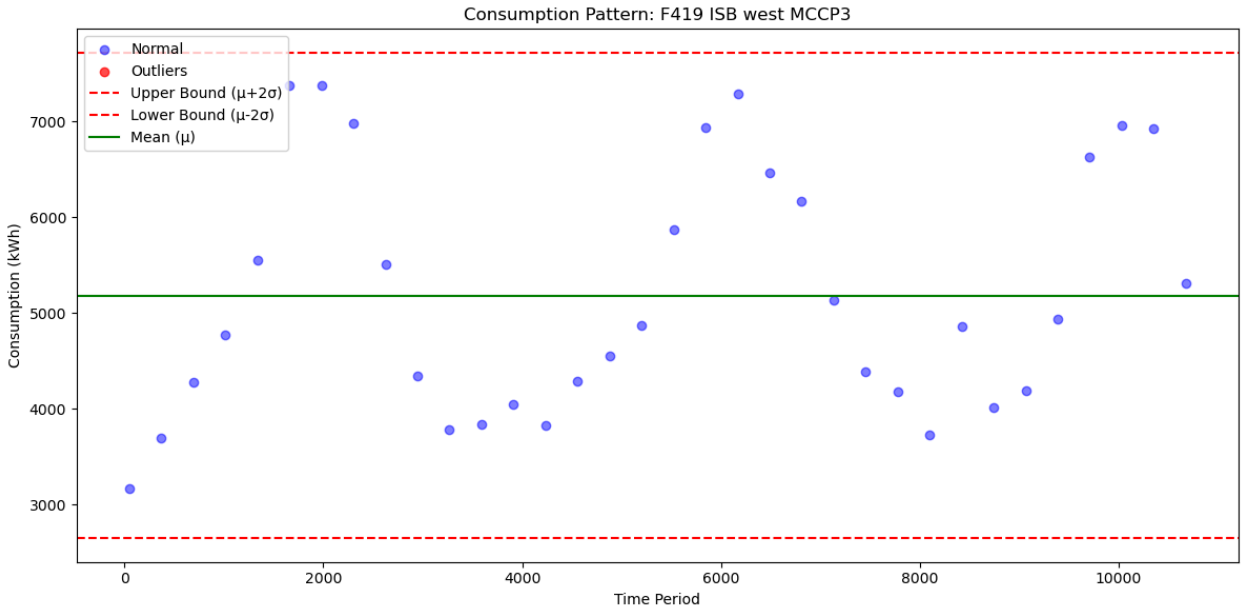

```
Statistics for F419 ISB west DB.L10:
Mean: 1053.90
Standard Deviation: 120.00
Upper Bound: 1293.89
Lower Bound: 813.90
Number of outliers: 1
```



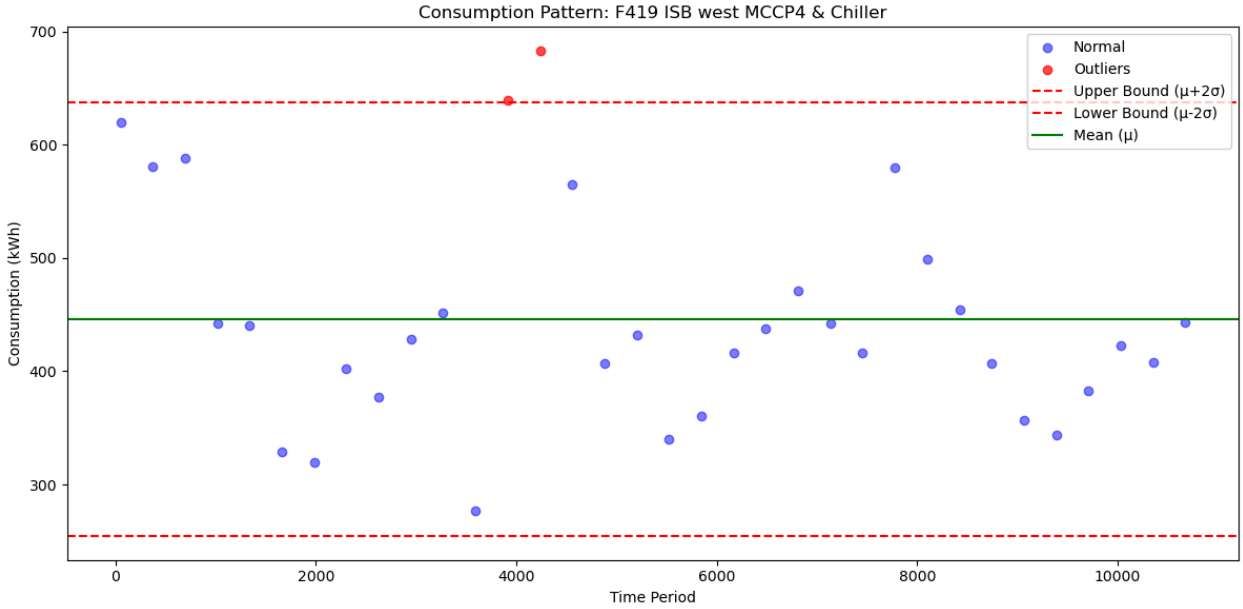

```
Statistics for F419 ISB west Humidifier:
Mean: 0.00
Standard Deviation: 0.00
Upper Bound: 0.00
Lower Bound: 0.00
Number of outliers: 0
```



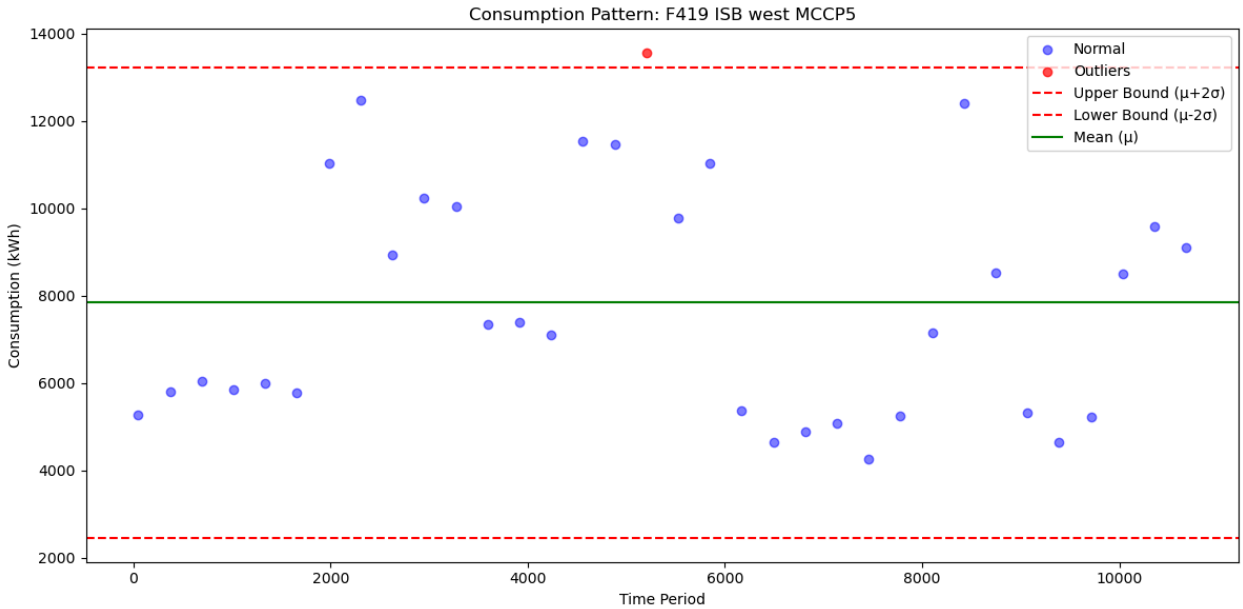

```
Statistics for F419 ISB west MCCP3:
Mean: 5183.60
Standard Deviation: 1266.08
Upper Bound: 7715.76
Lower Bound: 2651.44
Number of outliers: 0
```



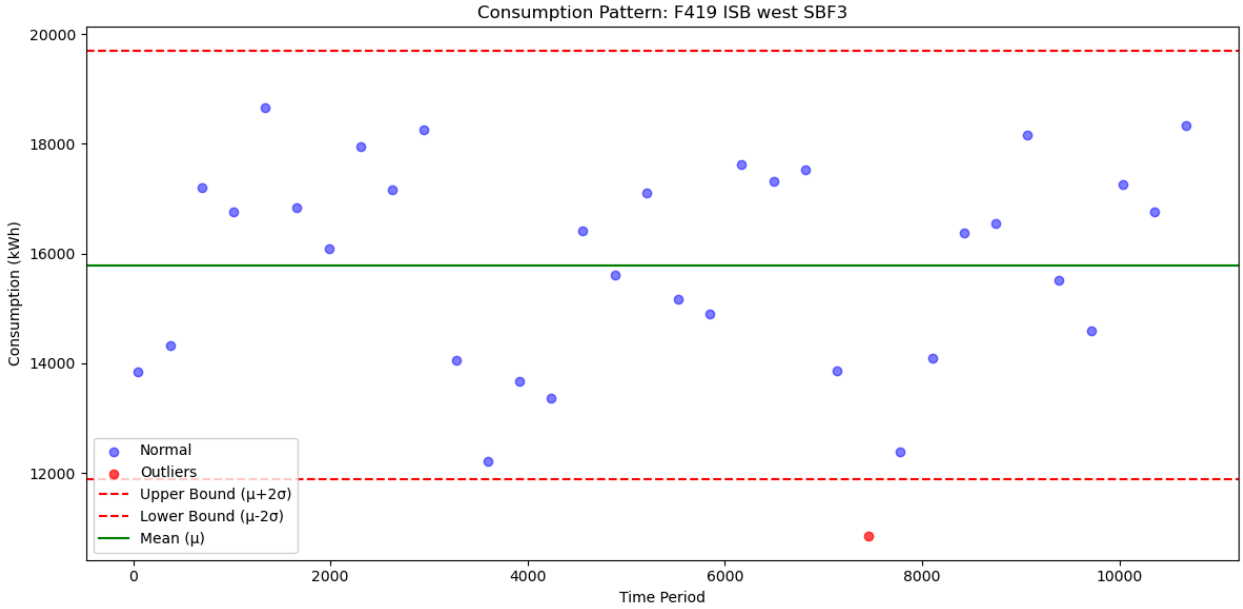

```
Statistics for F419 ISB west MCCP4 & Chiller:
Mean: 446.09
Standard Deviation: 95.63
Upper Bound: 637.35
Lower Bound: 254.82
Number of outliers: 2
```



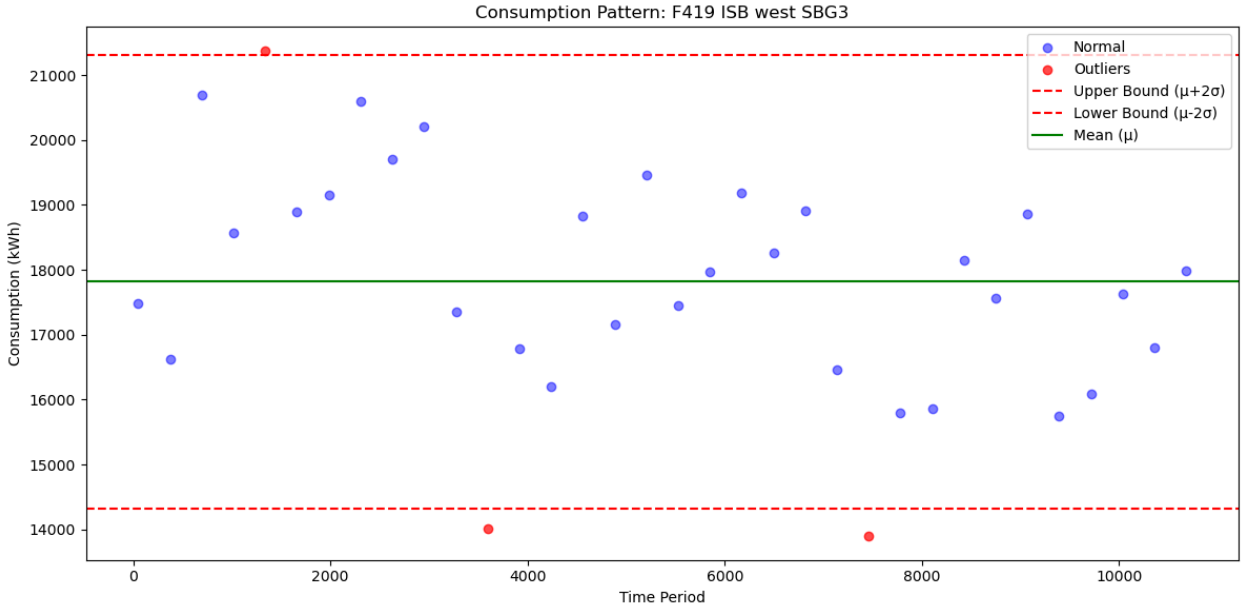

```
Statistics for F419 ISB west MCCP5:
Mean: 7835.98
Standard Deviation: 2690.93
Upper Bound: 13217.83
Lower Bound: 2454.12
Number of outliers: 1
```



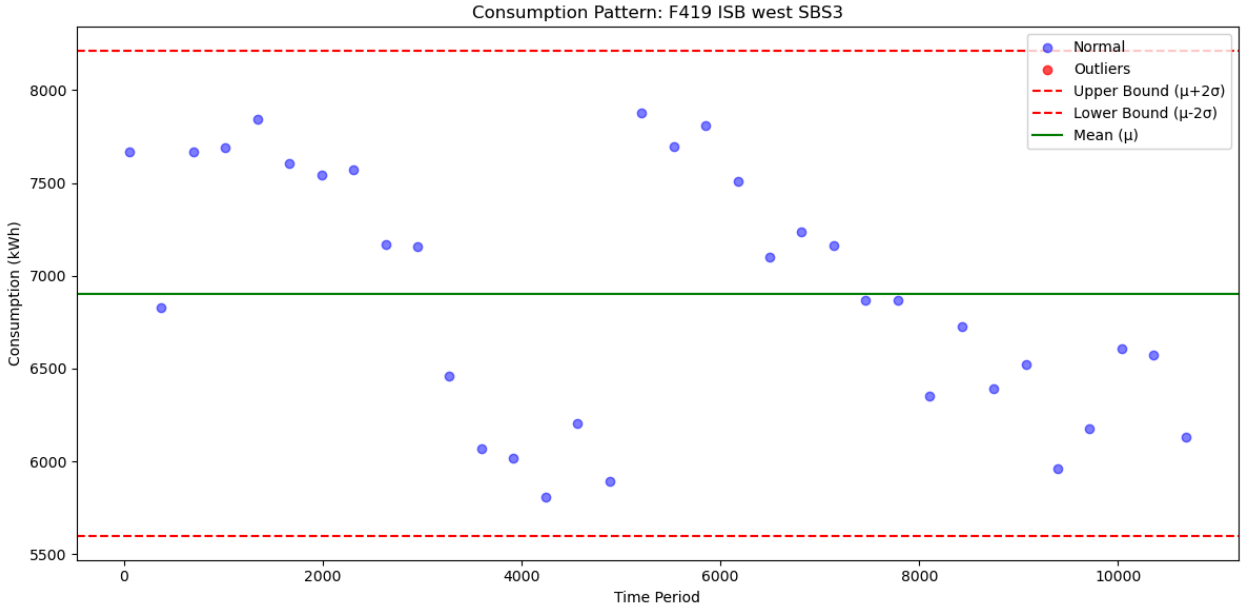

```
Statistics for F419 ISB west SBF3:
Mean: 15790.34
Standard Deviation: 1949.62
Upper Bound: 19689.59
Lower Bound: 11891.10
Number of outliers: 1
```



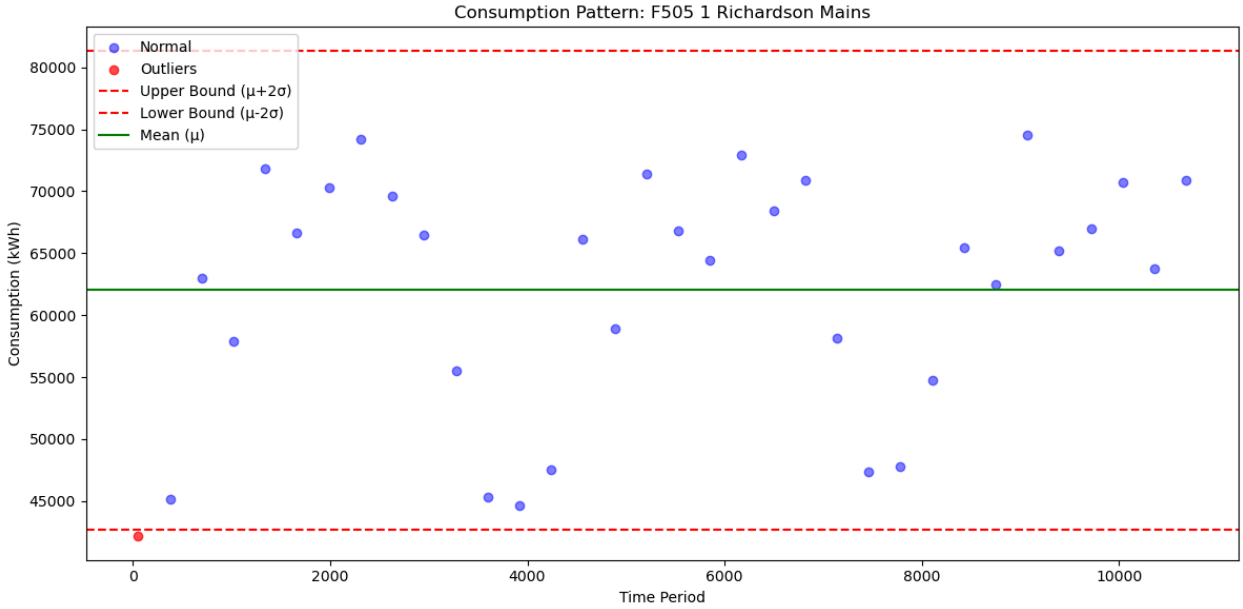

```
Statistics for F419 ISB west SBG3:
Mean: 17816.78
Standard Deviation: 1745.99
Upper Bound: 21308.76
Lower Bound: 14324.80
Number of outliers: 3
```



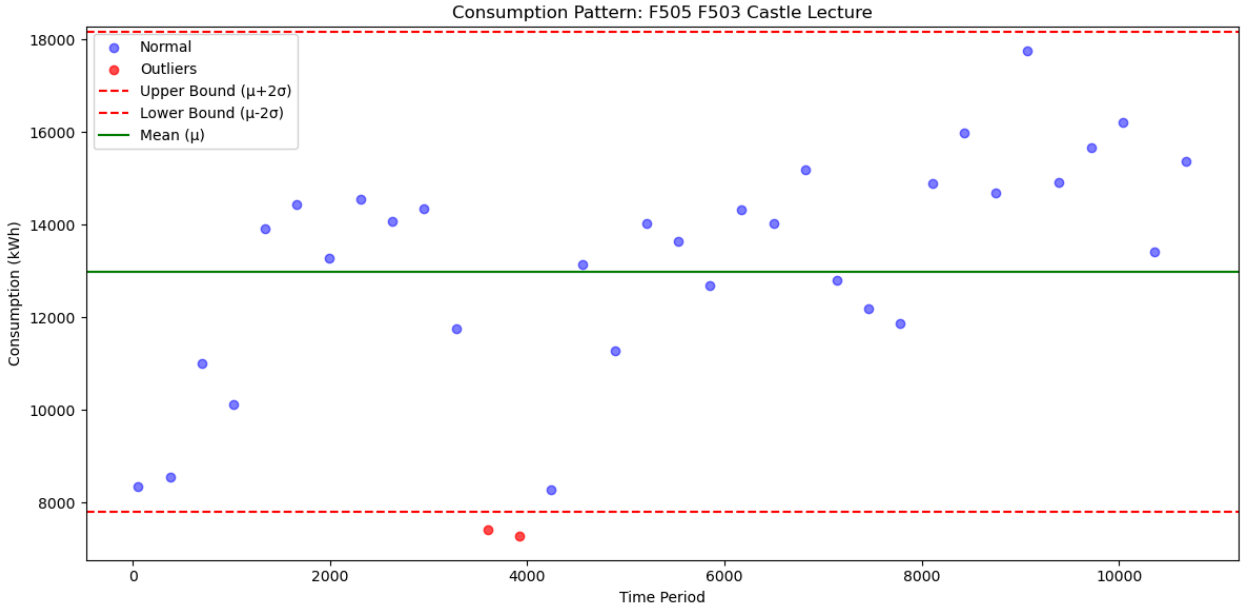

```
Statistics for F419 ISB west SBS3:
Mean: 6904.74
Standard Deviation: 652.62
Upper Bound: 8209.98
Lower Bound: 5599.51
Number of outliers: 0
```



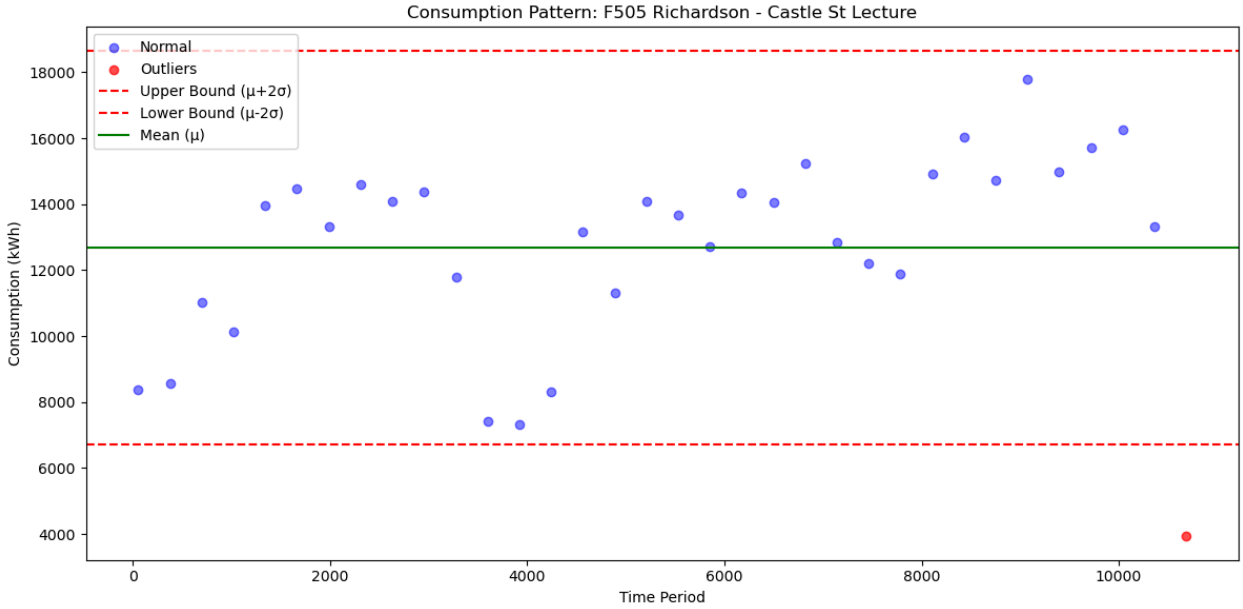

```
Statistics for F505 1 Richardson Mains:
Mean: 62012.27
Standard Deviation: 9648.10
Upper Bound: 81308.48
Lower Bound: 42716.07
Number of outliers: 1
```



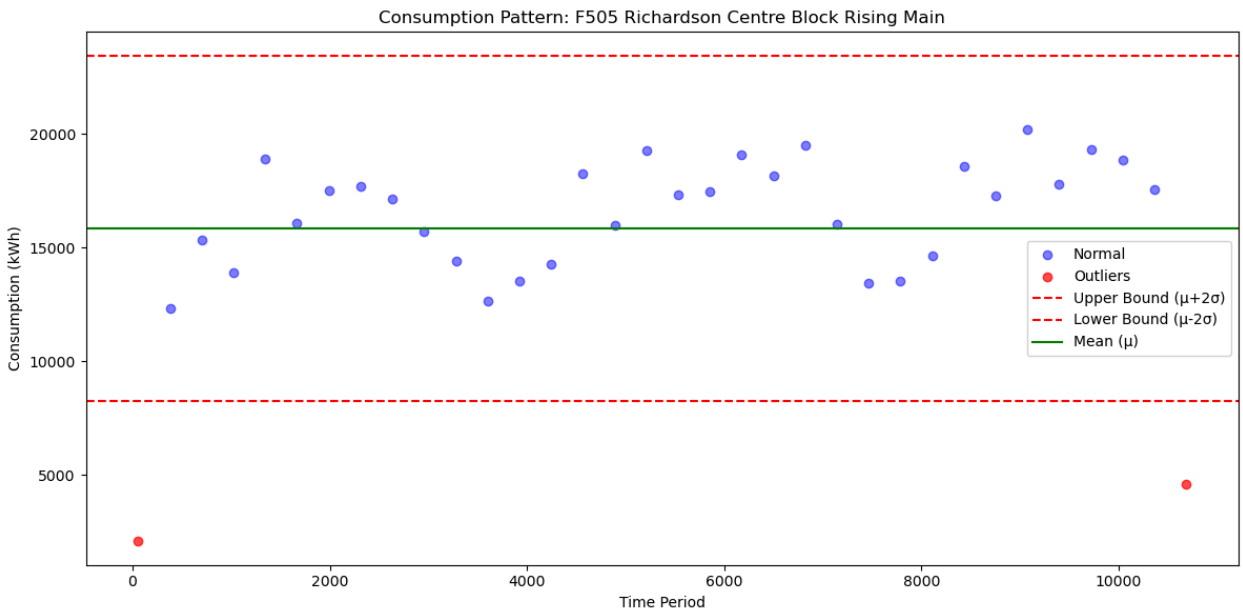

```
Statistics for F505 F503 Castle Lecture:
Mean: 12982.18
Standard Deviation: 2592.24
Upper Bound: 18166.66
Lower Bound: 7797.71
Number of outliers: 2
```



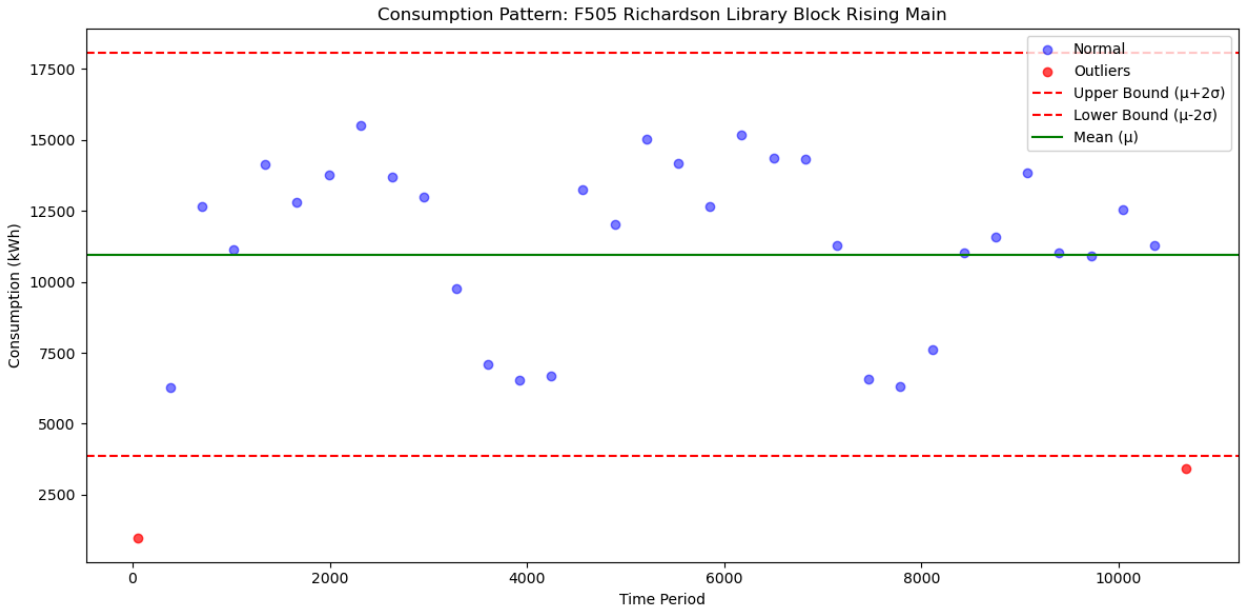

```
Statistics for F505 Richardson - Castle St Lecture:
Mean: 12675.19
Standard Deviation: 2983.58
Upper Bound: 18642.35
Lower Bound: 6708.03
Number of outliers: 1
```



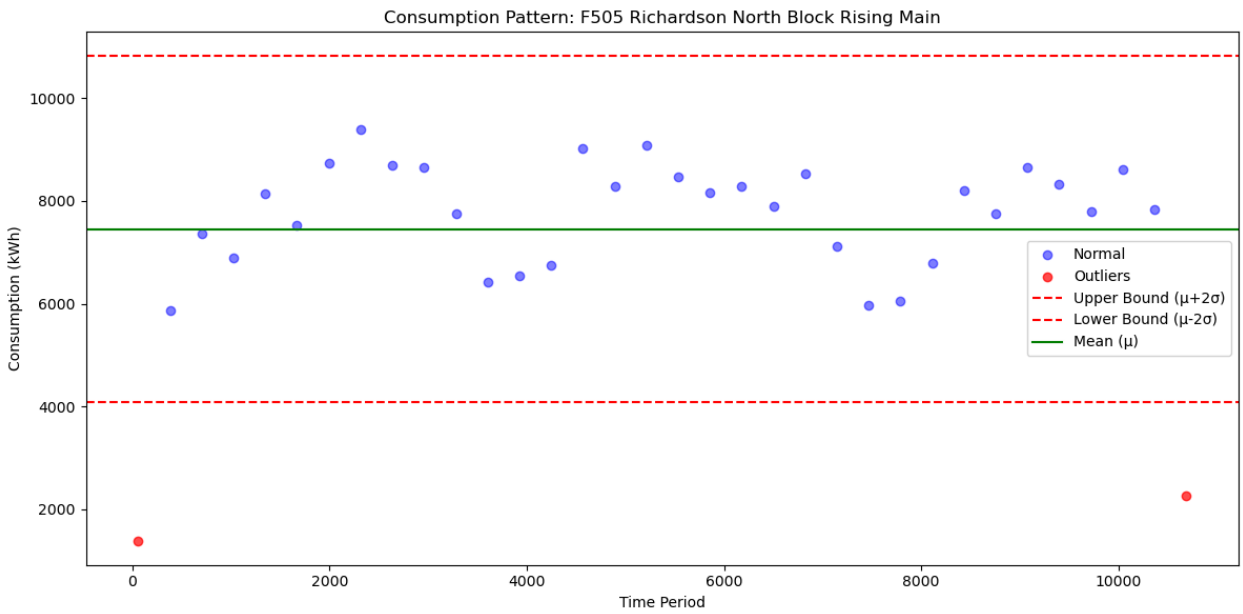

```
Statistics for F505 Richardson Centre Block Rising Main:
Mean: 15834.75
Standard Deviation: 3793.49
Upper Bound: 23421.73
Lower Bound: 8247.78
Number of outliers: 2
```



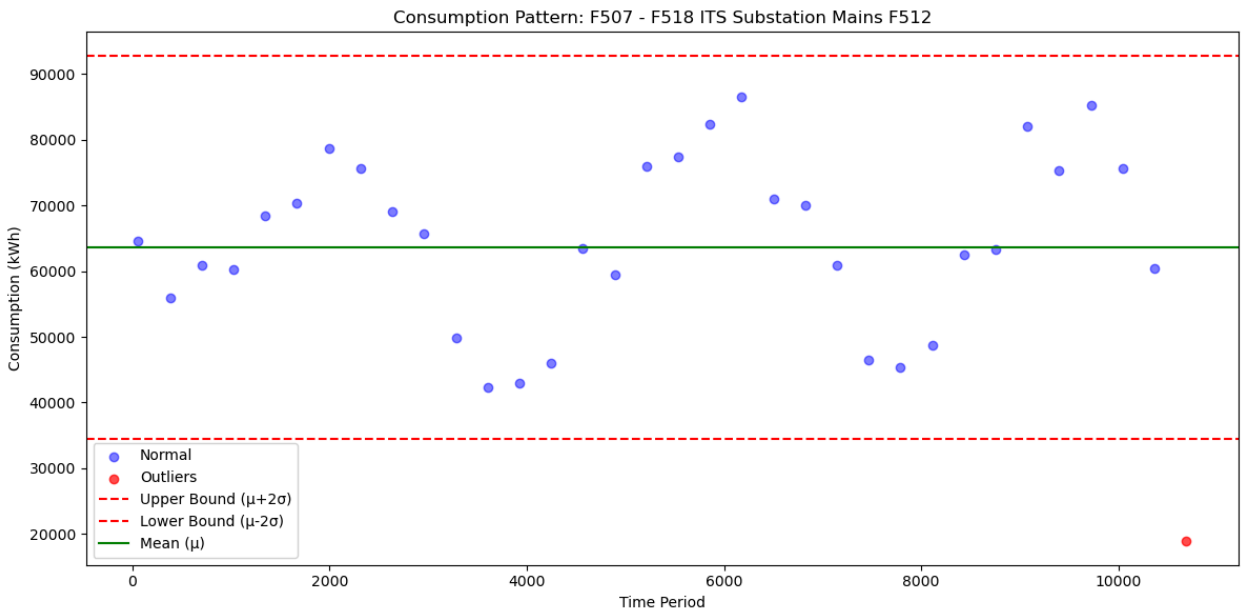

```
Statistics for F505 Richardson Library Block Rising Main:
Mean: 10948.80
Standard Deviation: 3548.41
Upper Bound: 18045.63
Lower Bound: 3851.97
Number of outliers: 2
```



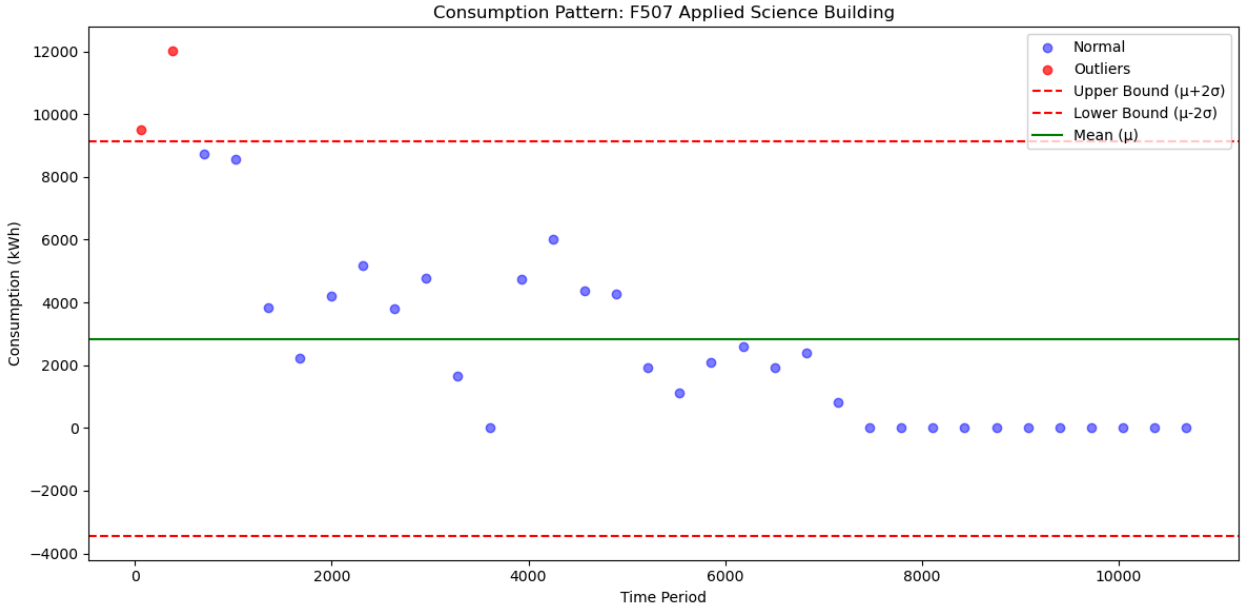

```
Statistics for F505 Richardson North Block Rising Main:
Mean: 7447.83
Standard Deviation: 1683.83
Upper Bound: 10815.49
Lower Bound: 4080.16
Number of outliers: 2
```



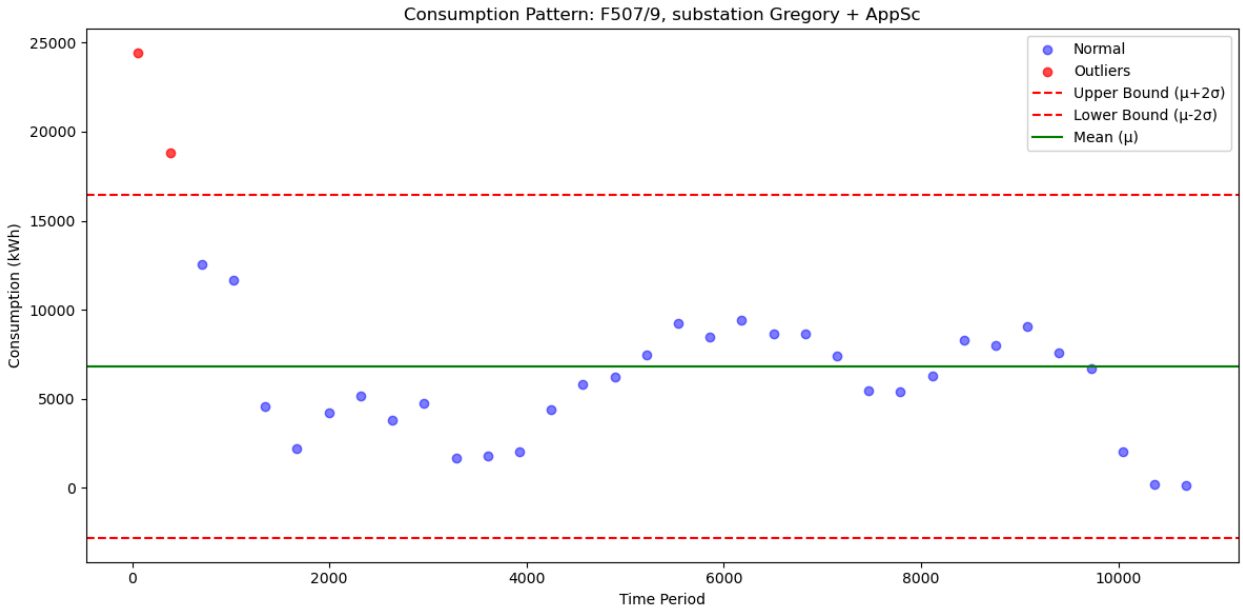

```
Statistics for F507 - F518 ITS Substation Mains F512:
Mean: 63583.77
Standard Deviation: 14572.09
Upper Bound: 92727.96
Lower Bound: 34439.59
Number of outliers: 1
```



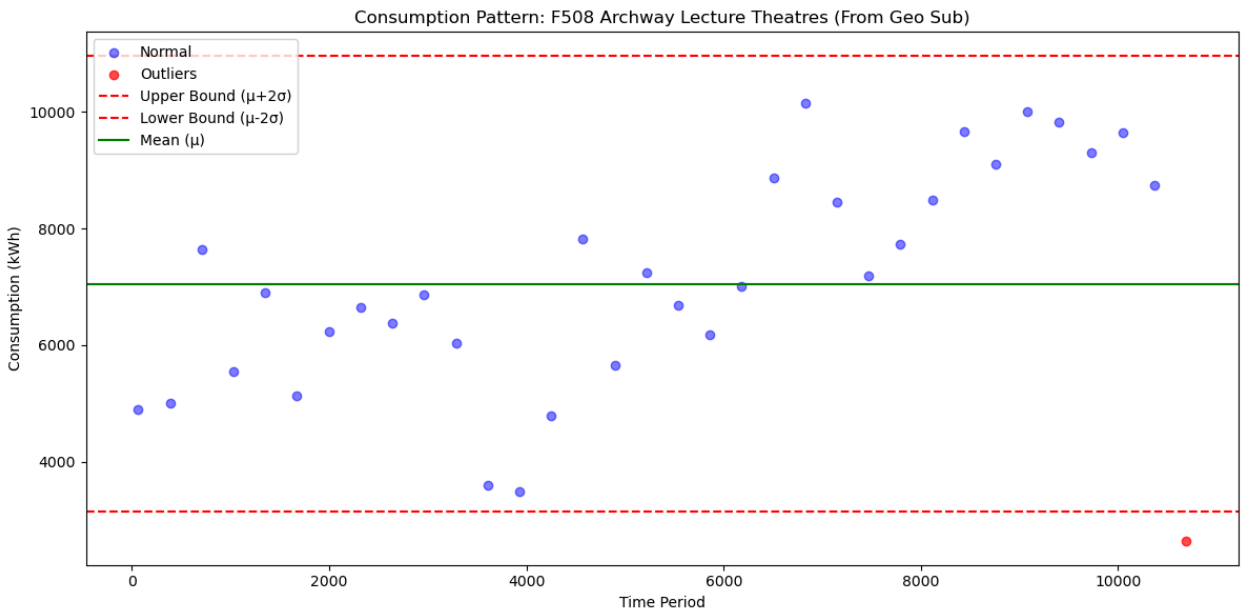

```
Statistics for F507 Applied Science Building:
Mean: 2844.71
Standard Deviation: 3139.58
Upper Bound: 9123.87
Lower Bound: -3434.46
Number of outliers: 2
```



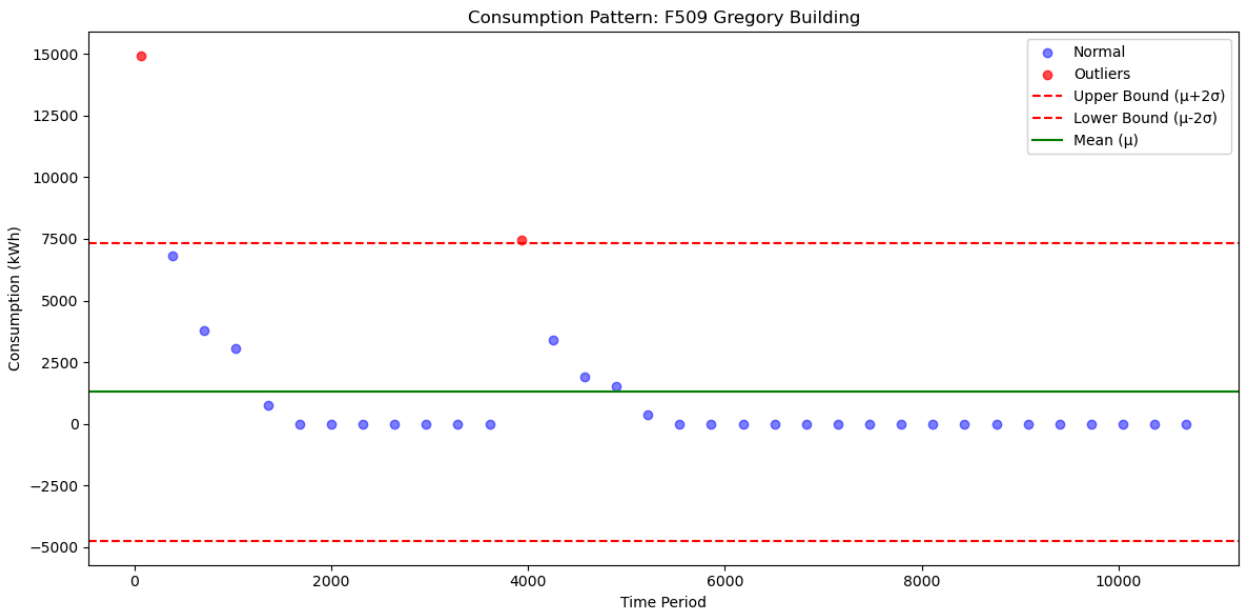

```
Statistics for F507/9, substation Gregory + AppSc:
Mean: 6840.24
Standard Deviation: 4817.29
Upper Bound: 16474.82
Lower Bound: -2794.34
Number of outliers: 2
```



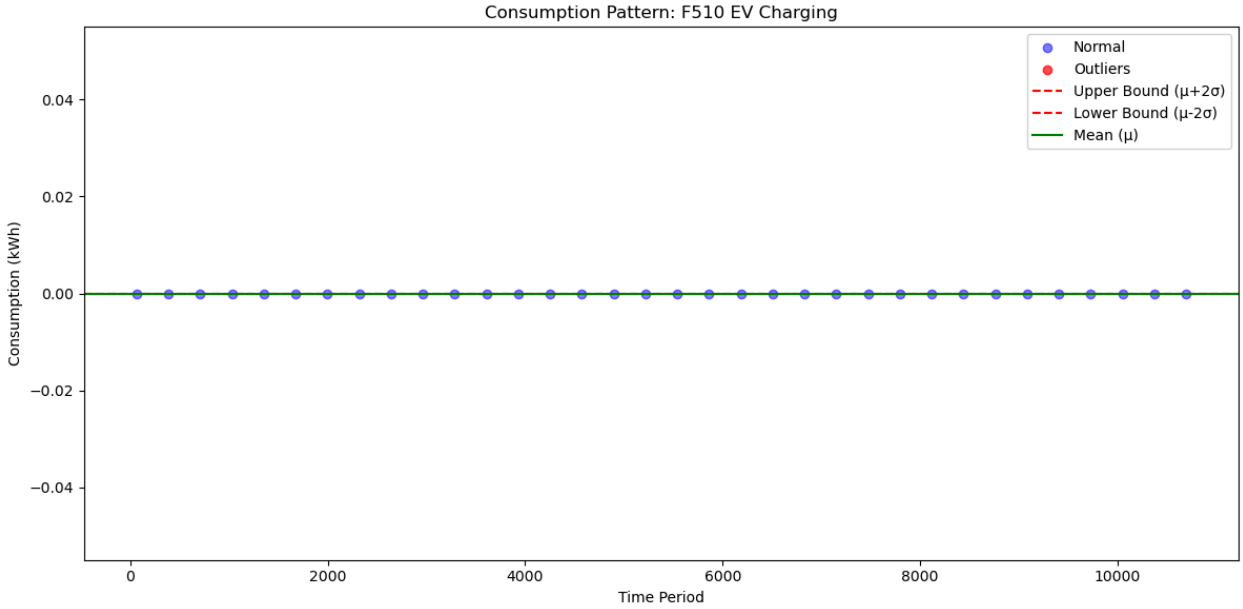

```
Statistics for F508 Archway Lecture Theatres (From Geo Sub):
Mean: 7046.32
Standard Deviation: 1954.81
Upper Bound: 10955.94
Lower Bound: 3136.71
Number of outliers: 1
```



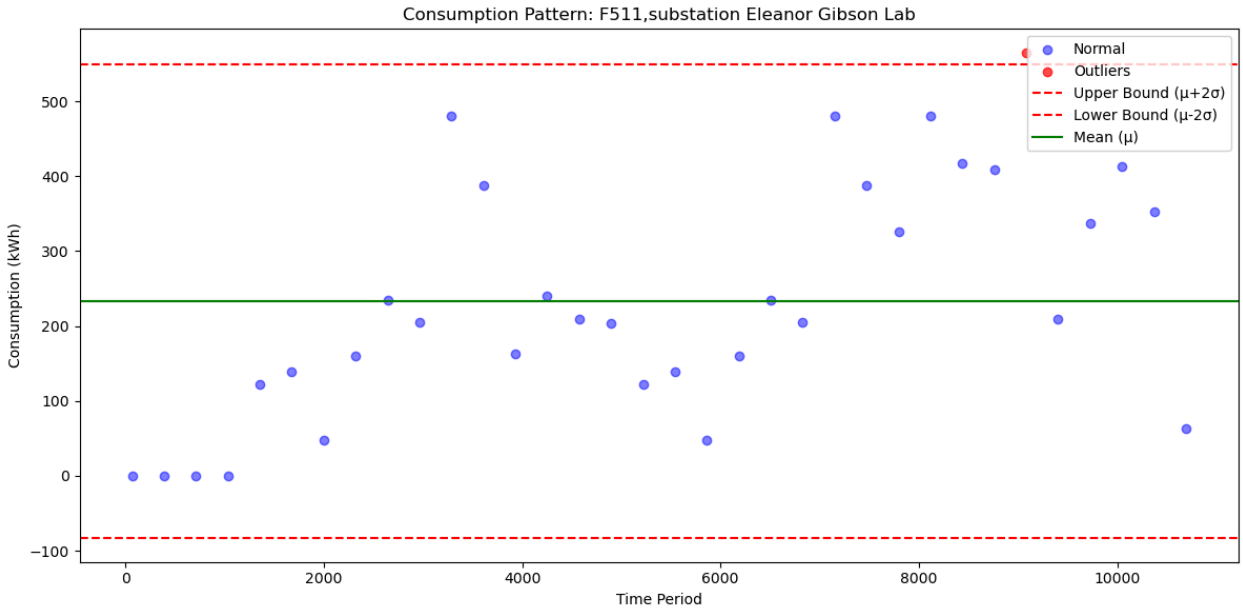

```
Statistics for F509 Gregory Building:
Mean: 1293.88
Standard Deviation: 3016.84
Upper Bound: 7327.56
Lower Bound: -4739.80
Number of outliers: 2
```



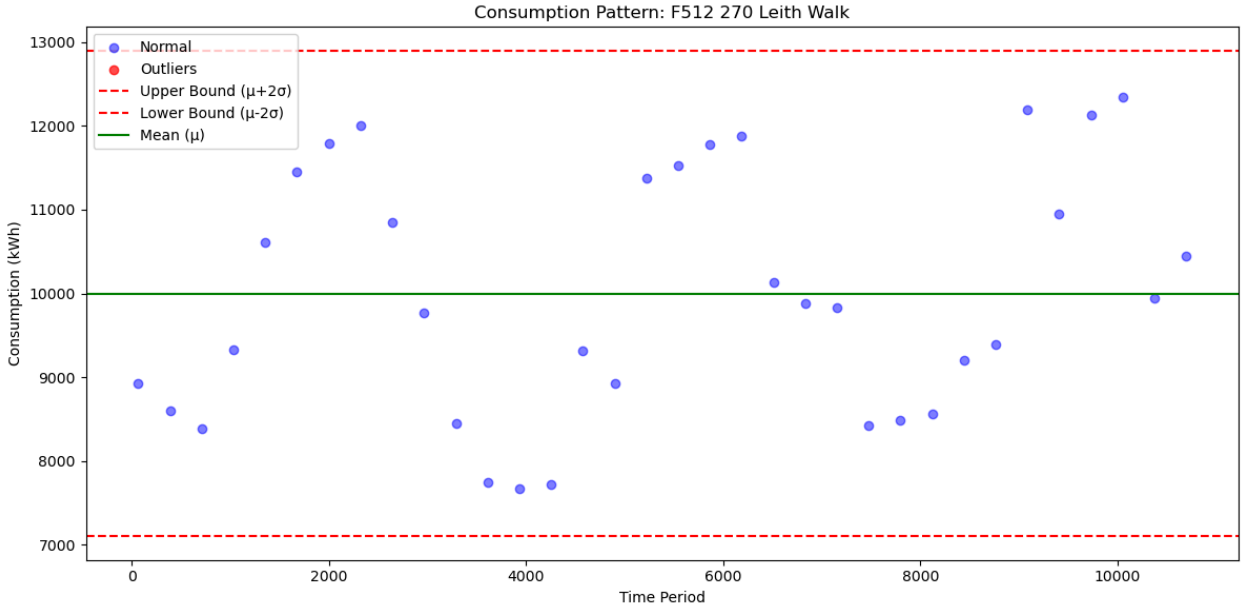

```
Statistics for F510 EV Charging:
Mean: 0.00
Standard Deviation: 0.00
Upper Bound: 0.00
Lower Bound: 0.00
Number of outliers: 0
```



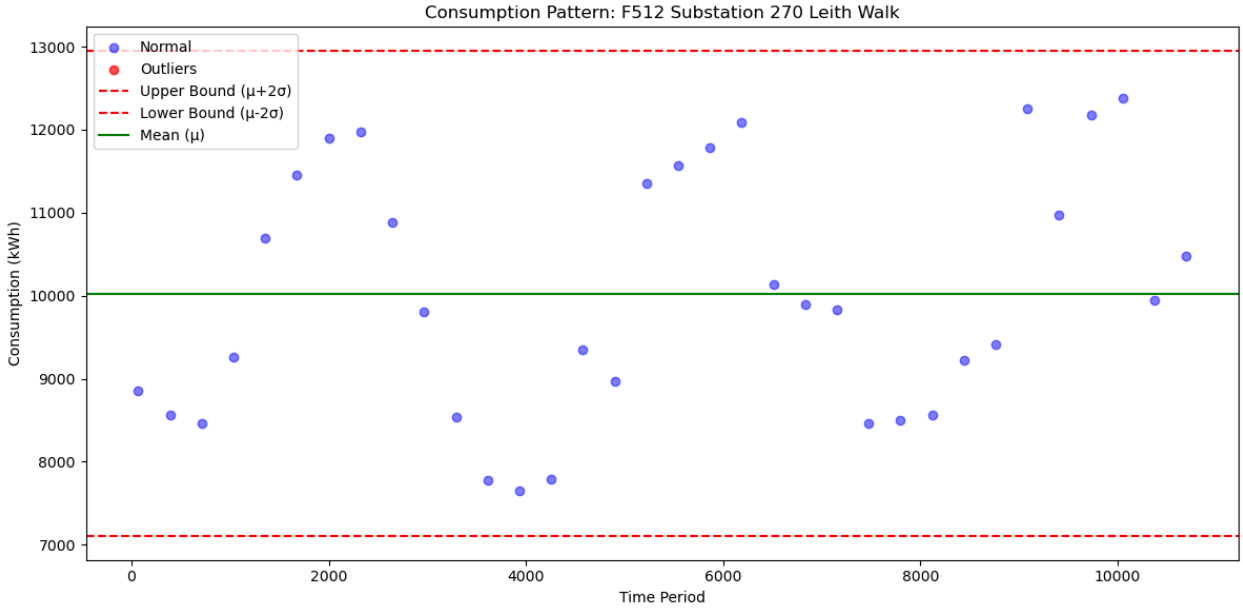

```
Statistics for F511,substation Eleanor Gibson Lab:
Mean: 233.58
Standard Deviation: 158.10
Upper Bound: 549.78
Lower Bound: -82.61
Number of outliers: 1
```



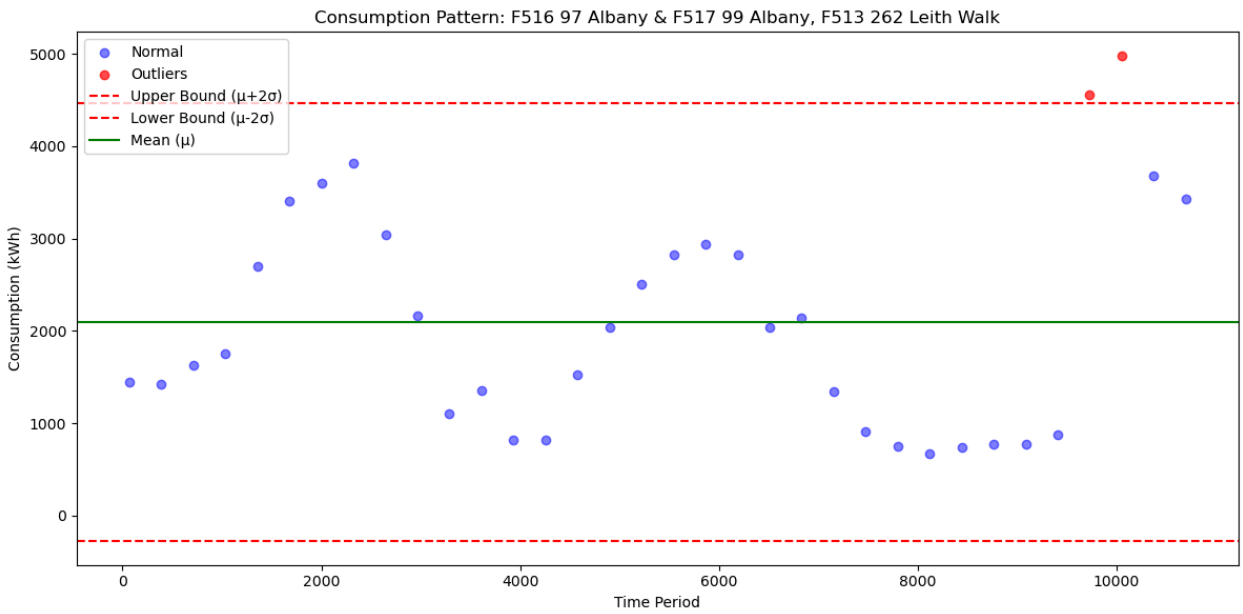

```
Statistics for F512 270 Leith Walk:
Mean: 10000.14
Standard Deviation: 1445.59
Upper Bound: 12891.31
Lower Bound: 7108.97
Number of outliers: 0
```



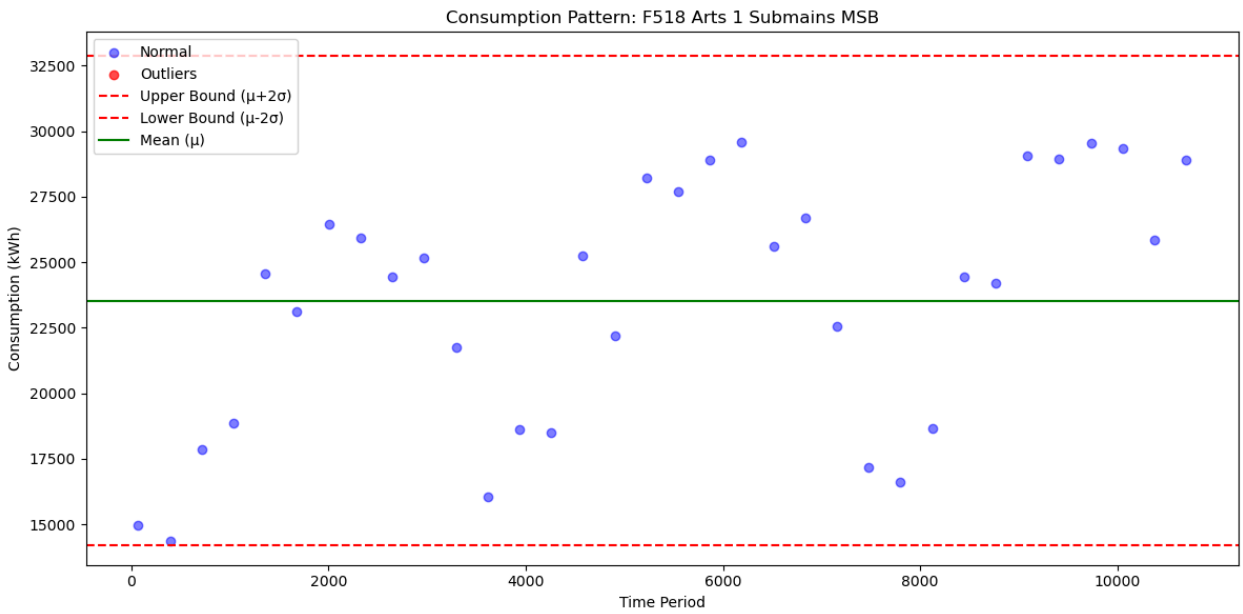

```
Statistics for F512 Substation 270 Leith Walk:
Mean: 10027.17
Standard Deviation: 1459.60
Upper Bound: 12946.37
Lower Bound: 7107.97
Number of outliers: 0
```



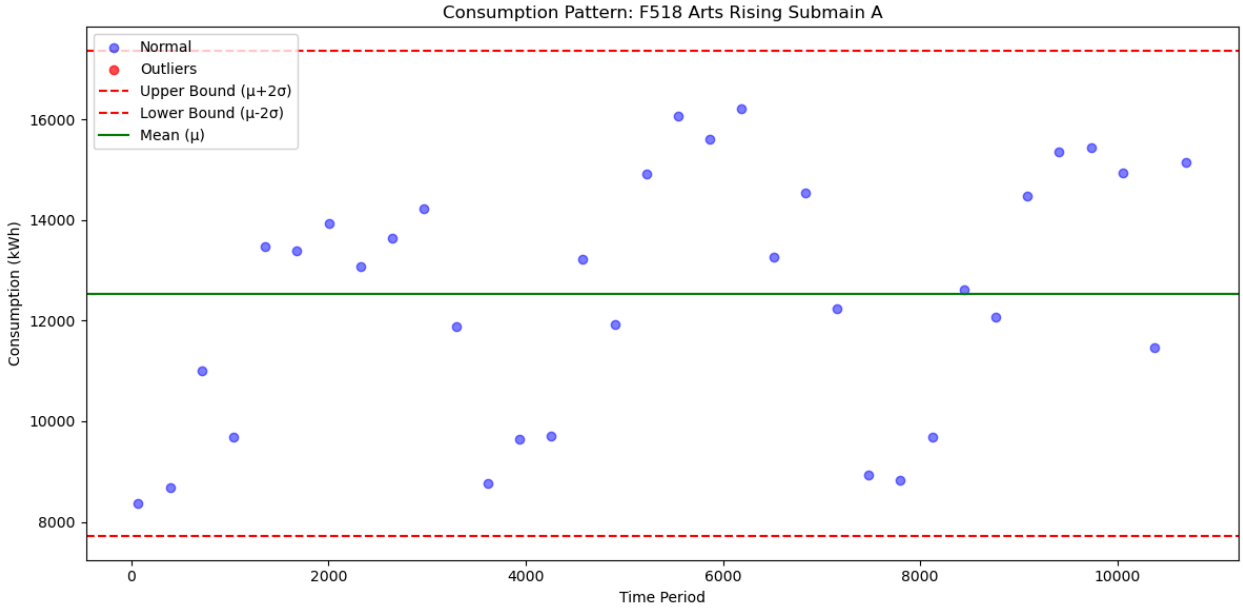

```
Statistics for F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:
Mean: 2099.60
Standard Deviation: 1184.93
Upper Bound: 4469.46
Lower Bound: -270.26
Number of outliers: 2
```



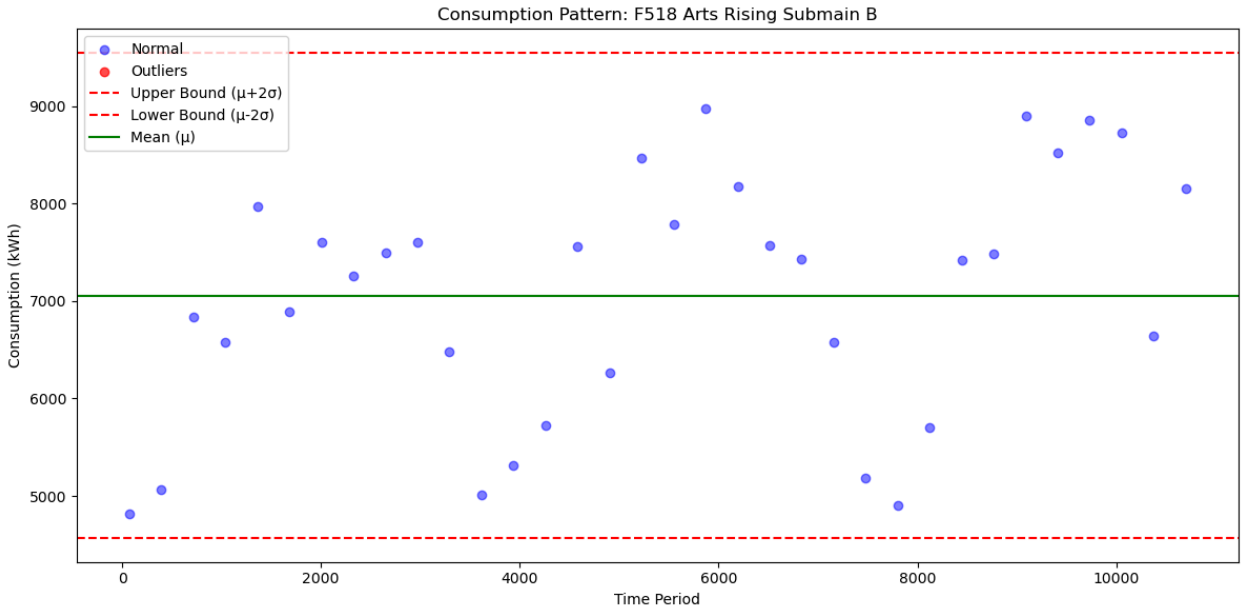

```
Statistics for F518 Arts 1 Submains MSB:
Mean: 23527.79
Standard Deviation: 4667.60
Upper Bound: 32862.99
Lower Bound: 14192.59
Number of outliers: 0
```



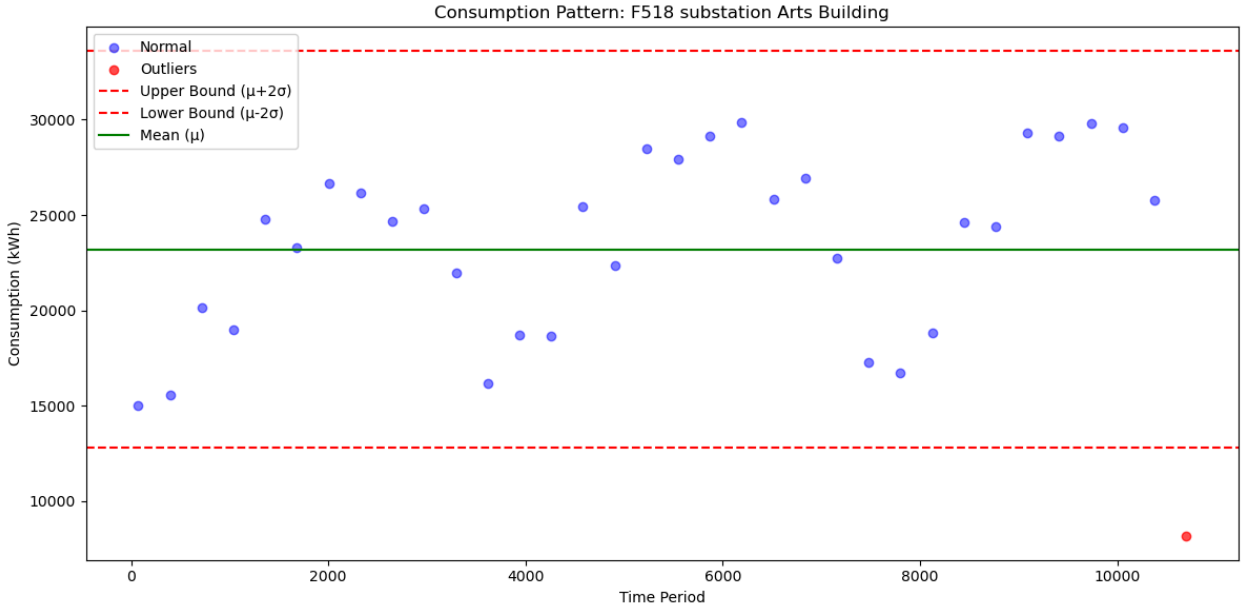

```
Statistics for F518 Arts Rising Submain A:
Mean: 12540.57
Standard Deviation: 2408.01
Upper Bound: 17356.58
Lower Bound: 7724.56
Number of outliers: 0
```



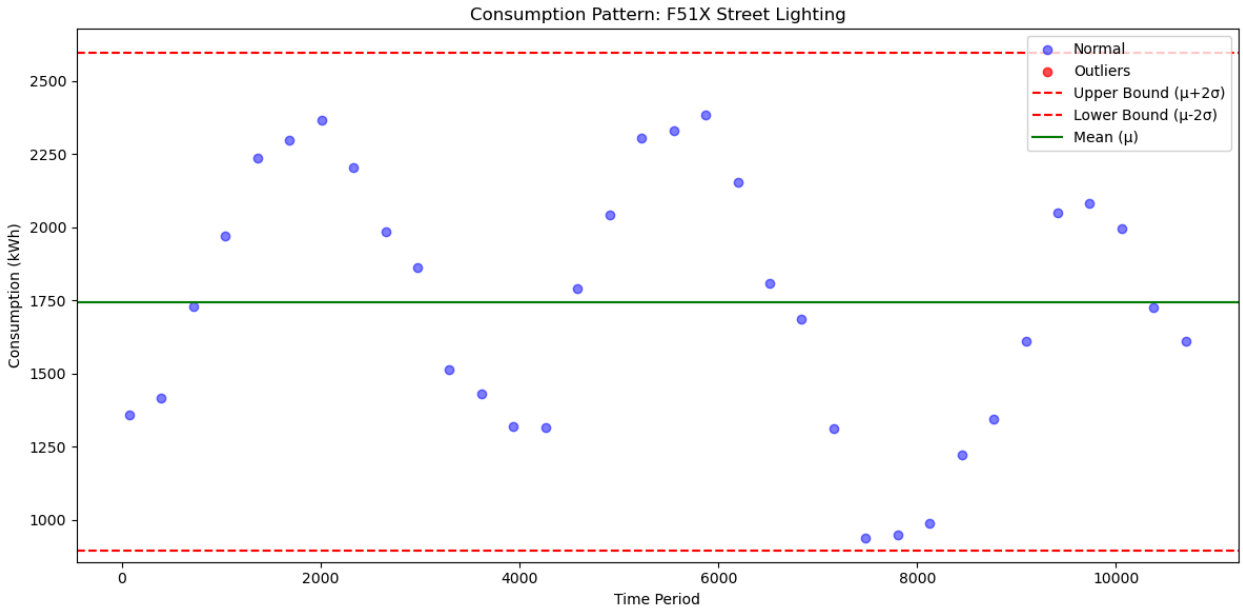

```
Statistics for F518 Arts Rising Submain B:
Mean: 7056.82
Standard Deviation: 1242.27
Upper Bound: 9541.35
Lower Bound: 4572.29
Number of outliers: 0
```



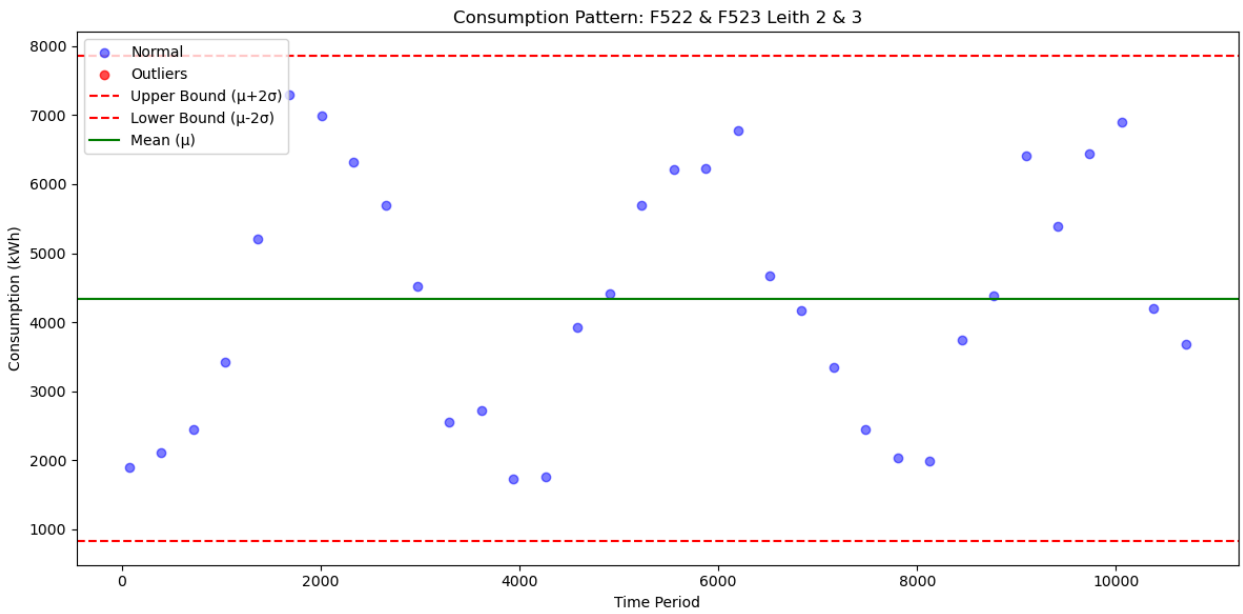

```
Statistics for F518 substation Arts Building:
Mean: 23189.45
Standard Deviation: 5203.27
Upper Bound: 33595.99
Lower Bound: 12782.90
Number of outliers: 1
```



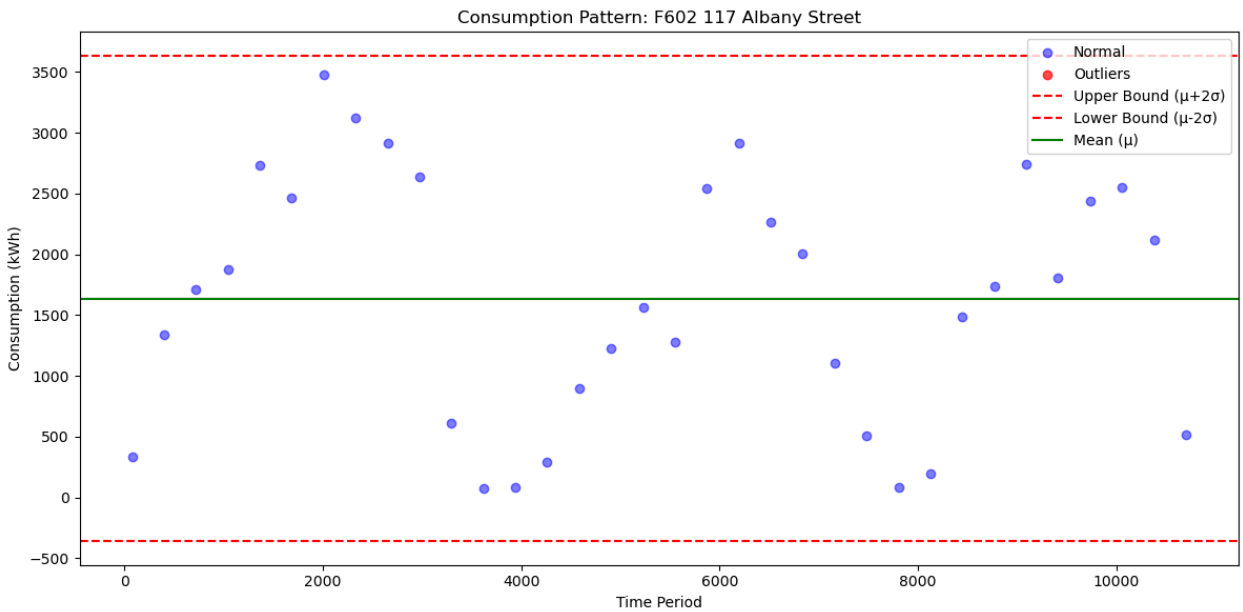

```
Statistics for F51X Street Lighting:
Mean: 1745.08
Standard Deviation: 425.11
Upper Bound: 2595.30
Lower Bound: 894.86
Number of outliers: 0
```




```
Statistics for F522 & F523 Leith 2 & 3:
Mean: 4344.05
Standard Deviation: 1754.52
Upper Bound: 7853.10
Lower Bound: 835.01
Number of outliers: 0
```



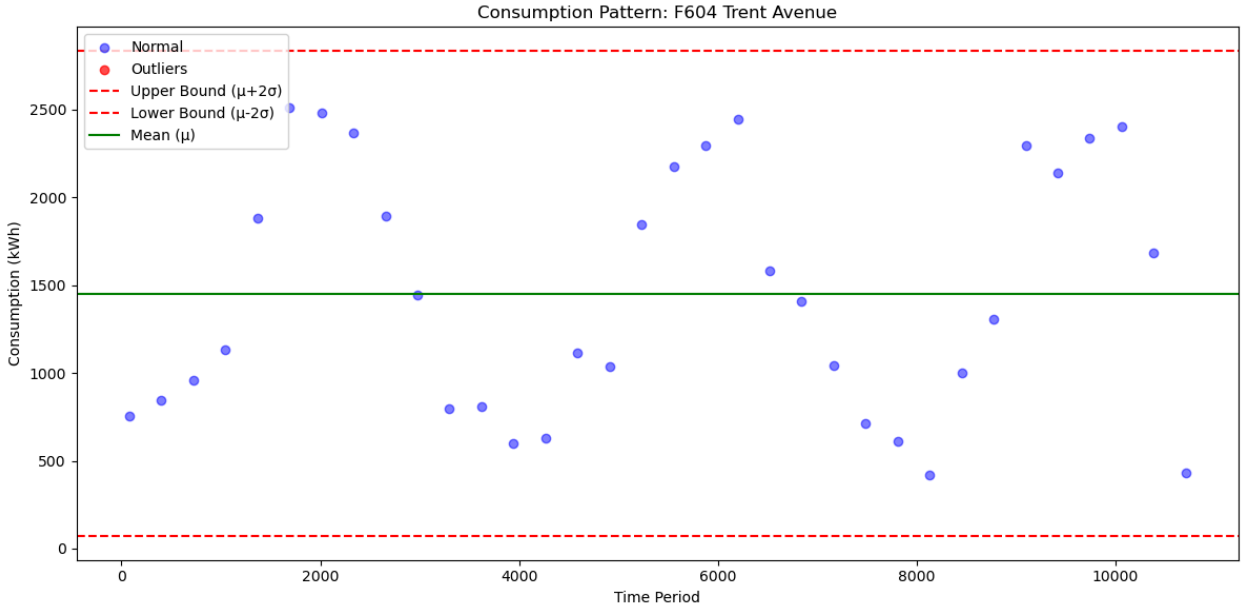

```
Statistics for F602 117 Albany Street:
Mean: 1636.74
Standard Deviation: 996.57
Upper Bound: 3629.88
Lower Bound: -356.39
Number of outliers: 0
```



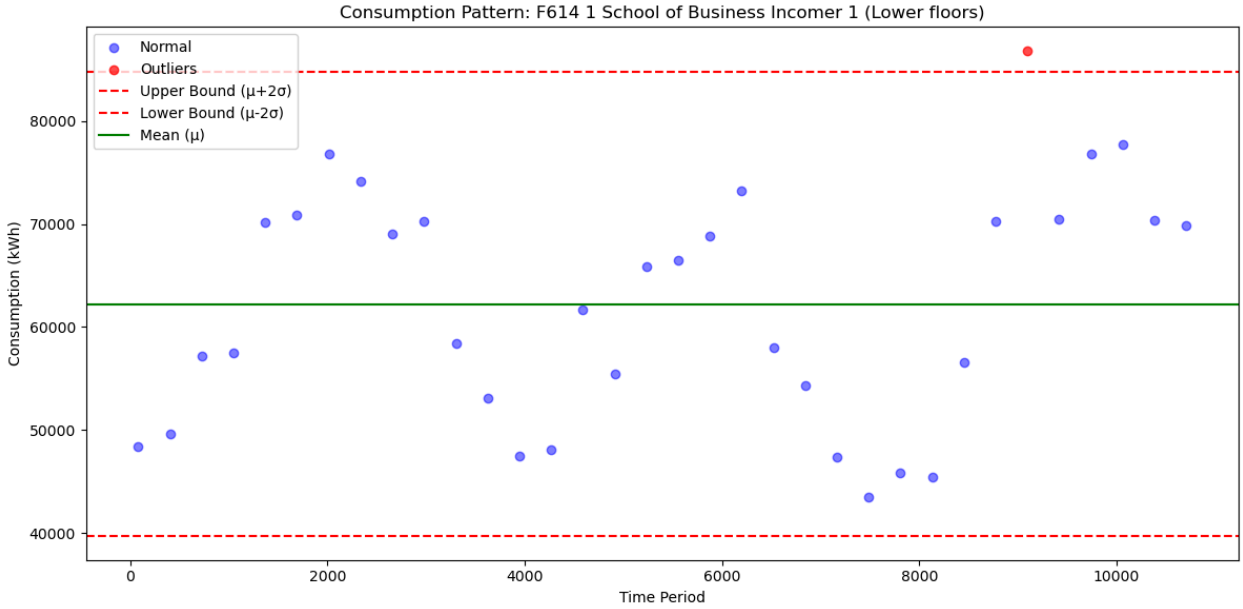

```
Statistics for F603 Property Services Building:
Mean: 15028.24
Standard Deviation: 3777.54
Upper Bound: 22583.33
Lower Bound: 7473.16
Number of outliers: 0
```

```
Statistics for F603 substation Property Services meter:
Mean: 15103.23
Standard Deviation: 3685.35
Upper Bound: 22473.93
Lower Bound: 7732.53
Number of outliers: 0
```

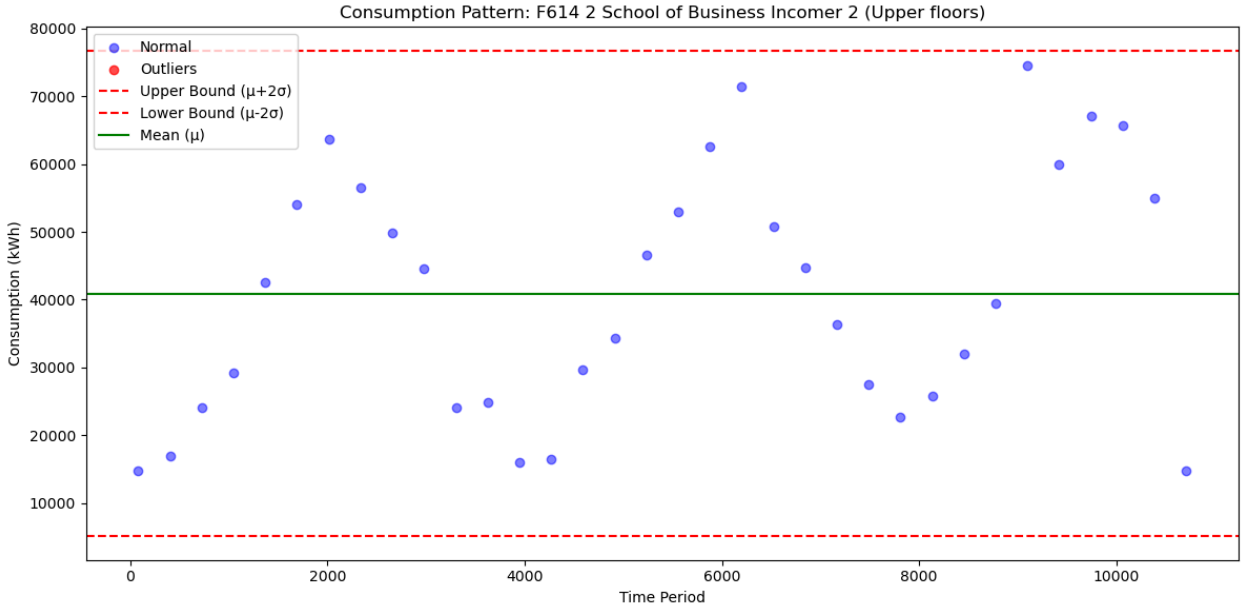
```
Statistics for F604 Trent Avenue:
Mean: 1453.36
Standard Deviation: 689.94
Upper Bound: 2833.25
Lower Bound: 73.47
Number of outliers: 0
```



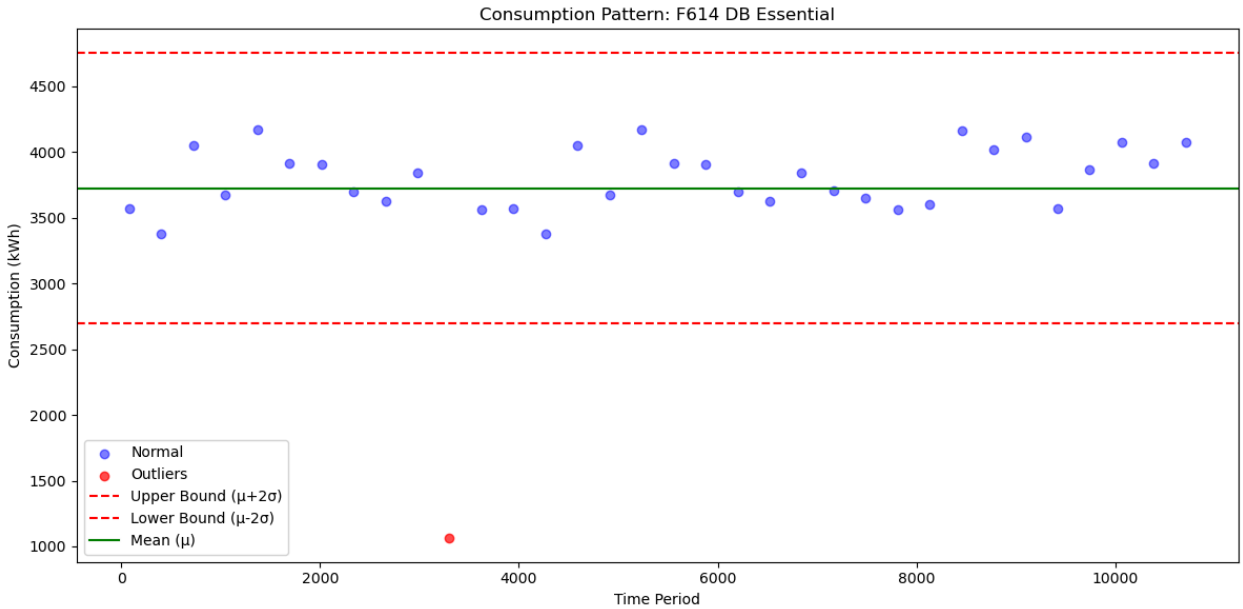

```
Statistics for F614 1 School of Business Incomer 1 (Lower floors):
Mean: 62236.19
Standard Deviation: 11251.36
Upper Bound: 84738.91
Lower Bound: 39733.47
Number of outliers: 1
```



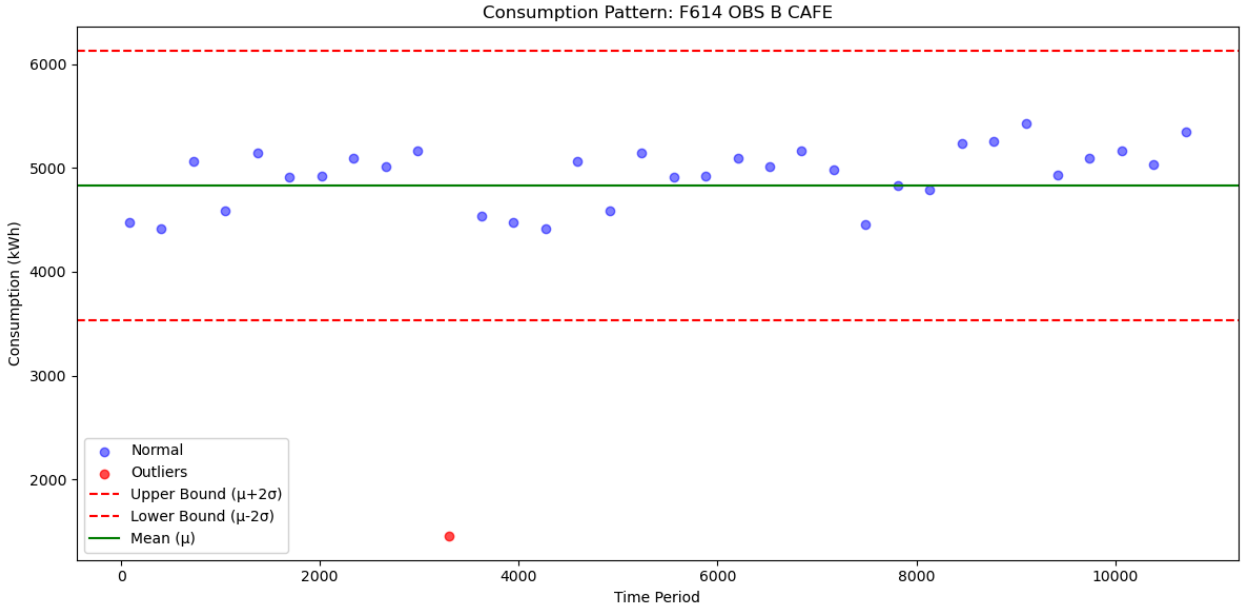

```
Statistics for F614 2 School of Business Incomer 2 (Upper floors):
Mean: 40916.12
Standard Deviation: 17871.05
Upper Bound: 76658.22
Lower Bound: 5174.02
Number of outliers: 0
```



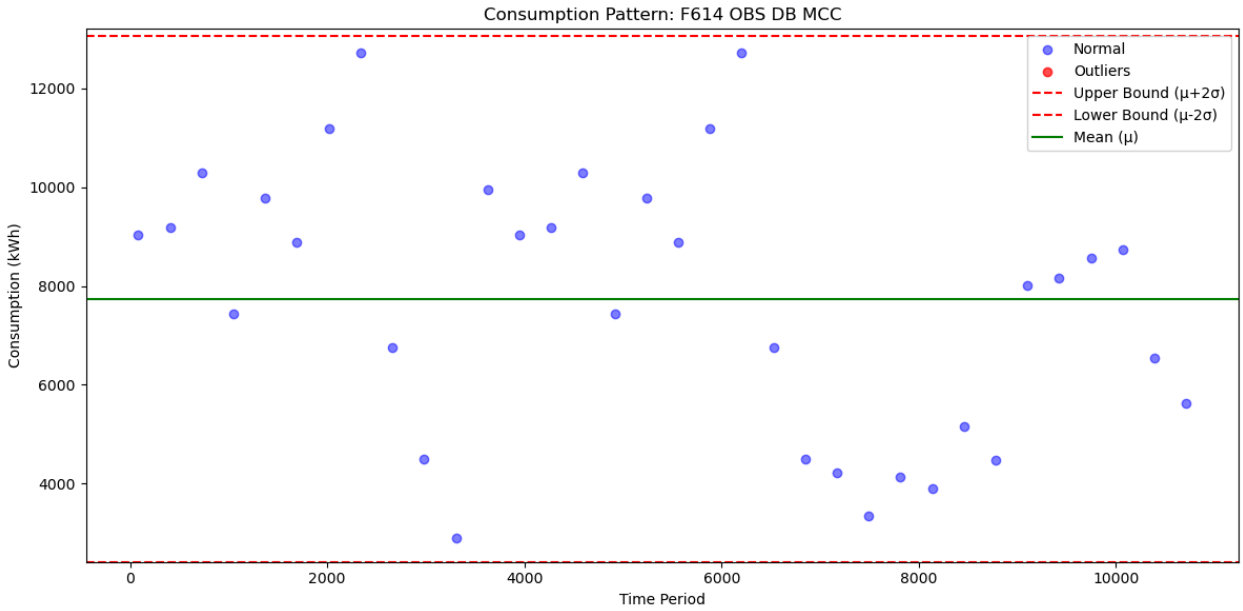

```
Statistics for F614 DB Essential:
Mean: 3724.54
Standard Deviation: 514.48
Upper Bound: 4753.51
Lower Bound: 2695.57
Number of outliers: 1
```



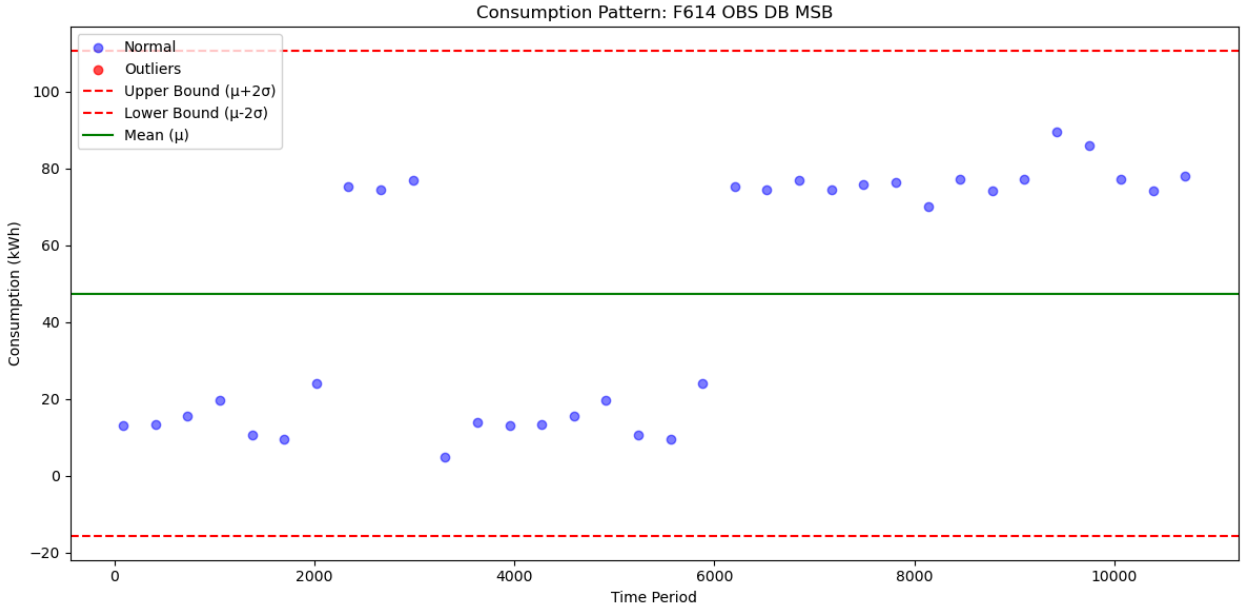

```
Statistics for F614 OBS B CAFE:
Mean: 4826.89
Standard Deviation: 648.21
Upper Bound: 6123.32
Lower Bound: 3530.47
Number of outliers: 1
```



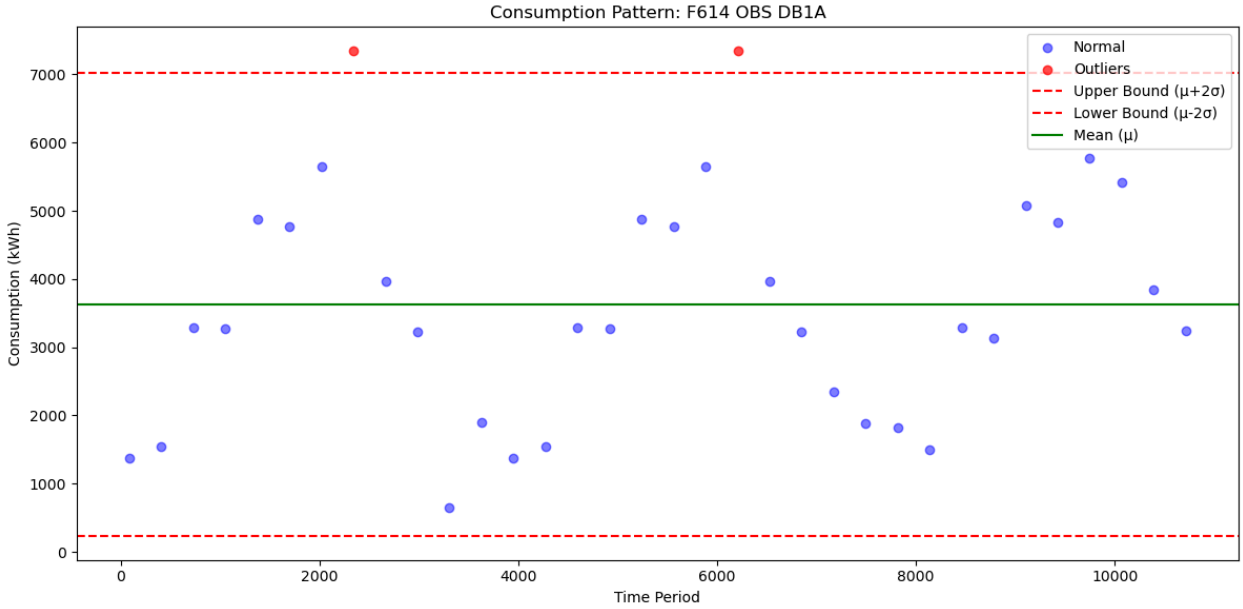

```
Statistics for F614 OBS DB MCC:
Mean: 7742.62
Standard Deviation: 2663.13
Upper Bound: 13068.89
Lower Bound: 2416.36
Number of outliers: 0
```



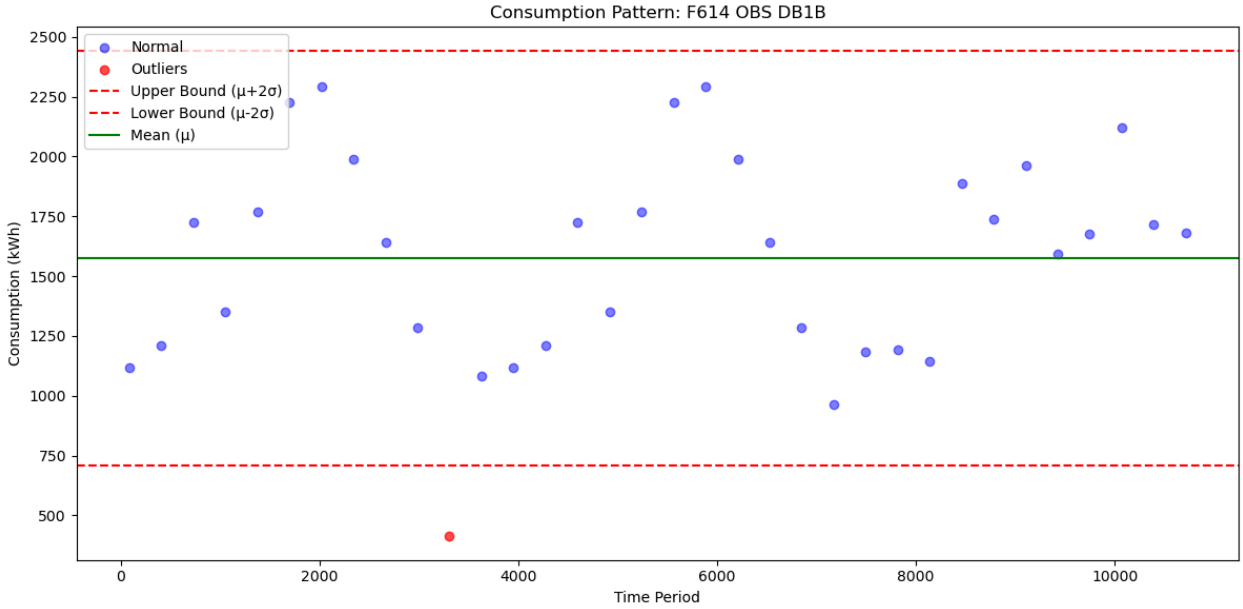

```
Statistics for F614 OBS DB MSB:
Mean: 47.49
Standard Deviation: 31.53
Upper Bound: 110.55
Lower Bound: -15.56
Number of outliers: 0
```



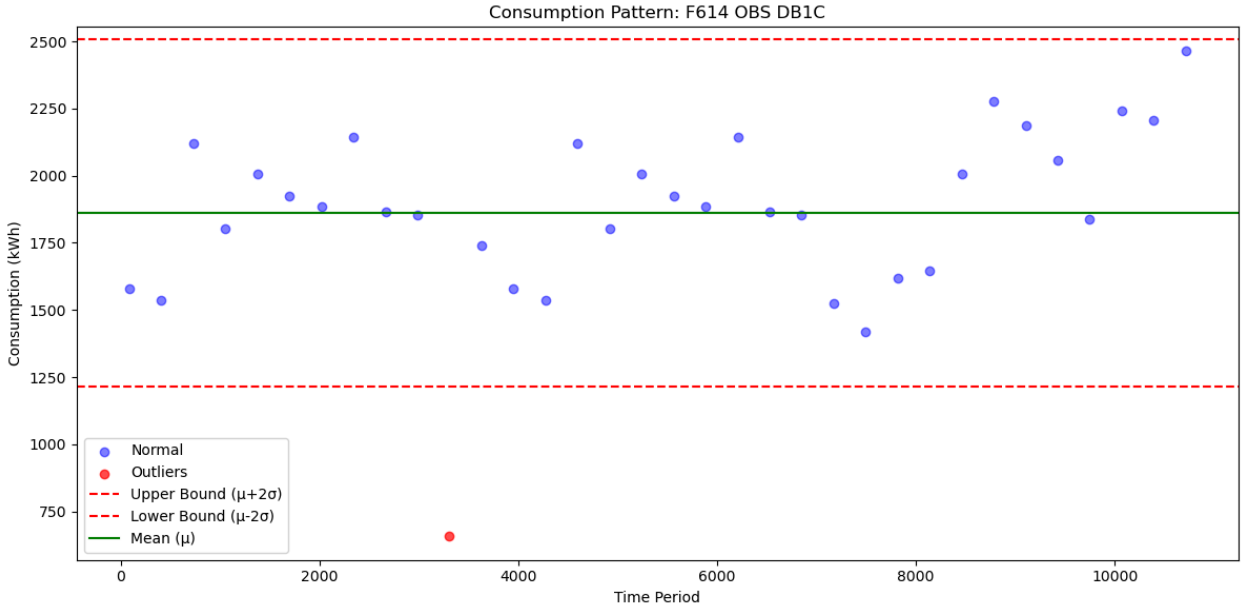

```
Statistics for F614 OBS DB1A:
Mean: 3625.97
Standard Deviation: 1694.36
Upper Bound: 7014.69
Lower Bound: 237.25
Number of outliers: 2
```



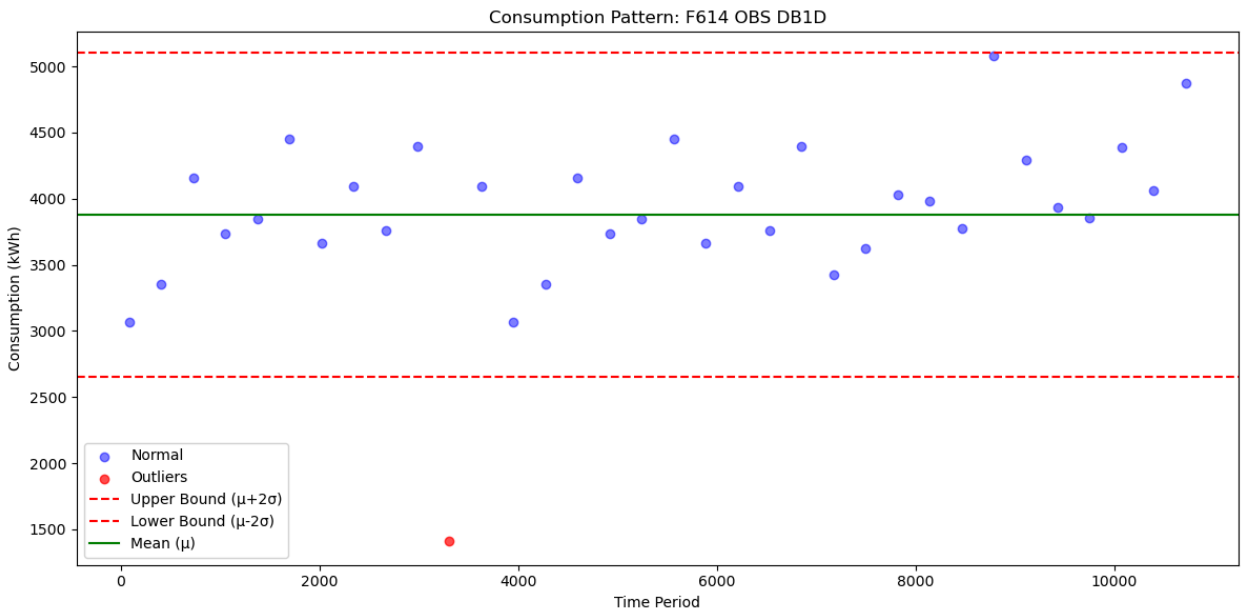

```
Statistics for F614 OBS DB1B:
Mean: 1575.58
Standard Deviation: 432.35
Upper Bound: 2440.28
Lower Bound: 710.89
Number of outliers: 1
```



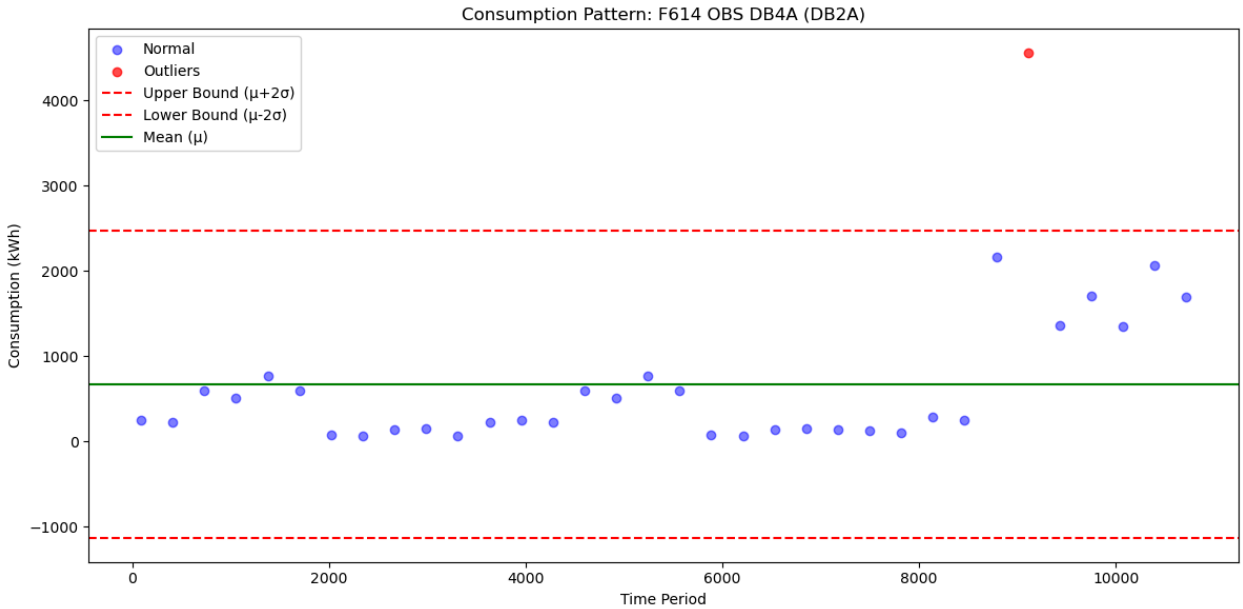

```
Statistics for F614 OBS DB1C:
Mean: 1862.30
Standard Deviation: 323.45
Upper Bound: 2509.20
Lower Bound: 1215.39
Number of outliers: 1
```



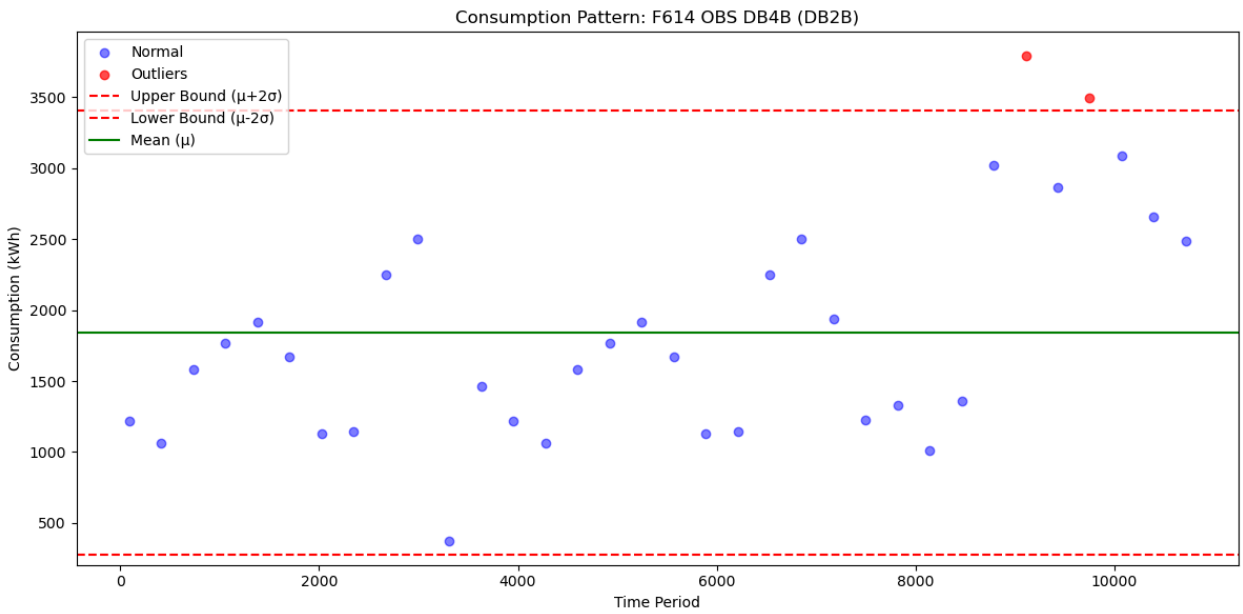

```
Statistics for F614 OBS DB1D:
Mean: 3878.40
Standard Deviation: 614.11
Upper Bound: 5106.62
Lower Bound: 2650.18
Number of outliers: 1
```



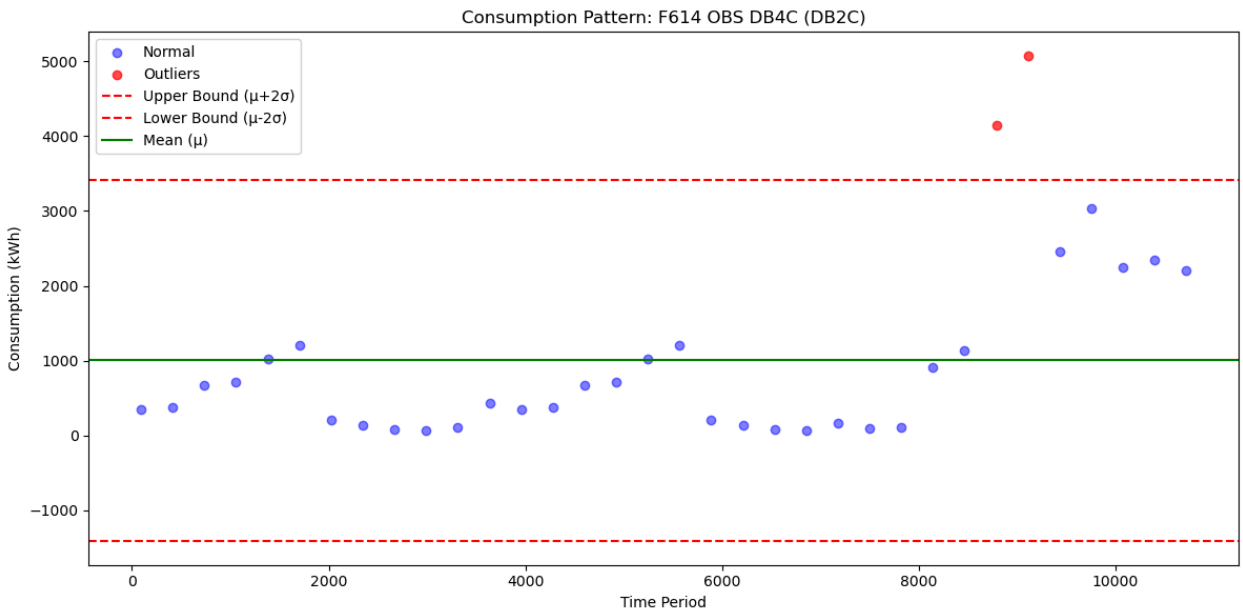

```
Statistics for F614 OBS DB4A (DB2A):
Mean: 669.94
Standard Deviation: 900.28
Upper Bound: 2470.50
Lower Bound: -1130.61
Number of outliers: 1
```



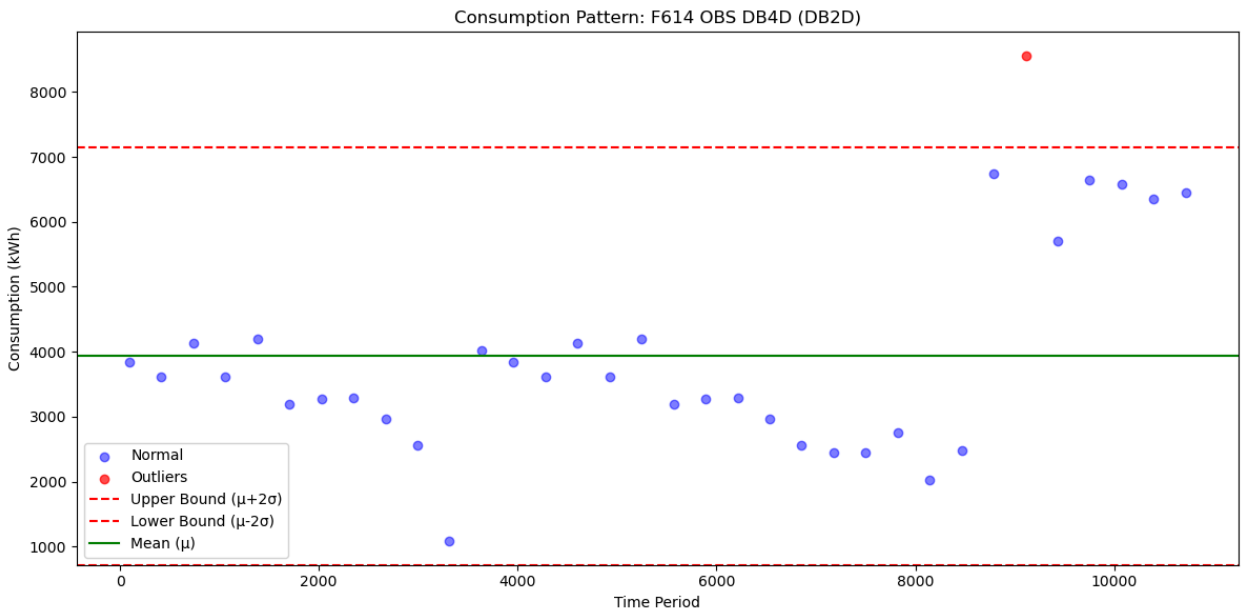

```
Statistics for F614 OBS DB4B (DB2B):
Mean: 1840.98
Standard Deviation: 783.53
Upper Bound: 3408.03
Lower Bound: 273.92
Number of outliers: 2
```



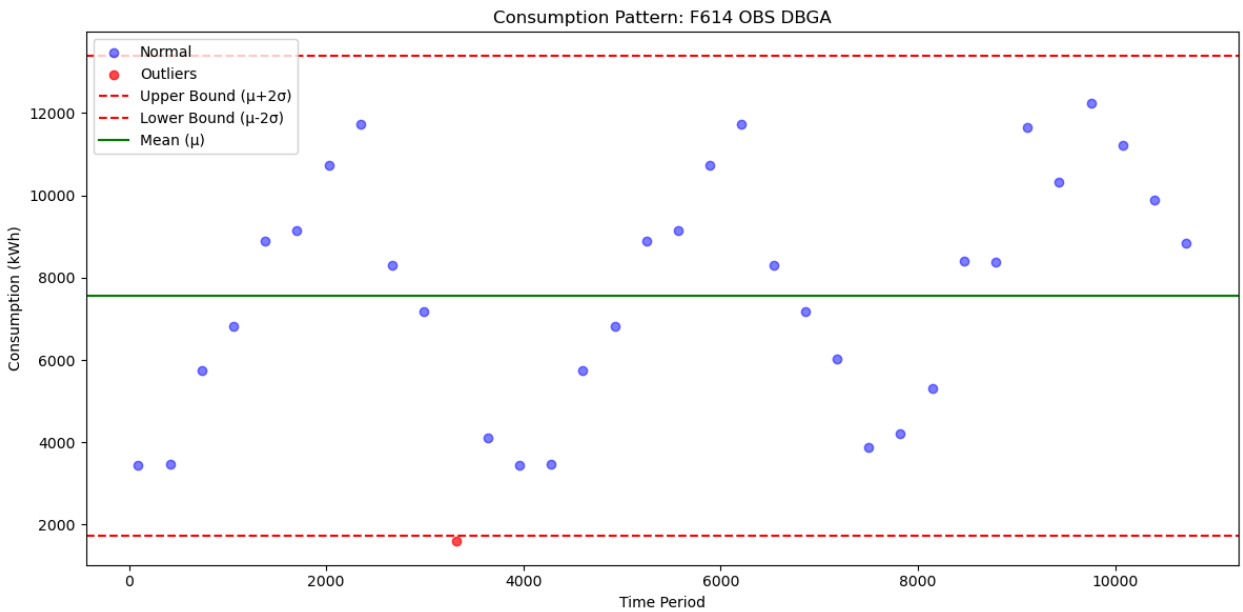

```
Statistics for F614 OBS DB4C (DB2C):
Mean: 1002.00
Standard Deviation: 1204.39
Upper Bound: 3410.78
Lower Bound: -1406.78
Number of outliers: 2
```



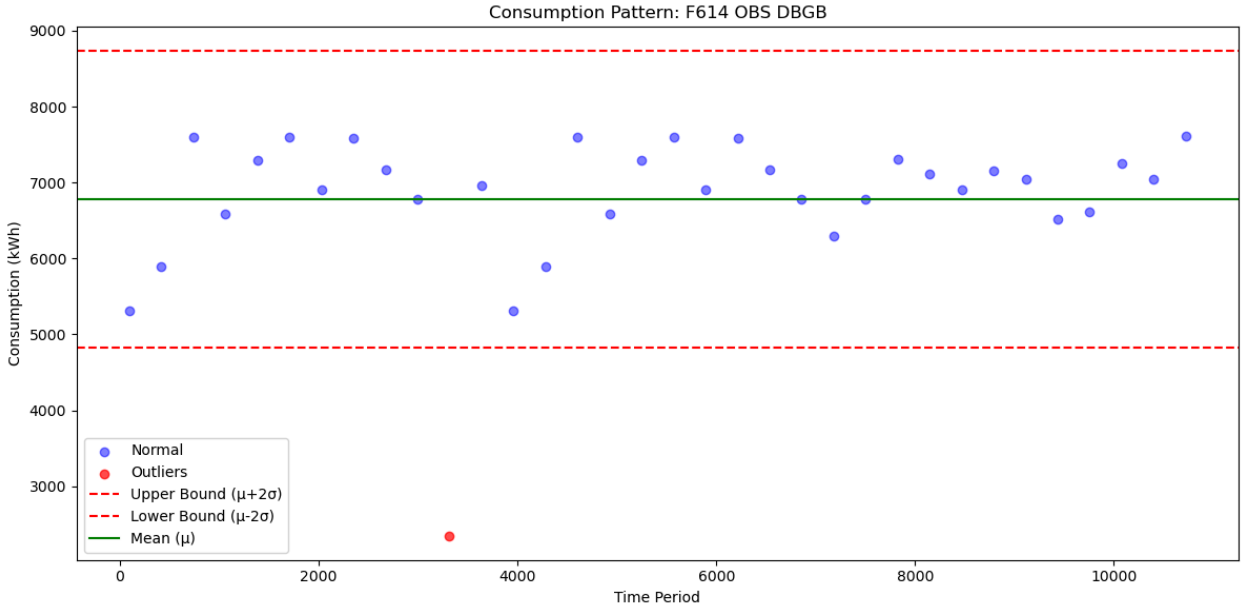

```
Statistics for F614 OBS DB4D (DB2D):
Mean: 3931.82
Standard Deviation: 1606.63
Upper Bound: 7145.08
Lower Bound: 718.56
Number of outliers: 1
```



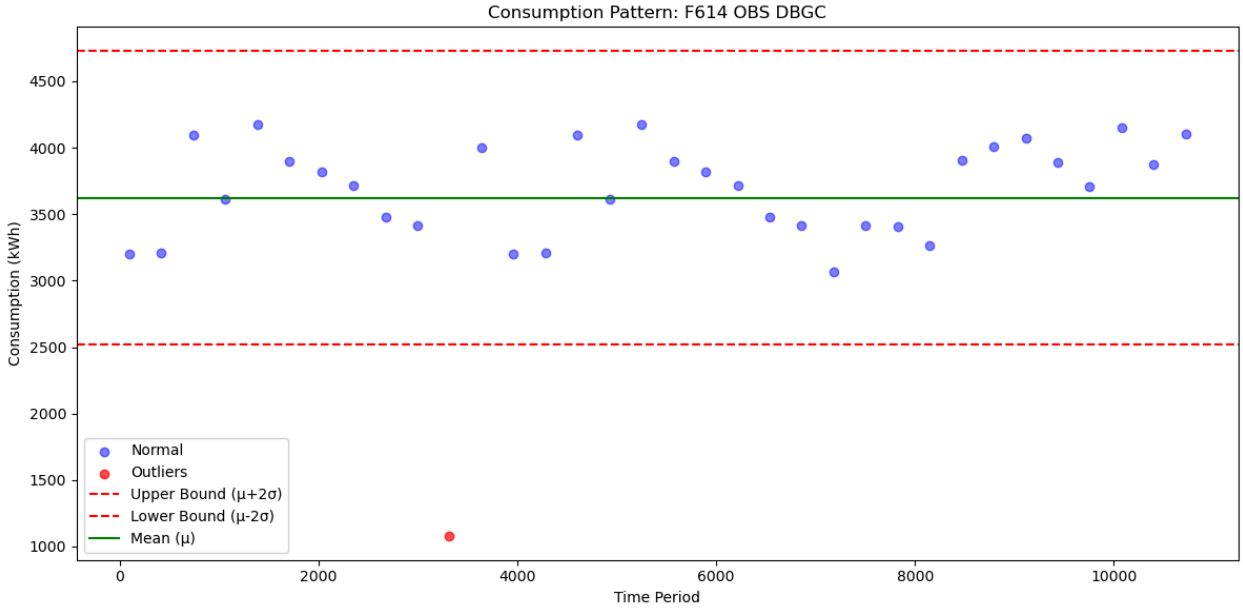

```
Statistics for F614 OBS DBGA:
Mean: 7560.86
Standard Deviation: 2911.14
Upper Bound: 13383.13
Lower Bound: 1738.59
Number of outliers: 1
```



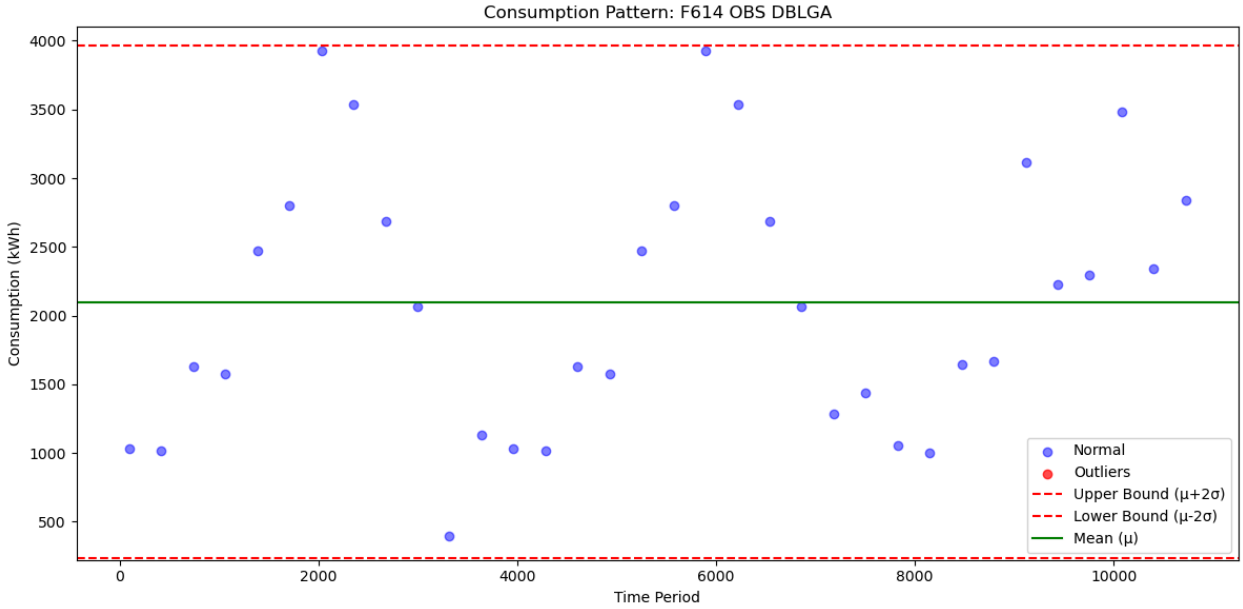

```
Statistics for F614 OBS DBGB:
Mean: 6776.19
Standard Deviation: 976.84
Upper Bound: 8729.88
Lower Bound: 4822.51
Number of outliers: 1
```



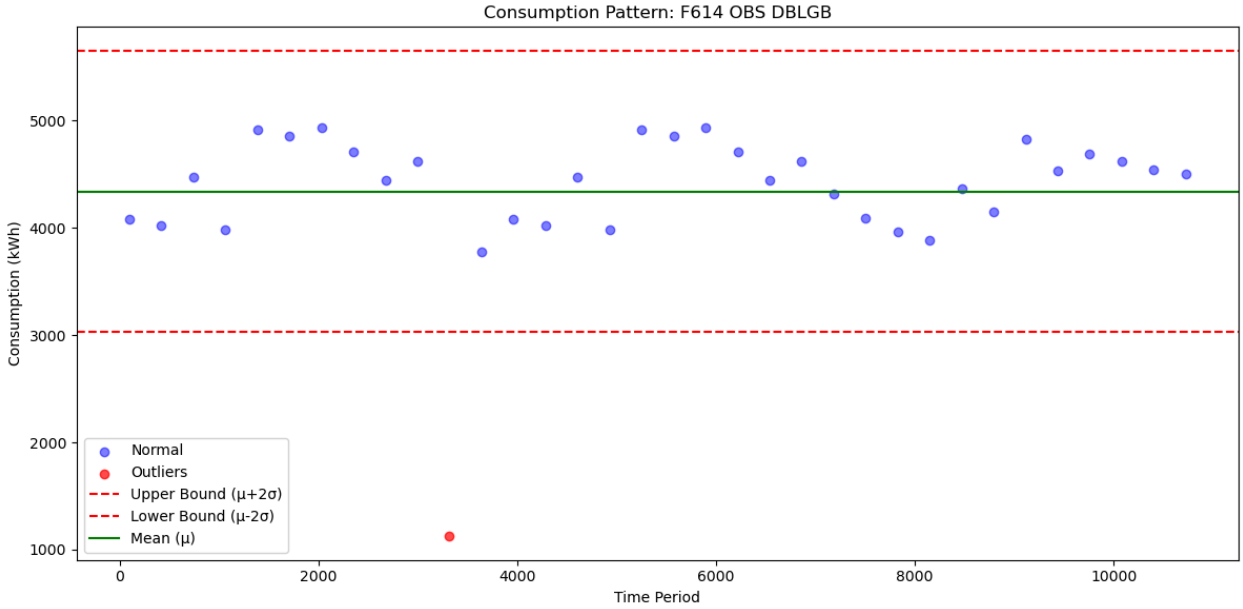

```
Statistics for F614 OBS DBGC:
Mean: 3623.00
Standard Deviation: 551.47
Upper Bound: 4725.94
Lower Bound: 2520.07
Number of outliers: 1
```



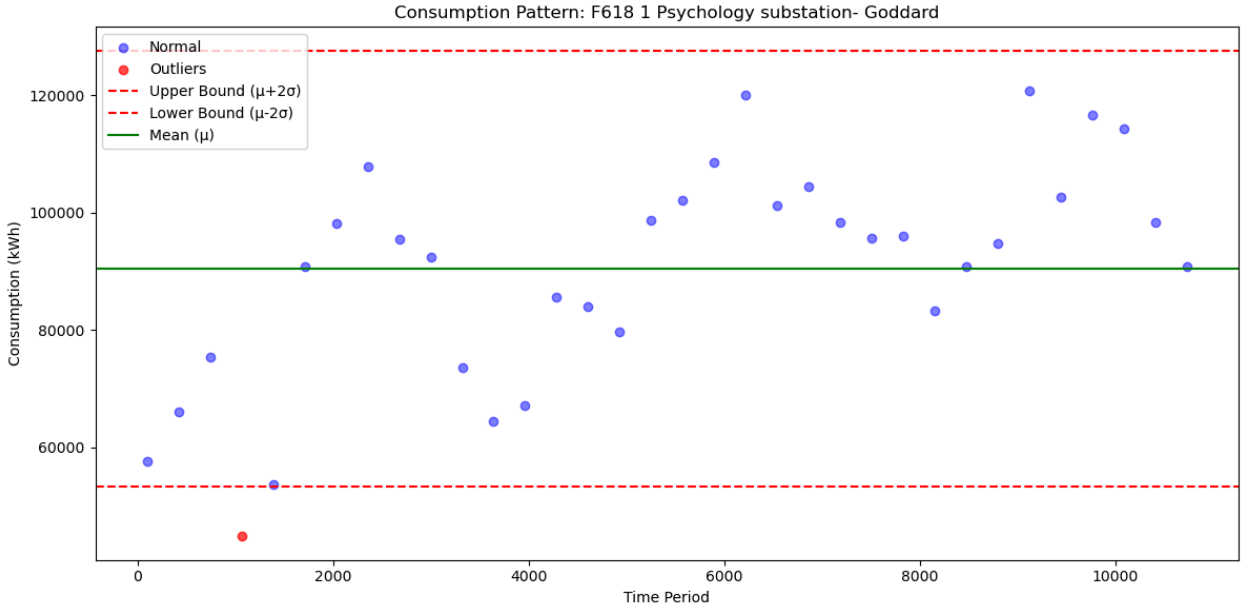

```
Statistics for F614 OBS DBLGA:
Mean: 2099.61
Standard Deviation: 933.54
Upper Bound: 3966.69
Lower Bound: 232.53
Number of outliers: 0
```



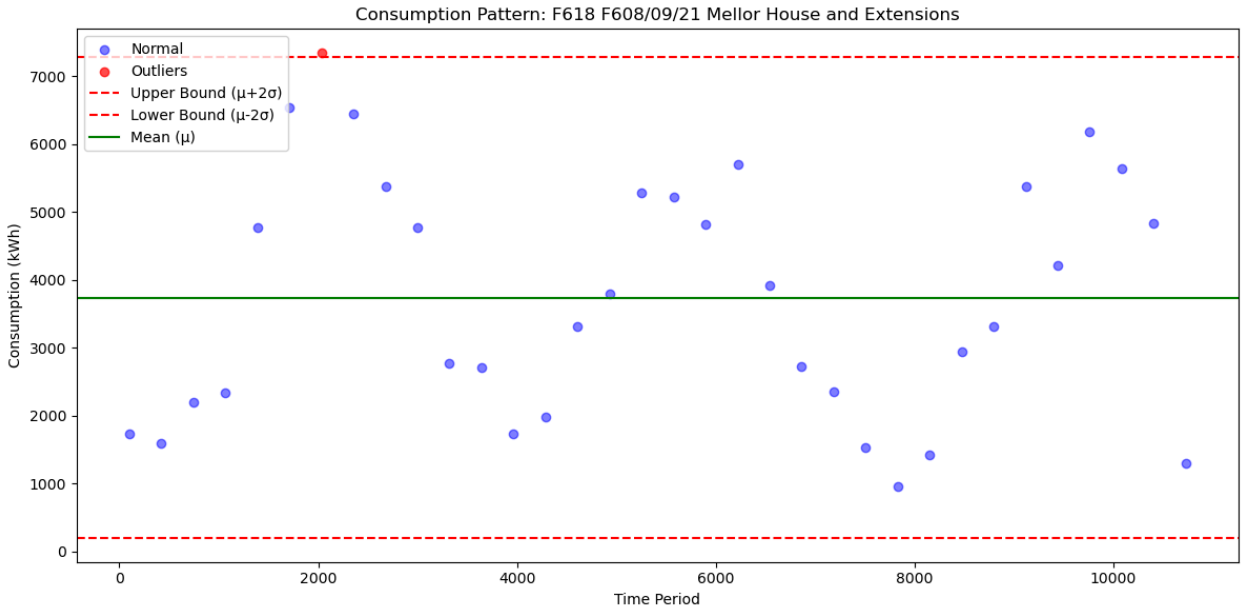

```
Statistics for F614 OBS DBLGB:
Mean: 4337.59
Standard Deviation: 655.13
Upper Bound: 5647.85
Lower Bound: 3027.33
Number of outliers: 1
```



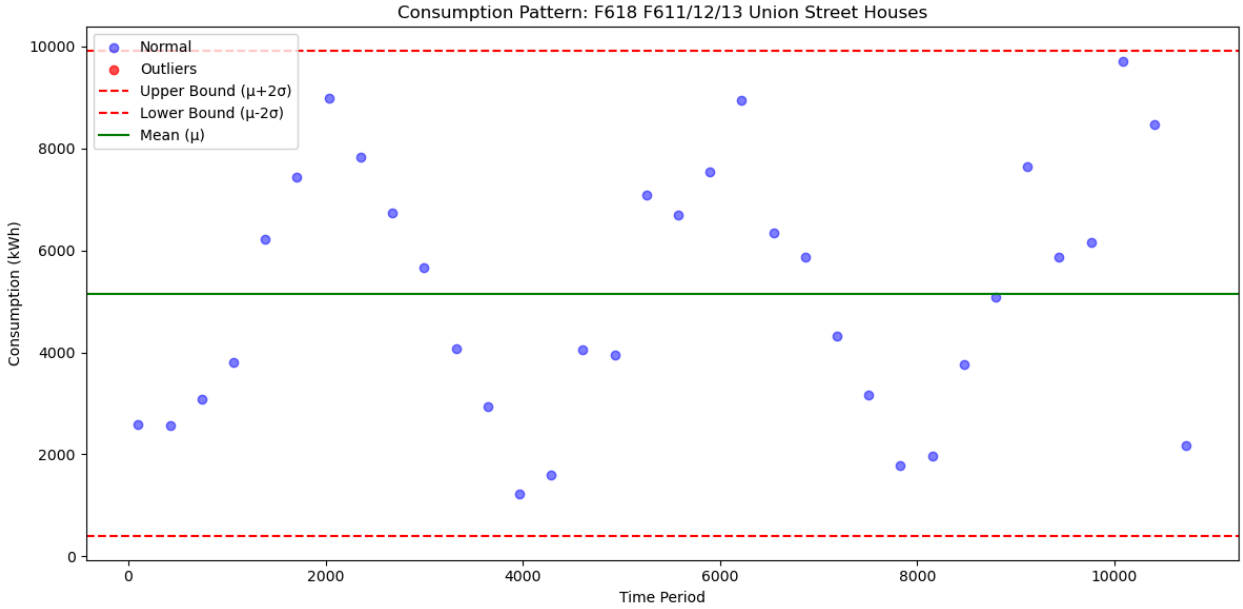

```
Statistics for F618 1 Psychology substation- Goddard:
Mean: 90395.38
Standard Deviation: 18559.75
Upper Bound: 127514.88
Lower Bound: 53275.87
Number of outliers: 1
```



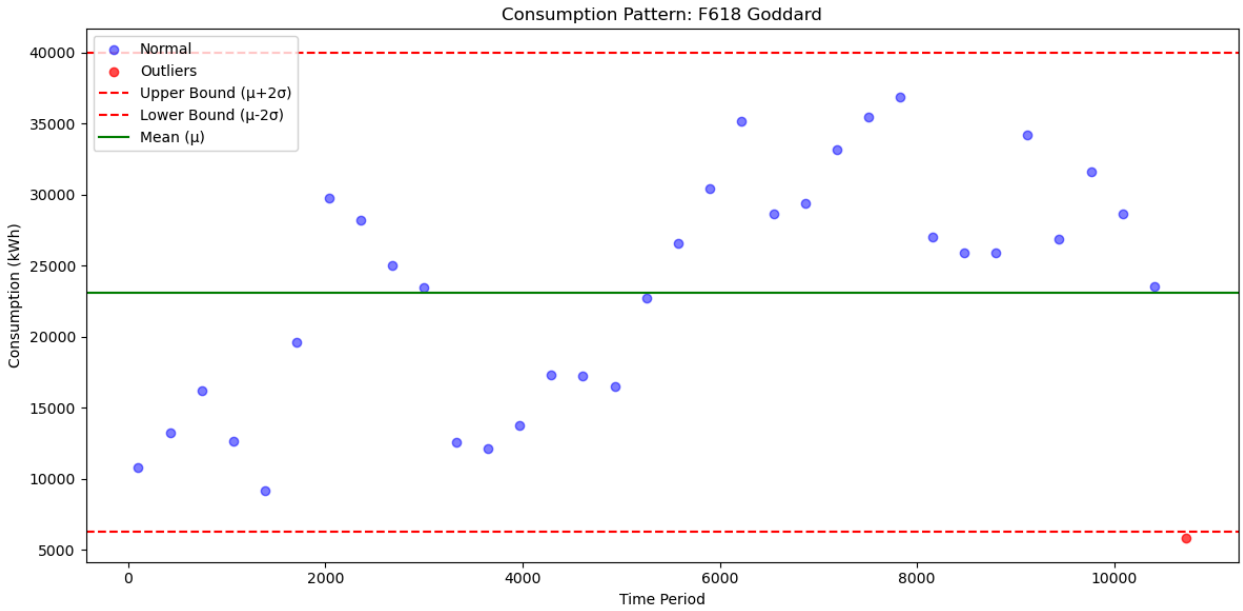

```
Statistics for F618 F608/09/21 Mellor House and Extensions:
Mean: 3737.12
Standard Deviation: 1767.85
Upper Bound: 7272.82
Lower Bound: 201.41
Number of outliers: 1
```



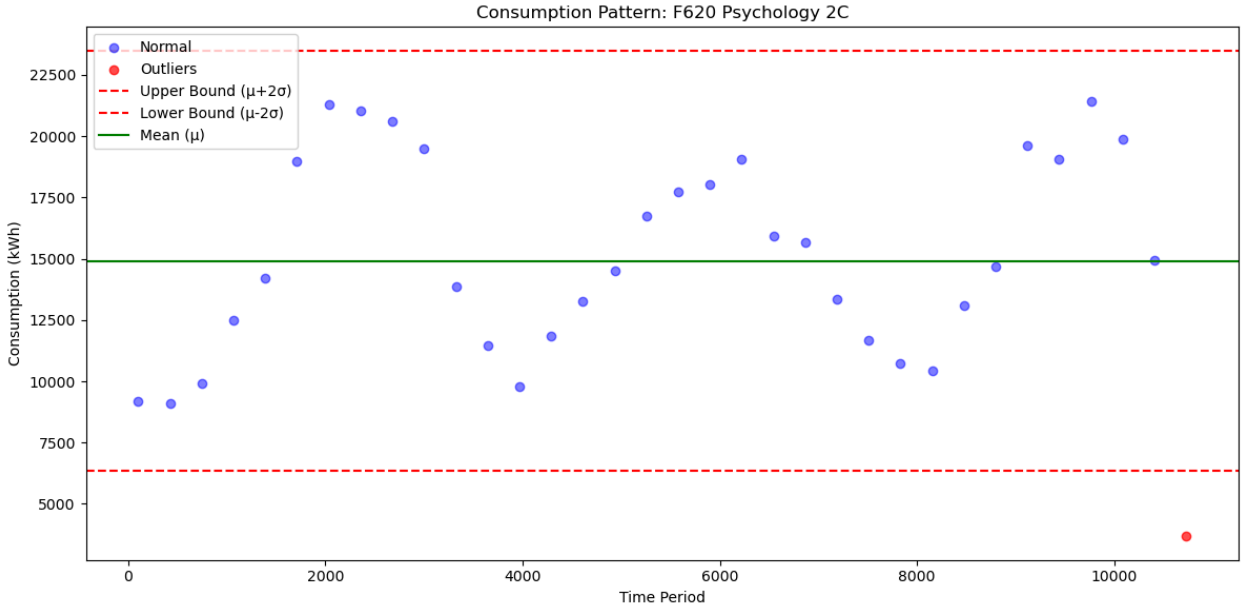

```
Statistics for F618 F611/12/13 Union Street Houses:
Mean: 5154.57
Standard Deviation: 2373.82
Upper Bound: 9902.22
Lower Bound: 406.93
Number of outliers: 0
```



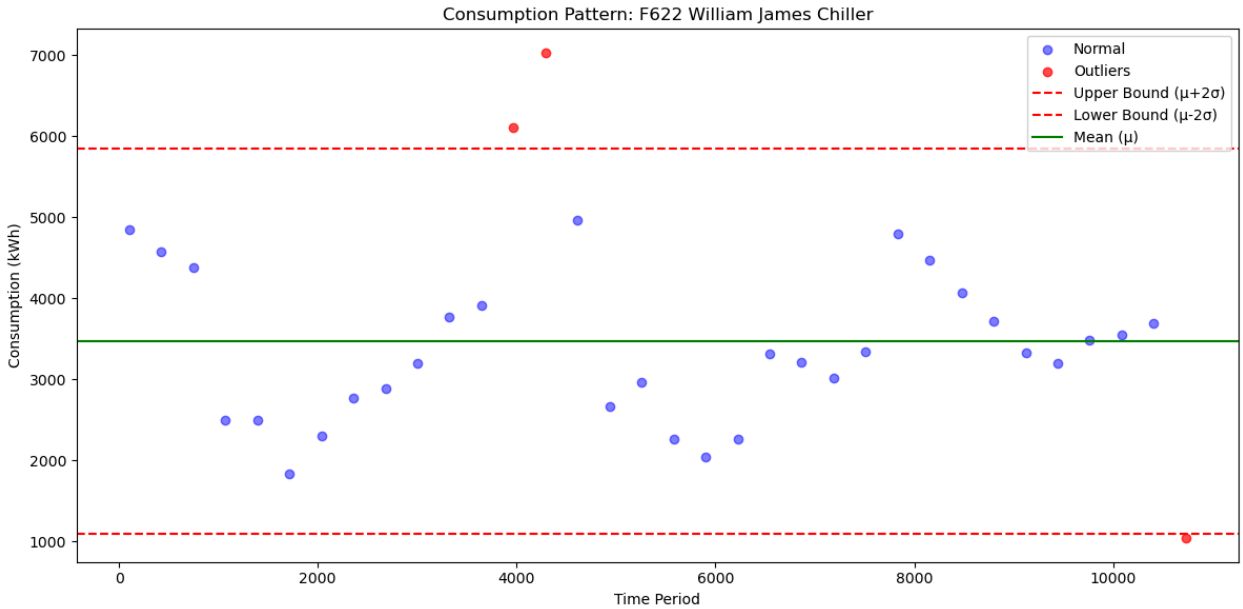

```
Statistics for F618 Goddard:
Mean: 23107.29
Standard Deviation: 8422.19
Upper Bound: 39951.67
Lower Bound: 6262.92
Number of outliers: 1
```



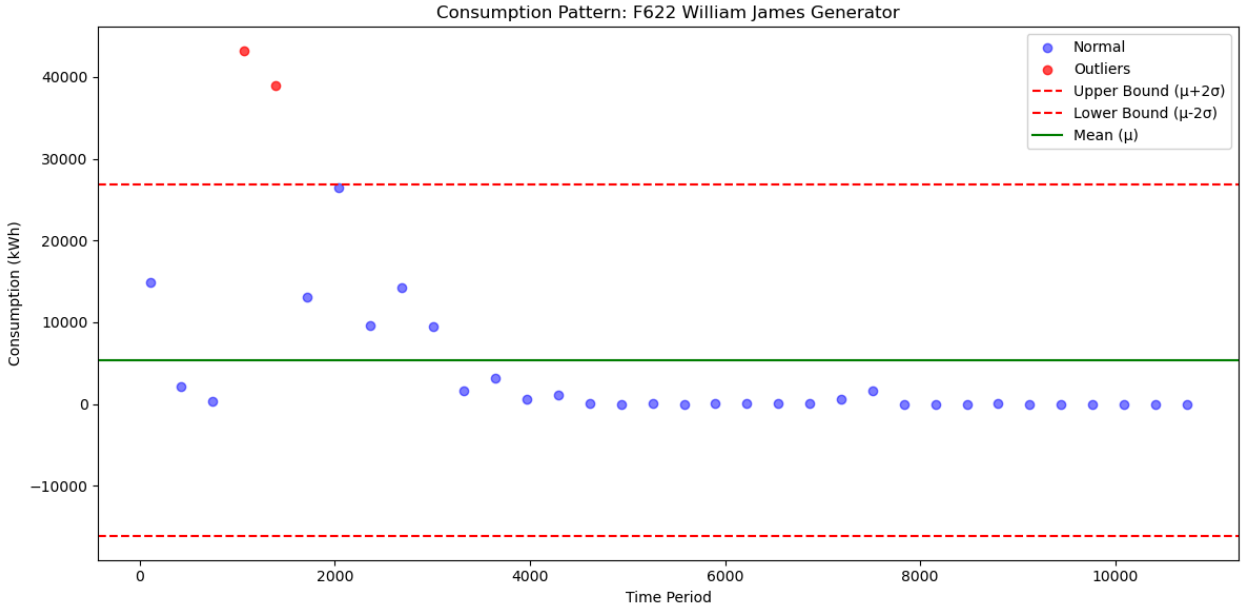

```
Statistics for F620 Psychology 2C:
Mean: 14905.00
Standard Deviation: 4283.90
Upper Bound: 23472.81
Lower Bound: 6337.20
Number of outliers: 1
```



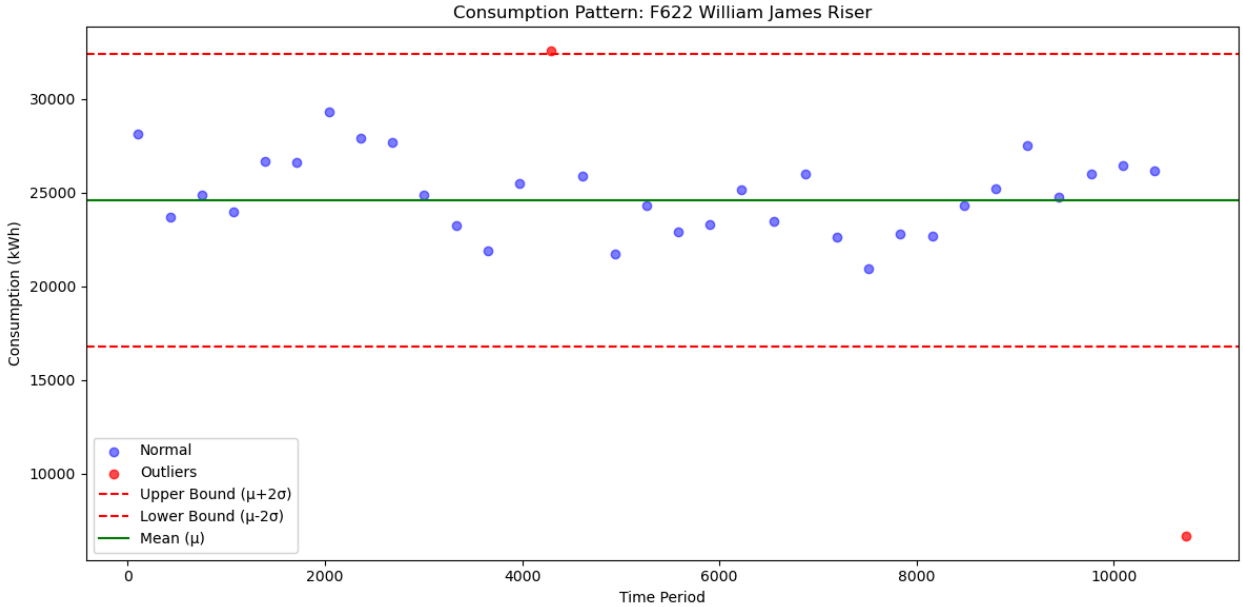

```
Statistics for F622 William James Chiller:
Mean: 3469.32
Standard Deviation: 1190.77
Upper Bound: 5850.86
Lower Bound: 1087.78
Number of outliers: 3
```



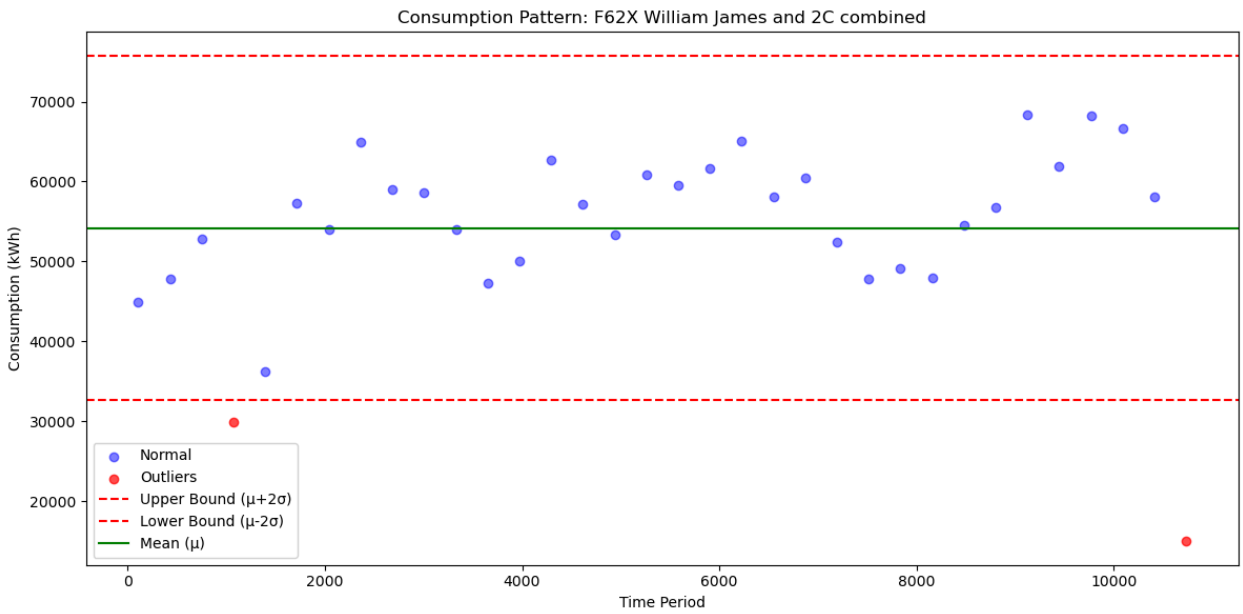

```
Statistics for F622 William James Generator:
Mean: 5341.77
Standard Deviation: 10724.23
Upper Bound: 26790.24
Lower Bound: -16106.70
Number of outliers: 2
```



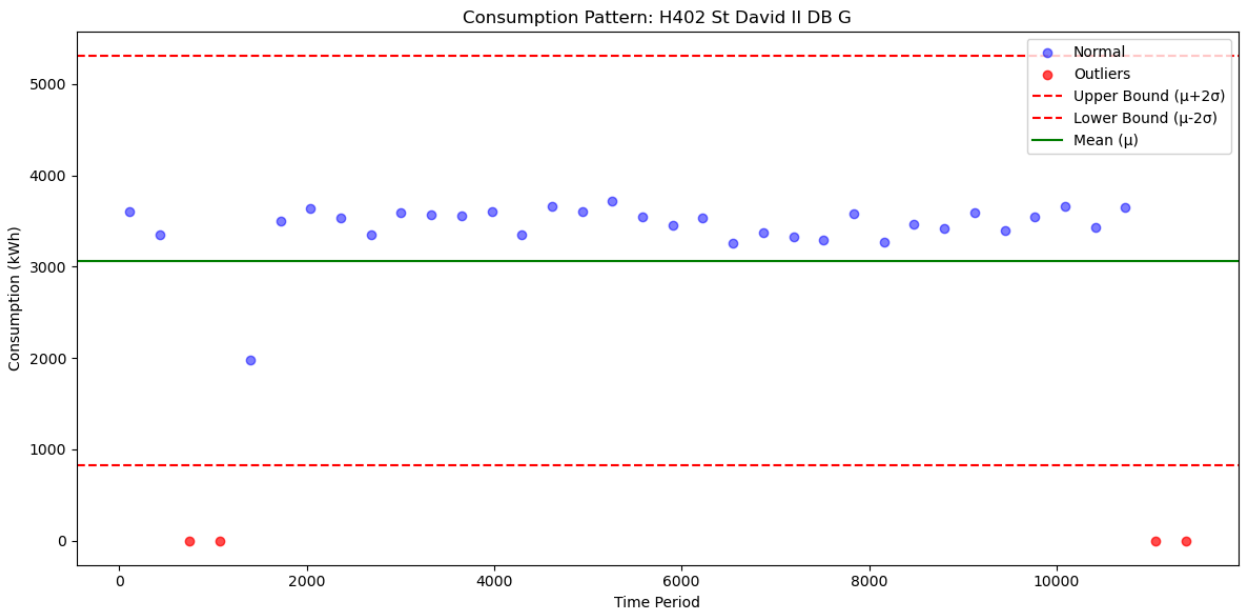

```
Statistics for F622 William James Riser:
Mean: 24580.49
Standard Deviation: 3897.55
Upper Bound: 32375.58
Lower Bound: 16785.40
Number of outliers: 2
```




```
Statistics for F62X William James and 2C combined:
Mean: 54199.90
Standard Deviation: 10739.60
Upper Bound: 75679.09
Lower Bound: 32720.70
Number of outliers: 2
```

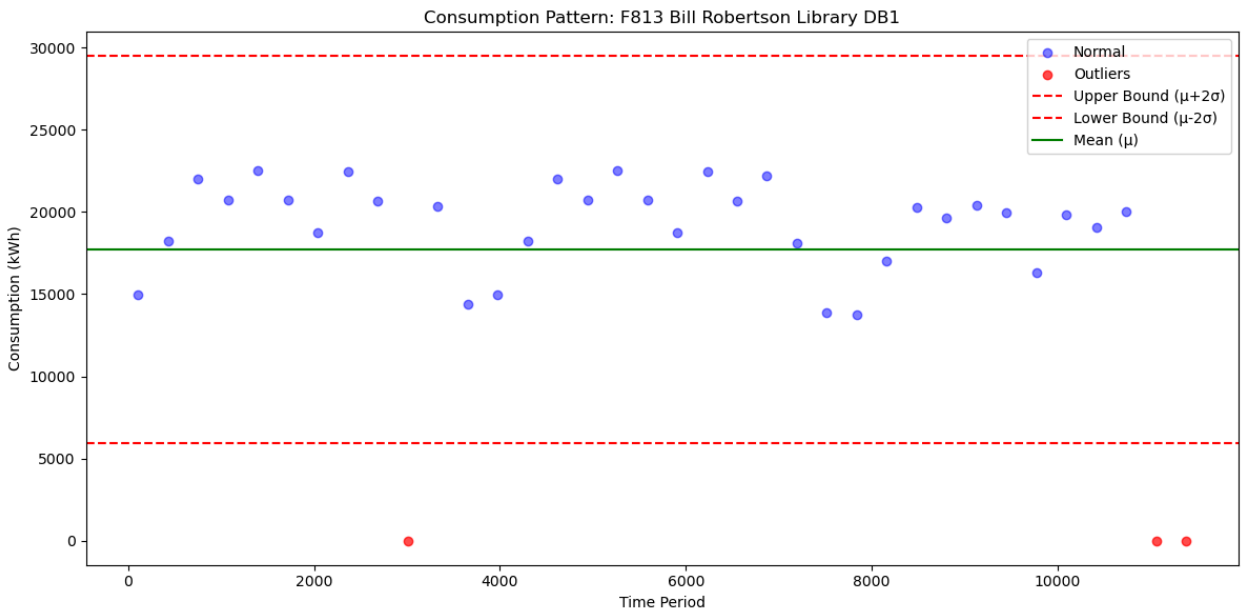



```
Statistics for H402 St David II DB G:
Mean: 3068.28
Standard Deviation: 1119.20
Upper Bound: 5306.68
Lower Bound: 829.88
Number of outliers: 4
```

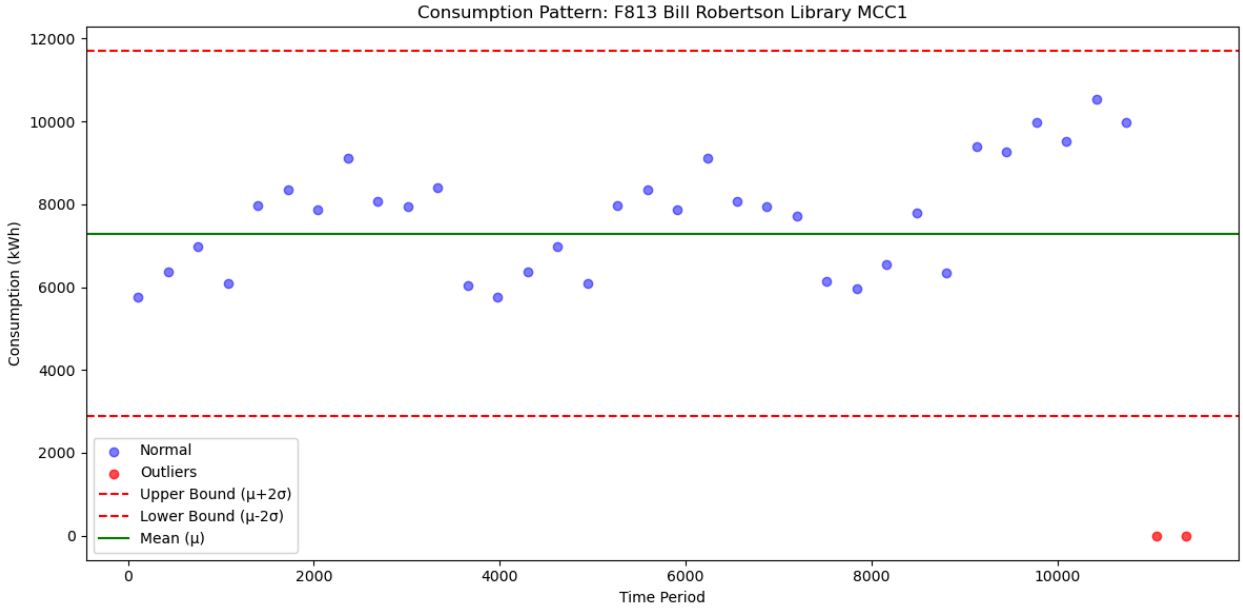



```
Statistics for F813 Bill Robertson Library 1 MSB:
Mean: 35273.01
Standard Deviation: 10907.94
Upper Bound: 57088.89
Lower Bound: 13457.14
Number of outliers: 2
```

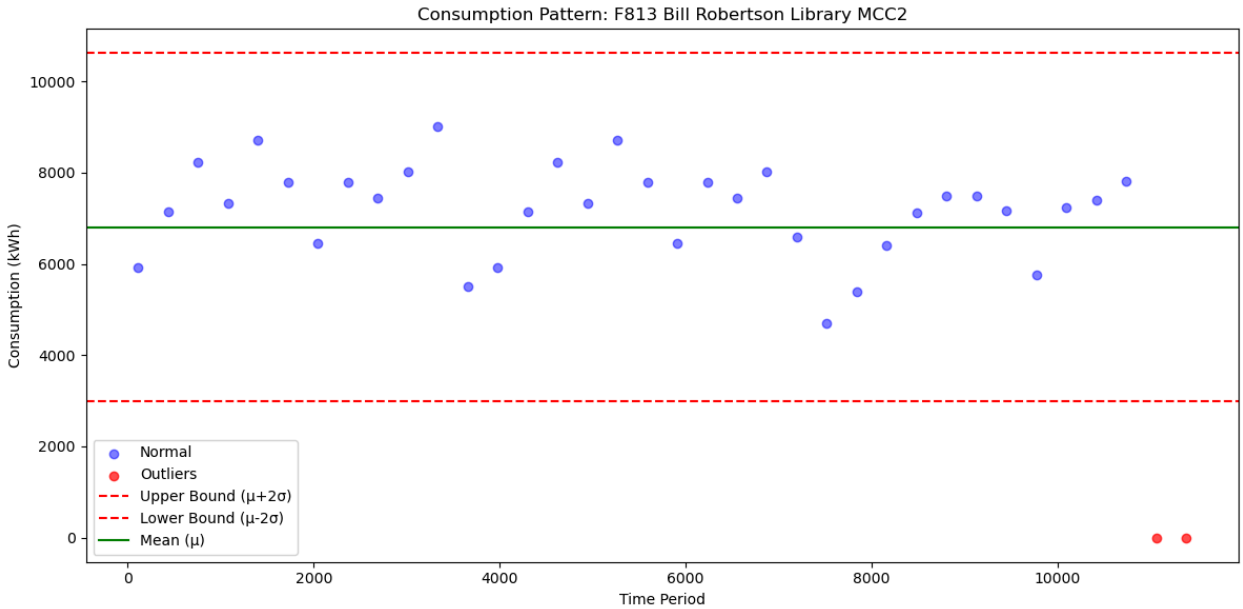
```
Statistics for F813 Bill Robertson Library DB1:
Mean: 17706.52
Standard Deviation: 5886.98
Upper Bound: 29480.48
Lower Bound: 5932.57
Number of outliers: 3
```



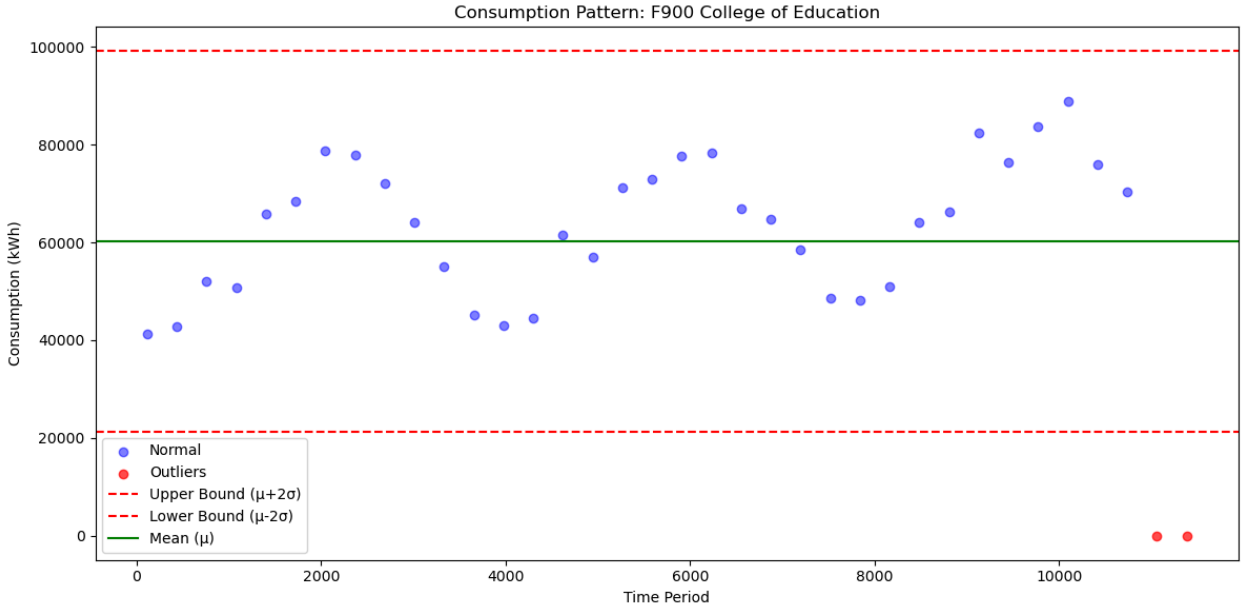

```
Statistics for F813 Bill Robertson Library MCC1:
Mean: 7295.79
Standard Deviation: 2201.05
Upper Bound: 11697.89
Lower Bound: 2893.68
Number of outliers: 2
```



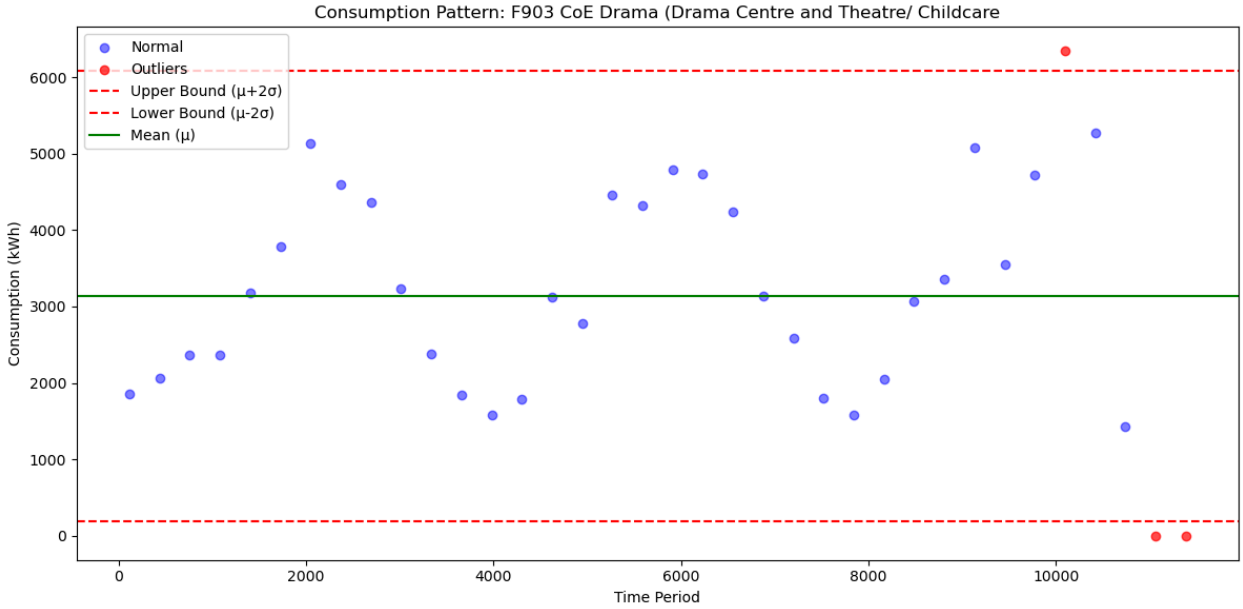

```
Statistics for F813 Bill Robertson Library MCC2:
Mean: 6799.55
Standard Deviation: 1909.05
Upper Bound: 10617.65
Lower Bound: 2981.46
Number of outliers: 2
```



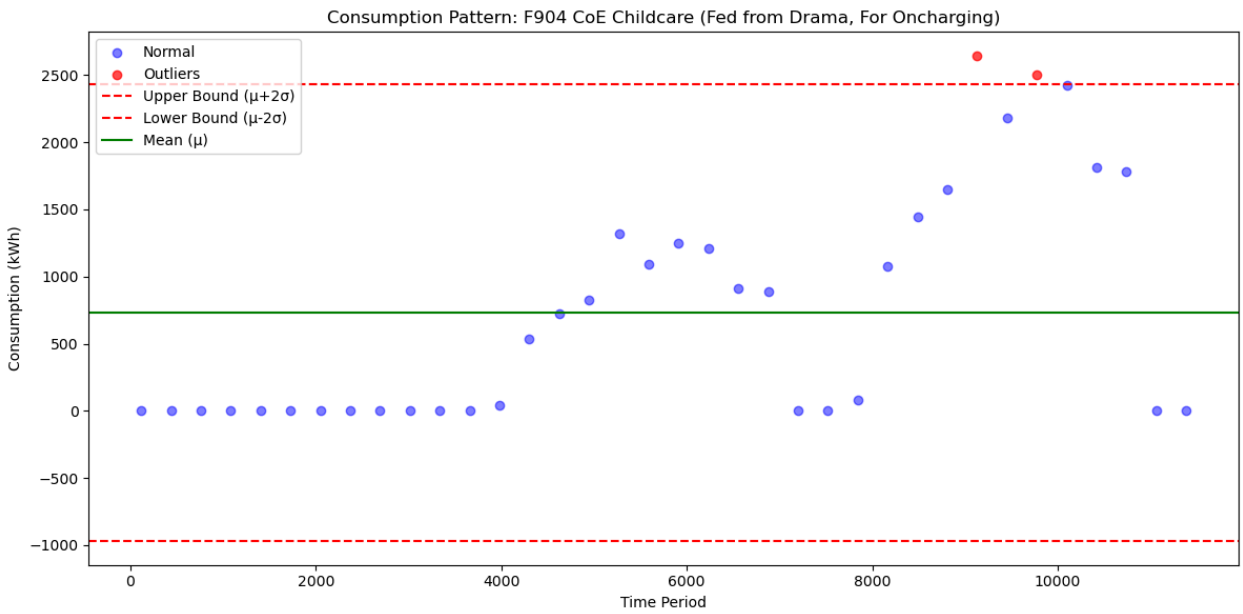

```
Statistics for F900 College of Education:
Mean: 60182.33
Standard Deviation: 19481.51
Upper Bound: 99145.35
Lower Bound: 21219.31
Number of outliers: 2
```



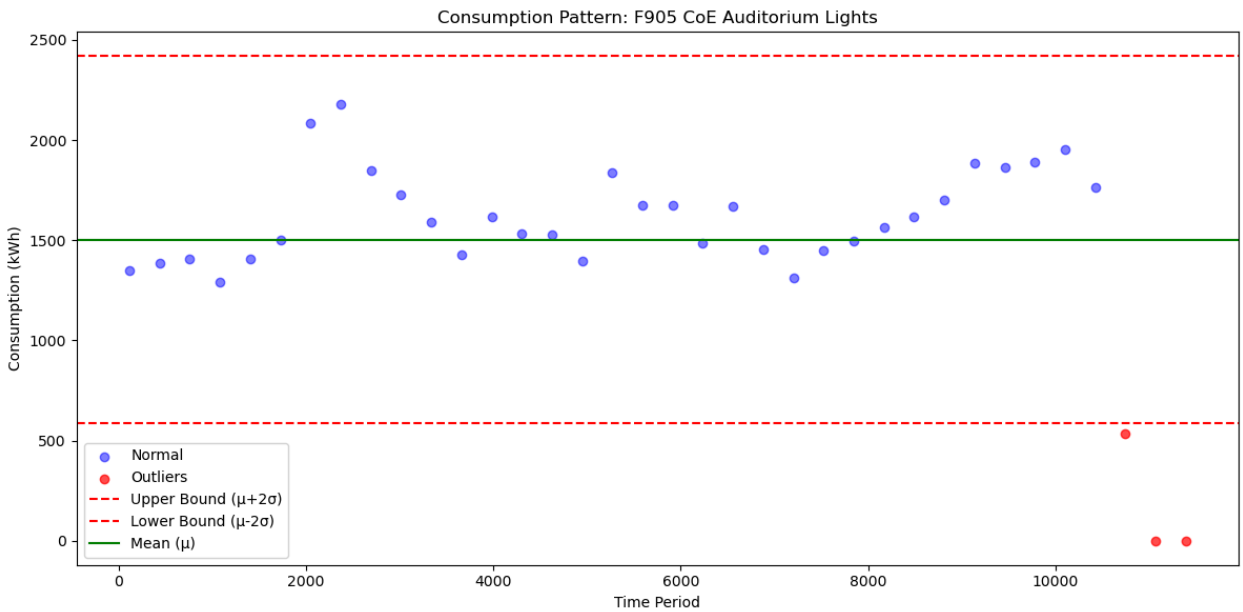

```
Statistics for F903 CoE Drama (Drama Centre and Theatre/ Childcare:
Mean: 3137.01
Standard Deviation: 1471.93
Upper Bound: 6080.87
Lower Bound: 193.14
Number of outliers: 3
```



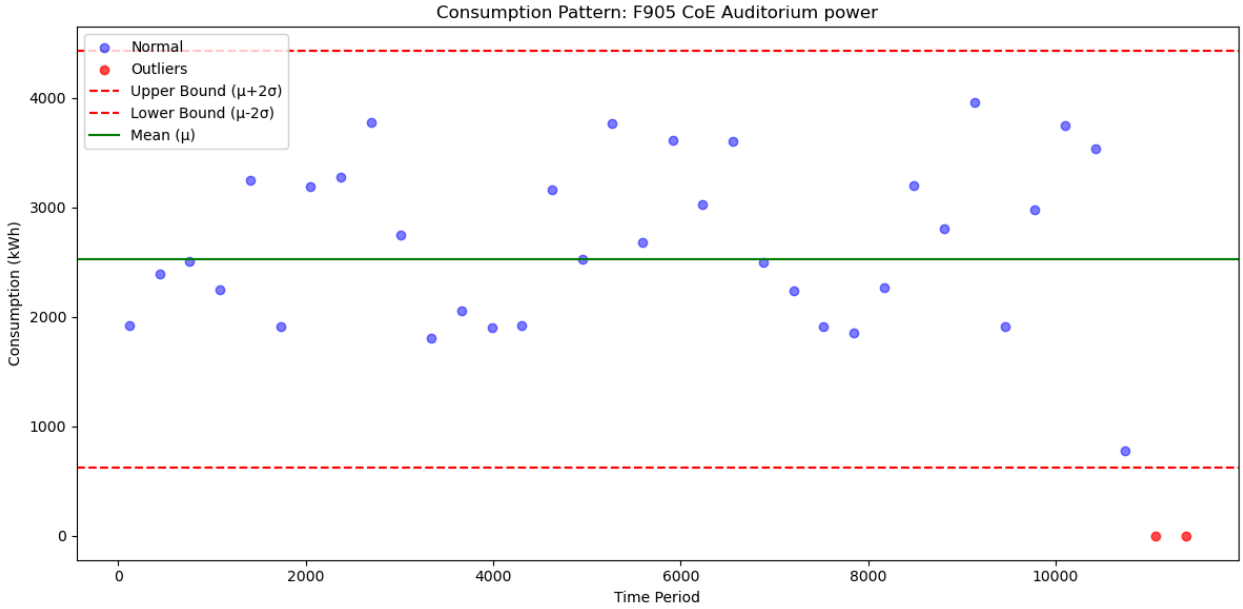

```
Statistics for F904 CoE Childcare (Fed from Drama, For Oncharging):
Mean: 732.54
Standard Deviation: 849.76
Upper Bound: 2432.06
Lower Bound: -966.98
Number of outliers: 2
```



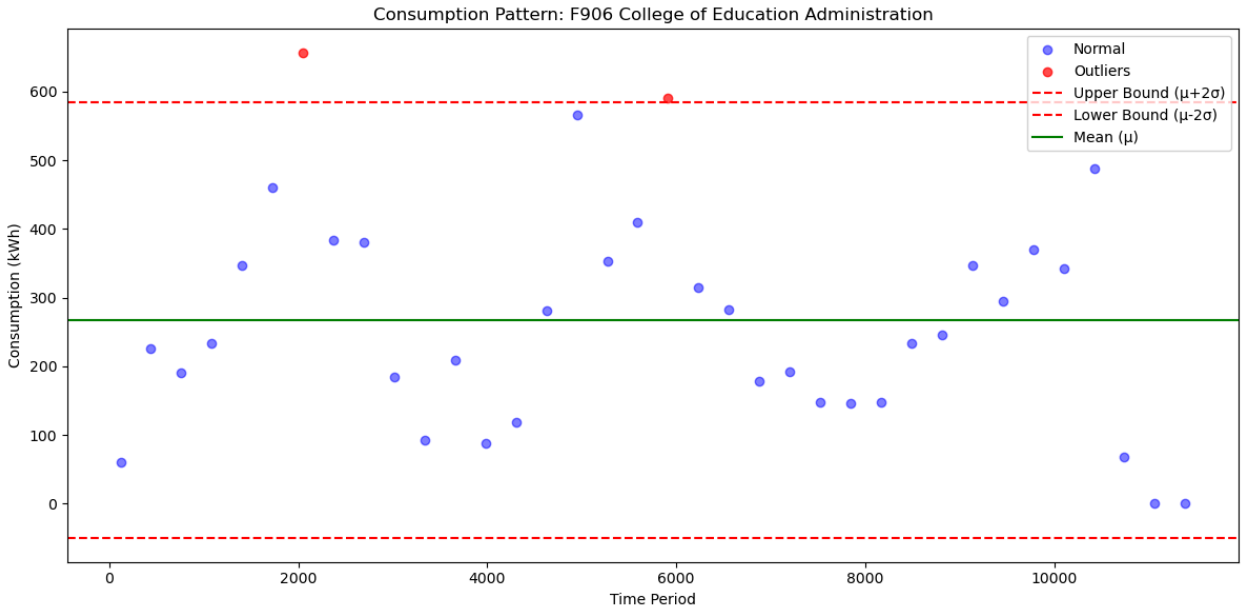

```
Statistics for F905 CoE Auditorium Lights:
Mean: 1503.27
Standard Deviation: 458.01
Upper Bound: 2419.30
Lower Bound: 587.25
Number of outliers: 3
```



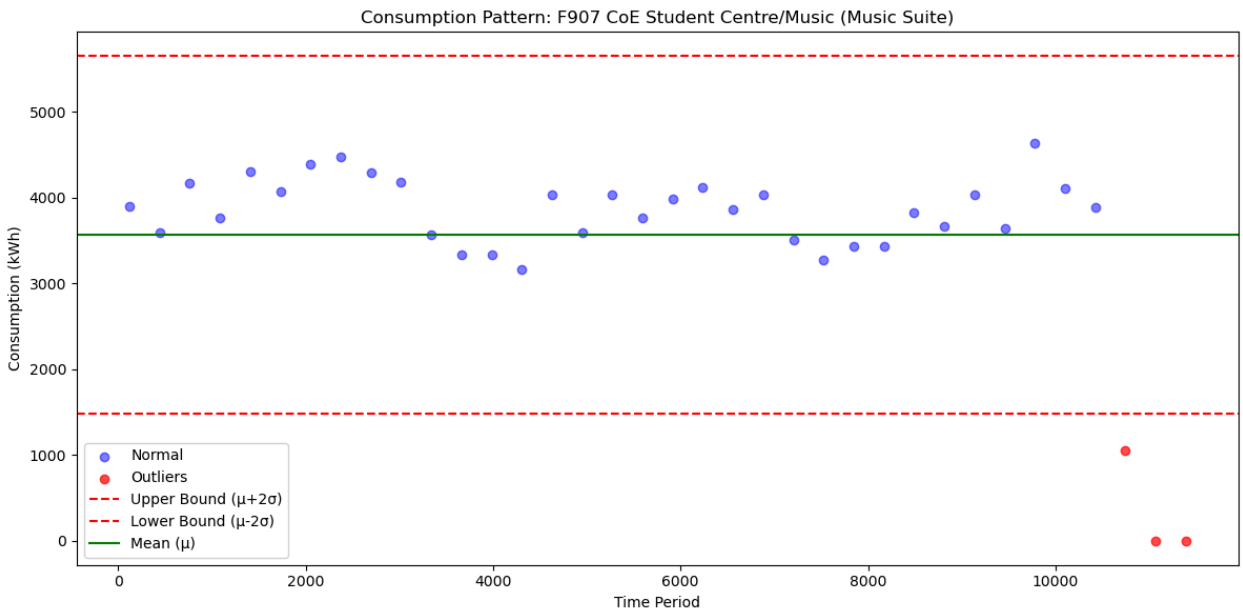

```
Statistics for F905 CoE Auditorium power:
Mean: 2525.77
Standard Deviation: 950.20
Upper Bound: 4426.17
Lower Bound: 625.38
Number of outliers: 2
```



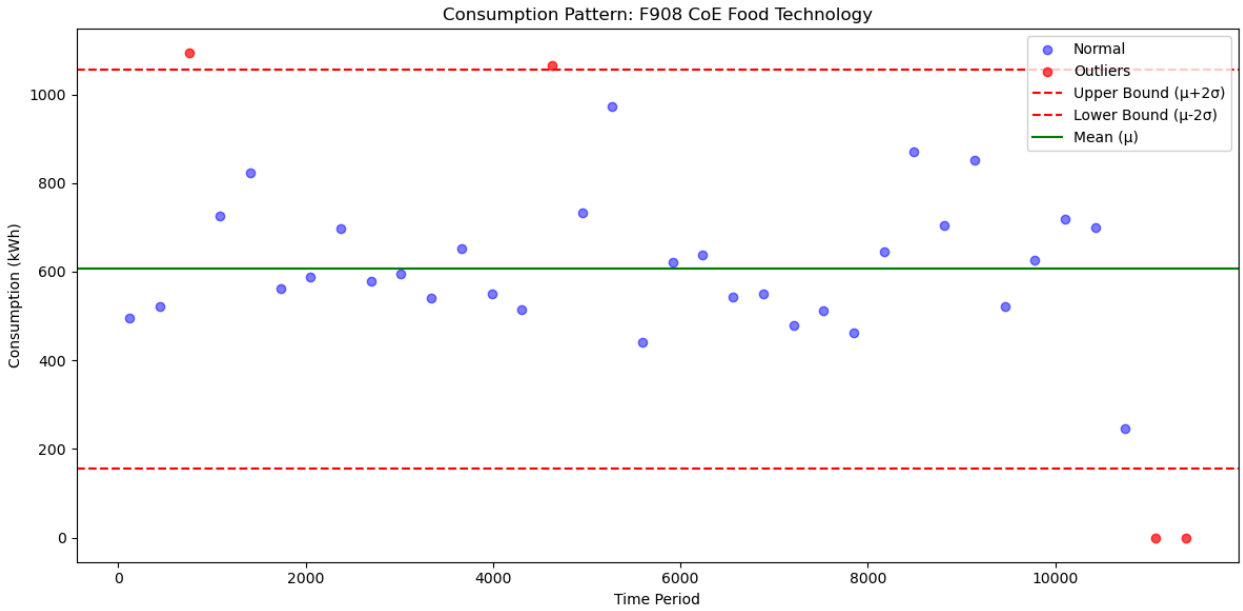

```
Statistics for F906 College of Education Administration:
Mean: 267.41
Standard Deviation: 158.53
Upper Bound: 584.47
Lower Bound: -49.66
Number of outliers: 2
```



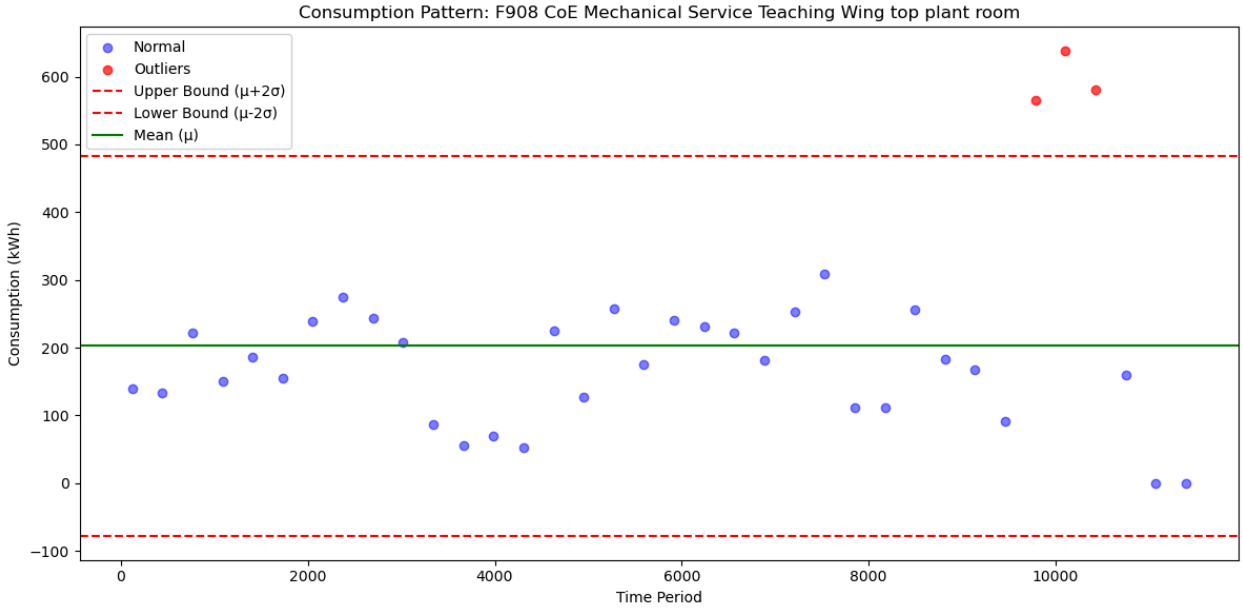

```
Statistics for F907 CoE Student Centre/Music (Music Suite):
Mean: 3567.76
Standard Deviation: 1040.99
Upper Bound: 5649.73
Lower Bound: 1485.79
Number of outliers: 3
```



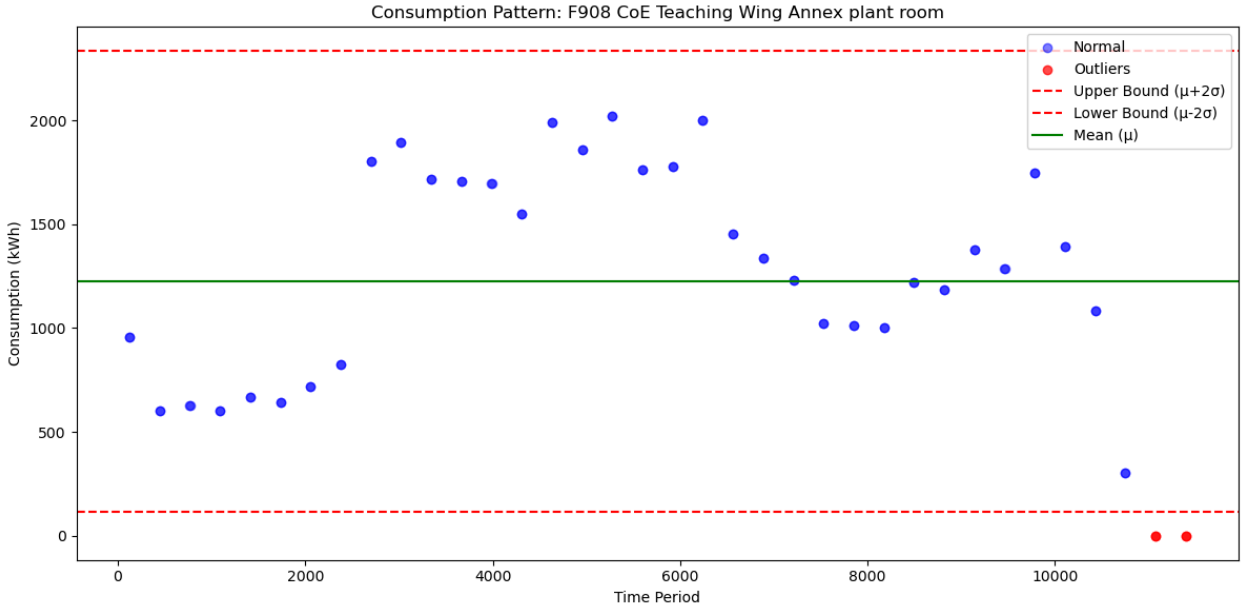

```
Statistics for F908 CoE Food Technology:
Mean: 606.67
Standard Deviation: 225.15
Upper Bound: 1056.97
Lower Bound: 156.38
Number of outliers: 4
```



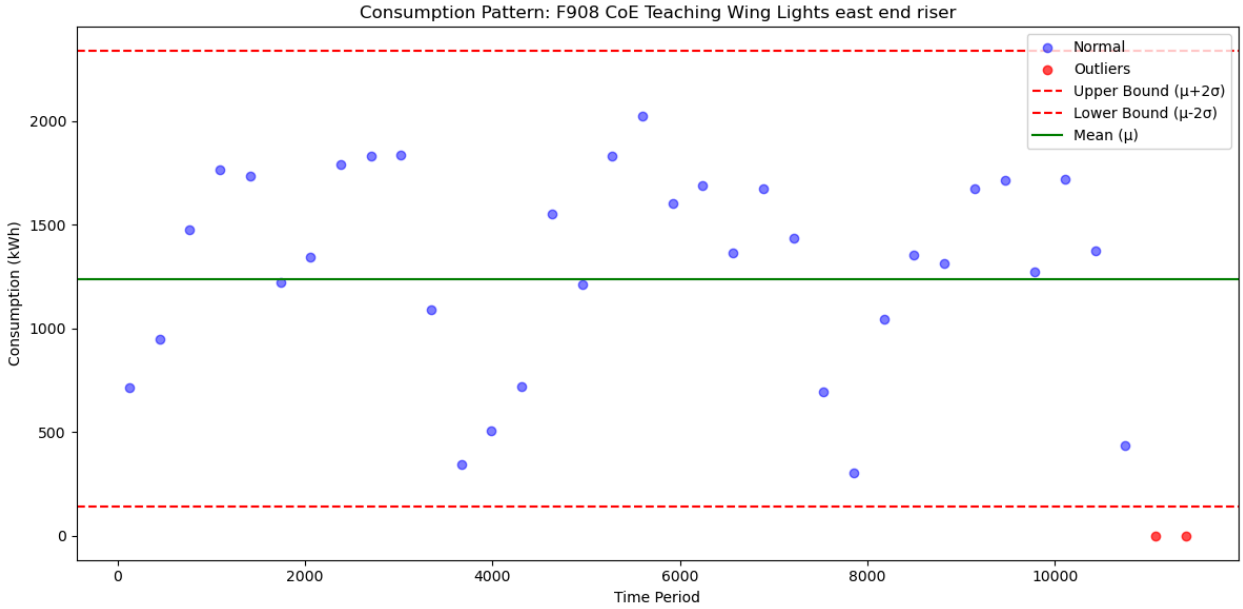

```
Statistics for F908 CoE Mechanical Service Teaching Wing top plant room:
Mean: 202.68
Standard Deviation: 140.15
Upper Bound: 482.99
Lower Bound: -77.62
Number of outliers: 3
```



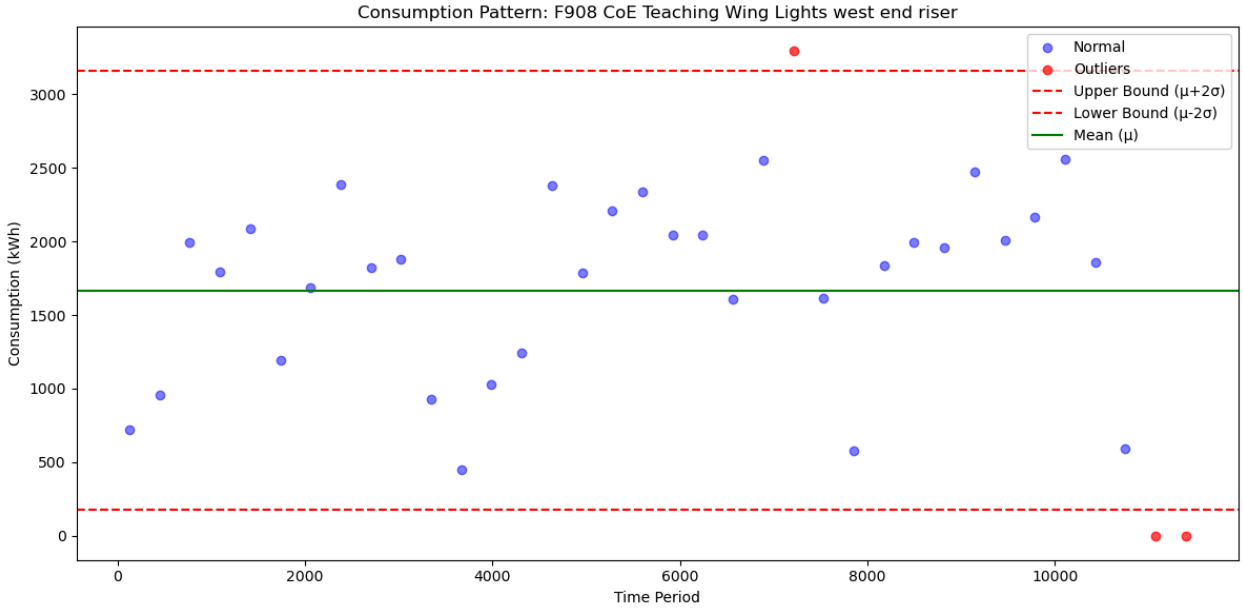

```
Statistics for F908 CoE Teaching Wing Annex plant room:
Mean: 1224.03
Standard Deviation: 554.65
Upper Bound: 2333.34
Lower Bound: 114.72
Number of outliers: 4
```



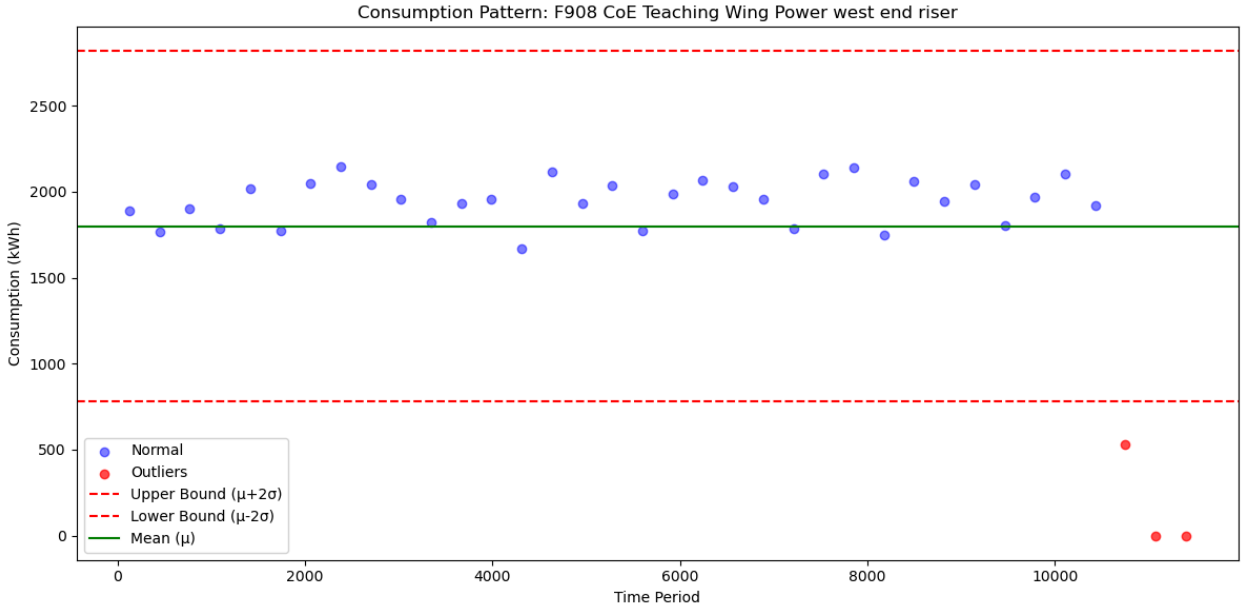

```
Statistics for F908 CoE Teaching Wing Lights east end riser:
Mean: 1238.91
Standard Deviation: 549.08
Upper Bound: 2337.07
Lower Bound: 140.75
Number of outliers: 2
```



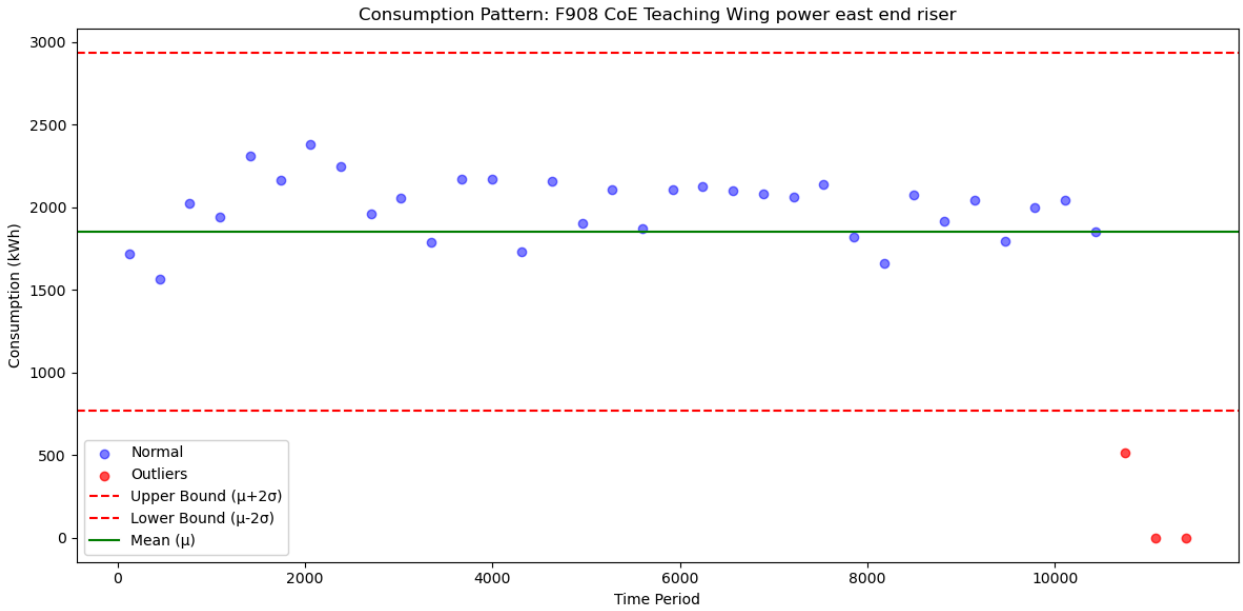

```
Statistics for F908 CoE Teaching Wing Lights west end riser:
Mean: 1668.43
Standard Deviation: 746.73
Upper Bound: 3161.88
Lower Bound: 174.98
Number of outliers: 3
```




```
Statistics for F908 CoE Teaching Wing Power west end riser:
Mean: 1798.52
Standard Deviation: 509.14
Upper Bound: 2816.81
Lower Bound: 780.24
Number of outliers: 3
```

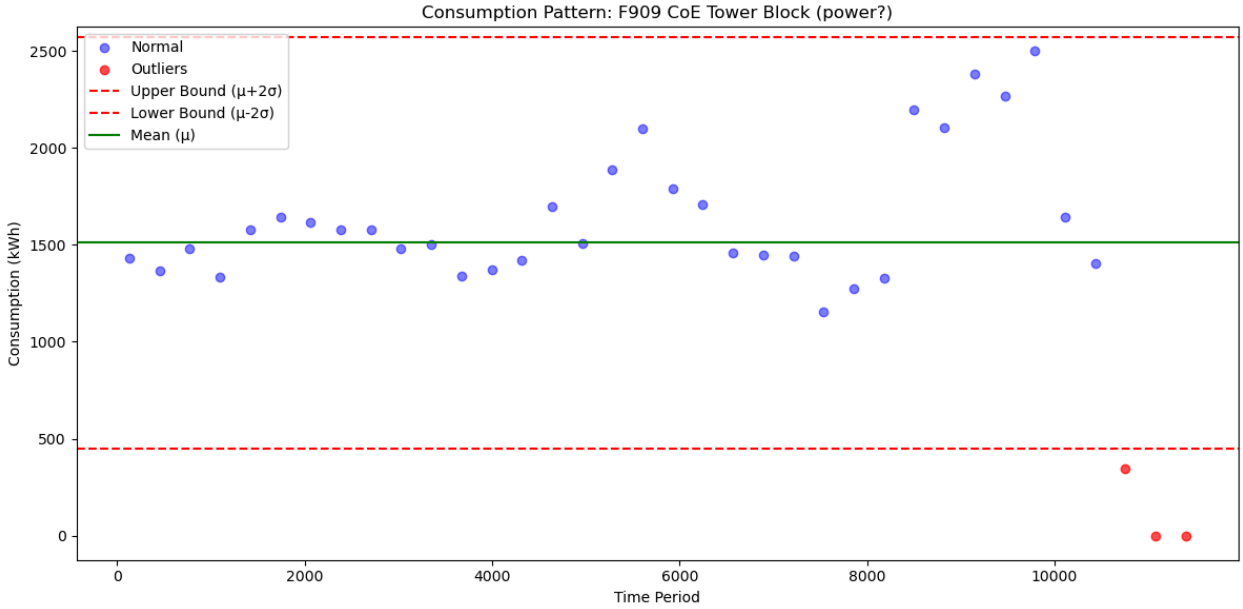



```
Statistics for F908 CoE Teaching Wing power east end riser:
Mean: 1850.53
Standard Deviation: 541.12
Upper Bound: 2932.77
Lower Bound: 768.30
Number of outliers: 3
```

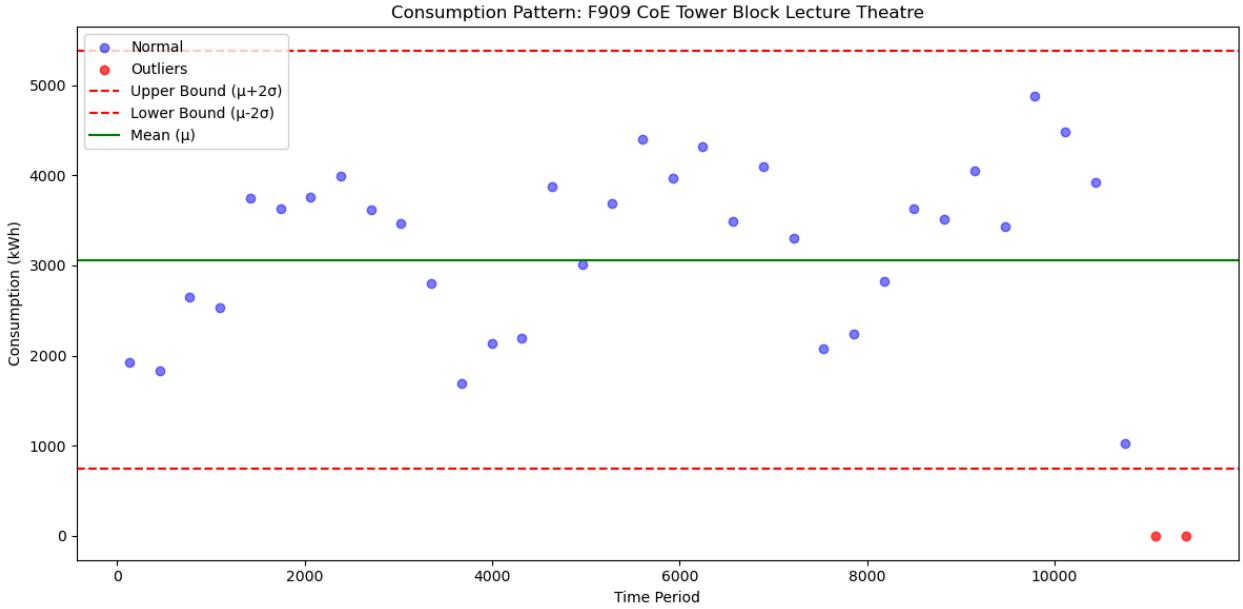



```
Statistics for F909 CoE Tower Block (lights?):
Mean: 1308.08
Standard Deviation: 546.54
Upper Bound: 2401.16
Lower Bound: 214.99
Number of outliers: 2
```

```
Statistics for F909 CoE Tower Block (power?):
Mean: 1510.35
Standard Deviation: 529.85
Upper Bound: 2570.05
Lower Bound: 450.65
Number of outliers: 3
```

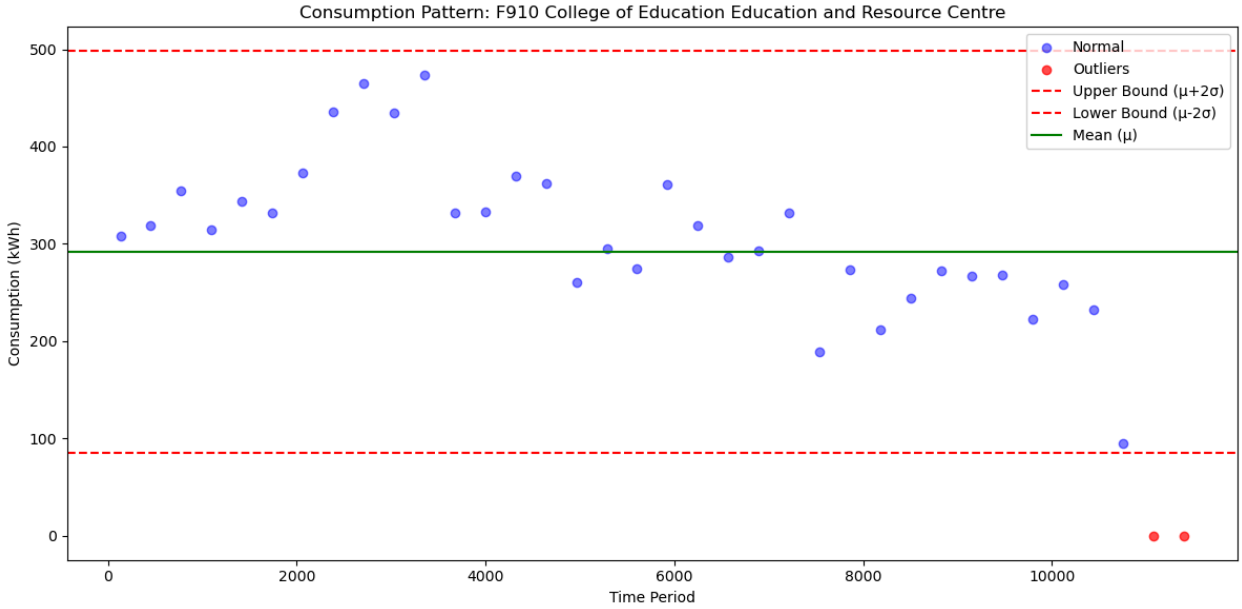



```
Statistics for F909 CoE Tower Block Lecture Theatre:
Mean: 3061.29
Standard Deviation: 1158.92
Upper Bound: 5379.12
Lower Bound: 743.46
Number of outliers: 2
```

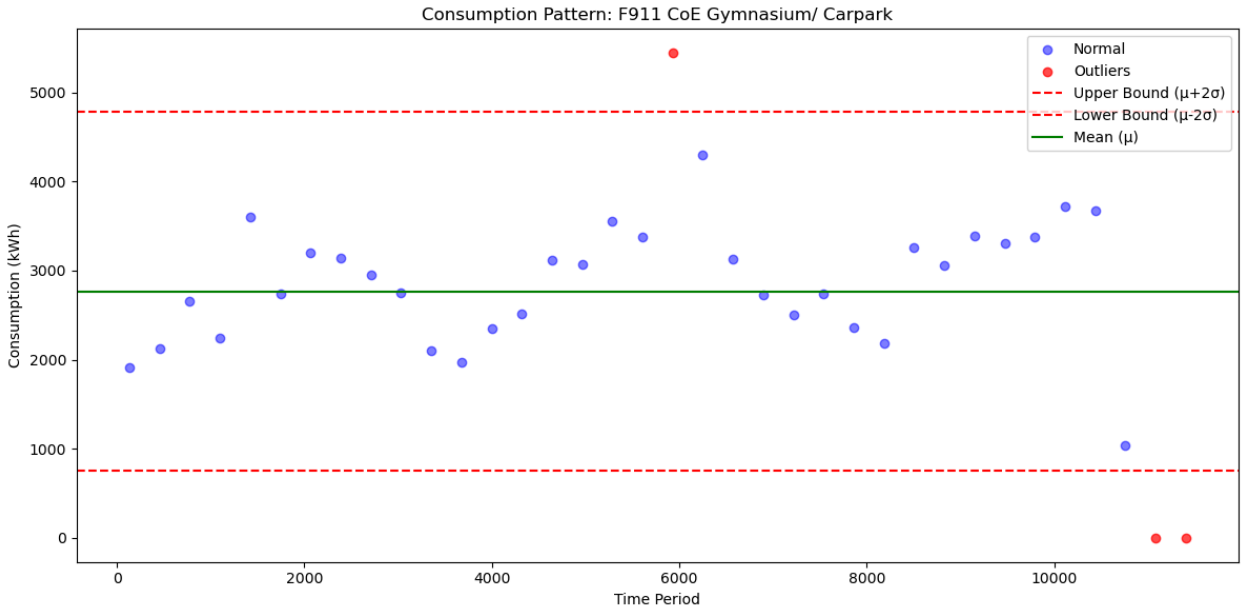



```
Statistics for F909 CoE Tower Block Lift:
Mean: 747.35
Standard Deviation: 233.87
Upper Bound: 1215.10
Lower Bound: 279.60
Number of outliers: 3
```

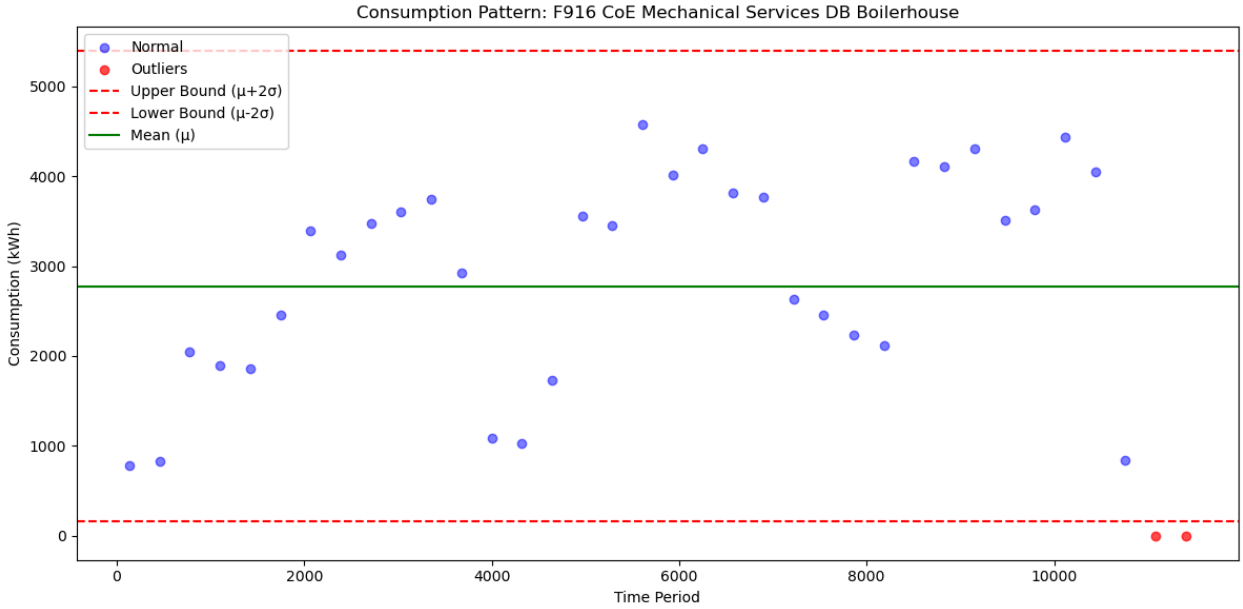
```
Statistics for F910 College of Education Education and Resource Centre:
Mean: 291.77
Standard Deviation: 103.12
Upper Bound: 498.01
Lower Bound: 85.53
Number of outliers: 2
```



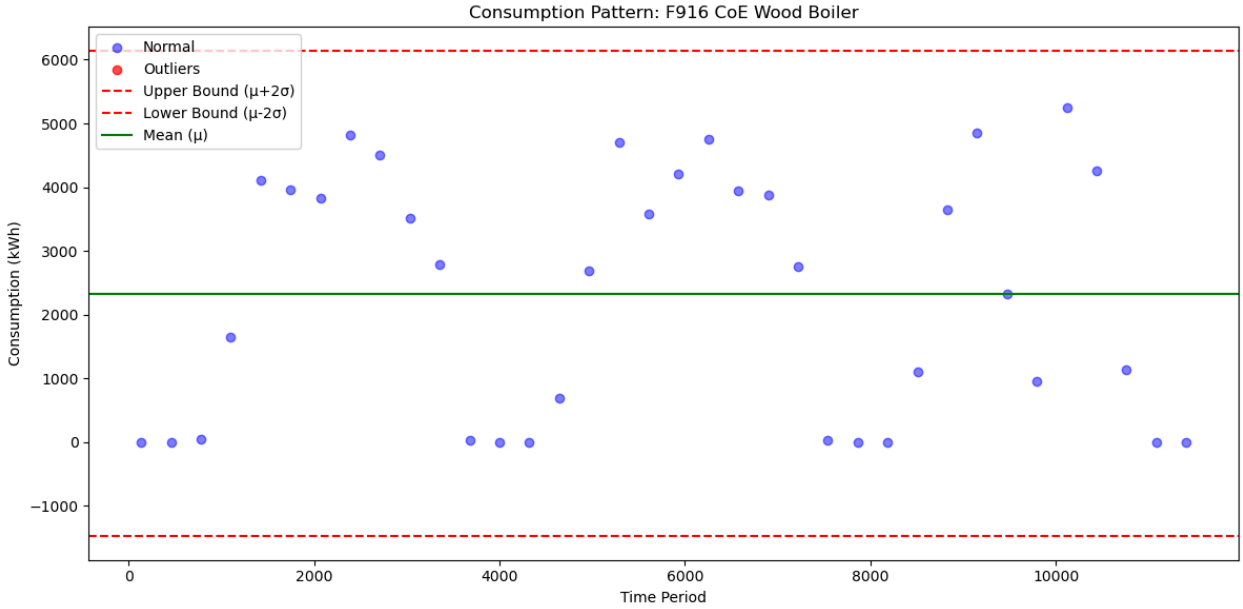

```
Statistics for F911 CoE Gymnasium/ Carpark:
Mean: 2766.18
Standard Deviation: 1008.54
Upper Bound: 4783.26
Lower Bound: 749.09
Number of outliers: 3
```



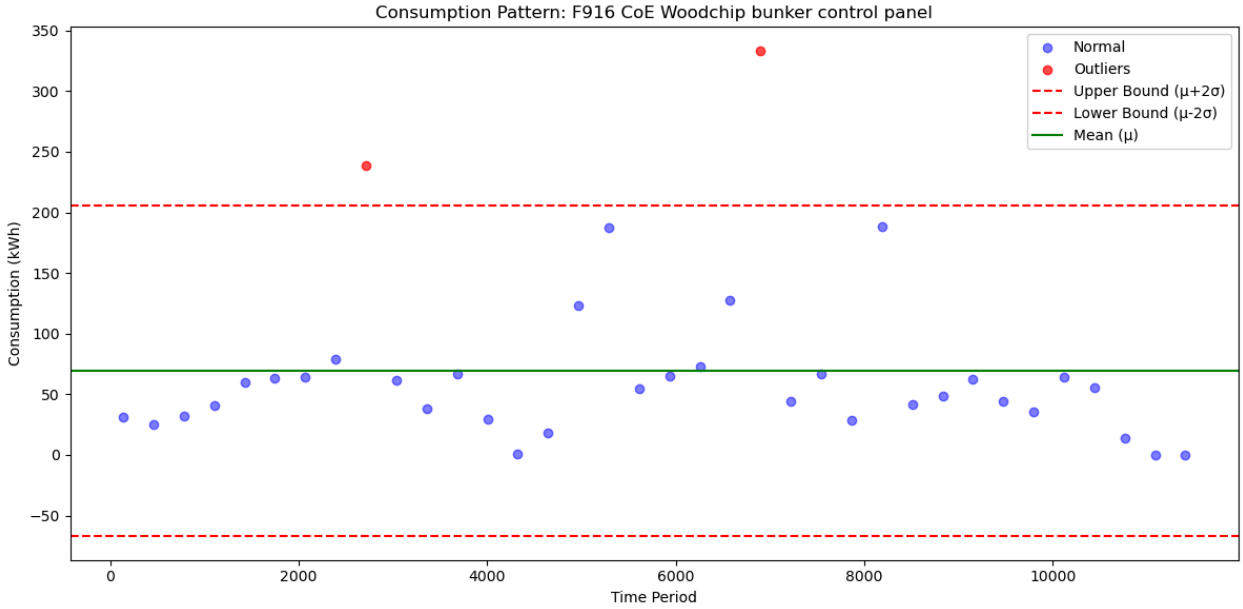

```
Statistics for F916 CoE Mechanical Services DB Boilerhouse:
Mean: 2775.93
Standard Deviation: 1307.61
Upper Bound: 5391.14
Lower Bound: 160.72
Number of outliers: 2
```



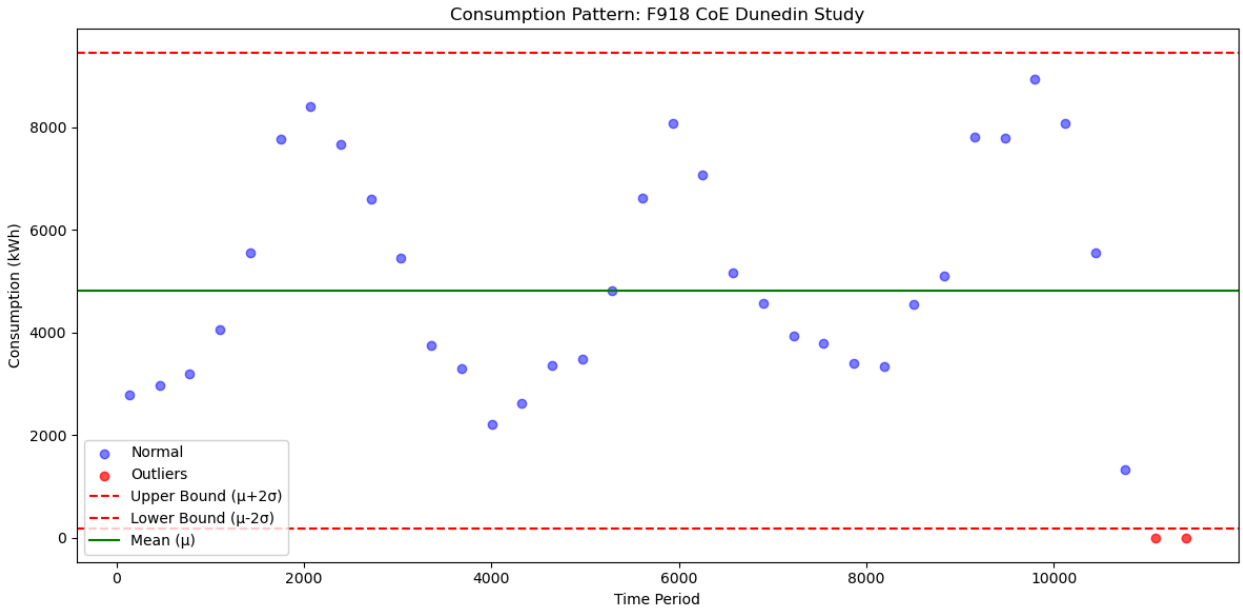

```
Statistics for F916 CoE Wood Boiler:
Mean: 2333.60
Standard Deviation: 1900.05
Upper Bound: 6133.70
Lower Bound: -1466.50
Number of outliers: 0
```



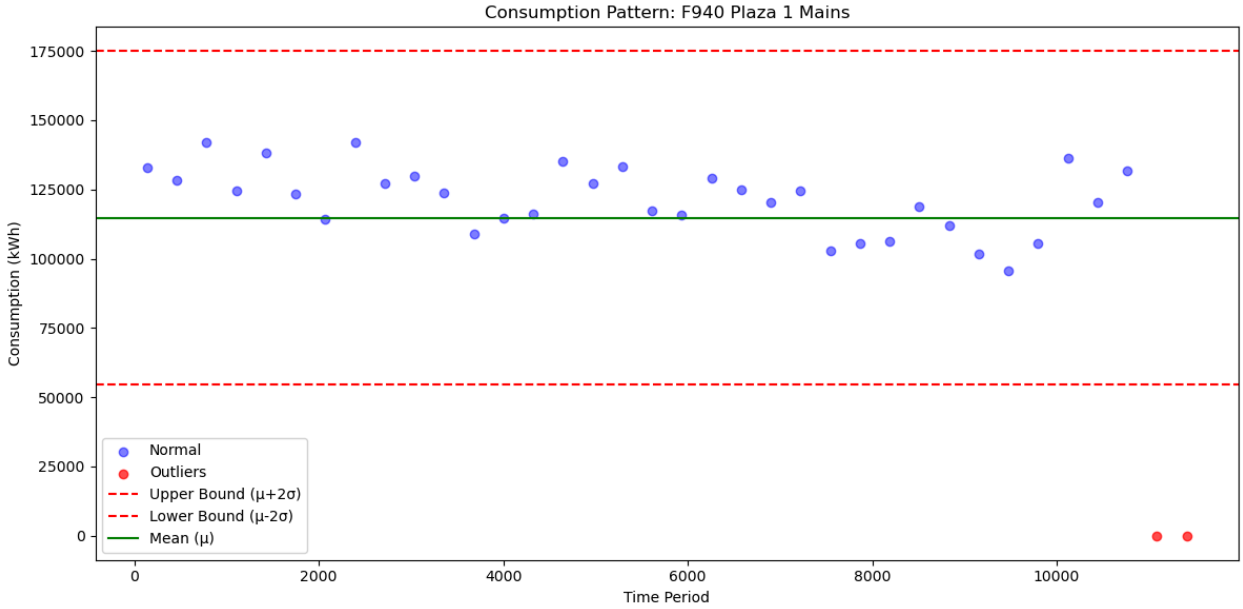

```
Statistics for F916 CoE Woodchip bunker control panel:
Mean: 69.59
Standard Deviation: 68.11
Upper Bound: 205.80
Lower Bound: -66.63
Number of outliers: 2
```



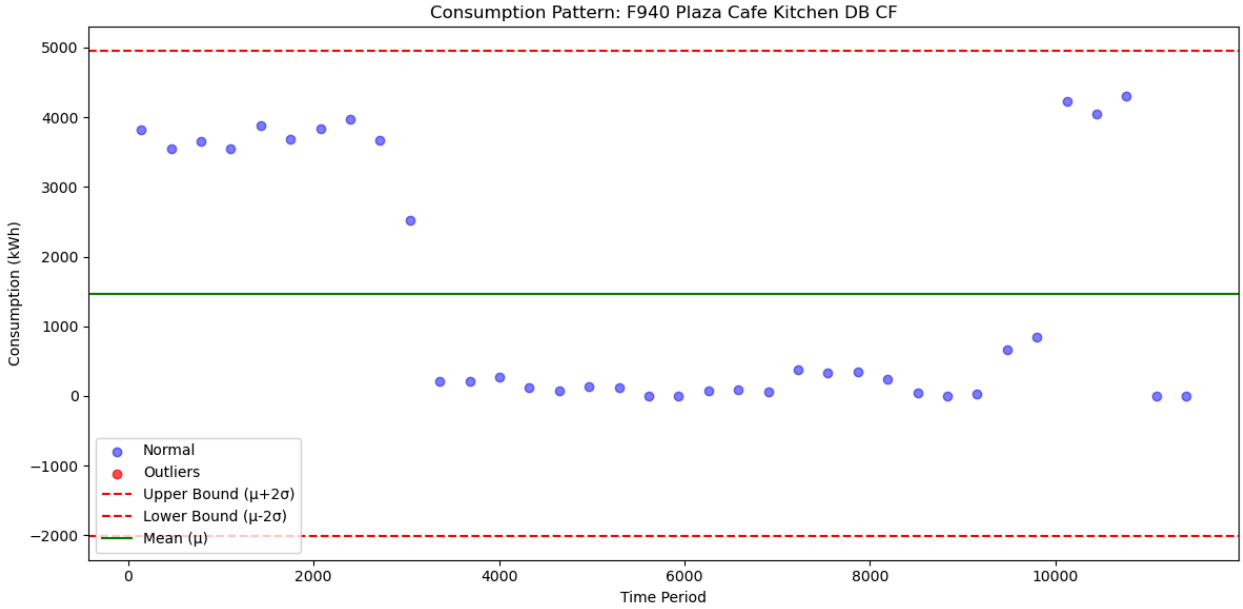

```
Statistics for F918 CoE Dunedin Study:
Mean: 4807.99
Standard Deviation: 2316.83
Upper Bound: 9441.65
Lower Bound: 174.32
Number of outliers: 2
```



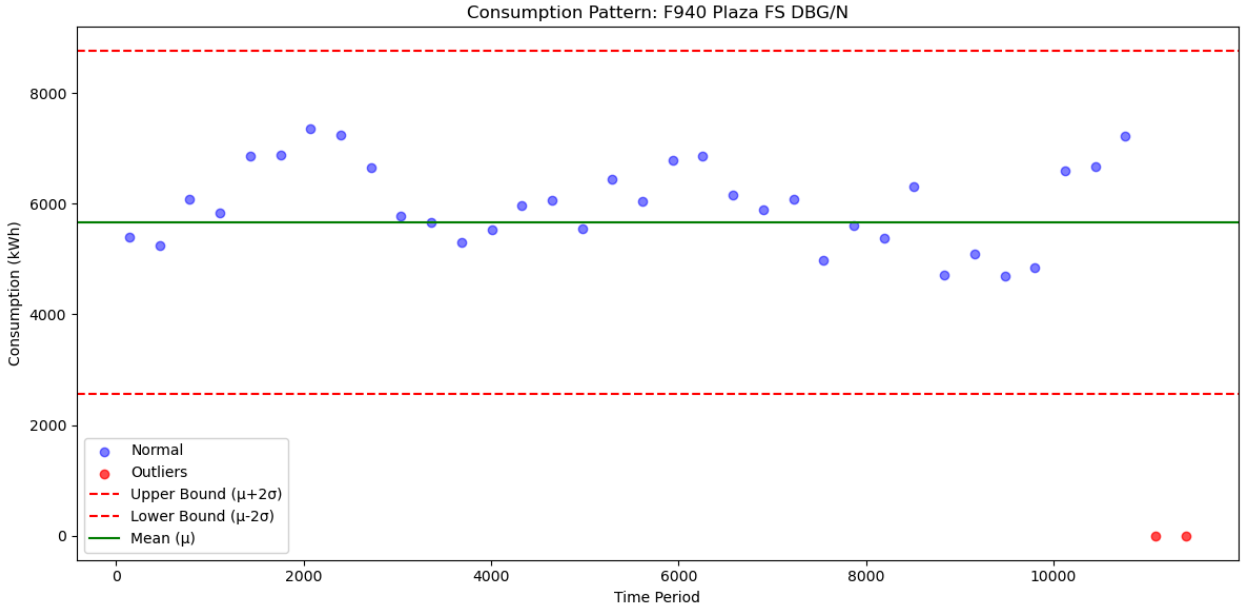

```
Statistics for F940 Plaza 1 Mains:
Mean: 114724.59
Standard Deviation: 30098.00
Upper Bound: 174920.60
Lower Bound: 54528.59
Number of outliers: 2
```



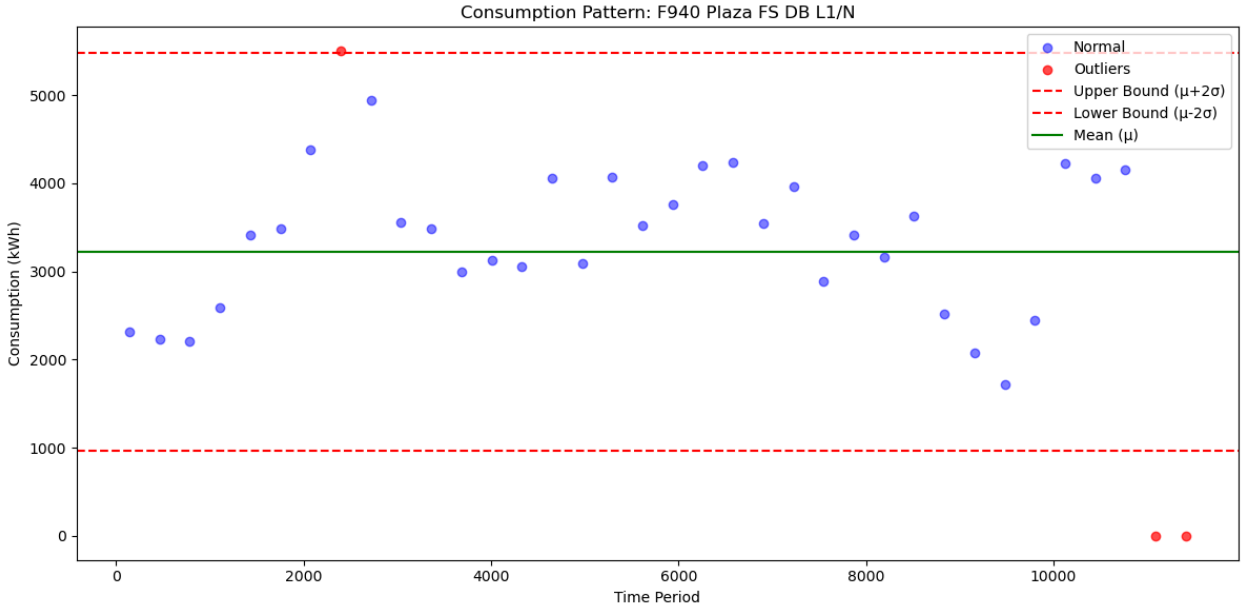

```
Statistics for F940 Plaza Cafe Kitchen DB CF:
Mean: 1471.44
Standard Deviation: 1738.27
Upper Bound: 4947.98
Lower Bound: -2005.10
Number of outliers: 0
```



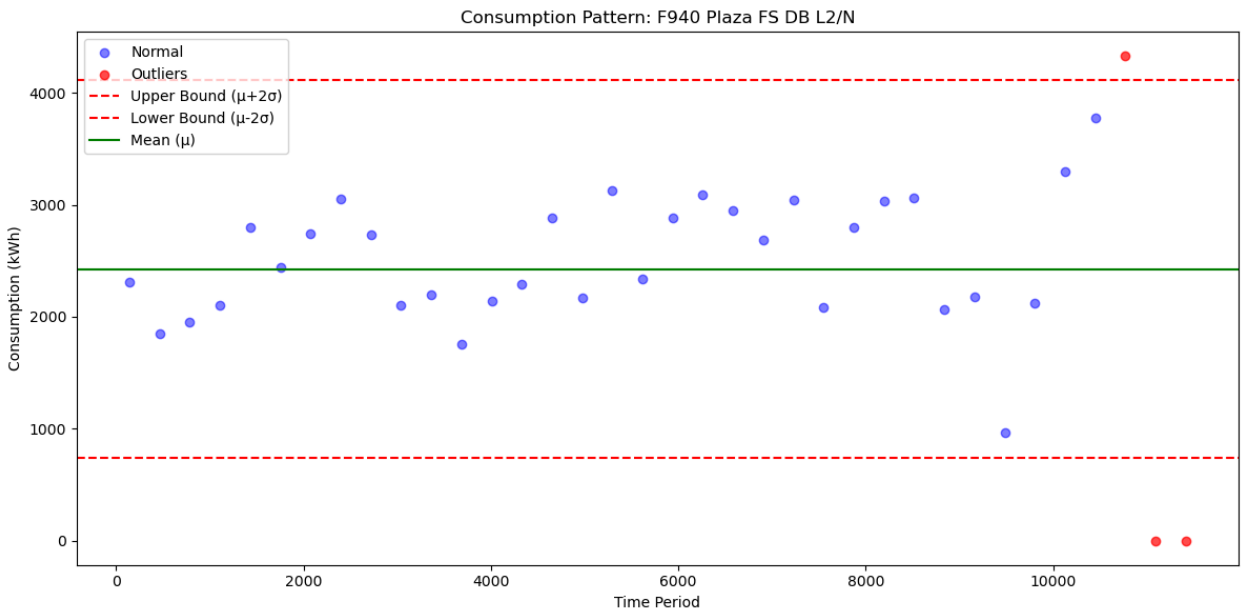

```
Statistics for F940 Plaza FS DBG/N:
Mean: 5662.52
Standard Deviation: 1548.68
Upper Bound: 8759.87
Lower Bound: 2565.17
Number of outliers: 2
```



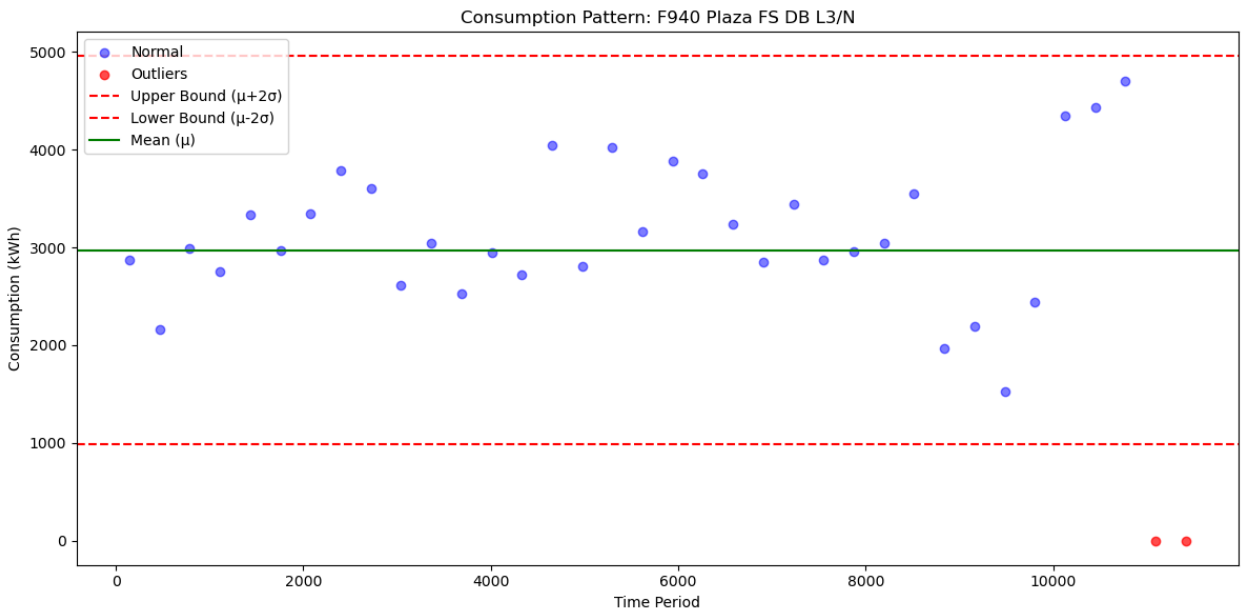

```
Statistics for F940 Plaza FS DB L1/N:
Mean: 3223.66
Standard Deviation: 1129.18
Upper Bound: 5482.01
Lower Bound: 965.31
Number of outliers: 3
```



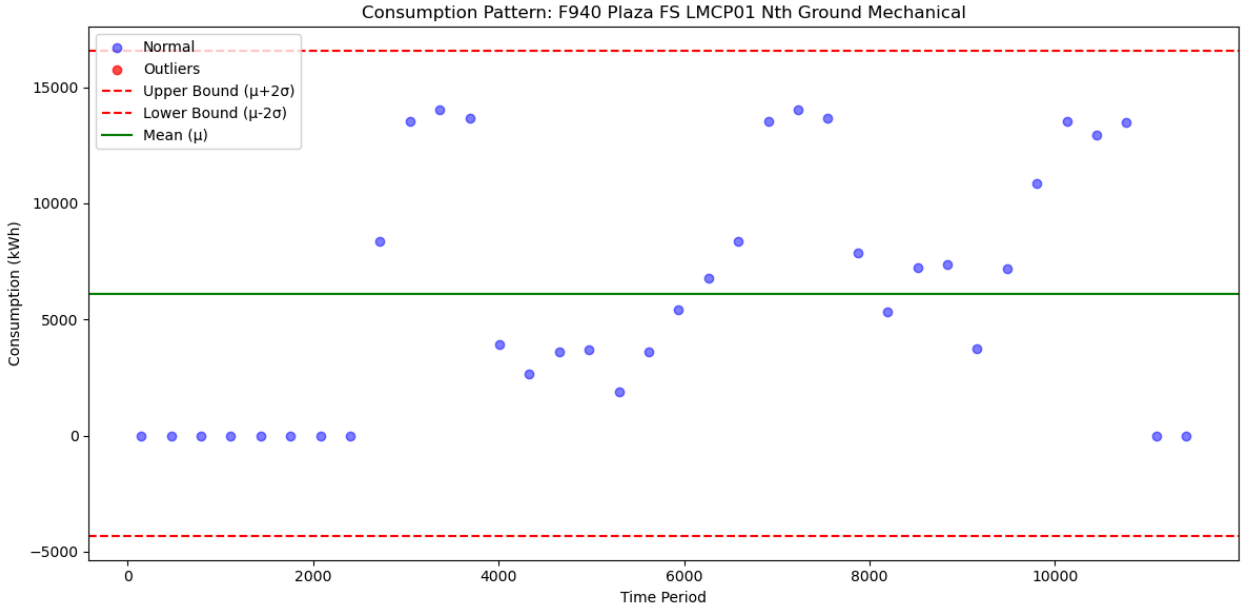

```
Statistics for F940 Plaza FS DB L2/N:
Mean: 2426.57
Standard Deviation: 844.30
Upper Bound: 4115.16
Lower Bound: 737.98
Number of outliers: 3
```



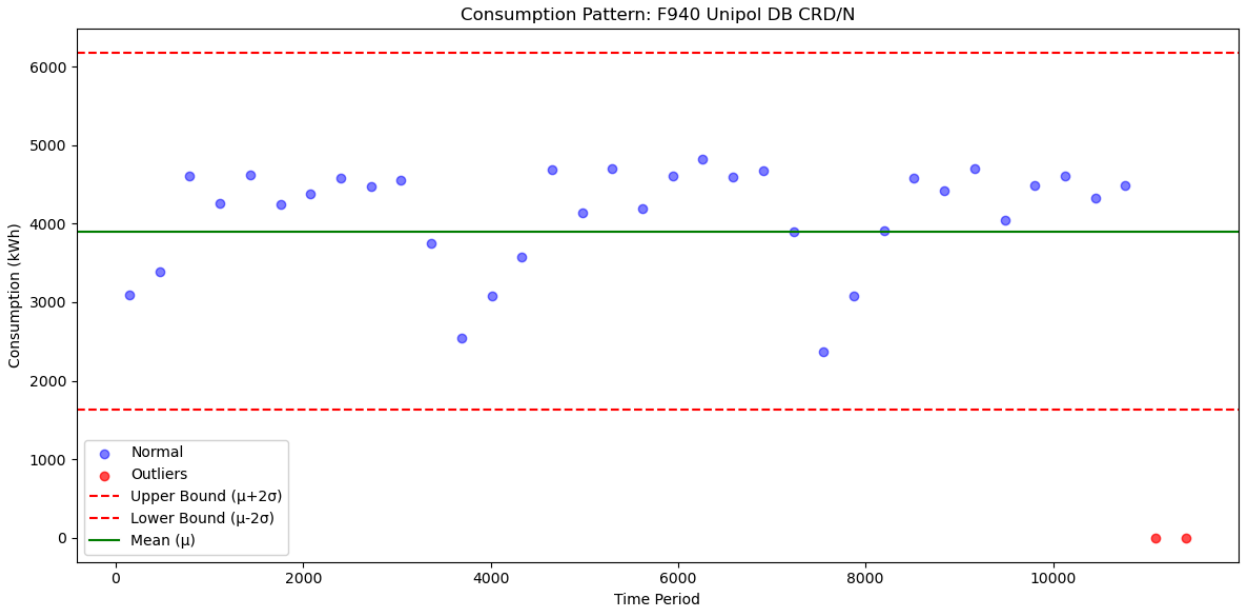

```
Statistics for F940 Plaza FS DB L3/N:
Mean: 2969.31
Standard Deviation: 993.47
Upper Bound: 4956.25
Lower Bound: 982.37
Number of outliers: 2
```



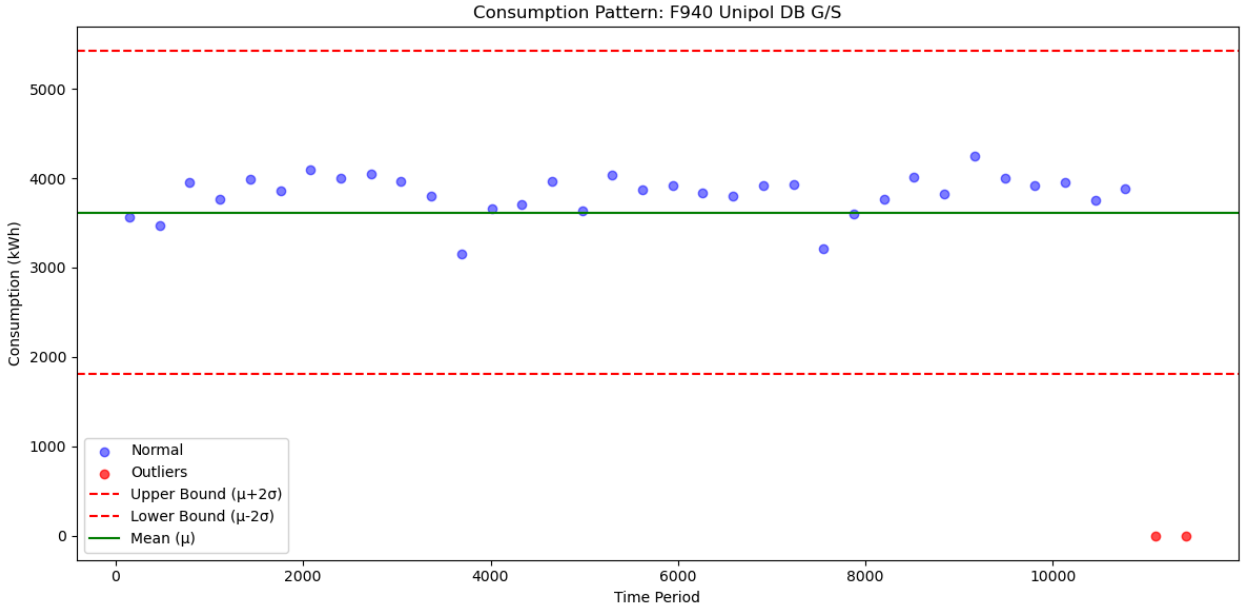

```
Statistics for F940 Plaza FS LMCP01 Nth Ground Mechanical:
Mean: 6120.43
Standard Deviation: 5218.38
Upper Bound: 16557.20
Lower Bound: -4316.34
Number of outliers: 0
```



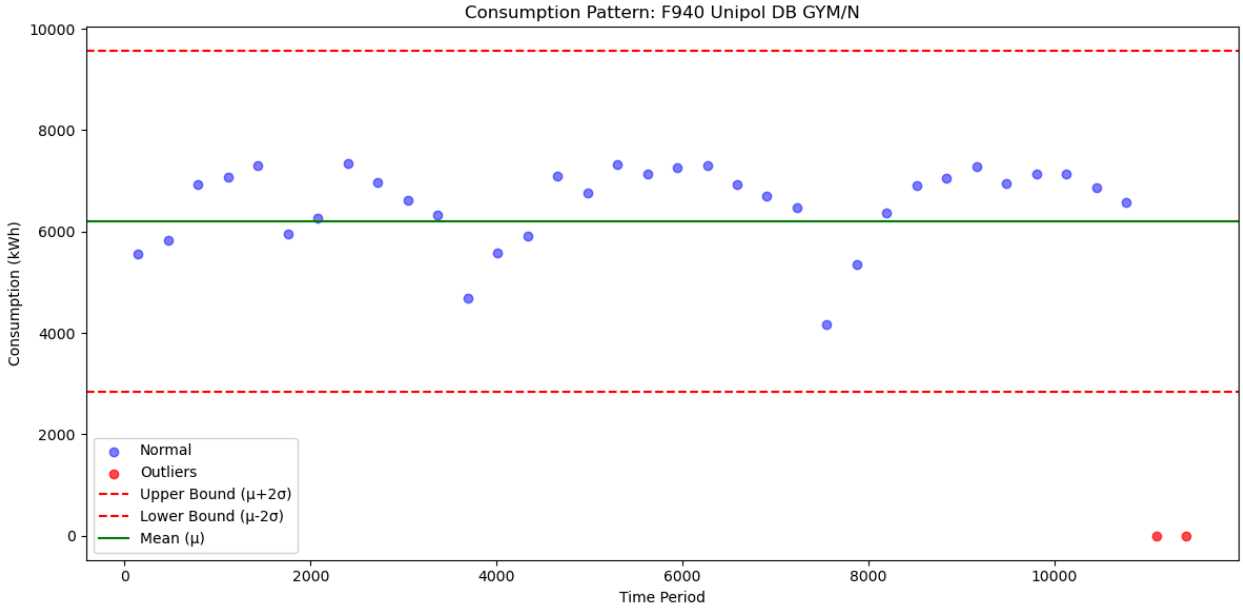

```
Statistics for F940 Unipol DB CRD/N:
Mean: 3902.67
Standard Deviation: 1134.48
Upper Bound: 6171.63
Lower Bound: 1633.72
Number of outliers: 2
```



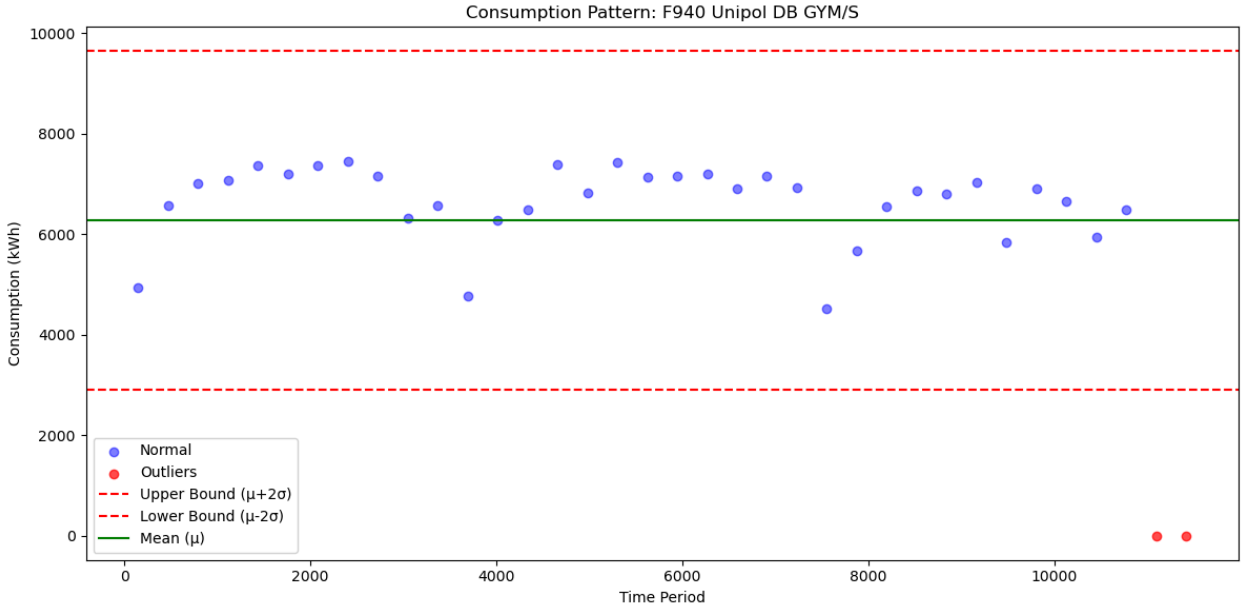

```
Statistics for F940 Unipol DB G/S:
Mean: 3613.39
Standard Deviation: 903.80
Upper Bound: 5420.98
Lower Bound: 1805.79
Number of outliers: 2
```



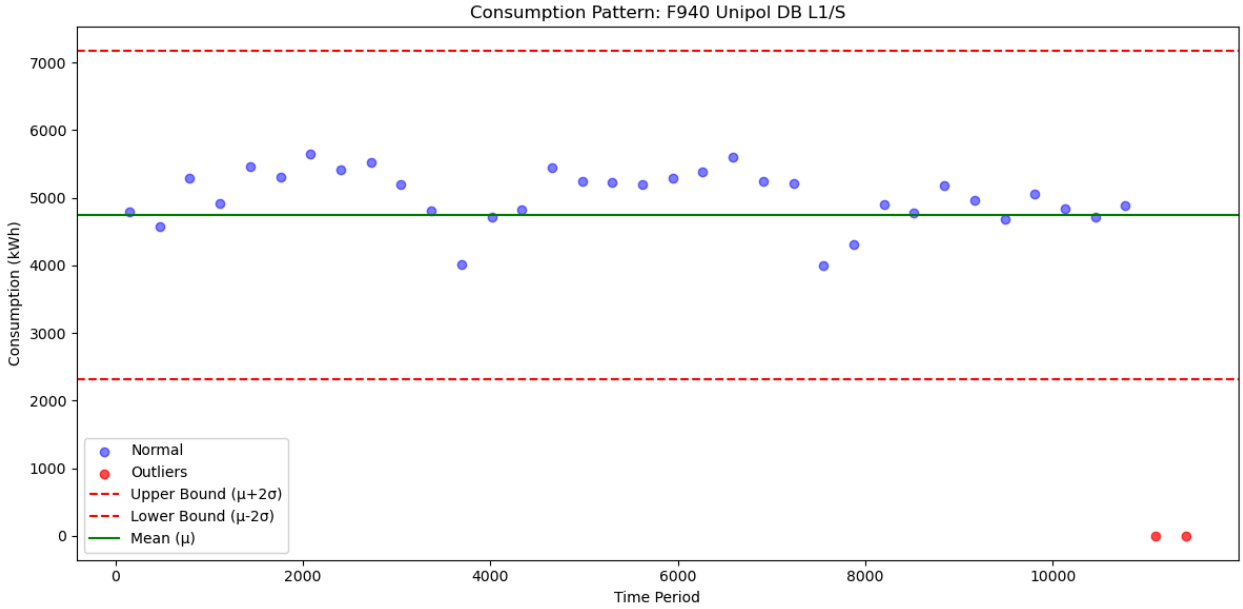

```
Statistics for F940 Unipol DB GYM/N:
Mean: 6198.32
Standard Deviation: 1679.62
Upper Bound: 9557.57
Lower Bound: 2839.08
Number of outliers: 2
```



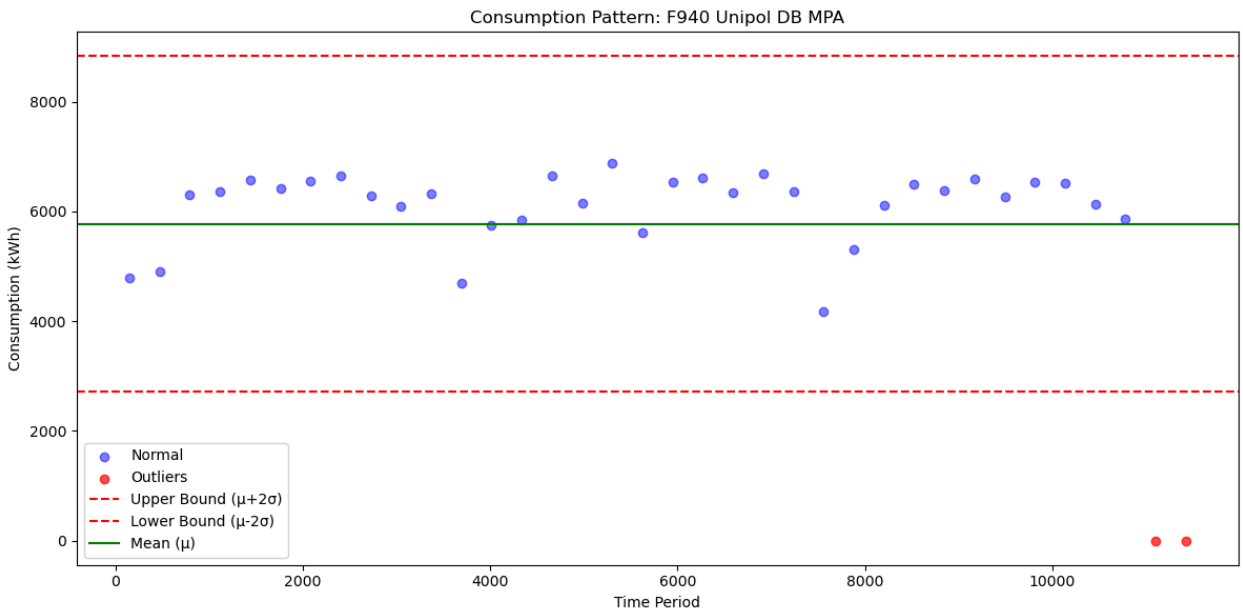

```
Statistics for F940 Unipol DB GYM/S:
Mean: 6278.26
Standard Deviation: 1683.90
Upper Bound: 9646.06
Lower Bound: 2910.46
Number of outliers: 2
```



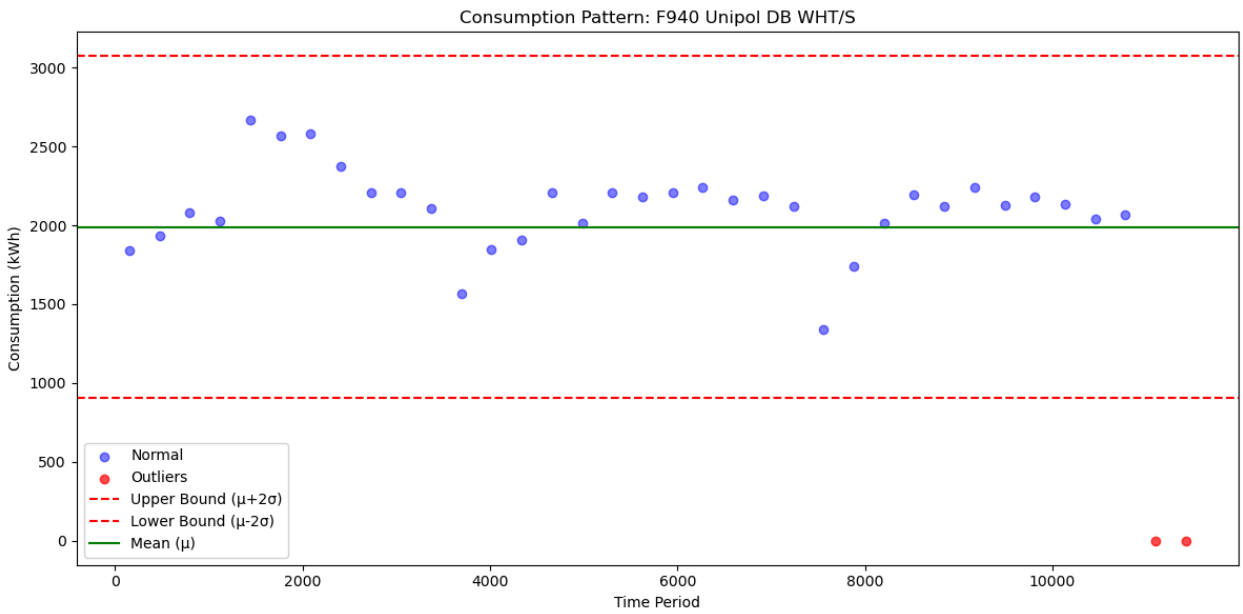

```
Statistics for F940 Unipol DB L1/S:
Mean: 4741.62
Standard Deviation: 1214.45
Upper Bound: 7170.53
Lower Bound: 2312.71
Number of outliers: 2
```



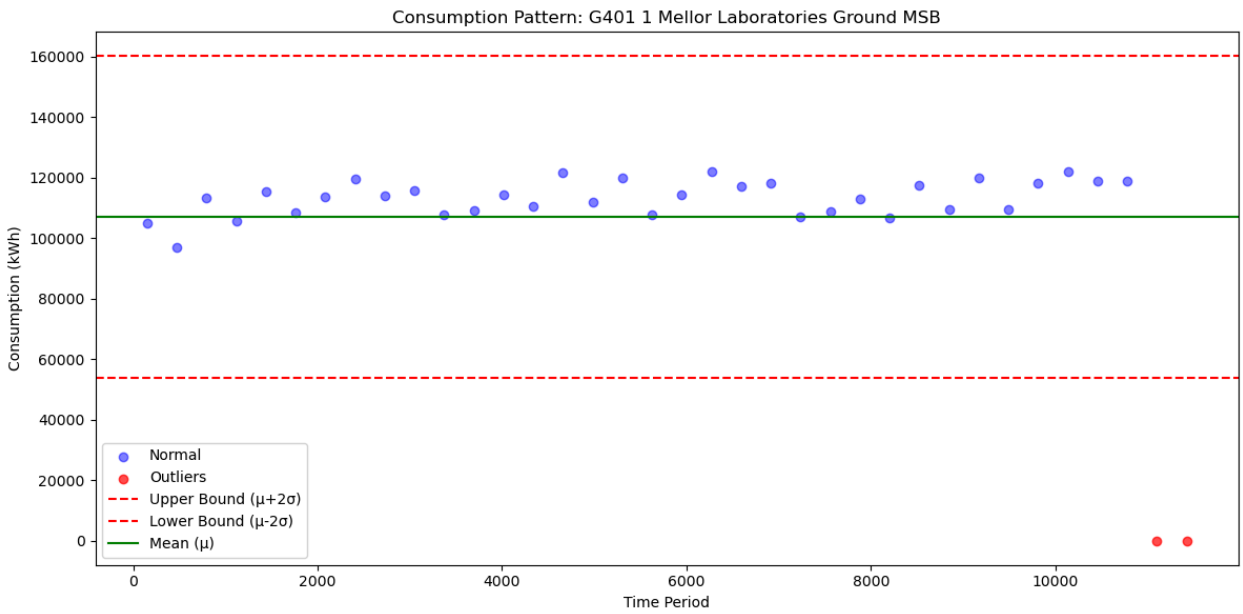

```
Statistics for F940 Unipol DB MPA:
Mean: 5769.35
Standard Deviation: 1529.14
Upper Bound: 8827.62
Lower Bound: 2711.07
Number of outliers: 2
```



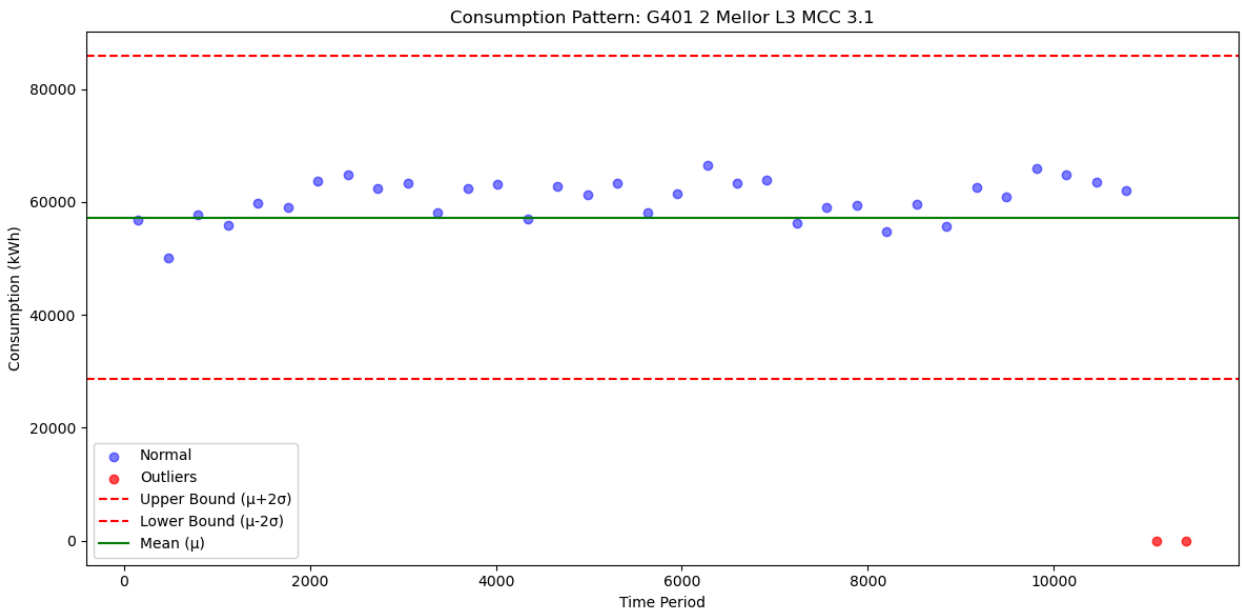

```
Statistics for F940 Unipol DB WHT/S:
Mean: 1988.79
Standard Deviation: 541.71
Upper Bound: 3072.21
Lower Bound: 905.37
Number of outliers: 2
```



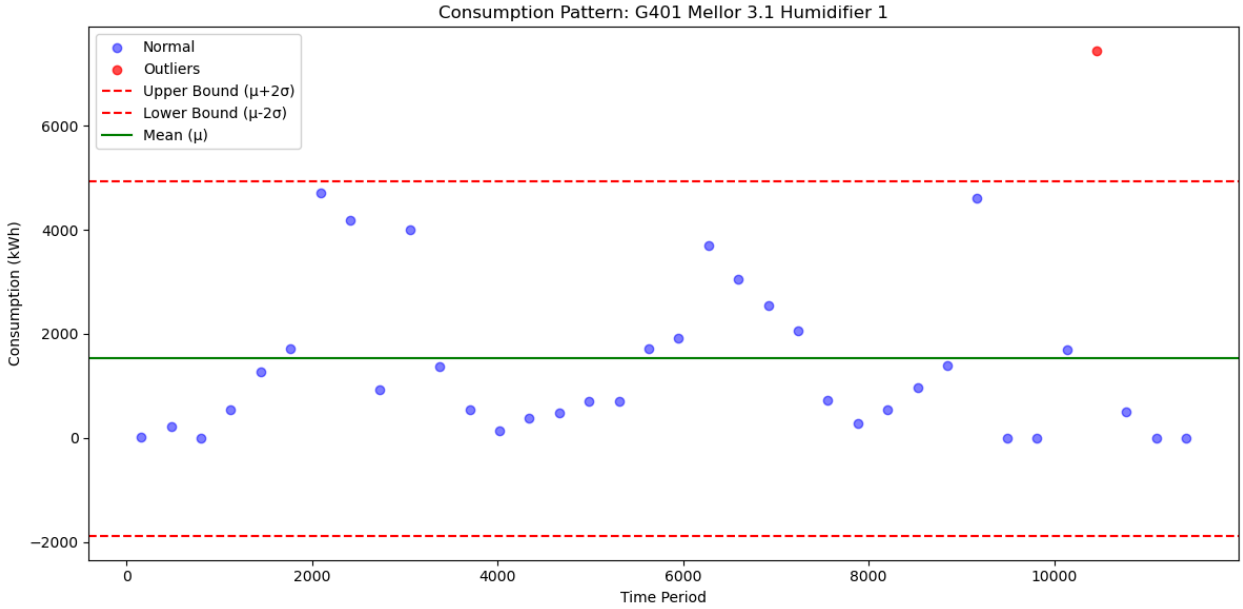

```
Statistics for G401 1 Mellor Laboratories Ground MSB:
Mean: 107062.81
Standard Deviation: 26575.23
Upper Bound: 160213.26
Lower Bound: 53912.36
Number of outliers: 2
```



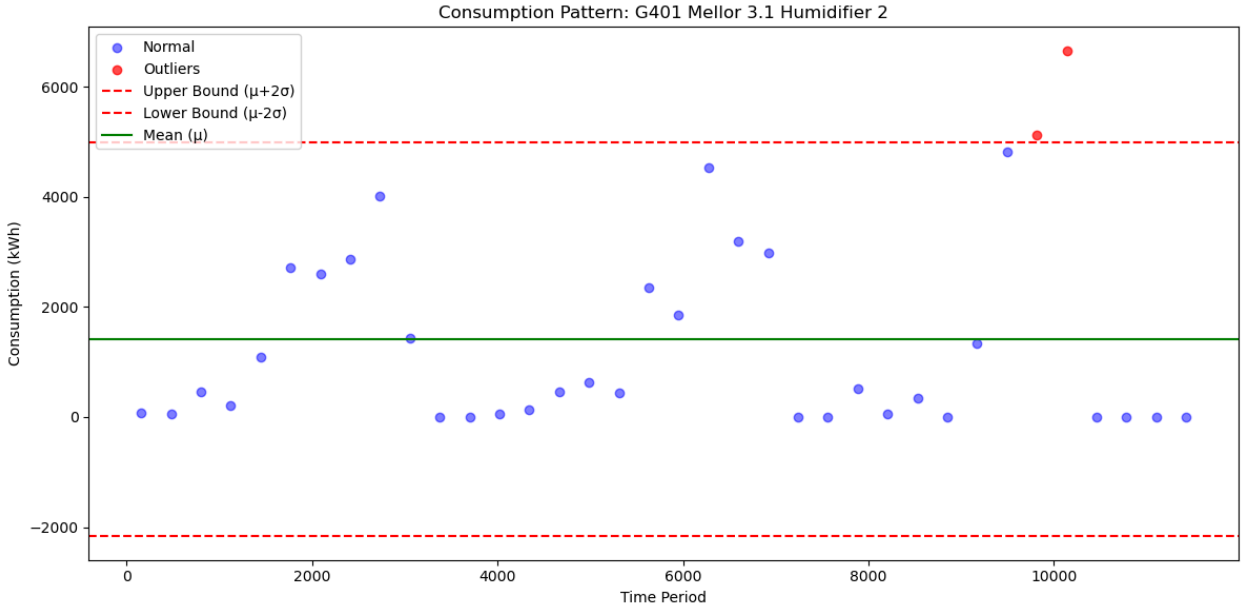

```
Statistics for G401 2 Mellor L3 MCC 3.1:
Mean: 57222.28
Standard Deviation: 14309.25
Upper Bound: 85840.79
Lower Bound: 28603.78
Number of outliers: 2
```



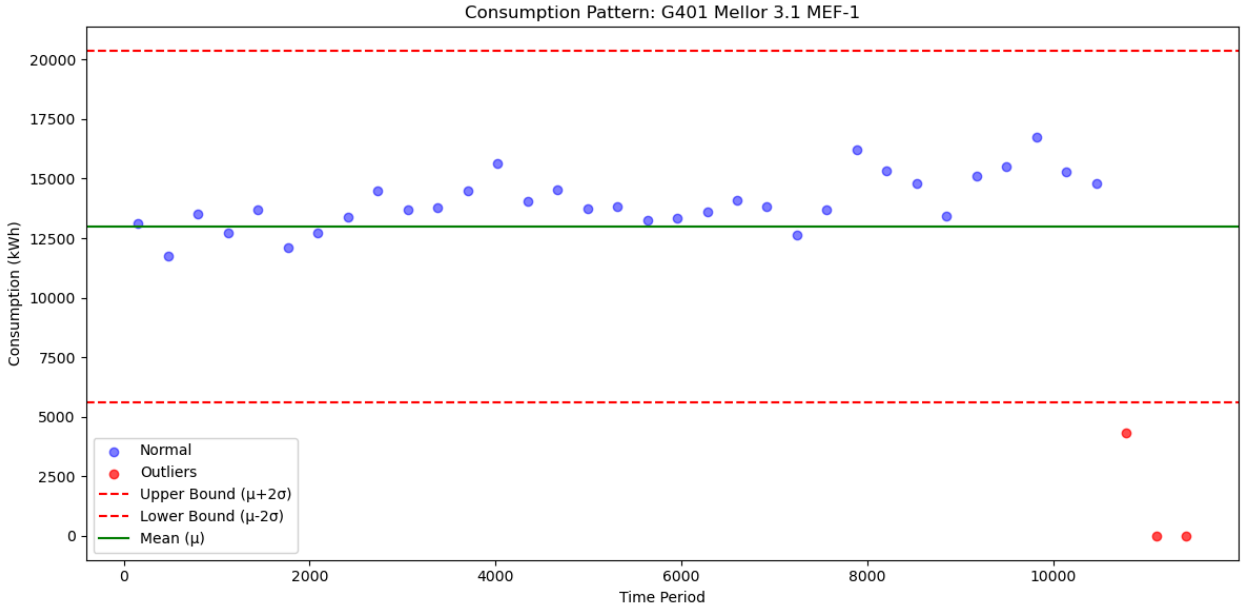

```
Statistics for G401 Mellor 3.1 Humidifier 1:
Mean: 1528.43
Standard Deviation: 1703.37
Upper Bound: 4935.16
Lower Bound: -1878.31
Number of outliers: 1
```



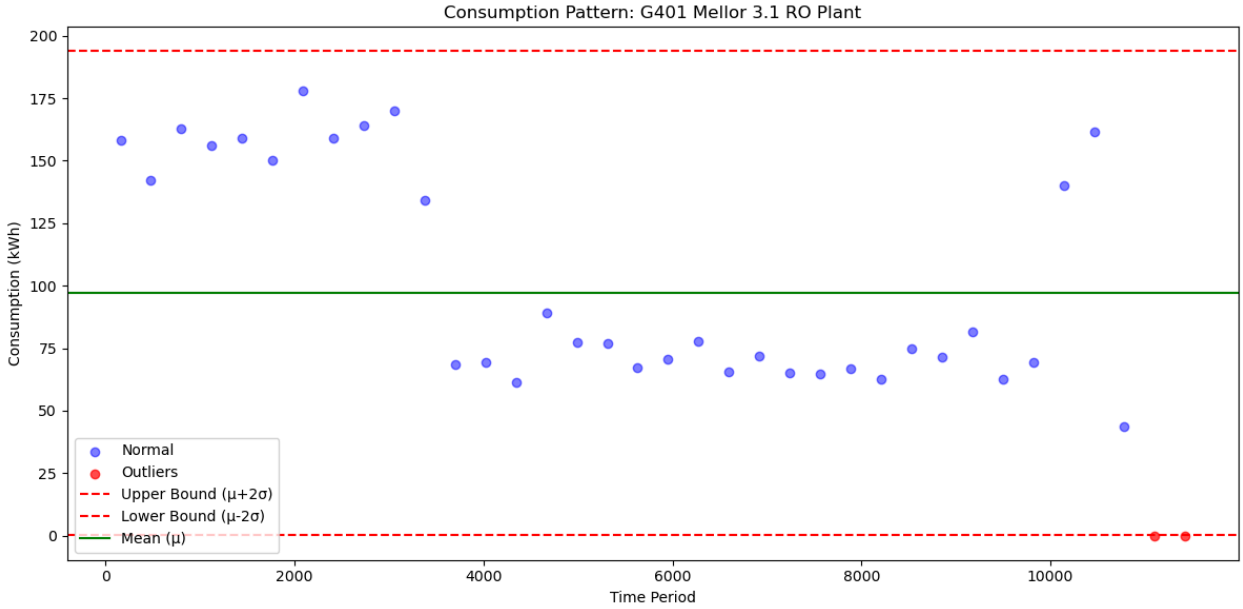

```
Statistics for G401 Mellor 3.1 Humidifier 2:
Mean: 1417.11
Standard Deviation: 1786.97
Upper Bound: 4991.05
Lower Bound: -2156.83
Number of outliers: 2
```



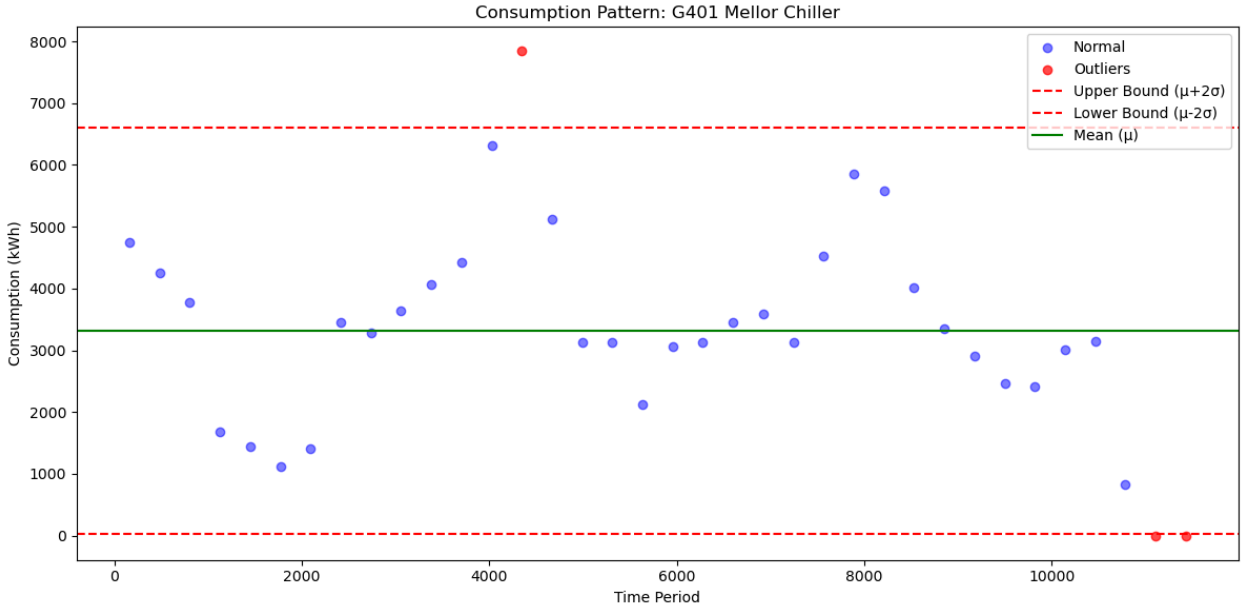

```
Statistics for G401 Mellor 3.1 MEF-1:
Mean: 12980.13
Standard Deviation: 3687.45
Upper Bound: 20355.04
Lower Bound: 5605.23
Number of outliers: 3
```



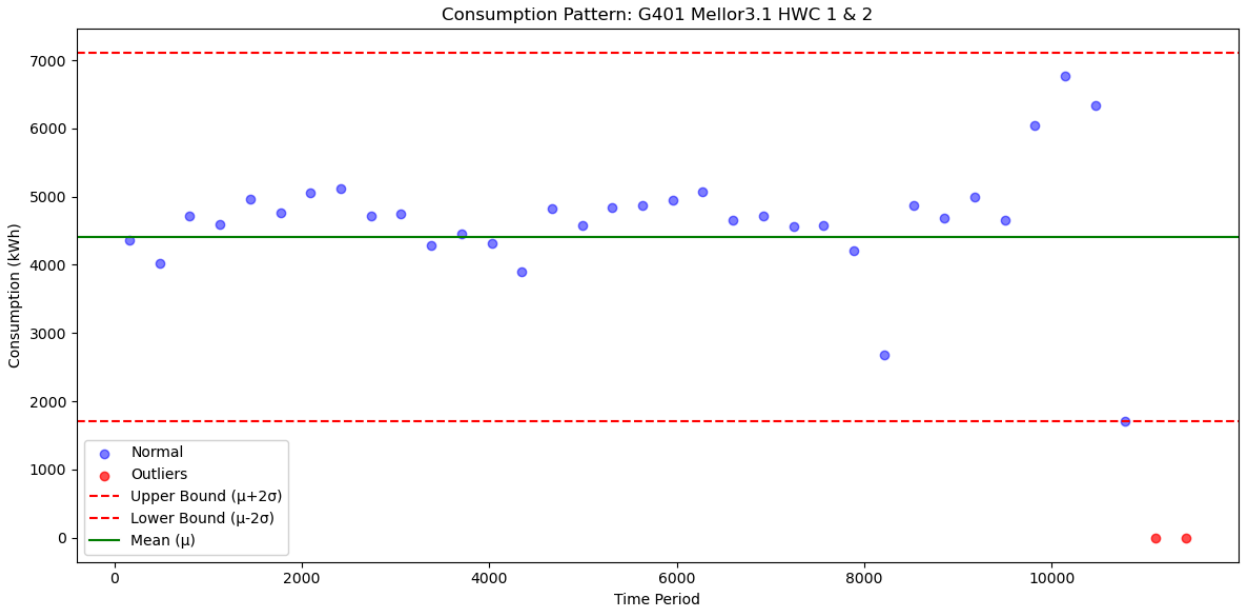

```
Statistics for G401 Mellor 3.1 RO Plant:
Mean: 97.10
Standard Deviation: 48.37
Upper Bound: 193.84
Lower Bound: 0.35
Number of outliers: 2
```



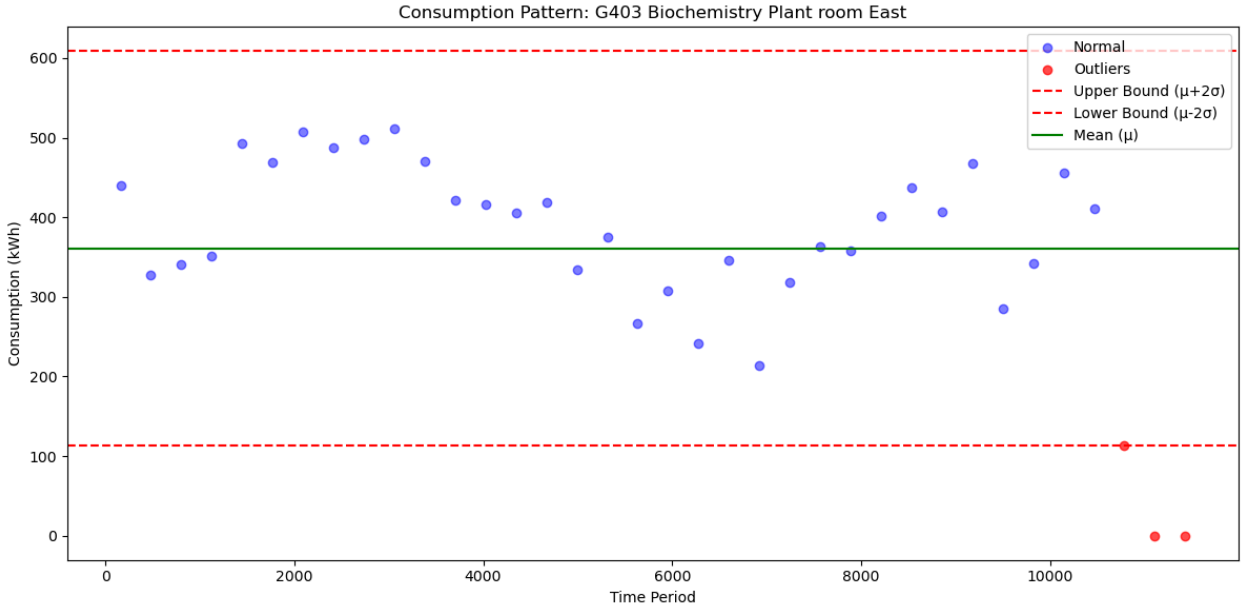

```
Statistics for G401 Mellor Chiller:
Mean: 3317.20
Standard Deviation: 1646.75
Upper Bound: 6610.70
Lower Bound: 23.70
Number of outliers: 3
```



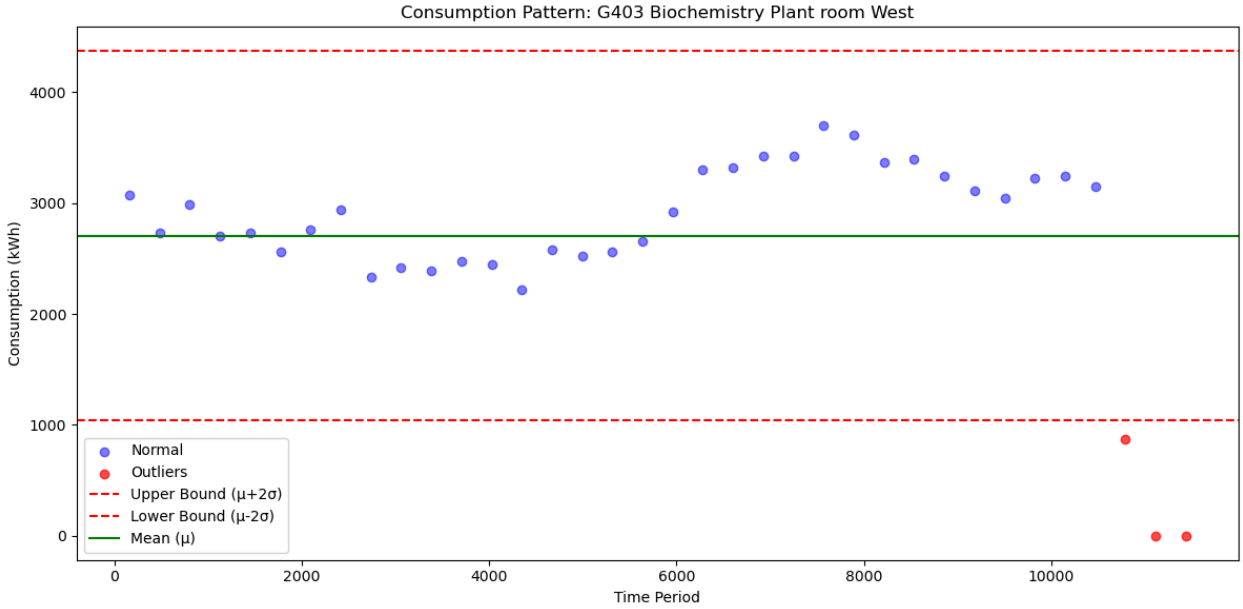

```
Statistics for G401 Mellor3.1 HWC 1 & 2:
Mean: 4406.63
Standard Deviation: 1348.39
Upper Bound: 7103.41
Lower Bound: 1709.85
Number of outliers: 2
```



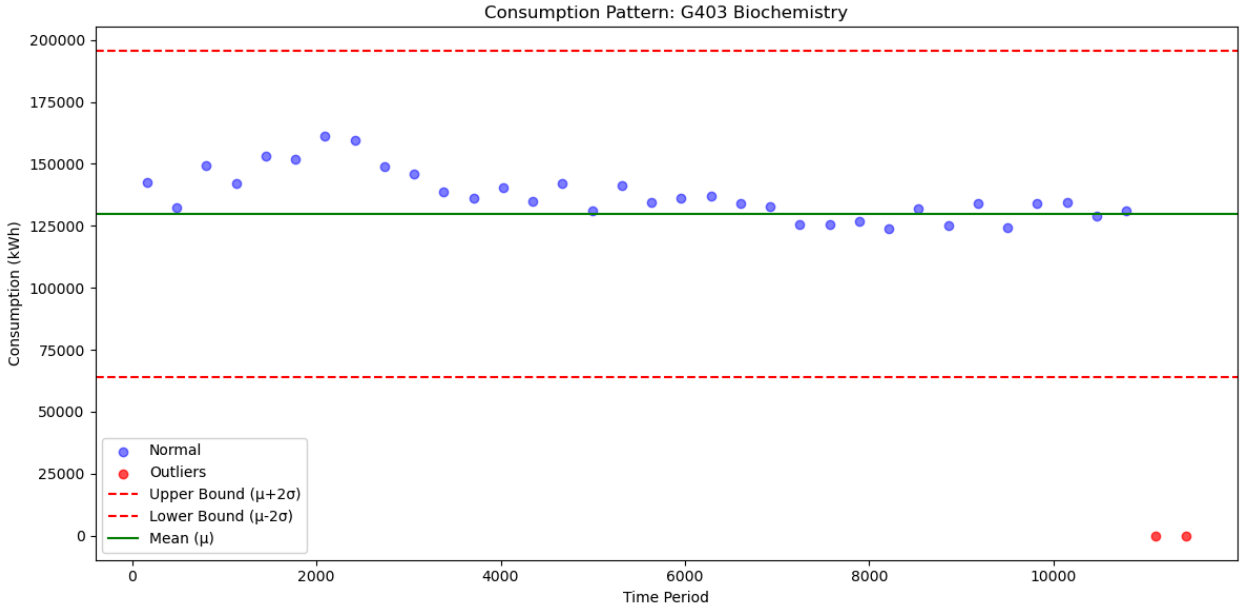

```
Statistics for G403 Biochemistry Plant room East:
Mean: 361.21
Standard Deviation: 123.78
Upper Bound: 608.77
Lower Bound: 113.66
Number of outliers: 3
```



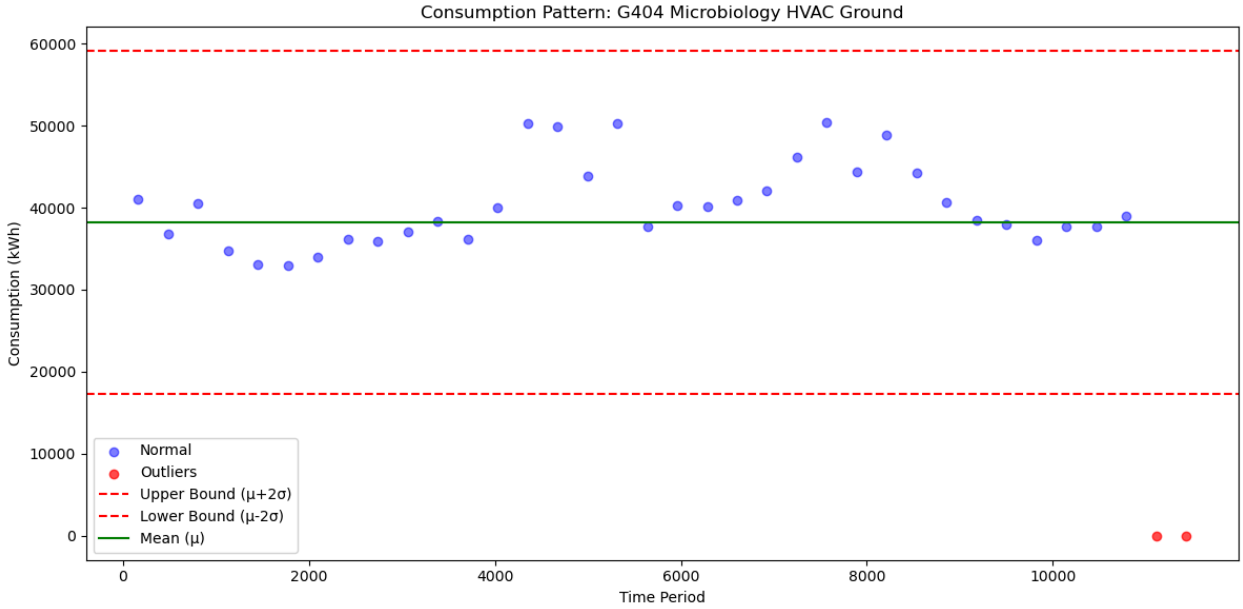

```
Statistics for G403 Biochemistry Plant room West:
Mean: 2706.03
Standard Deviation: 832.01
Upper Bound: 4370.04
Lower Bound: 1042.02
Number of outliers: 3
```



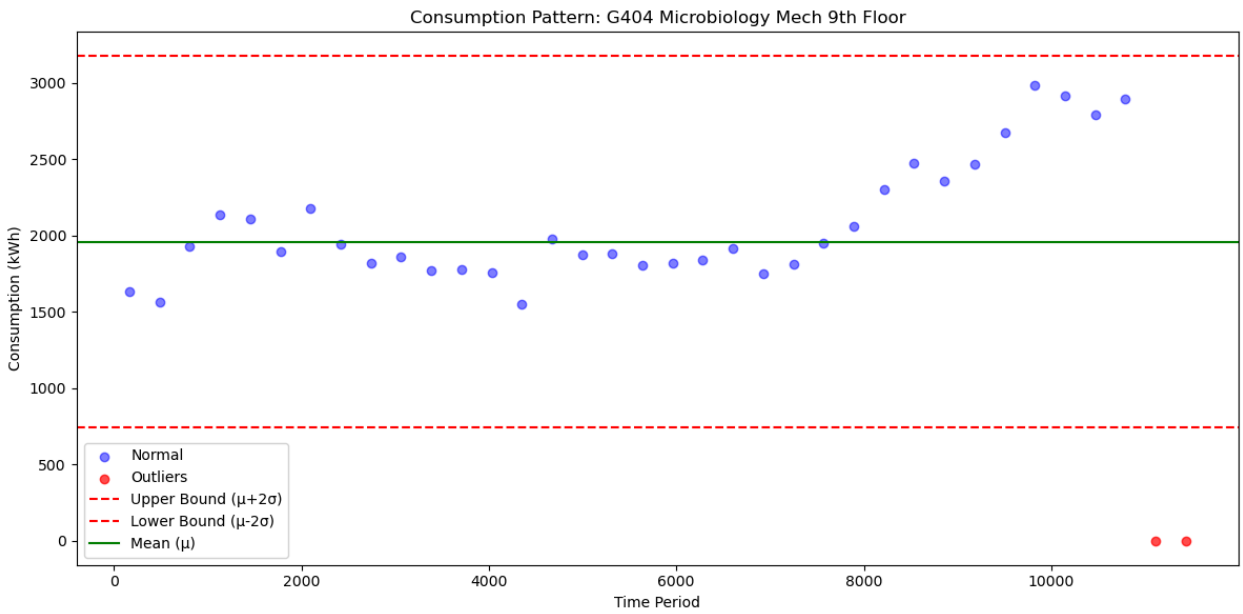

```
Statistics for G403 Biochemistry:
Mean: 129775.31
Standard Deviation: 32839.98
Upper Bound: 195455.27
Lower Bound: 64095.34
Number of outliers: 2
```



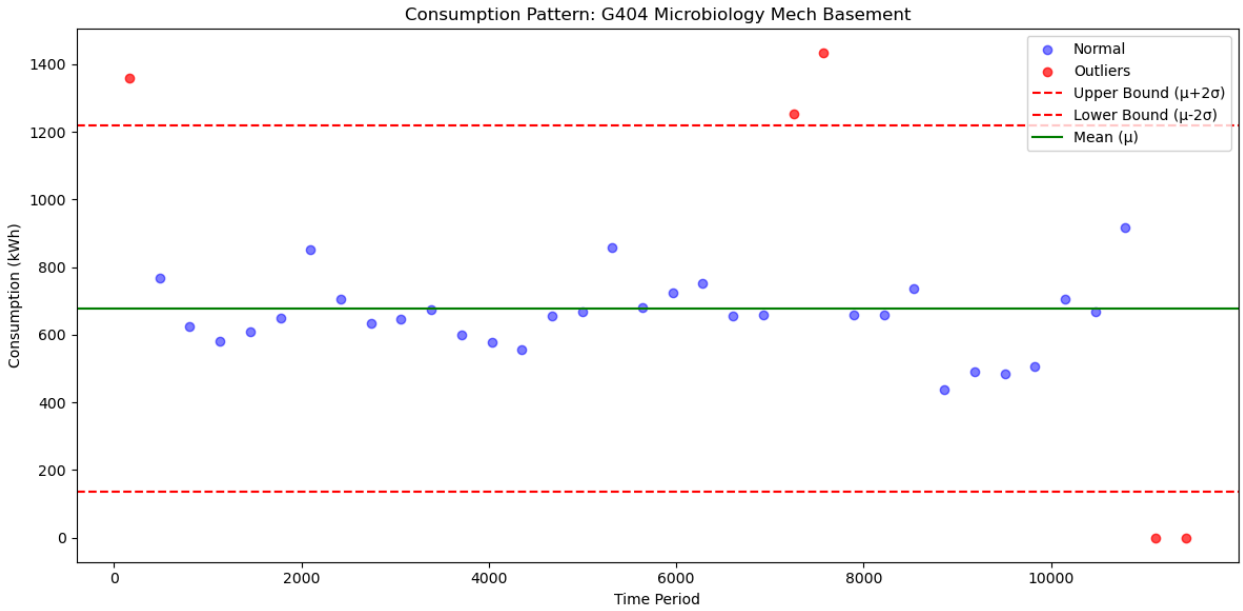

```
Statistics for G404 Microbiology HVAC Ground:
Mean: 38170.01
Standard Deviation: 10461.96
Upper Bound: 59093.93
Lower Bound: 17246.08
Number of outliers: 2
```




```
Statistics for G404 Microbiology Mech 9th Floor:
Mean: 1956.63
Standard Deviation: 608.58
Upper Bound: 3173.79
Lower Bound: 739.47
Number of outliers: 2
```

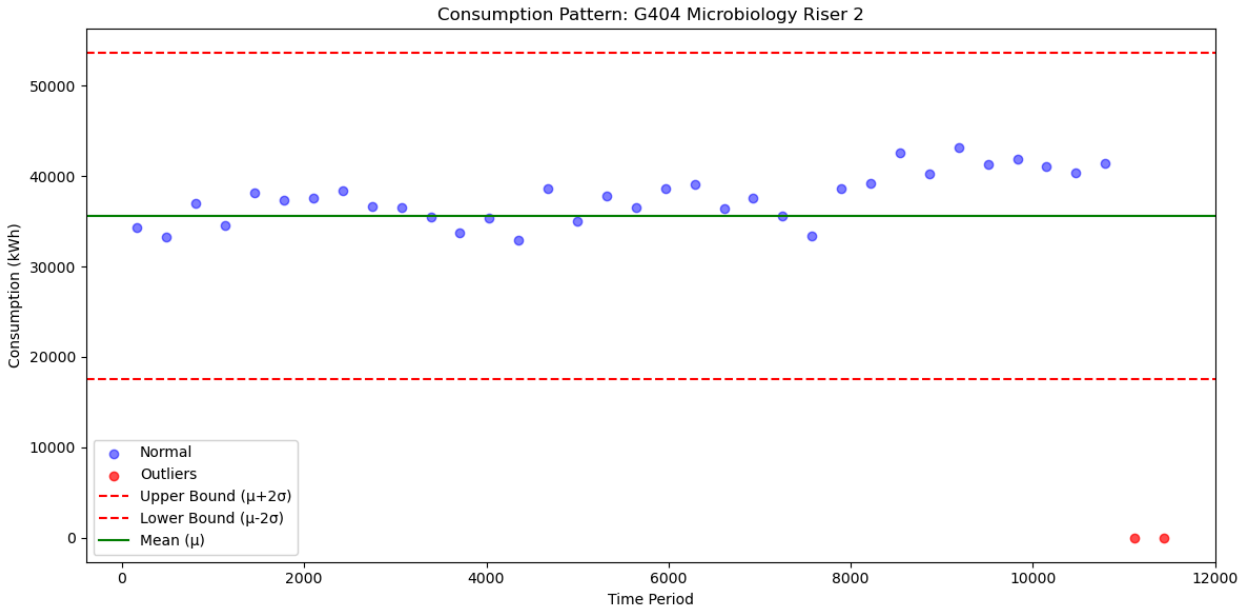



```
Statistics for G404 Microbiology Mech Basement:
Mean: 678.99
Standard Deviation: 270.77
Upper Bound: 1220.53
Lower Bound: 137.45
Number of outliers: 5
```

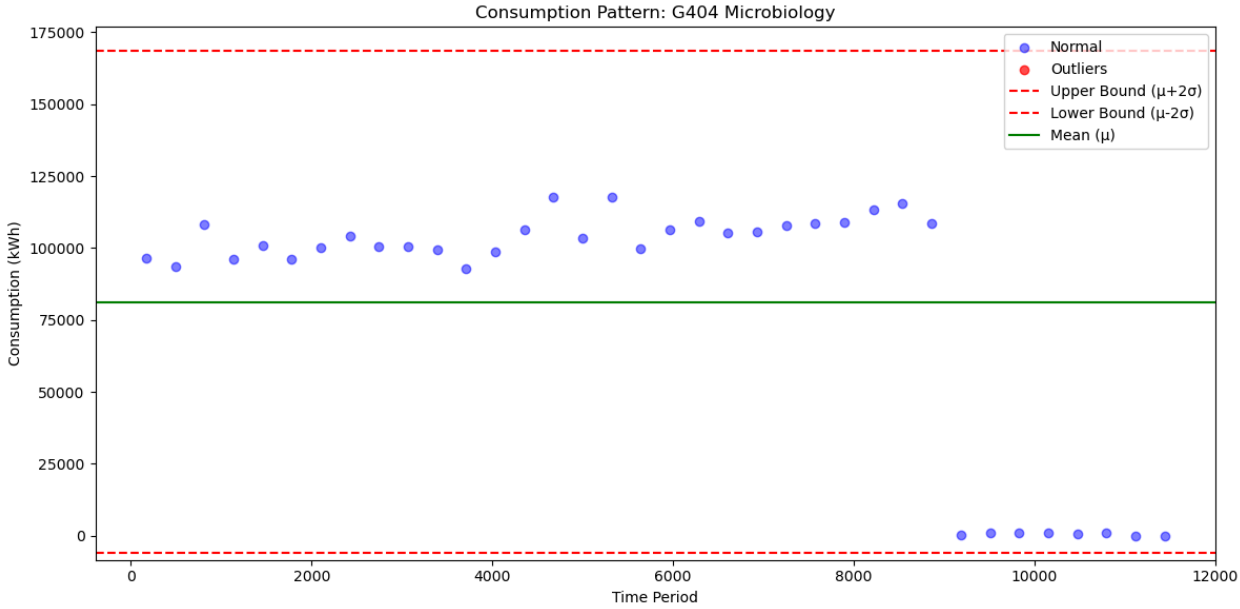



```
Statistics for G404 Microbiology Riser 1:
Mean: 1203.90
Standard Deviation: 441.78
Upper Bound: 2087.47
Lower Bound: 320.34
Number of outliers: 2
```

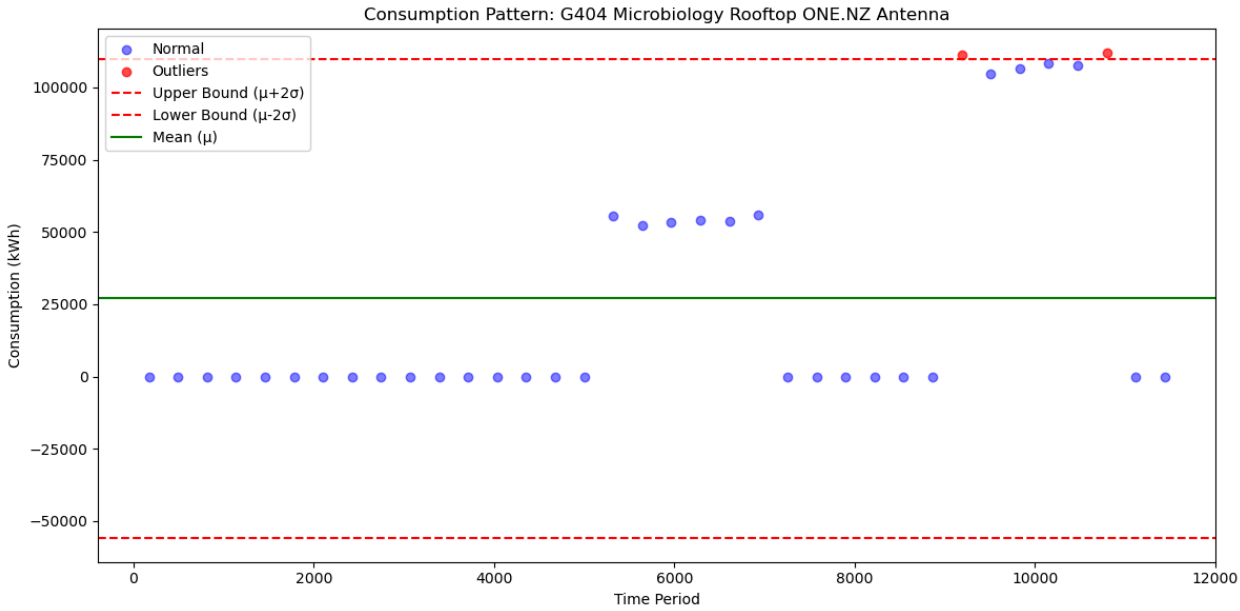
```
Statistics for G404 Microbiology Riser 2:
Mean: 35543.63
Standard Deviation: 9024.19
Upper Bound: 53592.01
Lower Bound: 17495.26
Number of outliers: 2
```



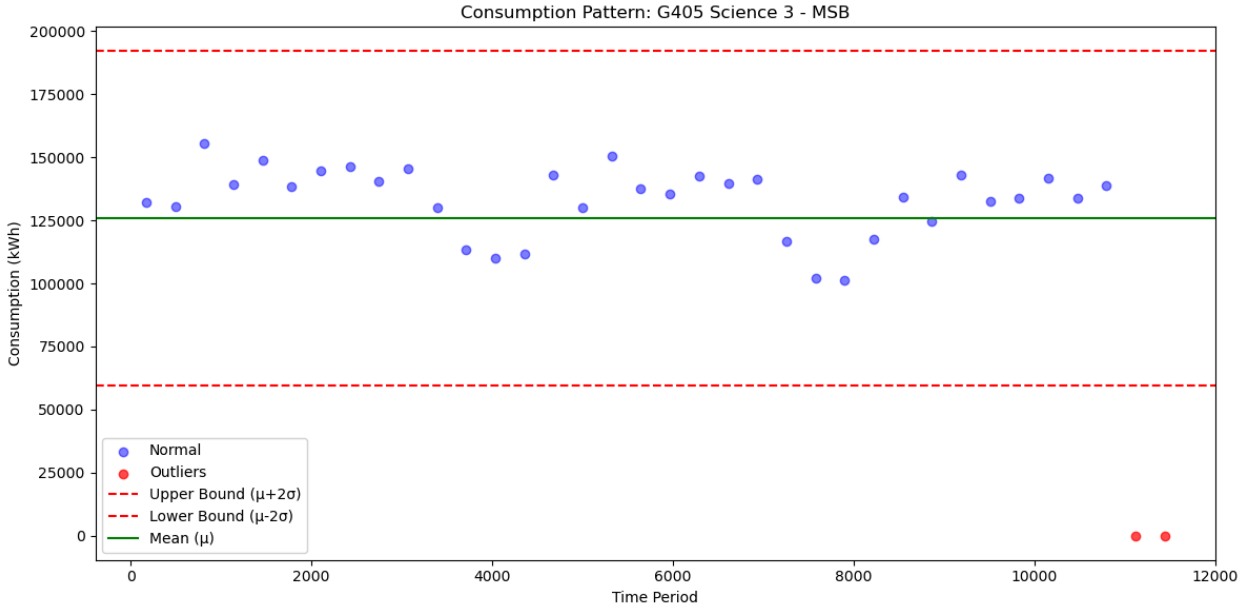

```
Statistics for G404 Microbiology:
Mean: 81301.31
Standard Deviation: 43580.37
Upper Bound: 168462.05
Lower Bound: -5859.42
Number of outliers: 0
```



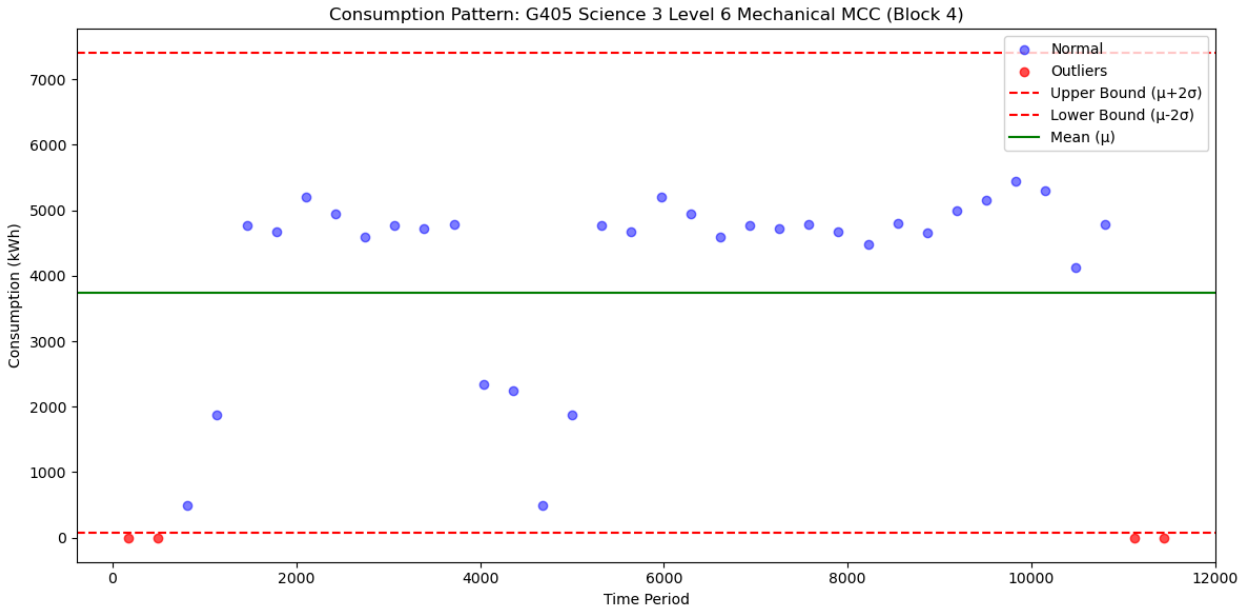

```
Statistics for G404 Microbiology Rooftop ONE.NZ Antenna:
Mean: 27111.30
Standard Deviation: 41428.57
Upper Bound: 109968.43
Lower Bound: -55745.84
Number of outliers: 2
```



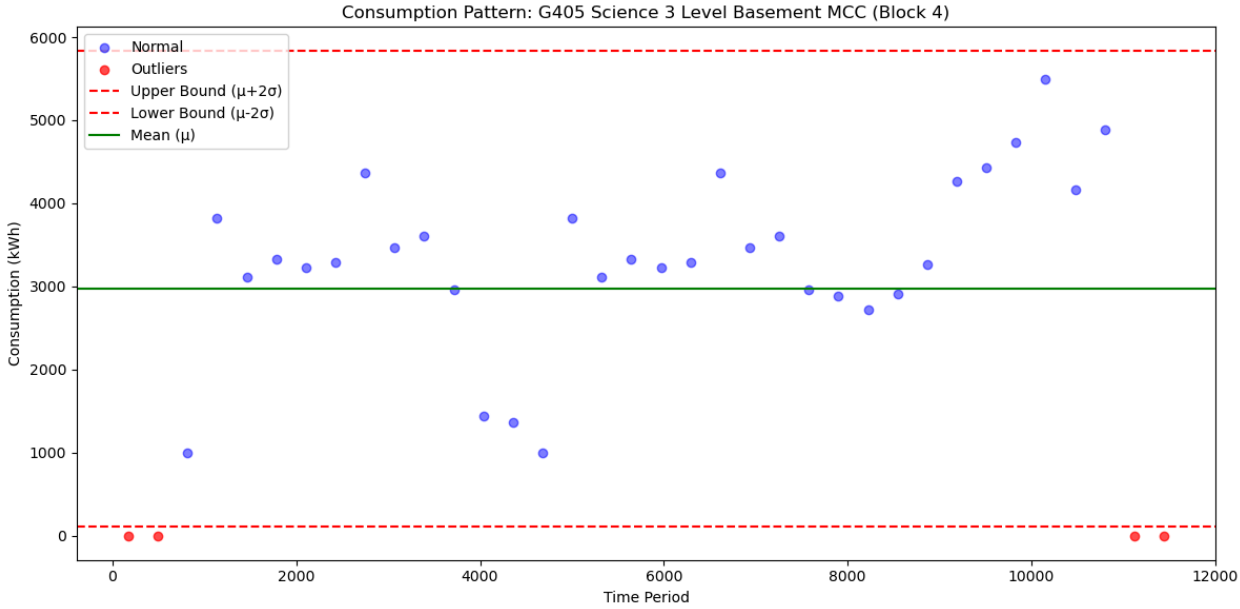

```
Statistics for G405 Science 3 - MSB:
Mean: 125763.04
Standard Deviation: 33179.37
Upper Bound: 192121.79
Lower Bound: 59404.29
Number of outliers: 2
```



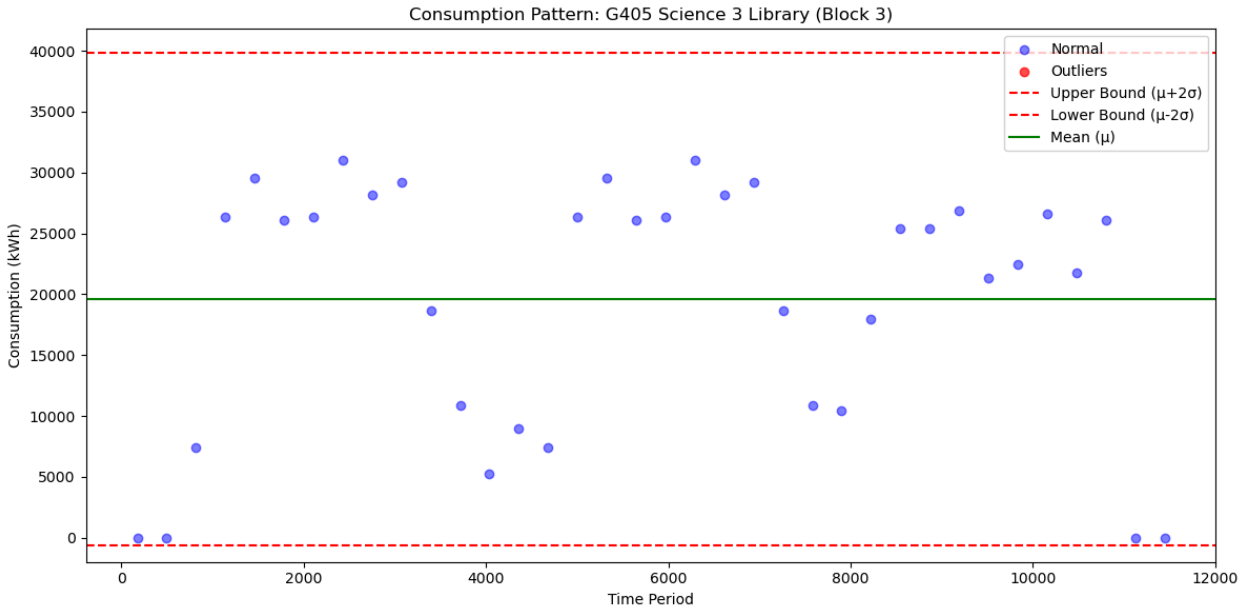

```
Statistics for G405 Science 3 Level 6 Mechanical MCC (Block 4):
Mean: 3739.87
Standard Deviation: 1828.78
Upper Bound: 7397.44
Lower Bound: 82.30
Number of outliers: 4
```



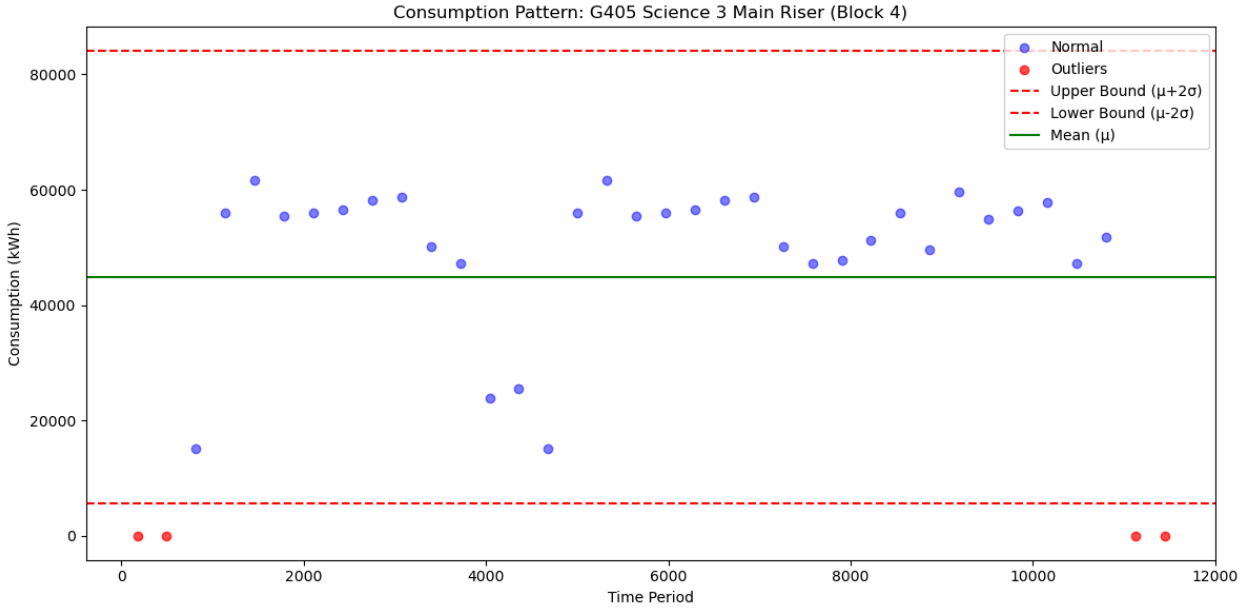

```
Statistics for G405 Science 3 Level Basement MCC (Block 4):
Mean: 2968.19
Standard Deviation: 1431.46
Upper Bound: 5831.11
Lower Bound: 105.27
Number of outliers: 4
```



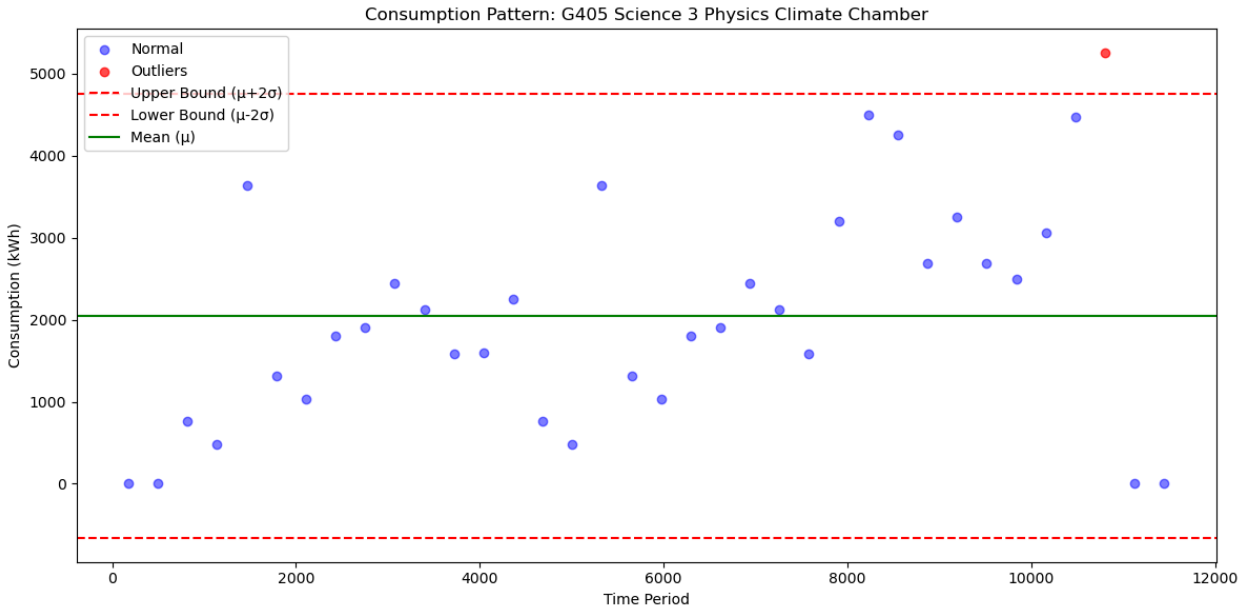

```
Statistics for G405 Science 3 Library (Block 3):
Mean: 19605.27
Standard Deviation: 10102.67
Upper Bound: 39810.61
Lower Bound: -600.06
Number of outliers: 0
```



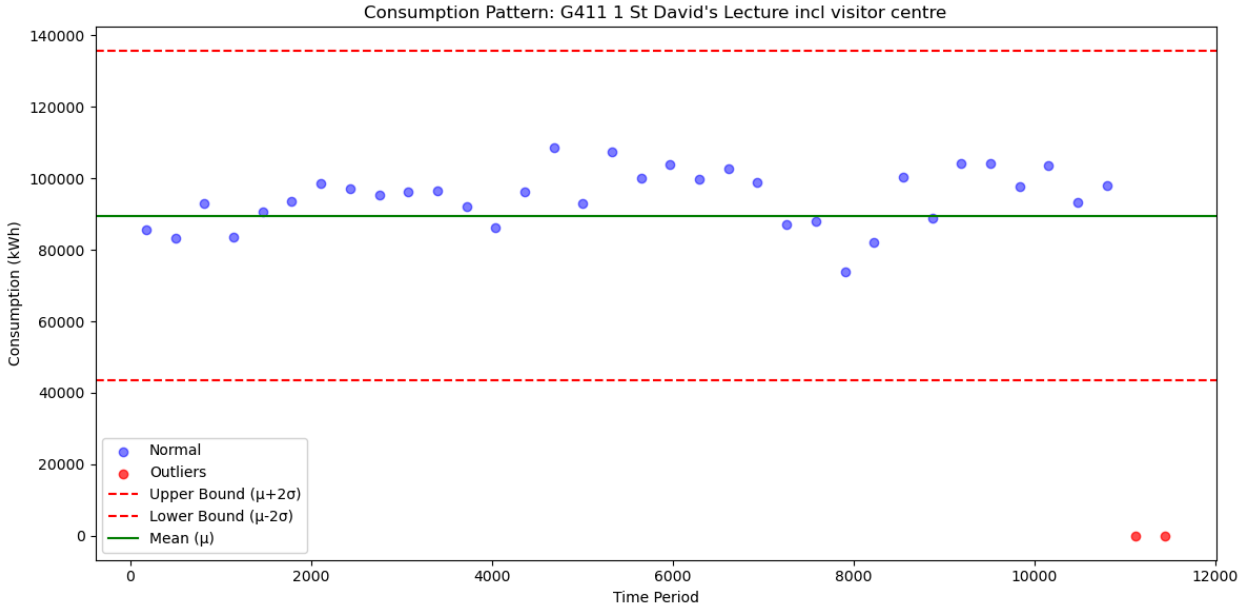

```
Statistics for G405 Science 3 Main Riser (Block 4):
Mean: 44777.09
Standard Deviation: 19630.29
Upper Bound: 84037.66
Lower Bound: 5516.52
Number of outliers: 4
```



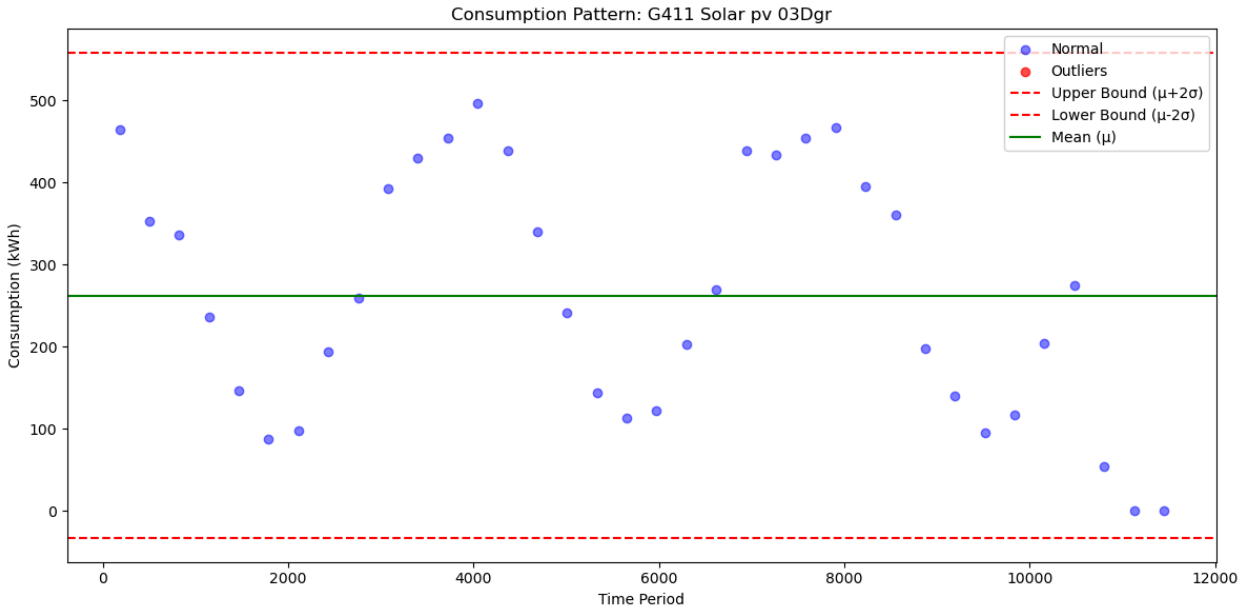

```
Statistics for G405 Science 3 Physics Climate Chamber:
Mean: 2050.61
Standard Deviation: 1354.12
Upper Bound: 4758.85
Lower Bound: -657.63
Number of outliers: 1
```



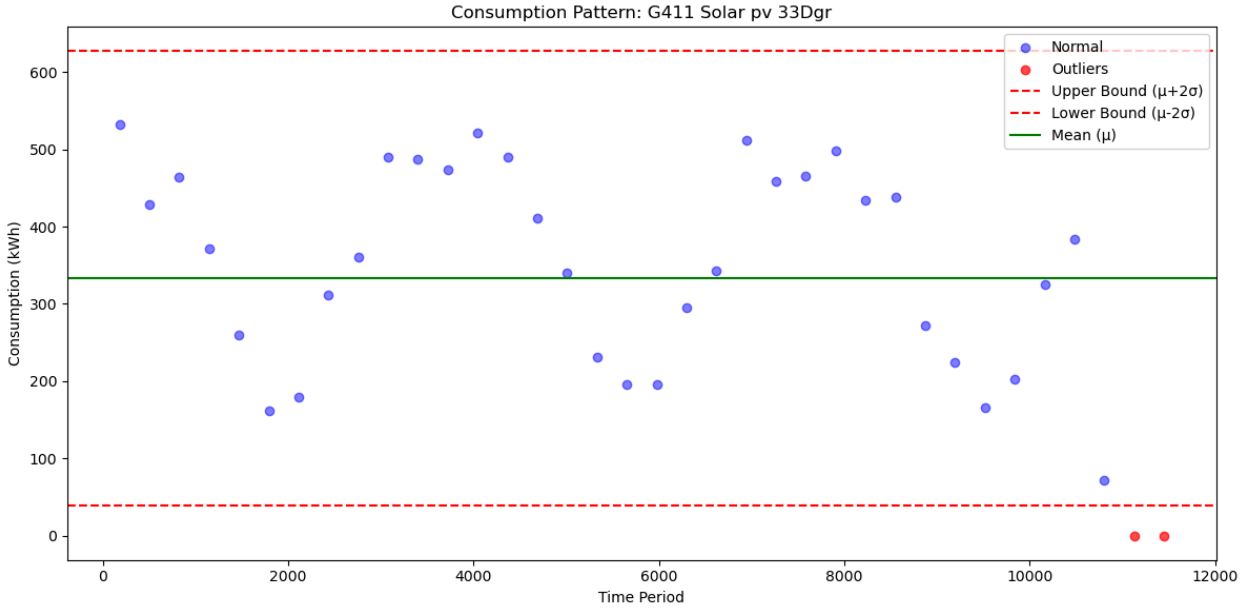

```
Statistics for G411 1 St David's Lecture incl visitor centre:
Mean: 89570.86
Standard Deviation: 23024.45
Upper Bound: 135619.75
Lower Bound: 43521.97
Number of outliers: 2
```



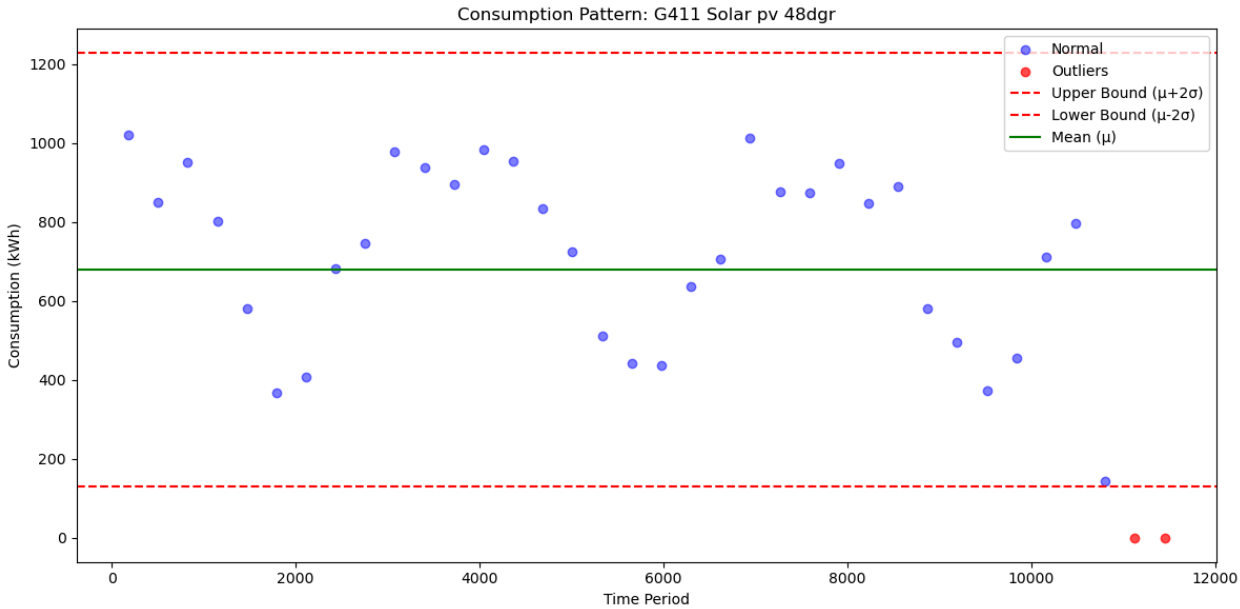

```
Statistics for G411 Solar pv 03Dgr:
Mean: 262.64
Standard Deviation: 147.58
Upper Bound: 557.80
Lower Bound: -32.52
Number of outliers: 0
```



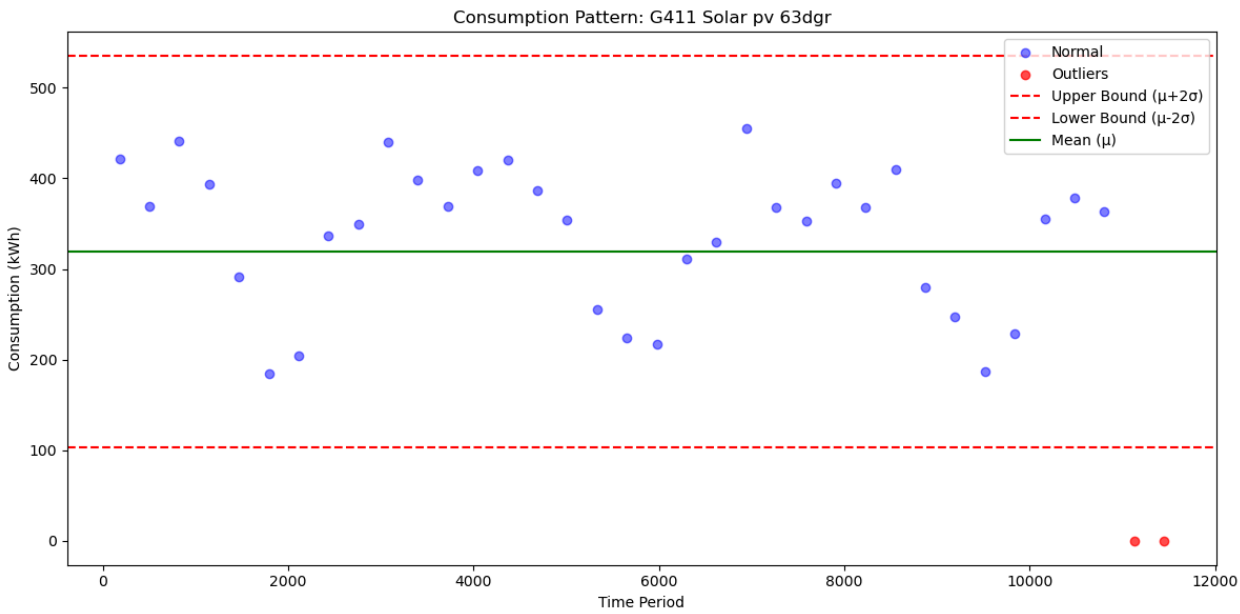

```
Statistics for G411 Solar pv 33Dgr:
Mean: 333.14
Standard Deviation: 146.95
Upper Bound: 627.03
Lower Bound: 39.25
Number of outliers: 2
```



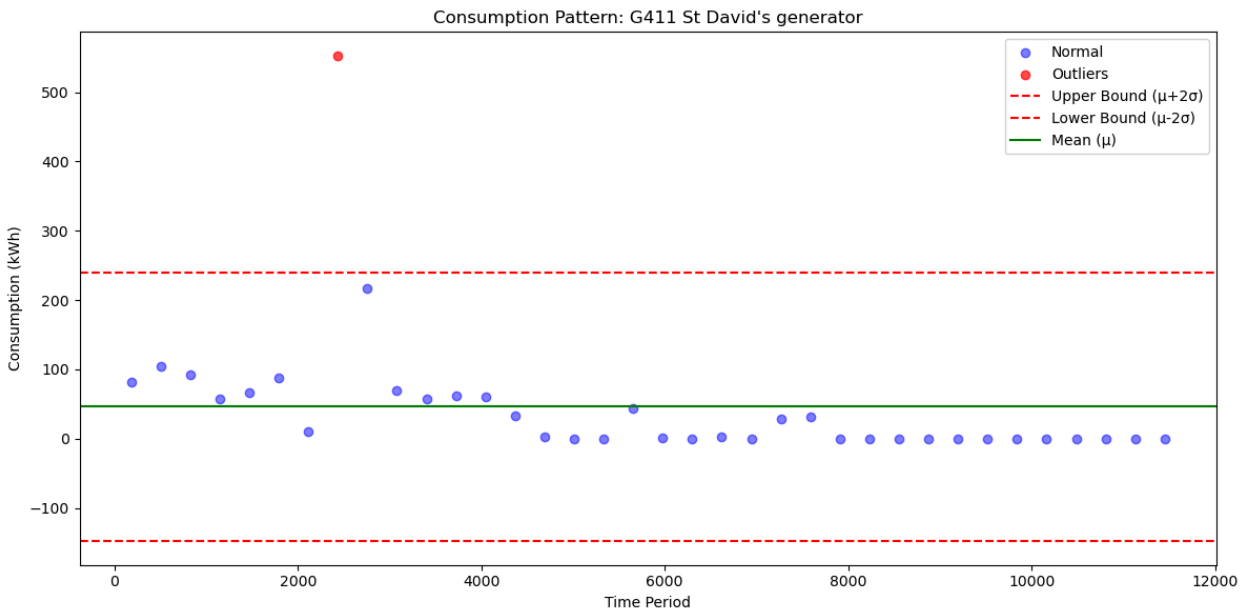

```
Statistics for G411 Solar pv 48dgr:
Mean: 678.90
Standard Deviation: 274.13
Upper Bound: 1227.17
Lower Bound: 130.64
Number of outliers: 2
```



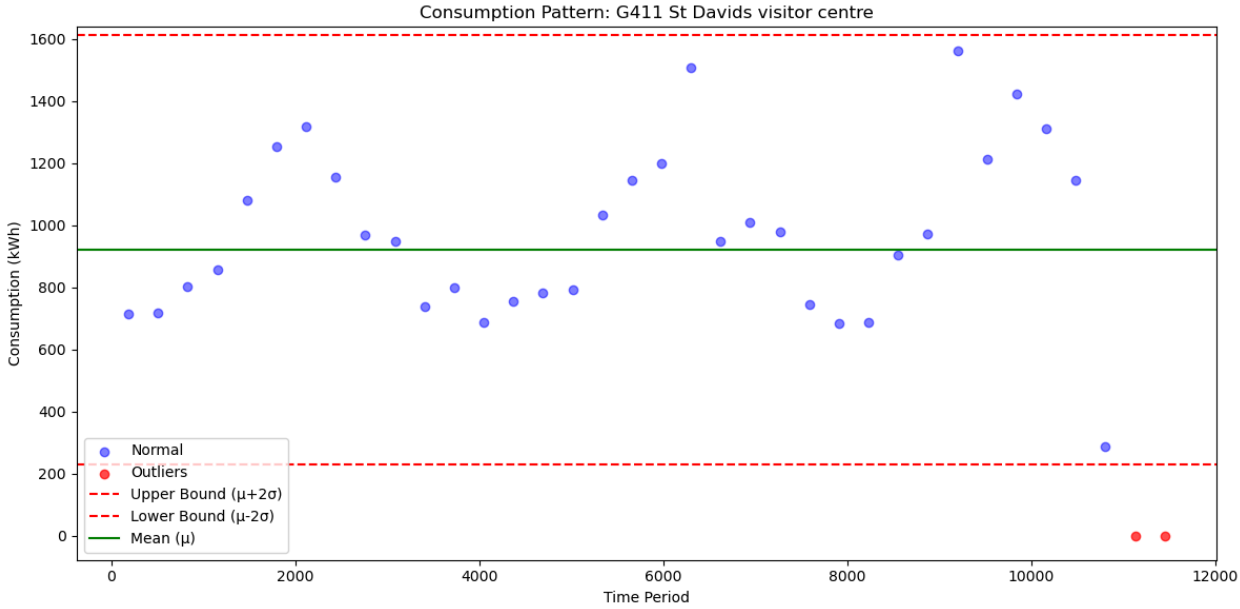

```
Statistics for G411 Solar pv 63dgr:
Mean: 319.15
Standard Deviation: 107.89
Upper Bound: 534.93
Lower Bound: 103.37
Number of outliers: 2
```



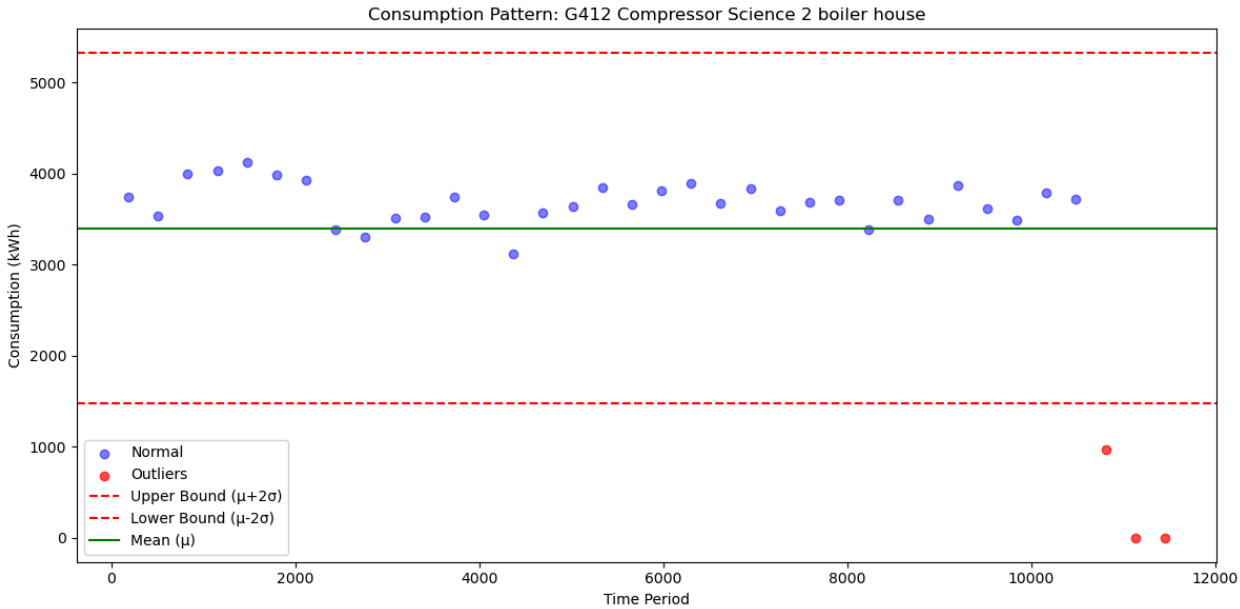

```
Statistics for G411 St David's generator:
Mean: 46.12
Standard Deviation: 96.75
Upper Bound: 239.63
Lower Bound: -147.39
Number of outliers: 1
```



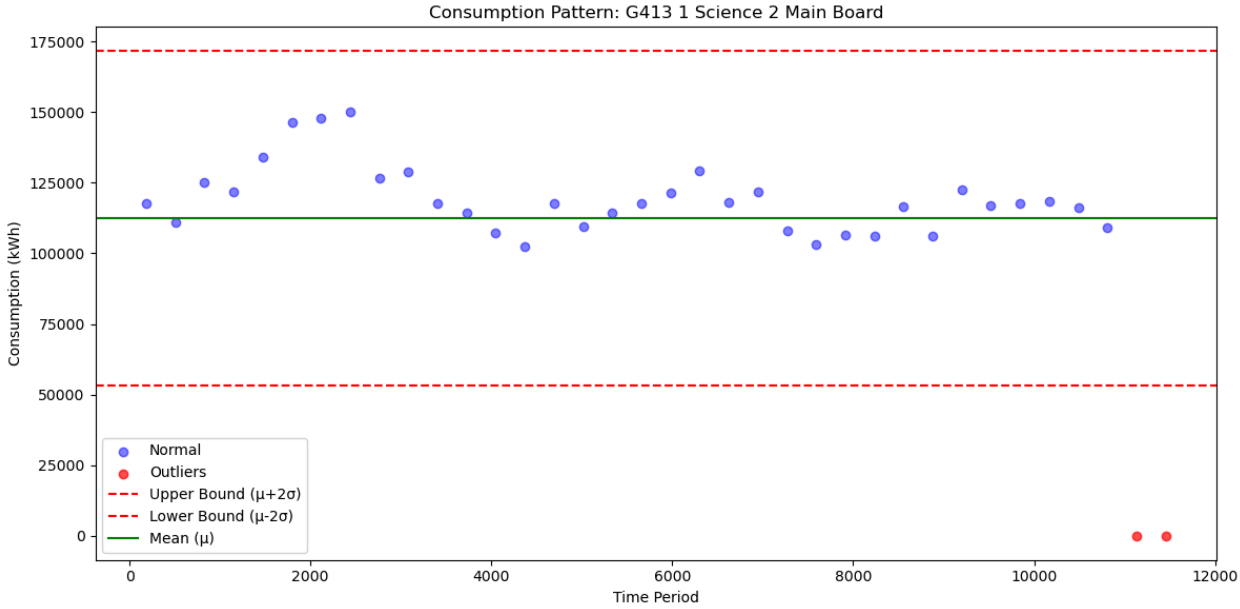

```
Statistics for G411 St Davids visitor centre:
Mean: 919.60
Standard Deviation: 345.84
Upper Bound: 1611.29
Lower Bound: 227.91
Number of outliers: 2
```



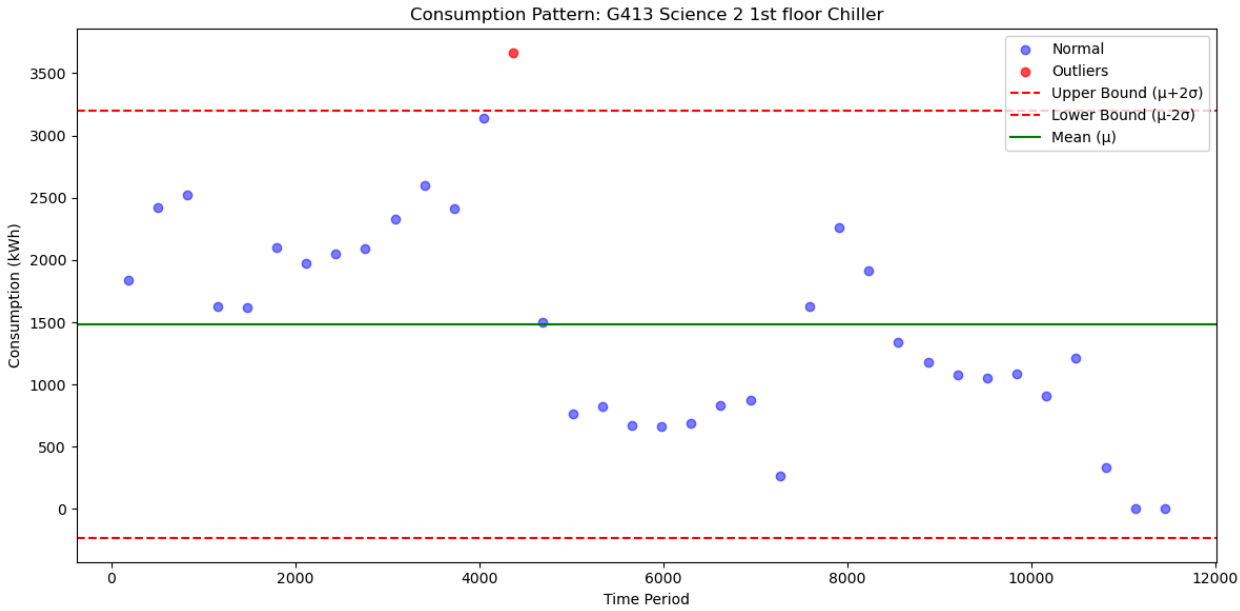

```
Statistics for G412 Compressor Science 2 boiler house:
Mean: 3399.39
Standard Deviation: 960.14
Upper Bound: 5319.66
Lower Bound: 1479.11
Number of outliers: 3
```



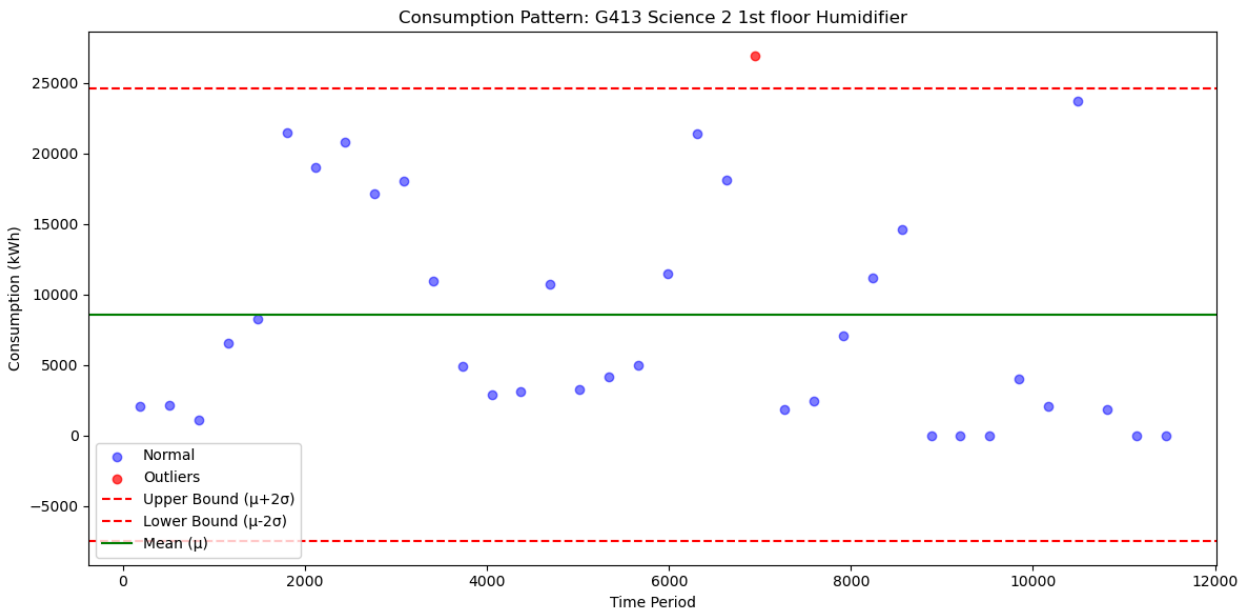

```
Statistics for G413 1 Science 2 Main Board:
Mean: 112476.03
Standard Deviation: 29575.13
Upper Bound: 171626.29
Lower Bound: 53325.77
Number of outliers: 2
```



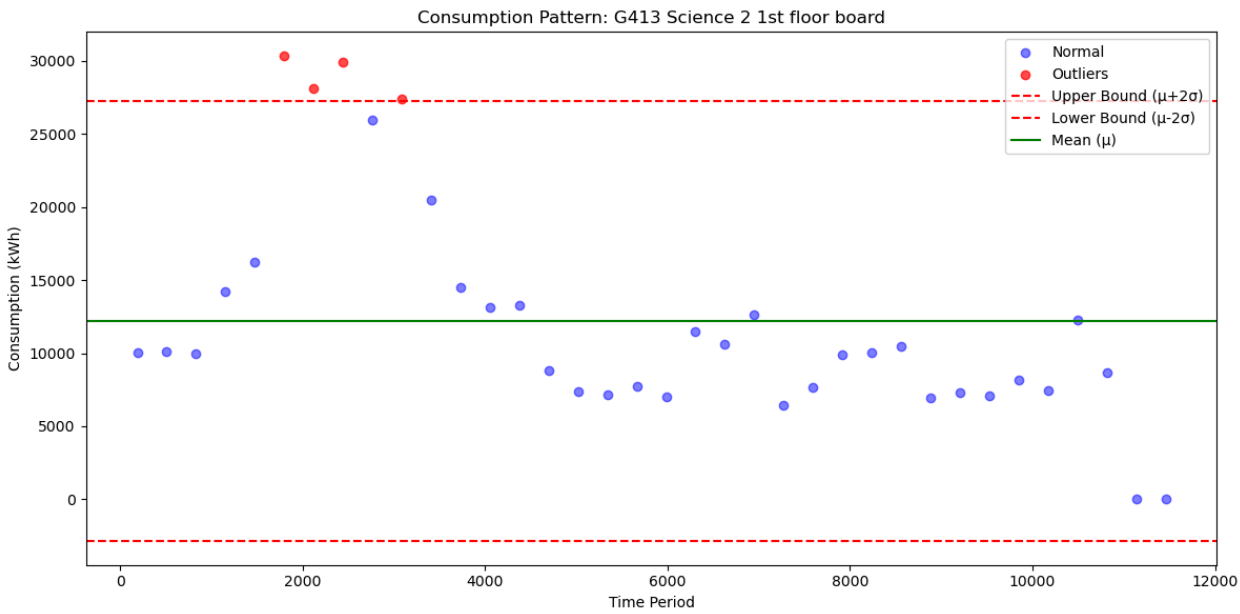

```
Statistics for G413 Science 2 1st floor Chiller:
Mean: 1485.25
Standard Deviation: 858.07
Upper Bound: 3201.39
Lower Bound: -230.88
Number of outliers: 1
```



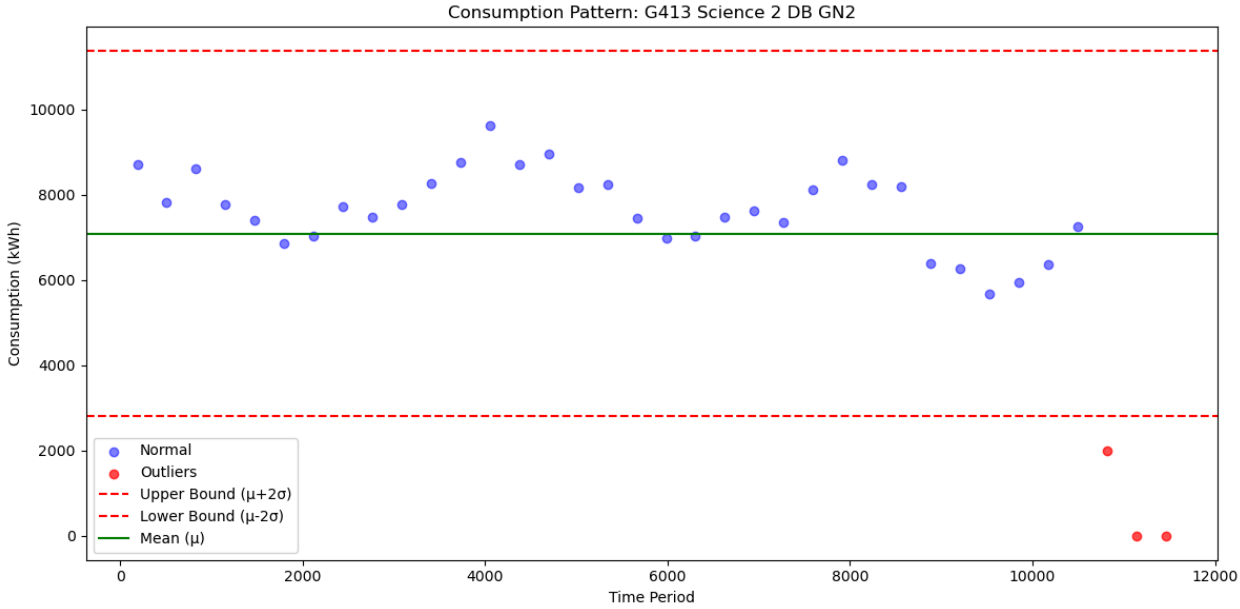

```
Statistics for G413 Science 2 1st floor Humidifier:
Mean: 8560.71
Standard Deviation: 8009.11
Upper Bound: 24578.93
Lower Bound: -7457.51
Number of outliers: 1
```



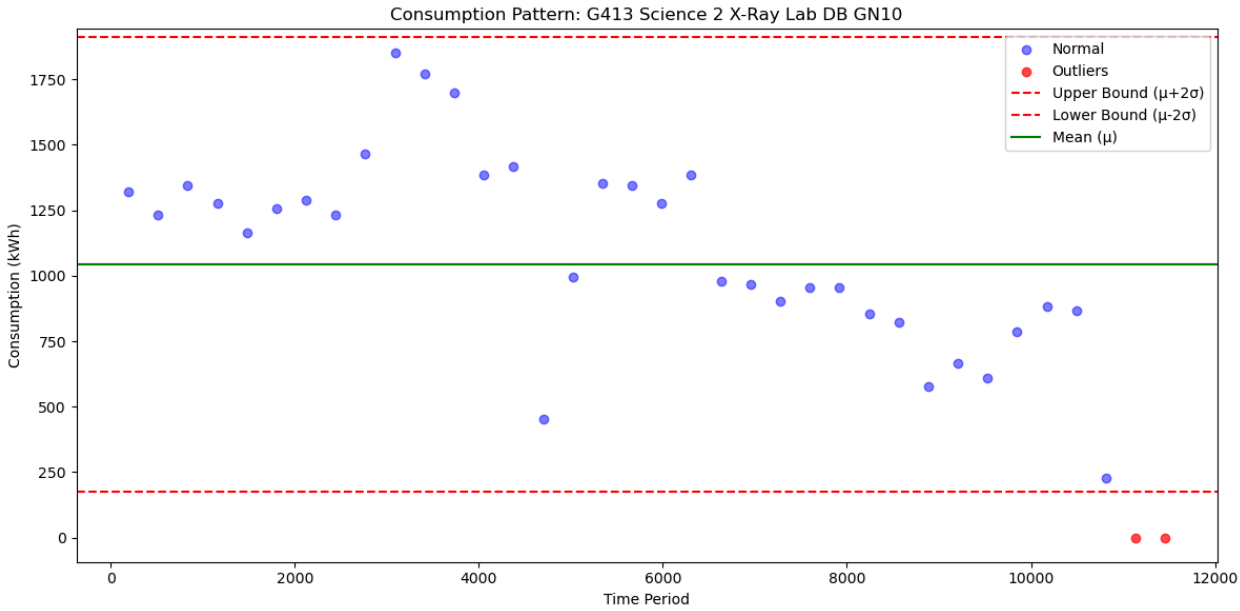

```
Statistics for G413 Science 2 1st floor board:
Mean: 12192.99
Standard Deviation: 7510.57
Upper Bound: 27214.12
Lower Bound: -2828.15
Number of outliers: 4
```



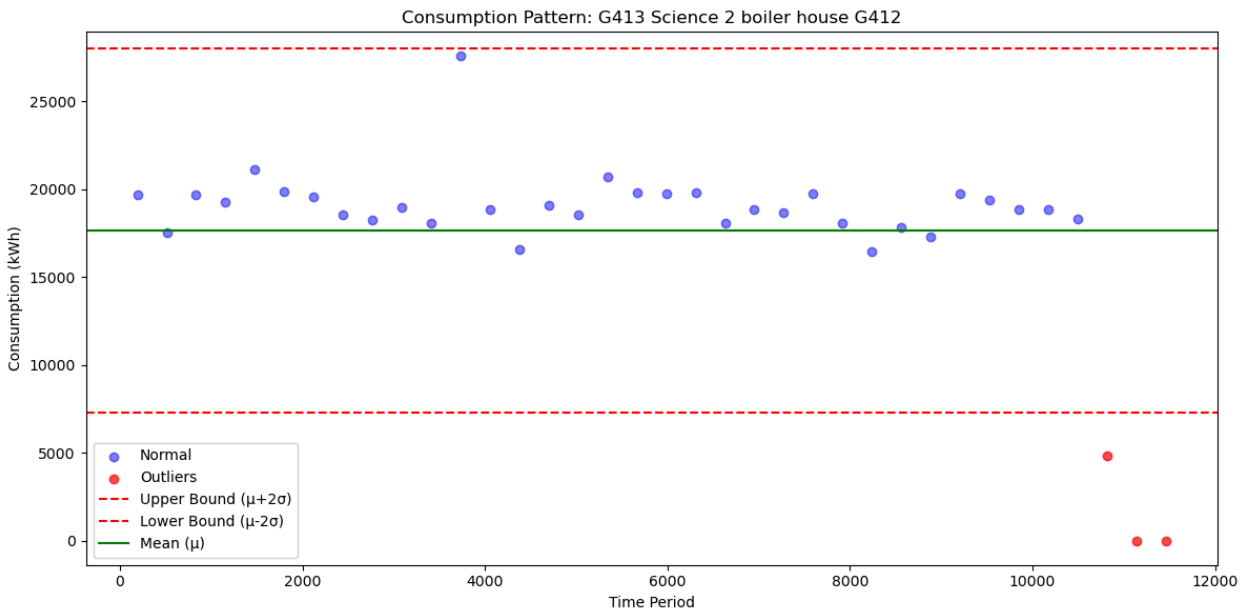

```
Statistics for G413 Science 2 DB GN2:
Mean: 7086.18
Standard Deviation: 2142.48
Upper Bound: 11371.15
Lower Bound: 2801.21
Number of outliers: 3
```



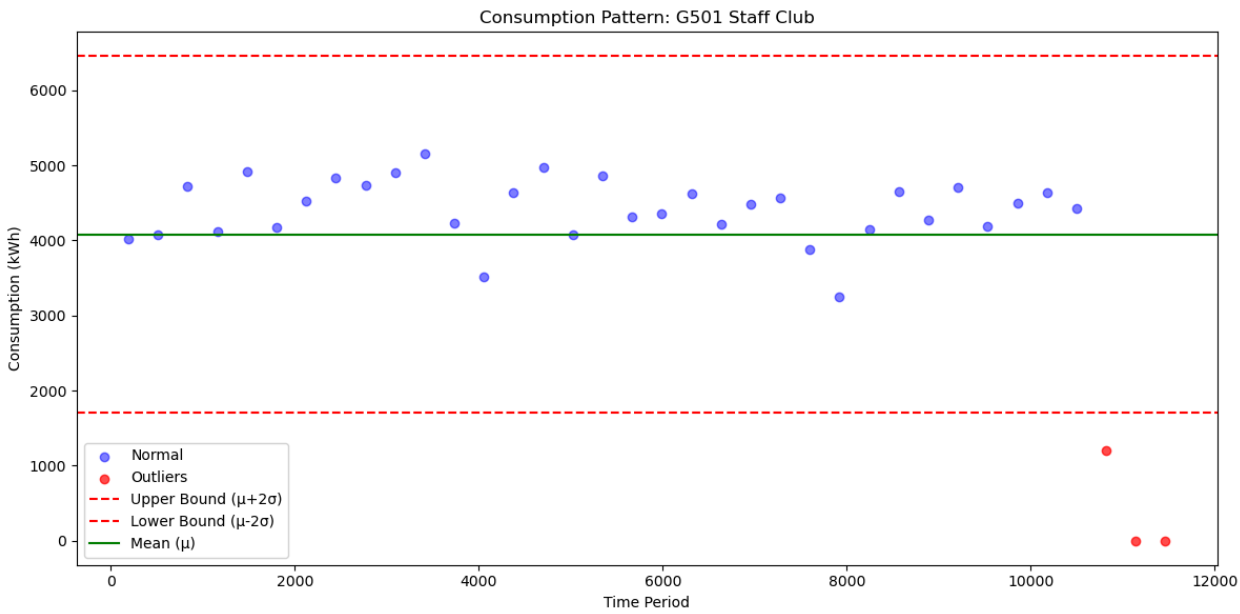

```
Statistics for G413 Science 2 X-Ray Lab DB GN10:
Mean: 1043.41
Standard Deviation: 434.88
Upper Bound: 1913.16
Lower Bound: 173.66
Number of outliers: 2
```



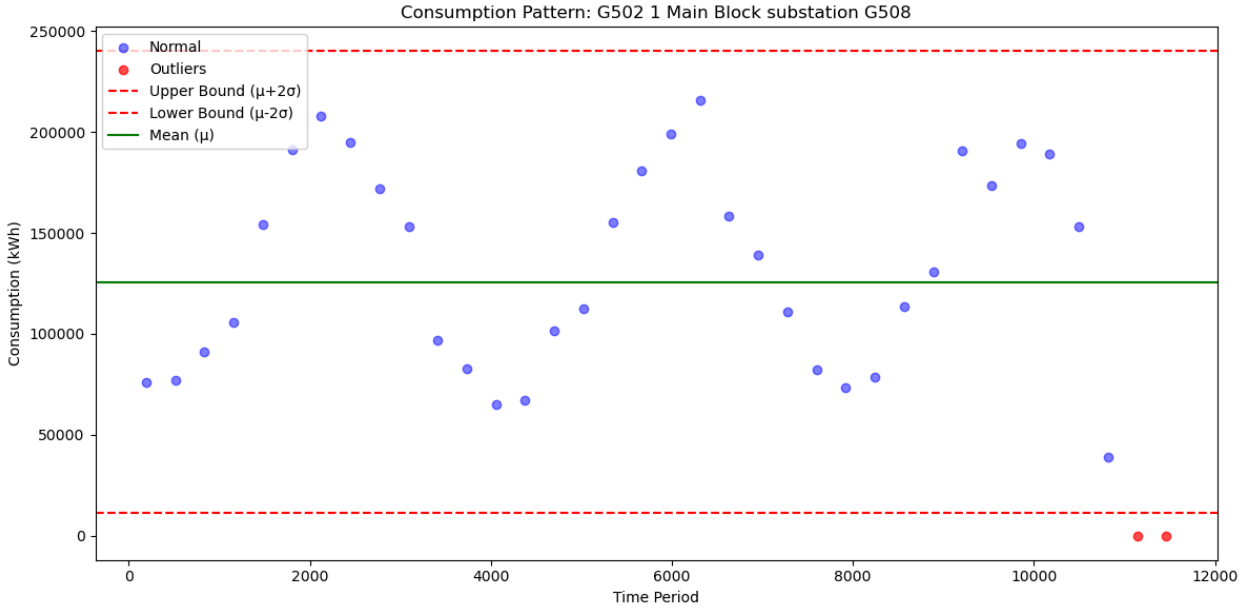

```
Statistics for G413 Science 2 boiler house G412:
Mean: 17661.21
Standard Deviation: 5183.43
Upper Bound: 28028.08
Lower Bound: 7294.34
Number of outliers: 3
```



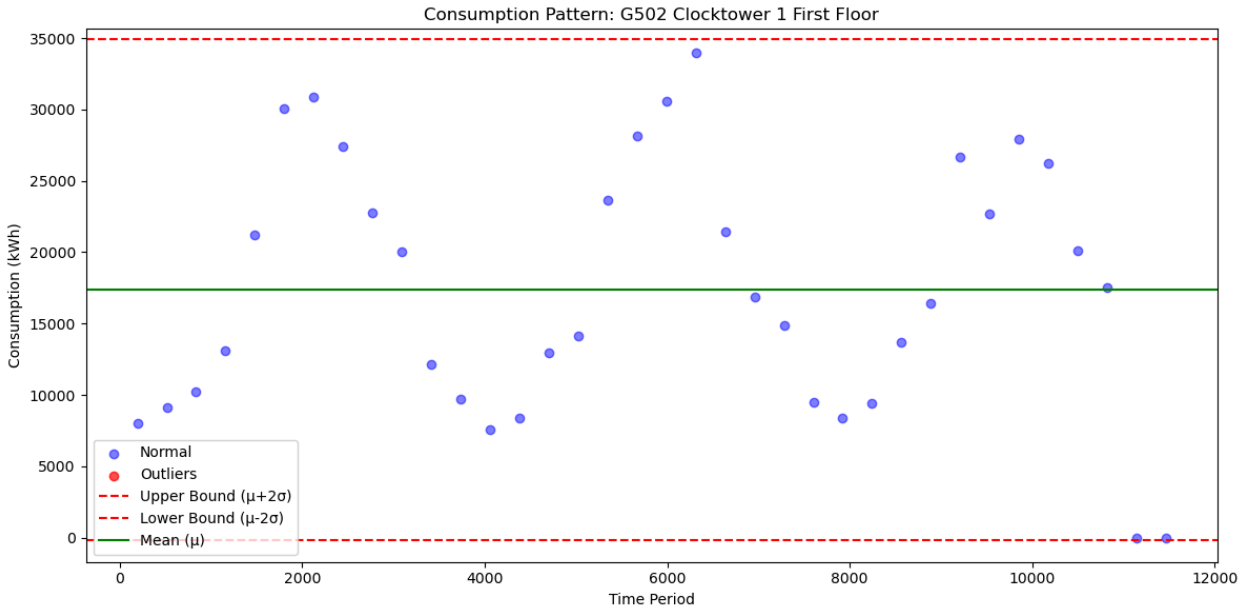

```
Statistics for G501 Staff Club:
Mean: 4081.56
Standard Deviation: 1187.27
Upper Bound: 6456.10
Lower Bound: 1707.01
Number of outliers: 3
```



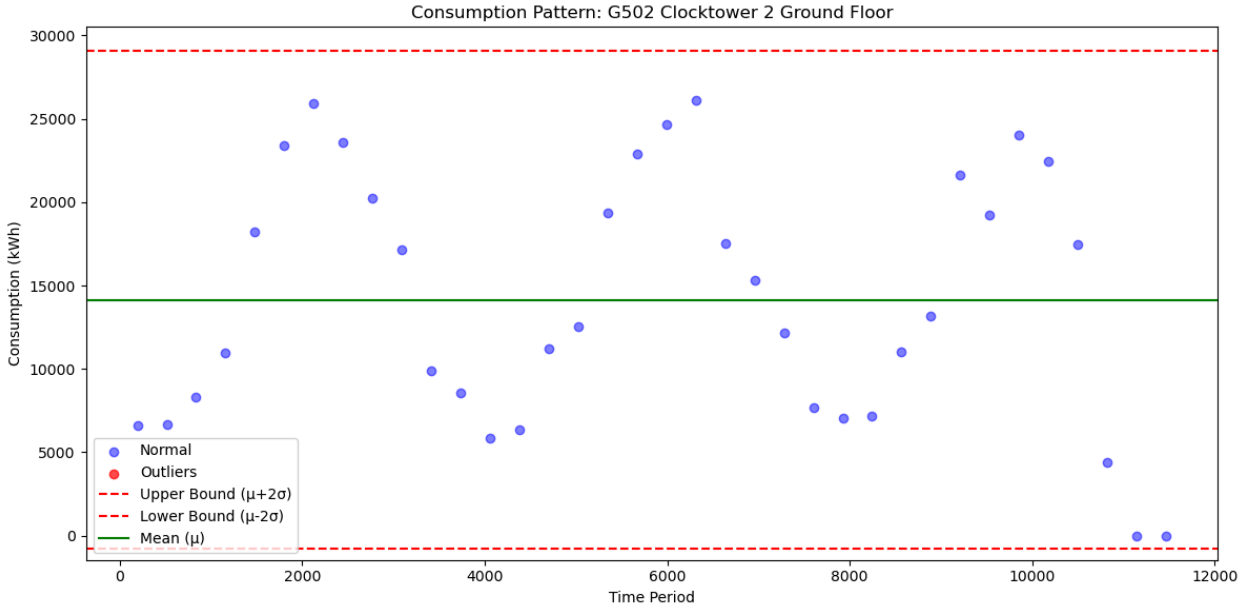

```
Statistics for G502 1 Main Block substation G508:
Mean: 125759.93
Standard Deviation: 57162.64
Upper Bound: 240085.21
Lower Bound: 11434.65
Number of outliers: 2
```




```
Statistics for G502 Clocktower 1 First Floor:
Mean: 17385.88
Standard Deviation: 8777.47
Upper Bound: 34940.82
Lower Bound: -169.07
Number of outliers: 0
```

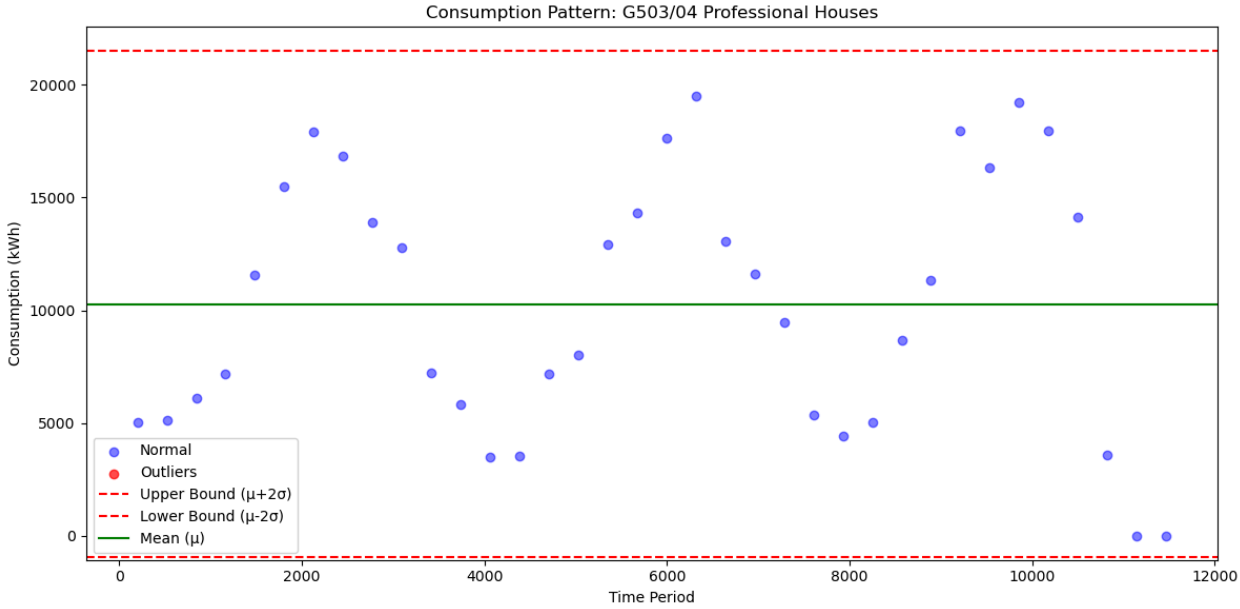



```
Statistics for G502 Clocktower 2 Ground Floor:
Mean: 14133.76
Standard Deviation: 7462.12
Upper Bound: 29058.00
Lower Bound: -790.49
Number of outliers: 0
```



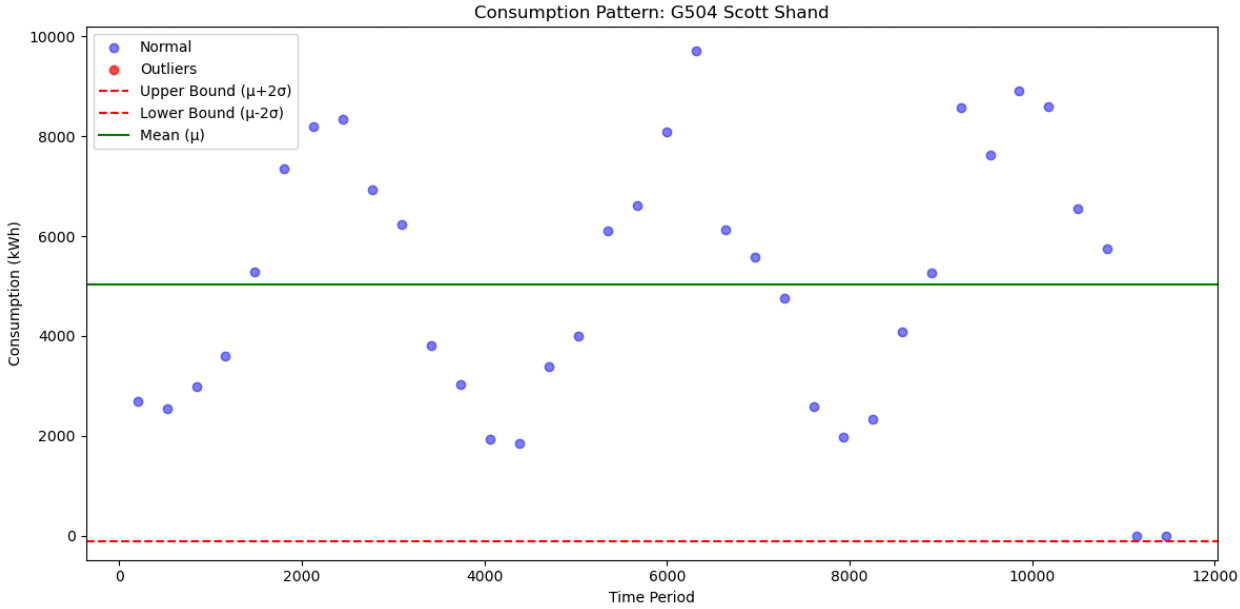

```
Statistics for G502 Clocktower 3 Basement:
Mean: 12592.24
Standard Deviation: 3884.30
Upper Bound: 20360.85
Lower Bound: 4823.63
Number of outliers: 3
```

```
Statistics for G503/04 Professional Houses:
Mean: 10268.94
Standard Deviation: 5612.22
Upper Bound: 21493.38
Lower Bound: -955.51
Number of outliers: 0
```

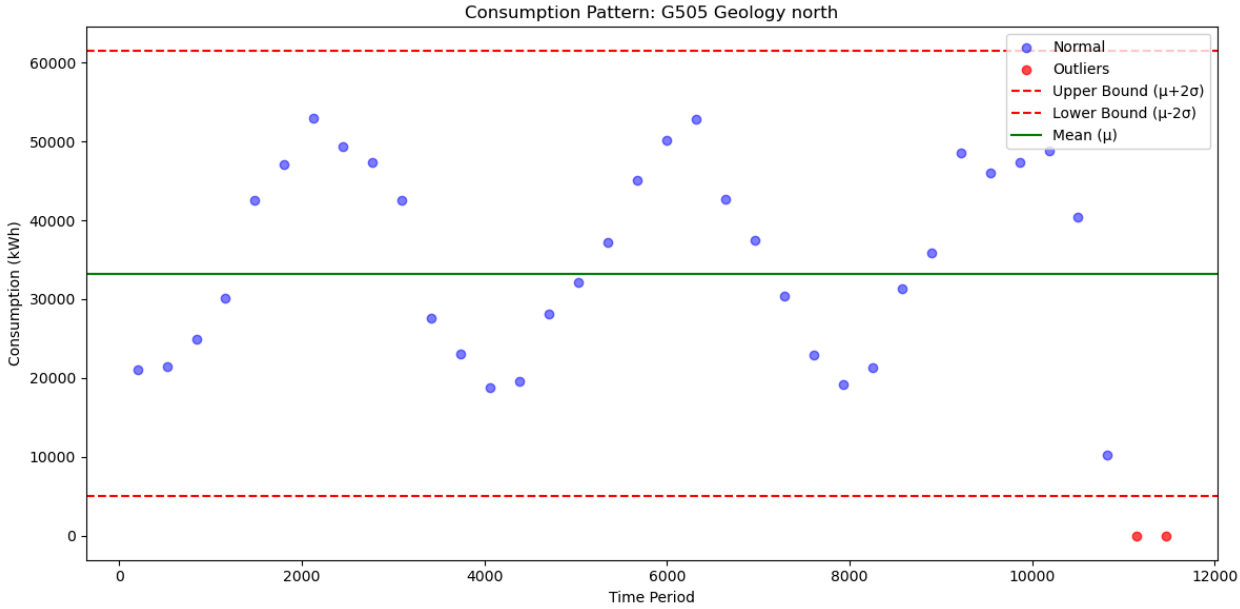



```
Statistics for G504 Black & Sale House:
Mean: 5363.43
Standard Deviation: 2879.45
Upper Bound: 11122.34
Lower Bound: -395.48
Number of outliers: 0
```

```
Statistics for G504 Scott Shand:
Mean: 5038.24
Standard Deviation: 2574.29
Upper Bound: 10186.82
Lower Bound: -110.35
Number of outliers: 0
```

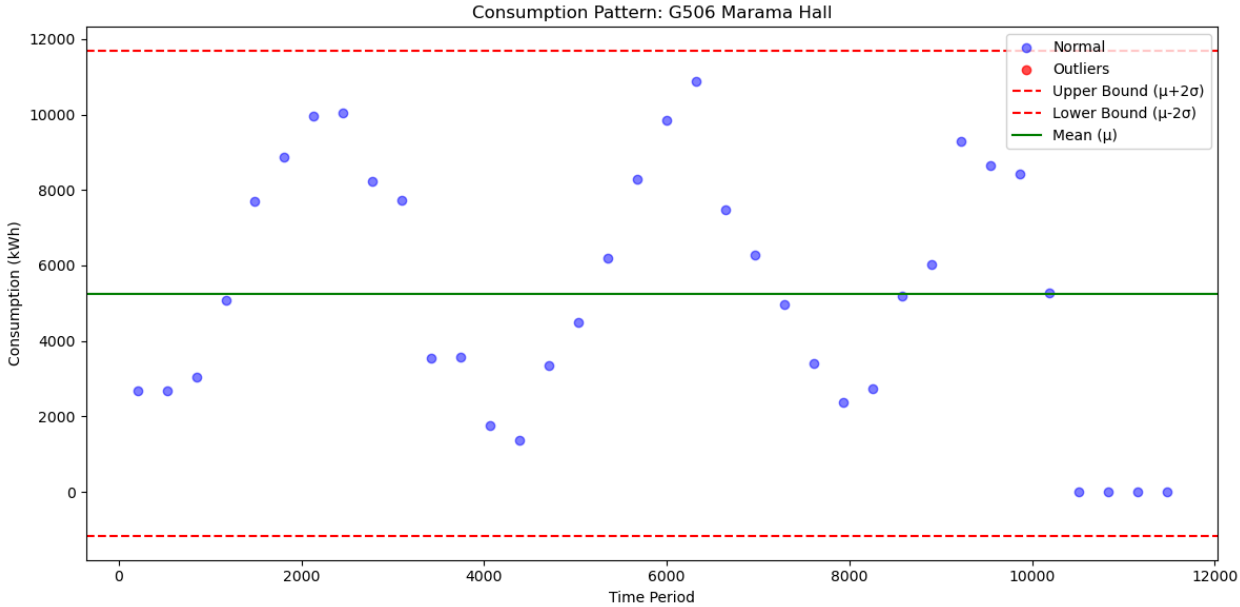



```
Statistics for G505 Geology north:
Mean: 33239.66
Standard Deviation: 14110.51
Upper Bound: 61460.68
Lower Bound: 5018.64
Number of outliers: 2
```

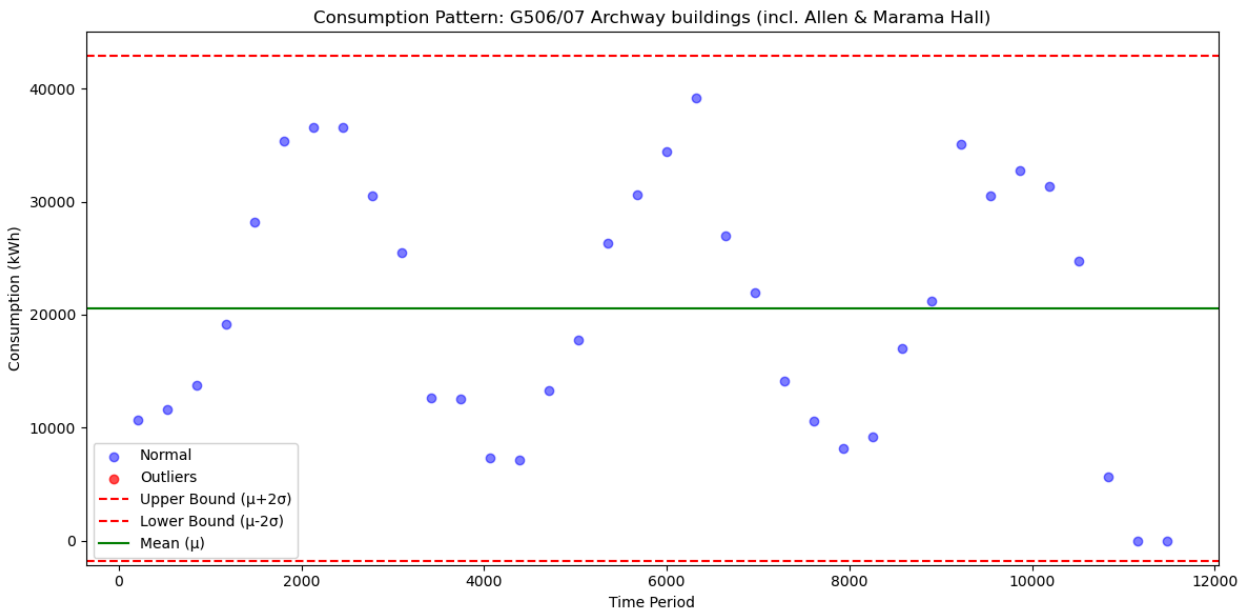



```
Statistics for G505 Geology south:
Mean: 10730.30
Standard Deviation: 5460.58
Upper Bound: 21651.47
Lower Bound: -190.86
Number of outliers: 0
```

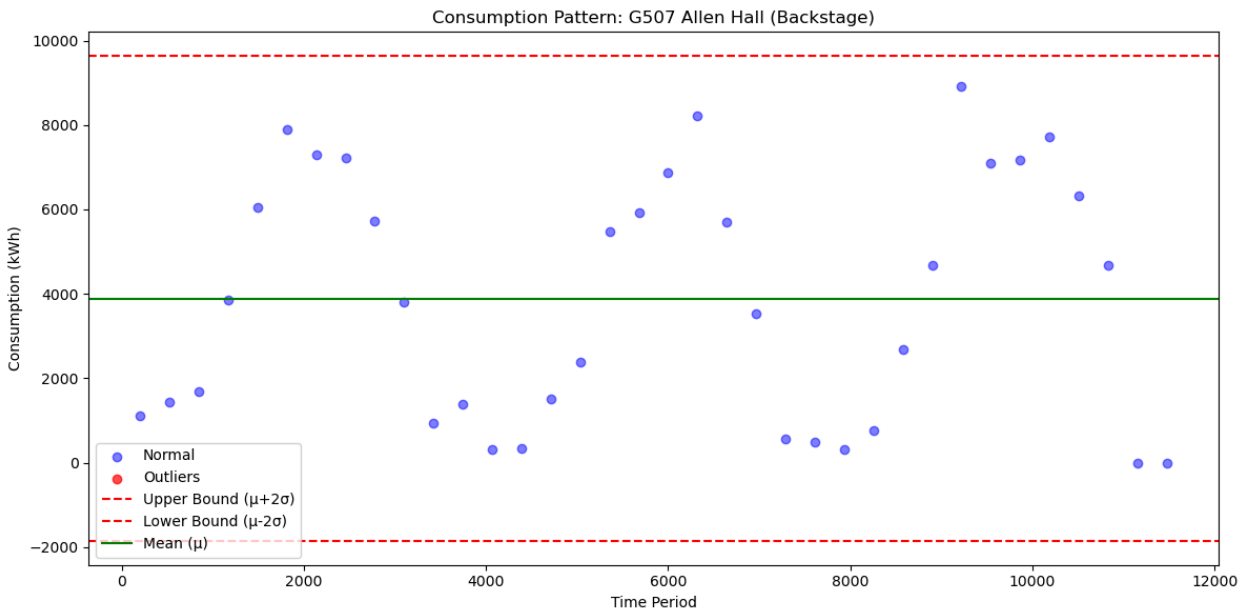
```
Statistics for G506 Marama Hall:
Mean: 5258.96
Standard Deviation: 3209.68
Upper Bound: 11678.31
Lower Bound: -1160.39
Number of outliers: 0
```



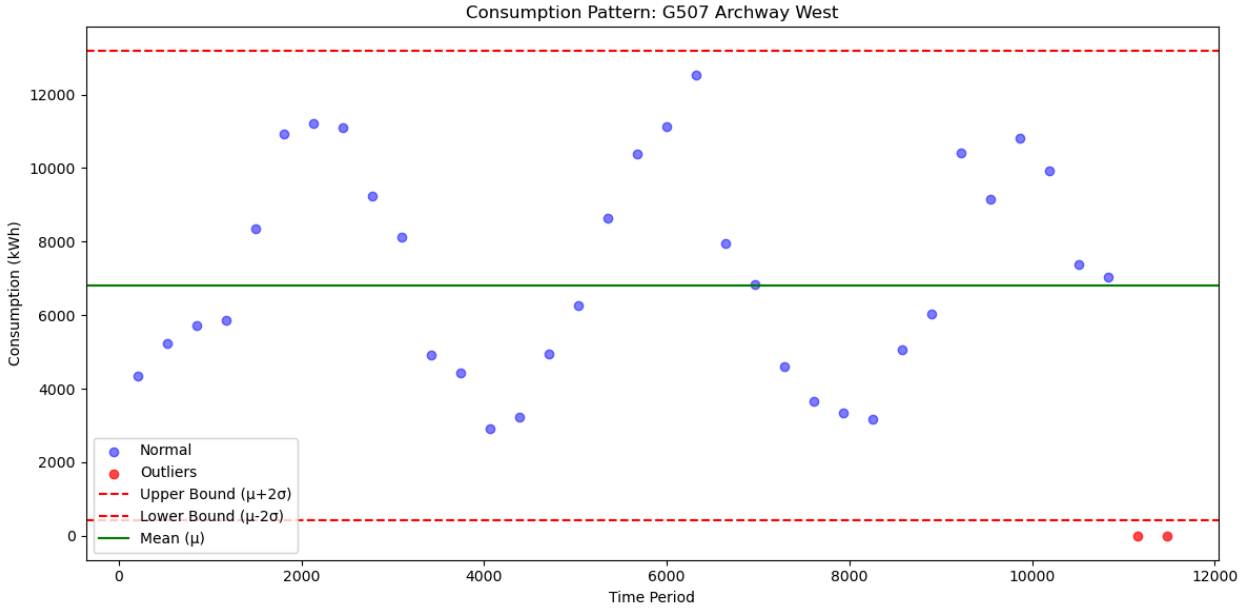

```
Statistics for G506/07 Archway buildings (incl. Allen & Marama Hall):
Mean: 20517.76
Standard Deviation: 11179.04
Upper Bound: 42875.83
Lower Bound: -1840.31
Number of outliers: 0
```



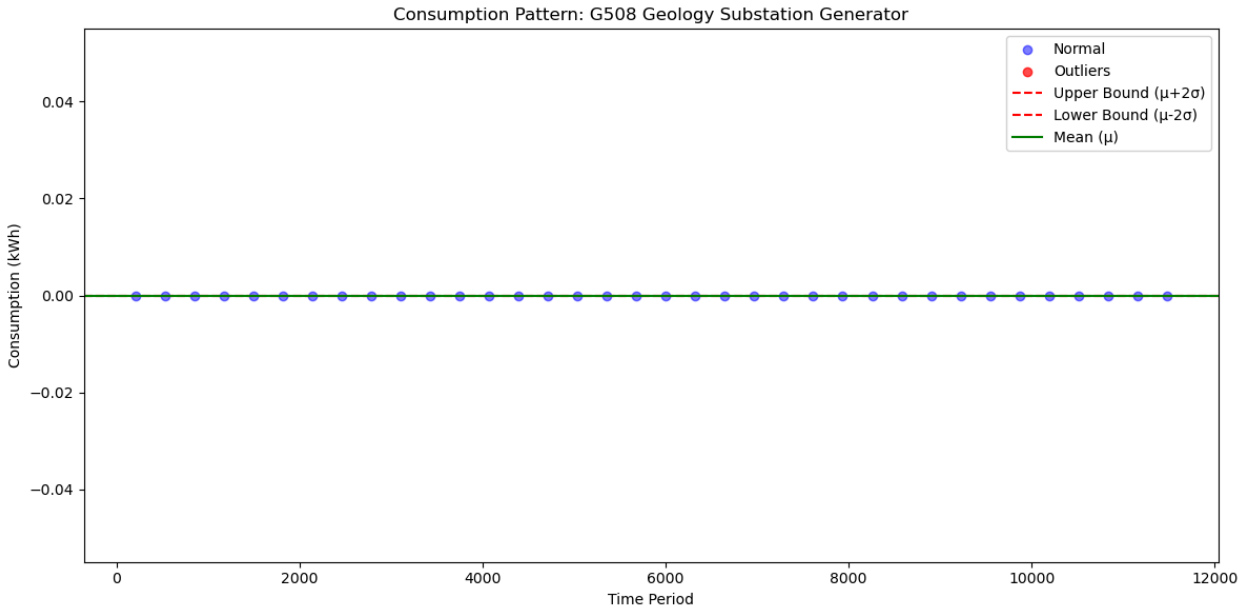

```
Statistics for G507 Allen Hall (Backstage):
Mean: 3891.44
Standard Deviation: 2870.52
Upper Bound: 9632.47
Lower Bound: -1849.60
Number of outliers: 0
```



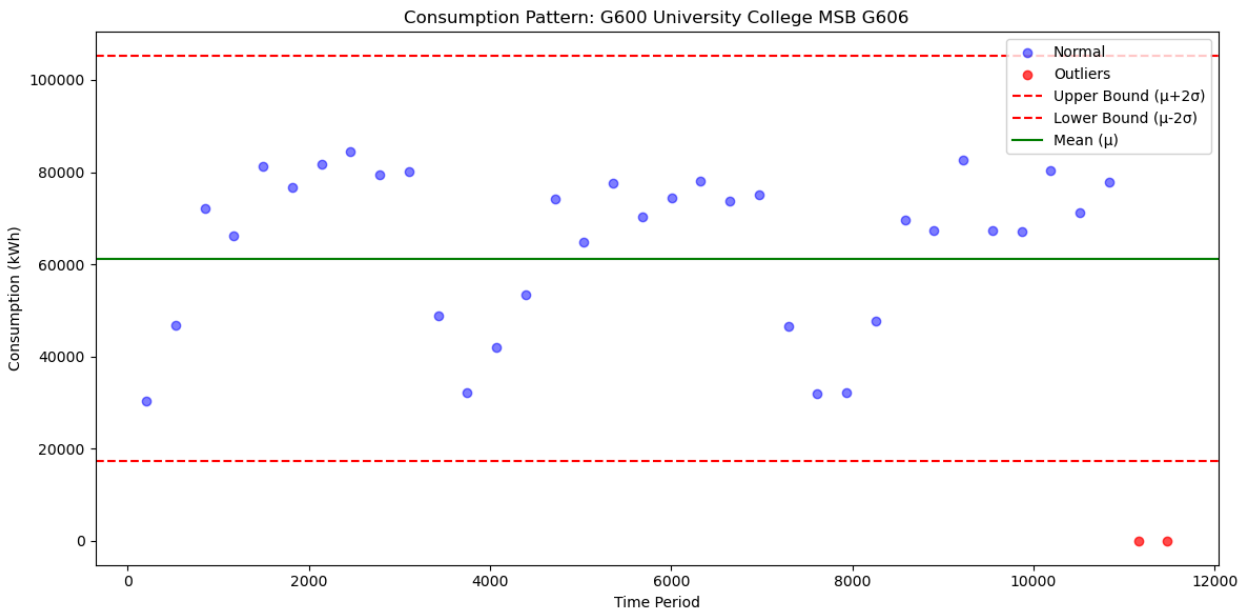

```
Statistics for G507 Archway West:
Mean: 6801.56
Standard Deviation: 3190.39
Upper Bound: 13182.34
Lower Bound: 420.78
Number of outliers: 2
```



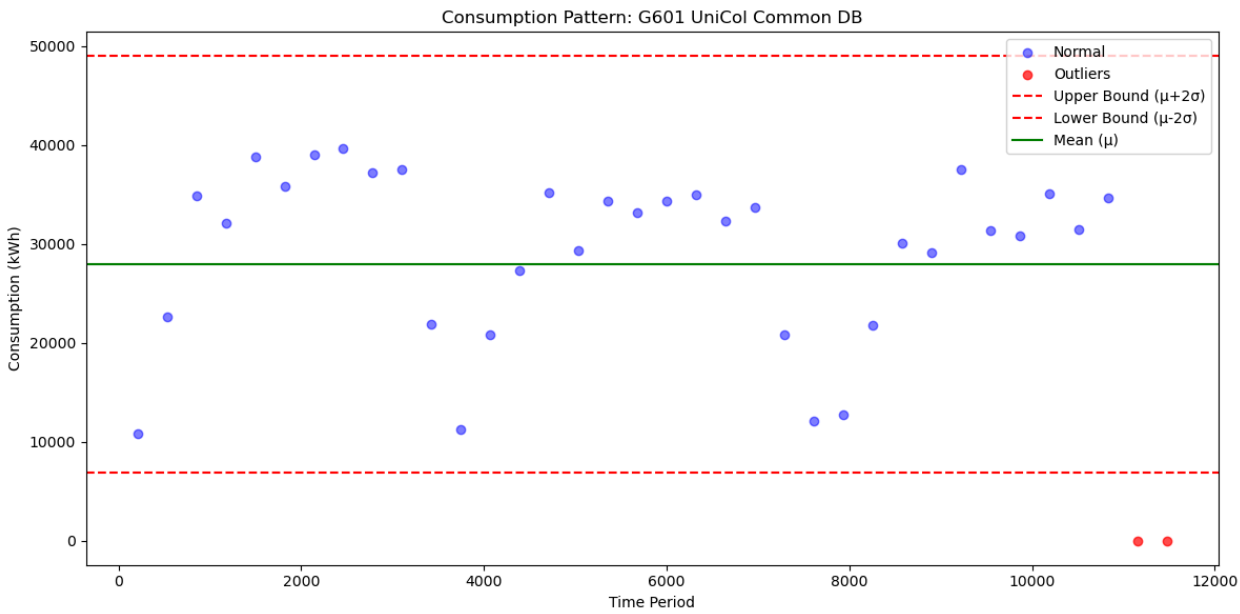

```
Statistics for G508 Geology Substation Generator:
Mean: 0.00
Standard Deviation: 0.00
Upper Bound: 0.00
Lower Bound: 0.00
Number of outliers: 0
```



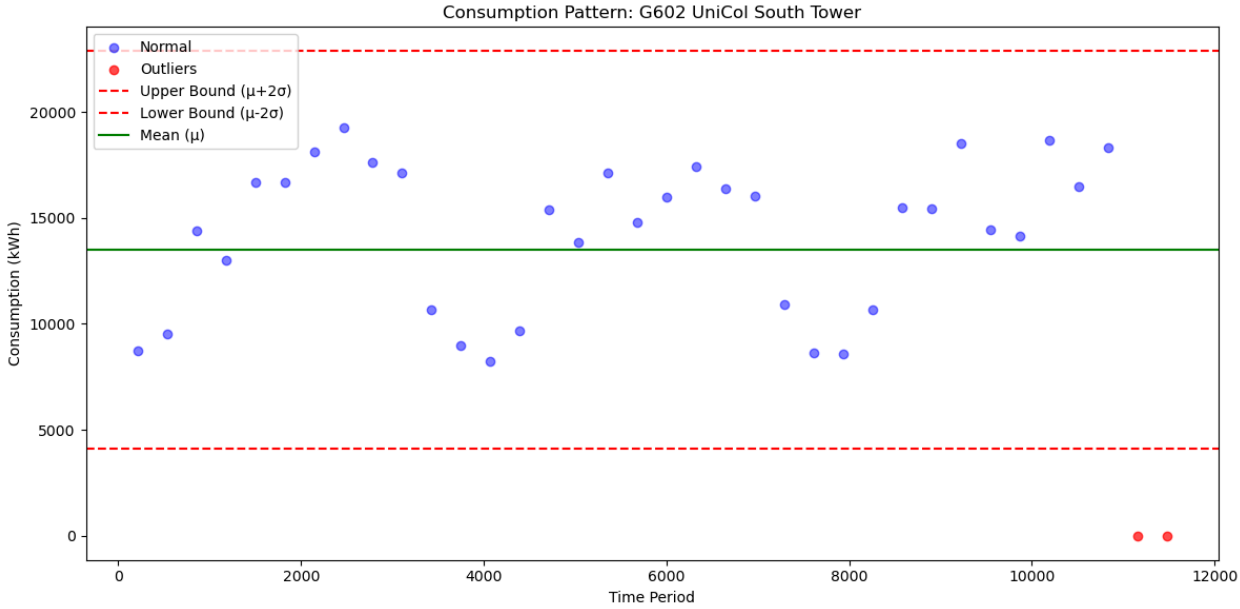

```
Statistics for G600 University College MSB G606:
Mean: 61253.01
Standard Deviation: 21959.35
Upper Bound: 105171.71
Lower Bound: 17334.30
Number of outliers: 2
```



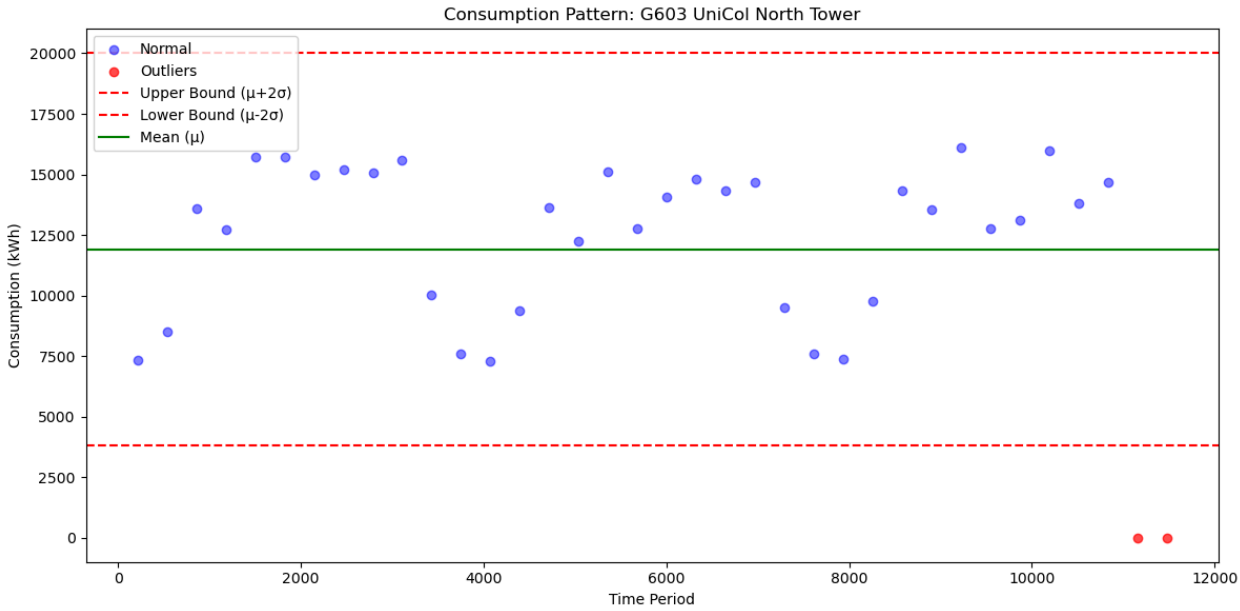

```
Statistics for G601 UniCol Common DB:
Mean: 27918.93
Standard Deviation: 10541.71
Upper Bound: 49002.35
Lower Bound: 6835.50
Number of outliers: 2
```



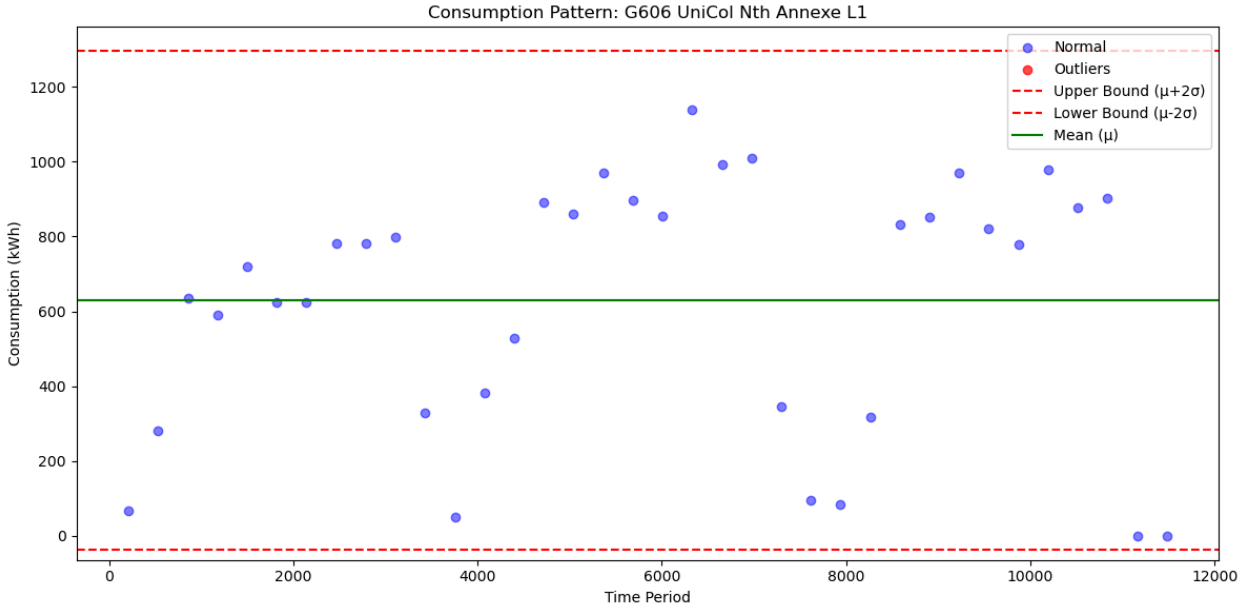

```
Statistics for G602 UniCol South Tower:
Mean: 13493.87
Standard Deviation: 4684.67
Upper Bound: 22863.21
Lower Bound: 4124.54
Number of outliers: 2
```



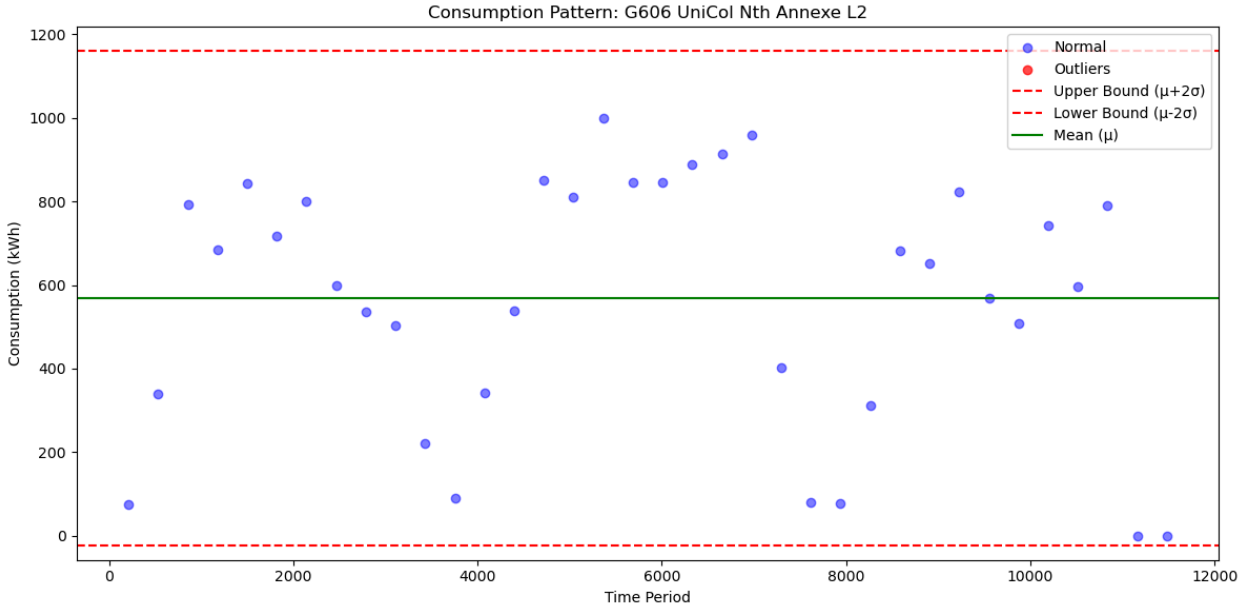

```
Statistics for G603 UniCol North Tower:
Mean: 11914.70
Standard Deviation: 4045.40
Upper Bound: 20005.49
Lower Bound: 3823.90
Number of outliers: 2
```



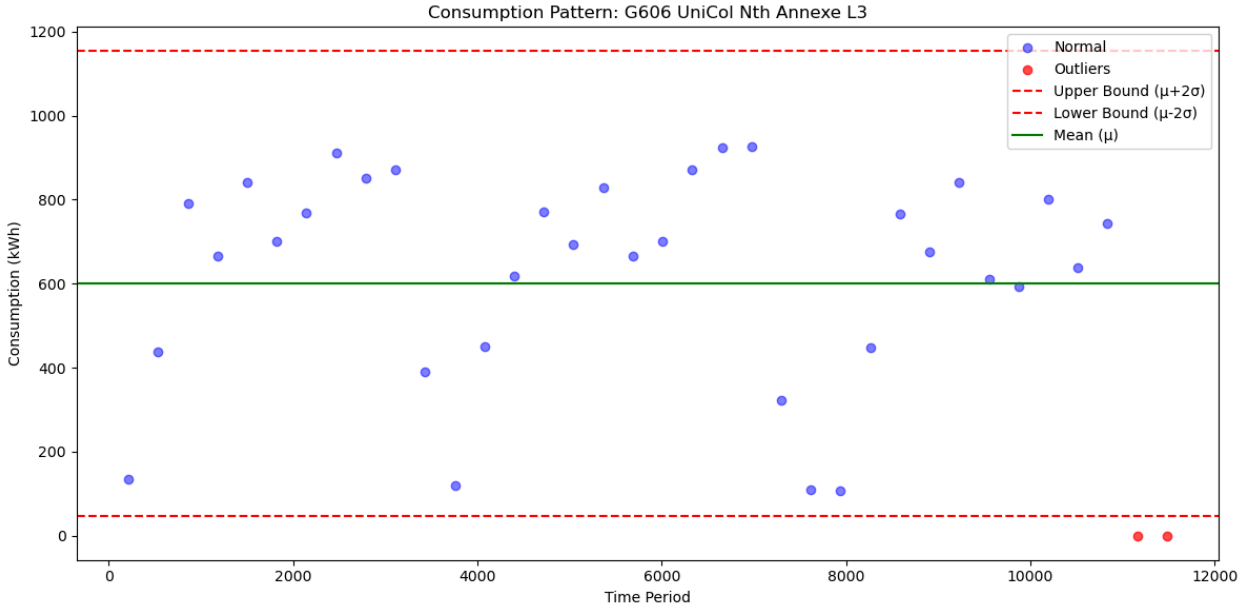

```
Statistics for G606 UniCol Nth Annexe L1:
Mean: 629.45
Standard Deviation: 333.19
Upper Bound: 1295.82
Lower Bound: -36.92
Number of outliers: 0
```



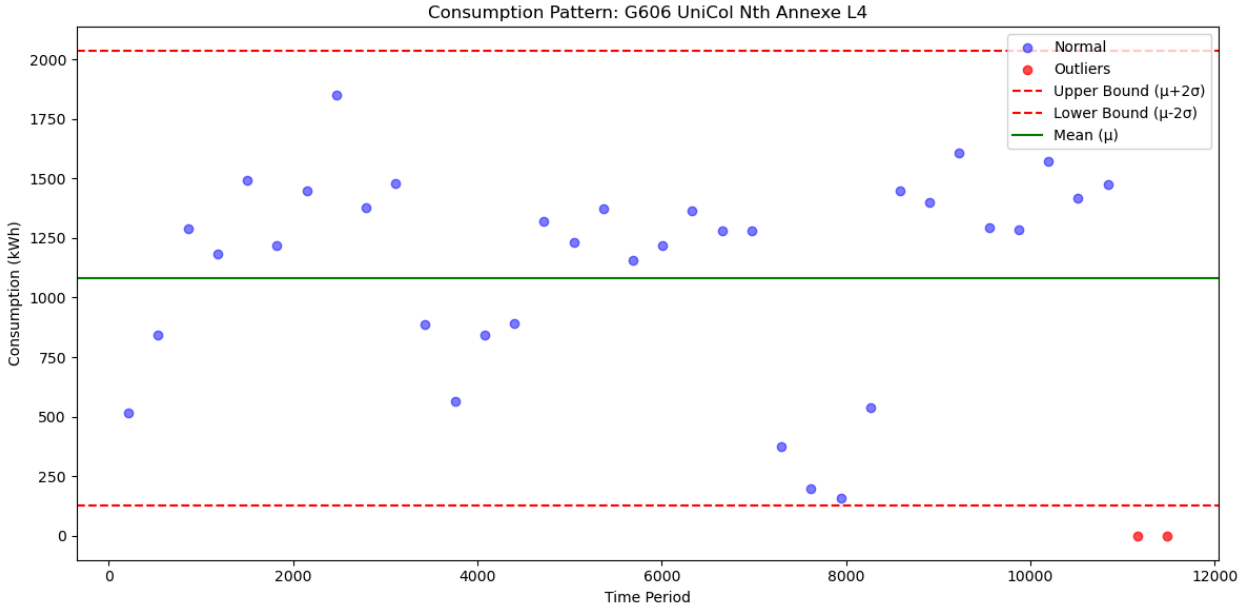

```
Statistics for G606 UniCol Nth Annexe L2:
Mean: 567.52
Standard Deviation: 296.04
Upper Bound: 1159.61
Lower Bound: -24.56
Number of outliers: 0
```



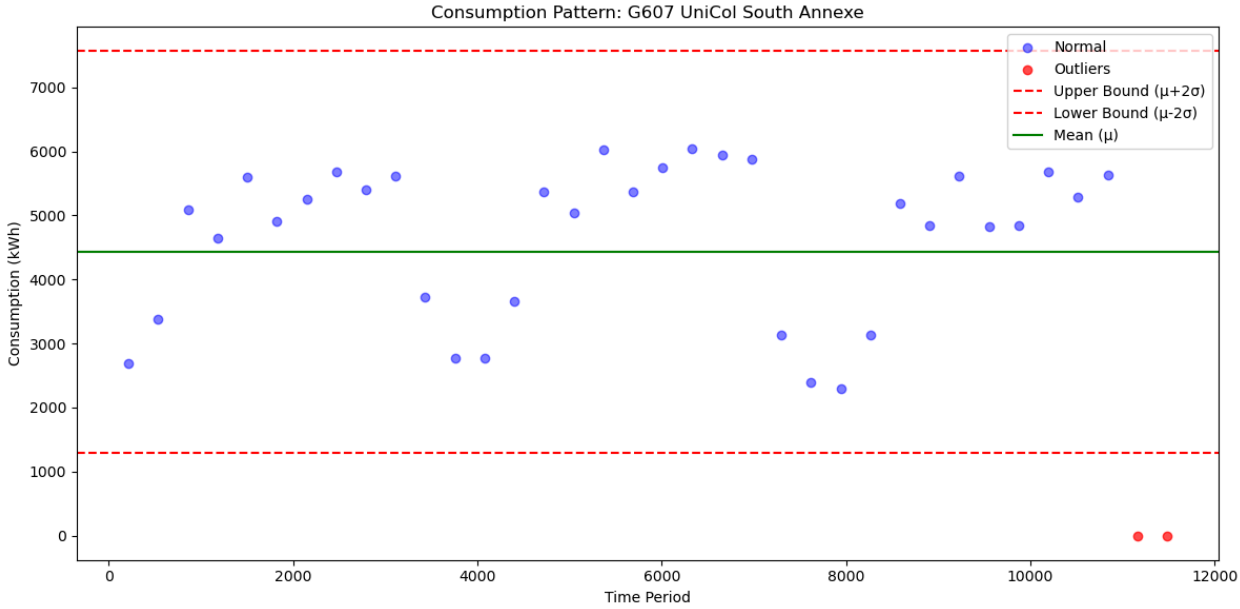

```
Statistics for G606 UniCol Nth Annexe L3:
Mean: 599.83
Standard Deviation: 276.83
Upper Bound: 1153.49
Lower Bound: 46.18
Number of outliers: 2
```



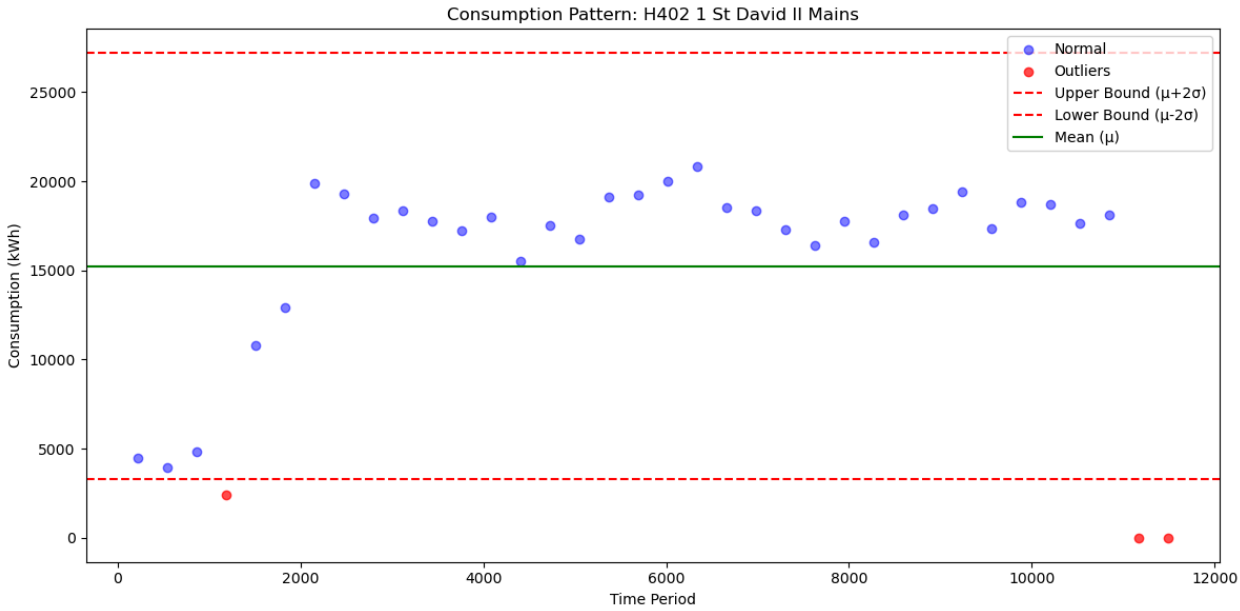

```
Statistics for G606 UniCol Nth Annexe L4:
Mean: 1080.19
Standard Deviation: 477.46
Upper Bound: 2035.11
Lower Bound: 125.27
Number of outliers: 2
```



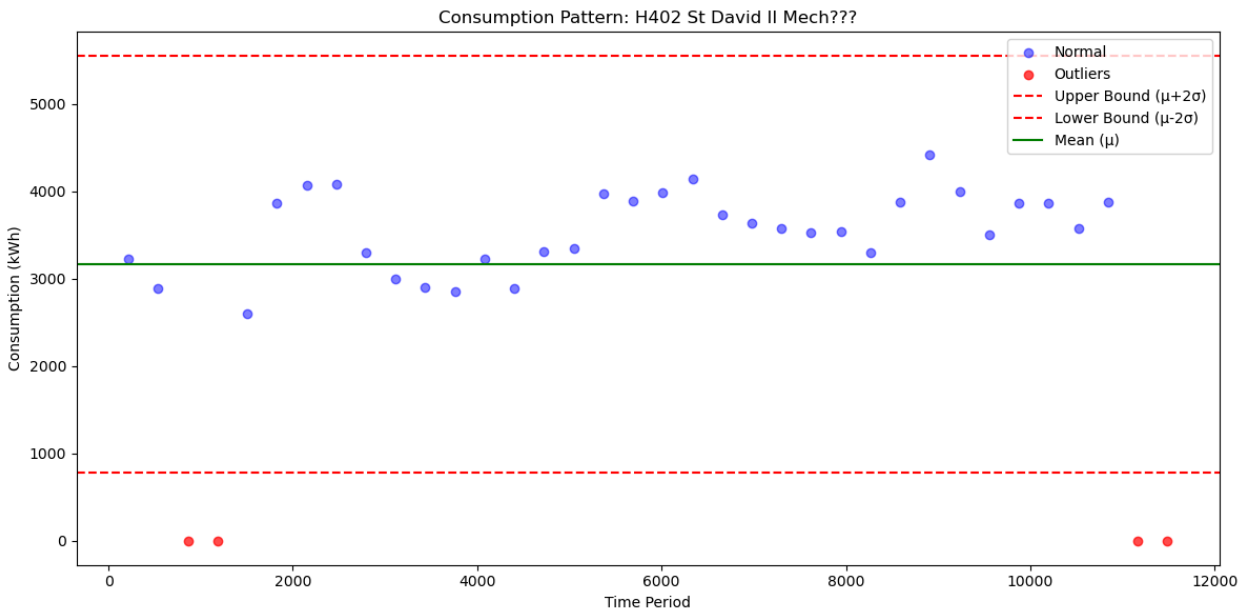

```
Statistics for G607 UniCol South Annexe:
Mean: 4429.77
Standard Deviation: 1566.84
Upper Bound: 7563.45
Lower Bound: 1296.09
Number of outliers: 2
```



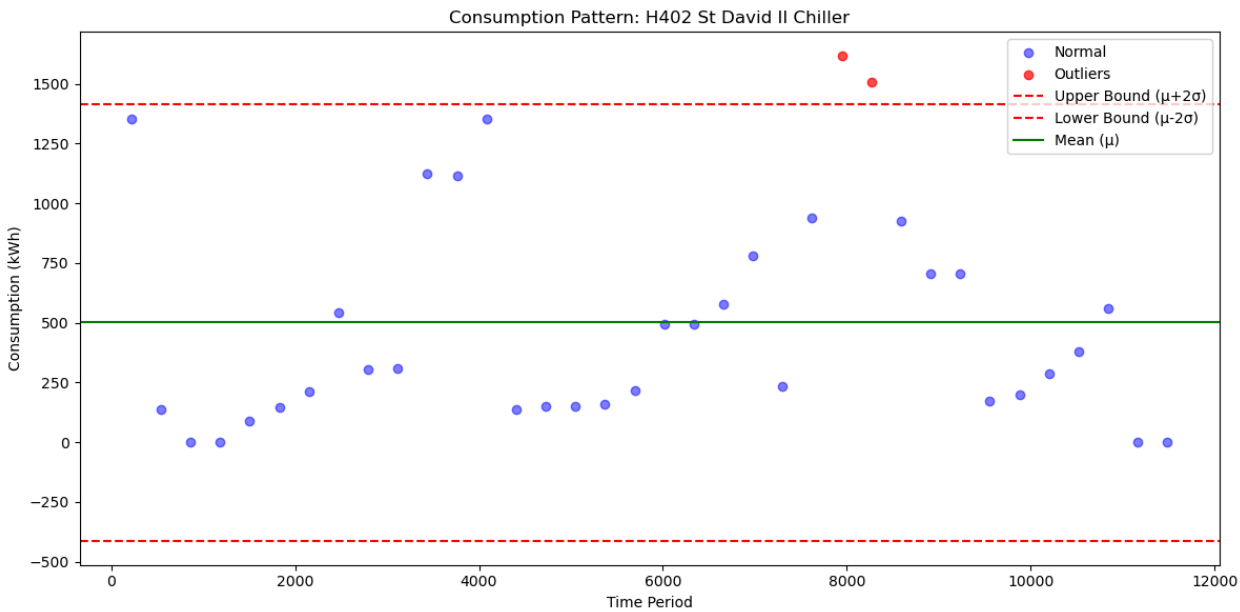

```
Statistics for H402 1 St David II Mains:
Mean: 15230.26
Standard Deviation: 5979.67
Upper Bound: 27189.59
Lower Bound: 3270.92
Number of outliers: 3
```



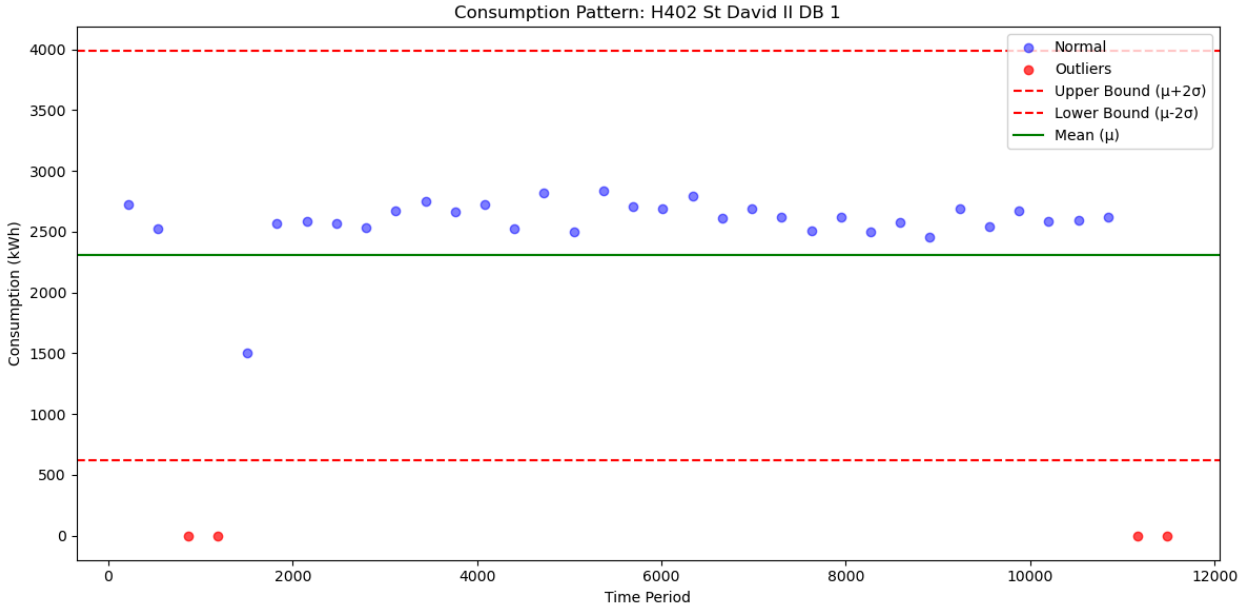

```
Statistics for H402 St David II Mech???:
Mean: 3159.42
Standard Deviation: 1192.71
Upper Bound: 5544.83
Lower Bound: 774.01
Number of outliers: 4
```



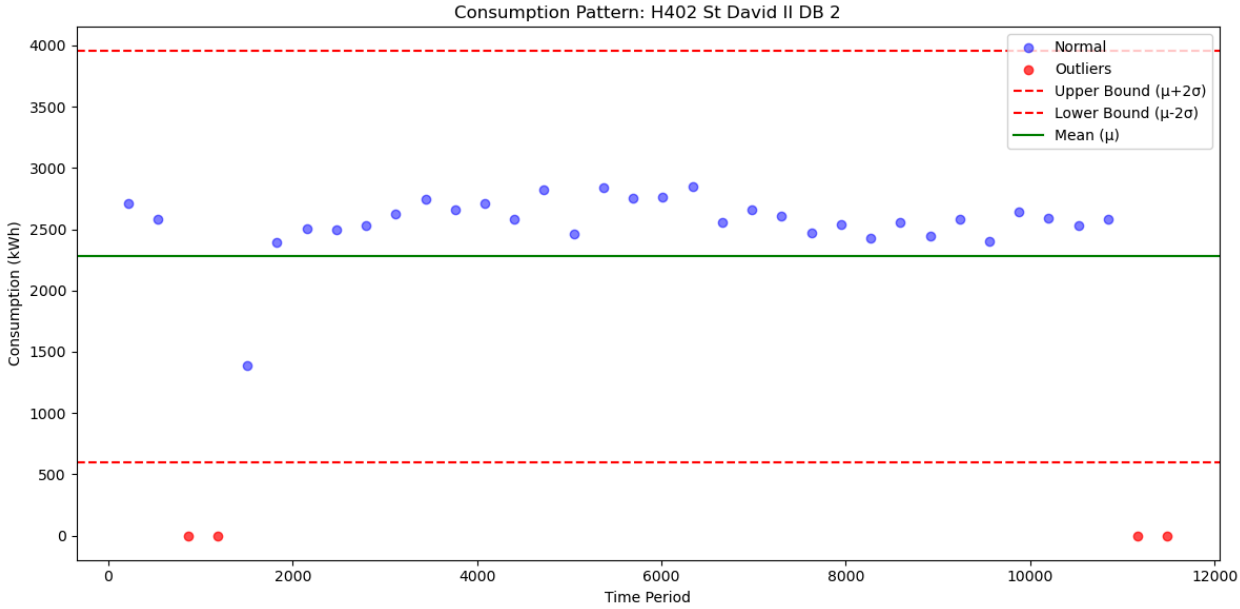

```
Statistics for H402 St David II Chiller:
Mean: 501.65
Standard Deviation: 457.00
Upper Bound: 1415.64
Lower Bound: -412.34
Number of outliers: 2
```



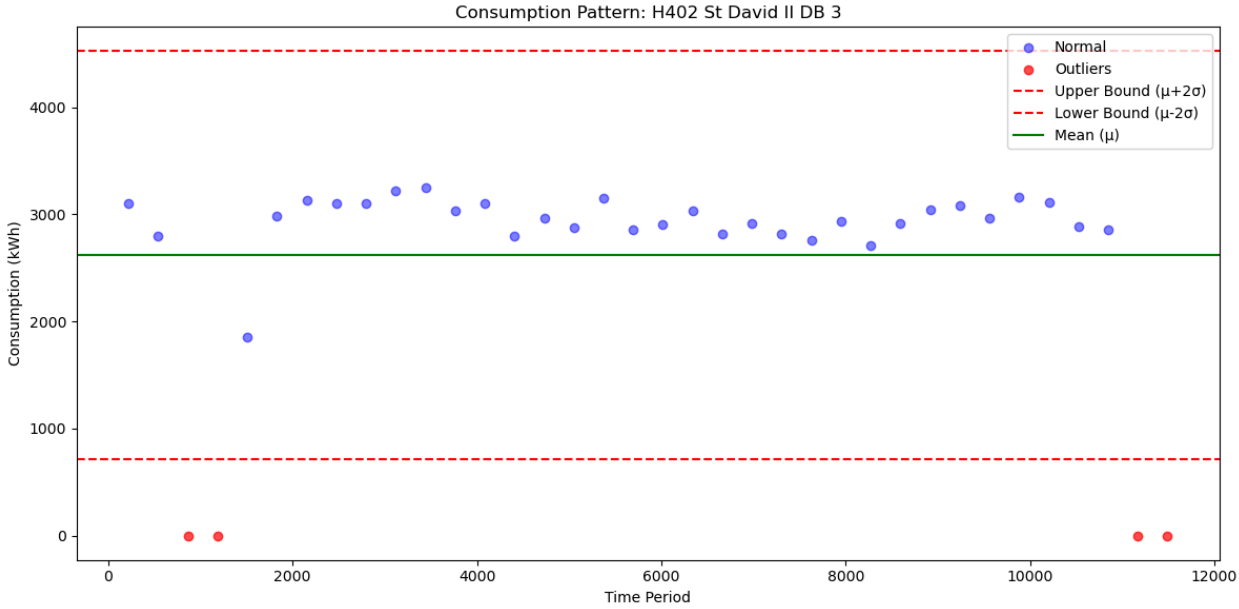

```
Statistics for H402 St David II DB 1:
Mean: 2304.35
Standard Deviation: 839.96
Upper Bound: 3984.28
Lower Bound: 624.42
Number of outliers: 4
```



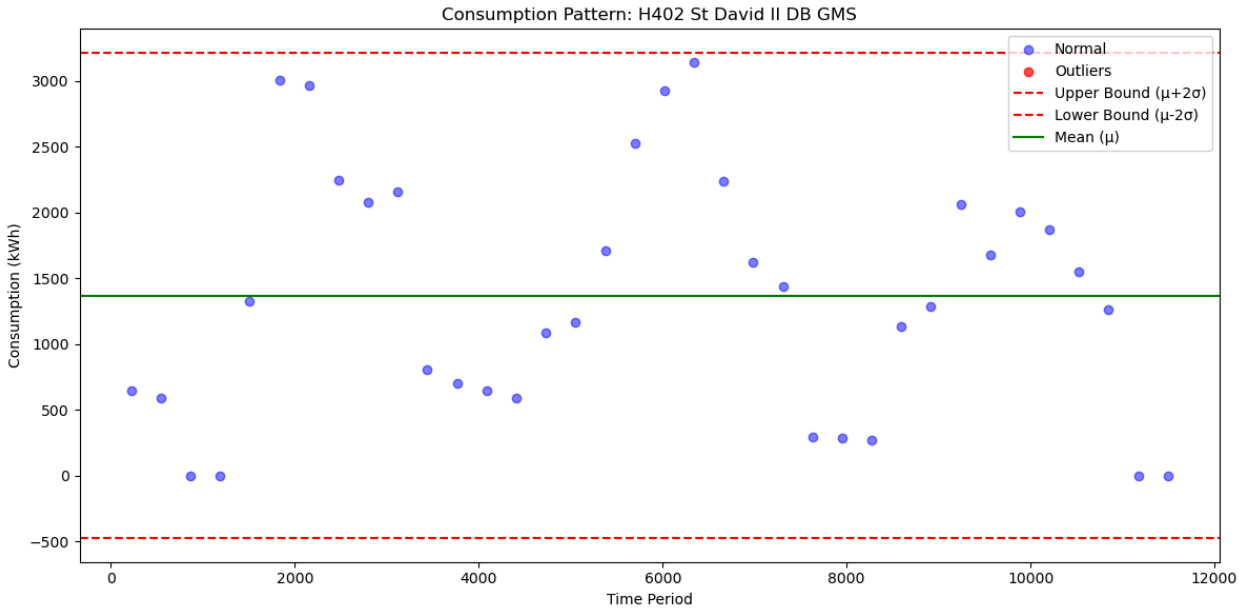

```
Statistics for H402 St David II DB 2:
Mean: 2277.39
Standard Deviation: 837.38
Upper Bound: 3952.16
Lower Bound: 602.62
Number of outliers: 4
```



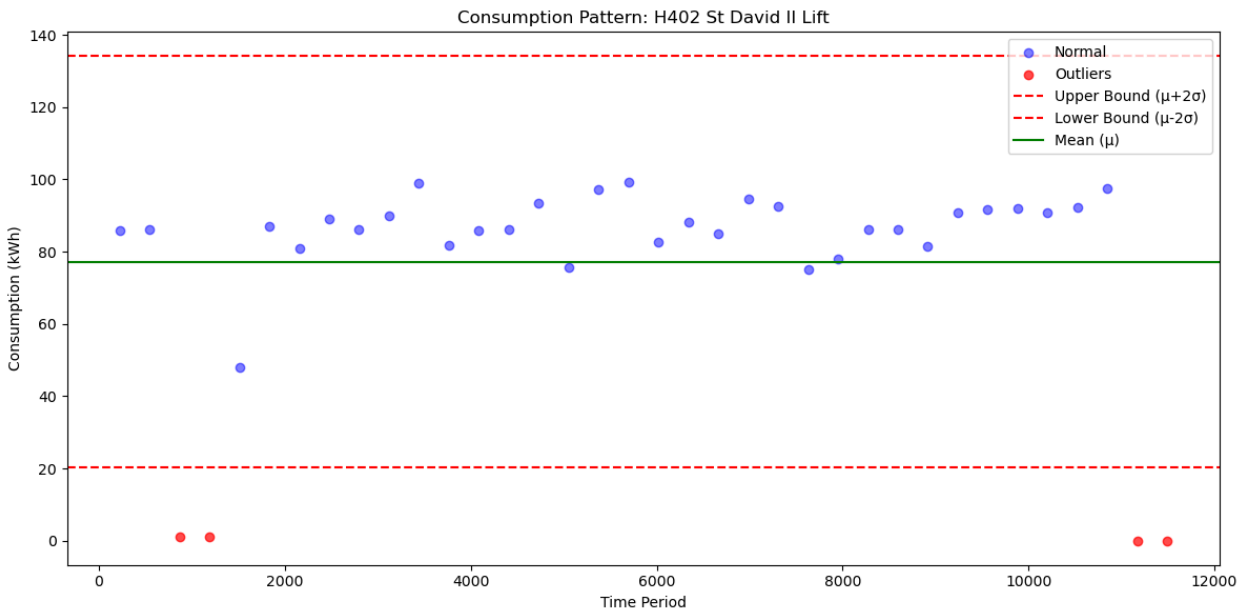

```
Statistics for H402 St David II DB 3:
Mean: 2618.44
Standard Deviation: 952.78
Upper Bound: 4524.01
Lower Bound: 712.87
Number of outliers: 4
```



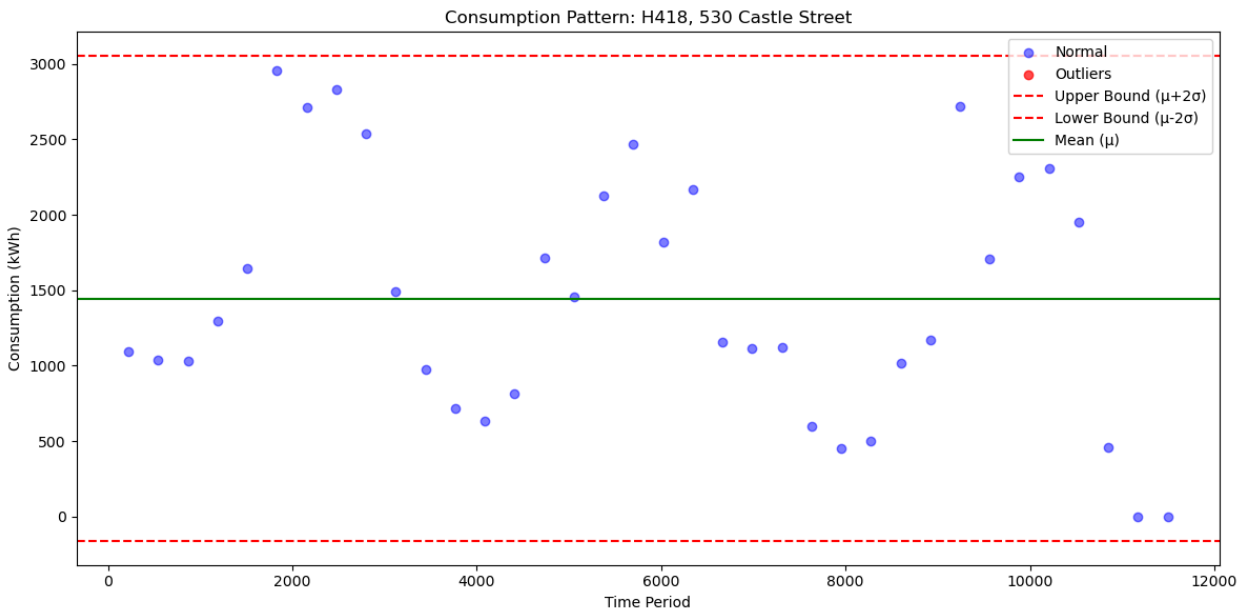

```
Statistics for H402 St David II DB GMS:
Mean: 1368.51
Standard Deviation: 920.45
Upper Bound: 3209.41
Lower Bound: -472.39
Number of outliers: 0
```




```
Statistics for H402 St David II Lift:
Mean: 77.15
Standard Deviation: 28.47
Upper Bound: 134.10
Lower Bound: 20.21
Number of outliers: 4
```

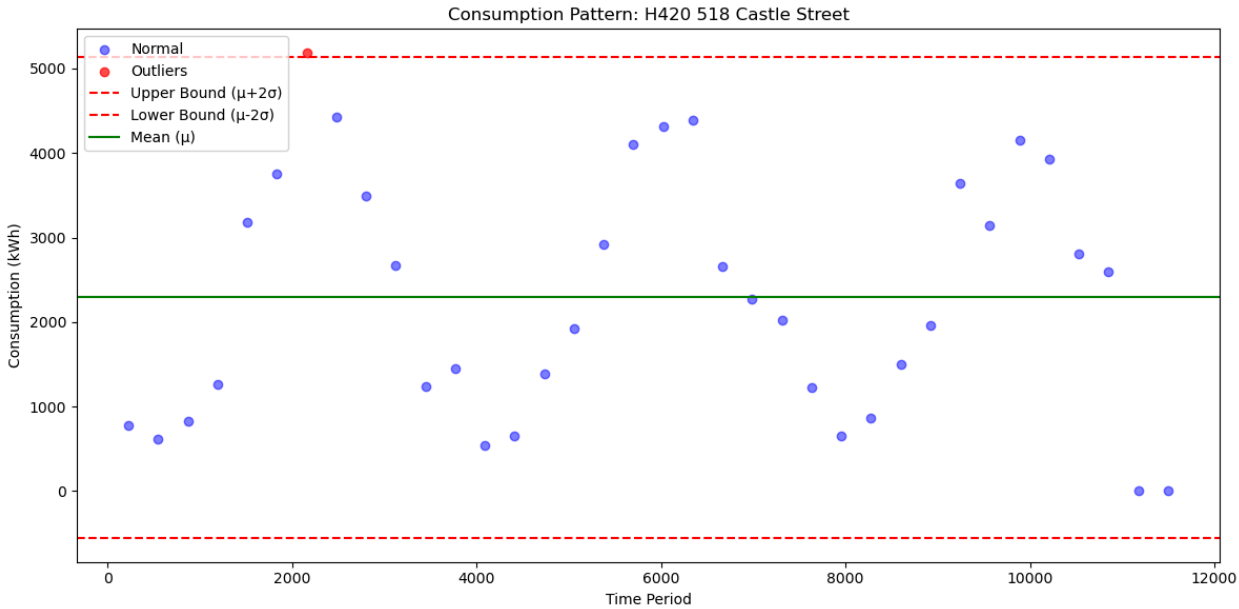



```
Statistics for H418, 530 Castle Street:
Mean: 1446.10
Standard Deviation: 802.75
Upper Bound: 3051.59
Lower Bound: -159.39
Number of outliers: 0
```

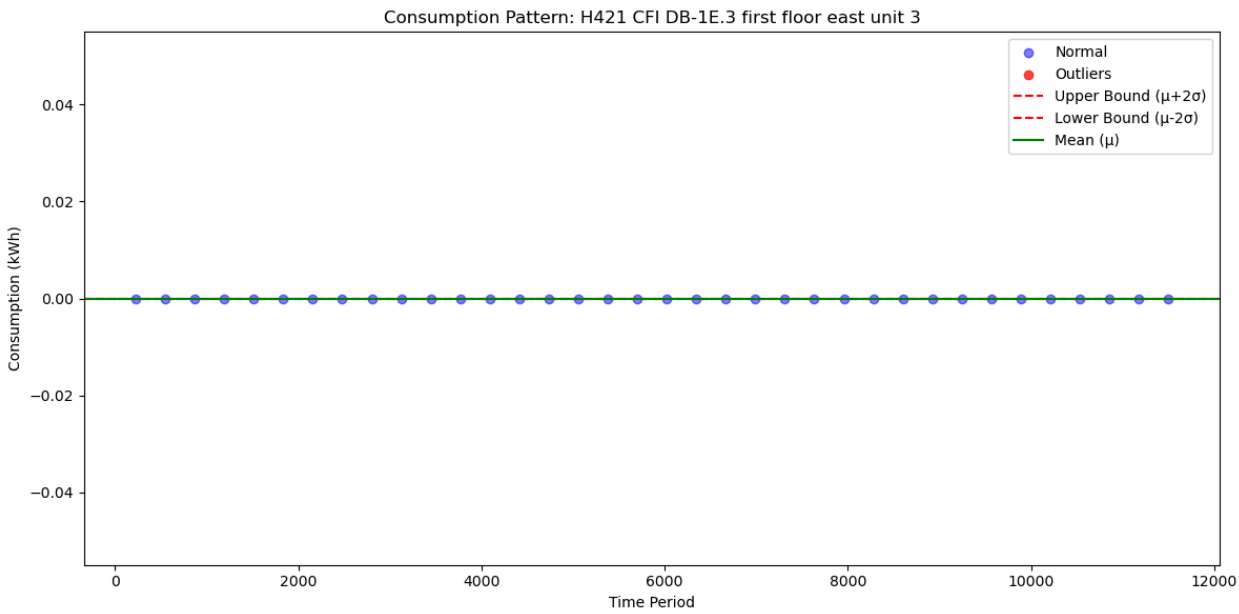



```
Statistics for H419, 520 Castle Street:
Mean: 3317.74
Standard Deviation: 2087.84
Upper Bound: 7493.41
Lower Bound: -857.94
Number of outliers: 0
```

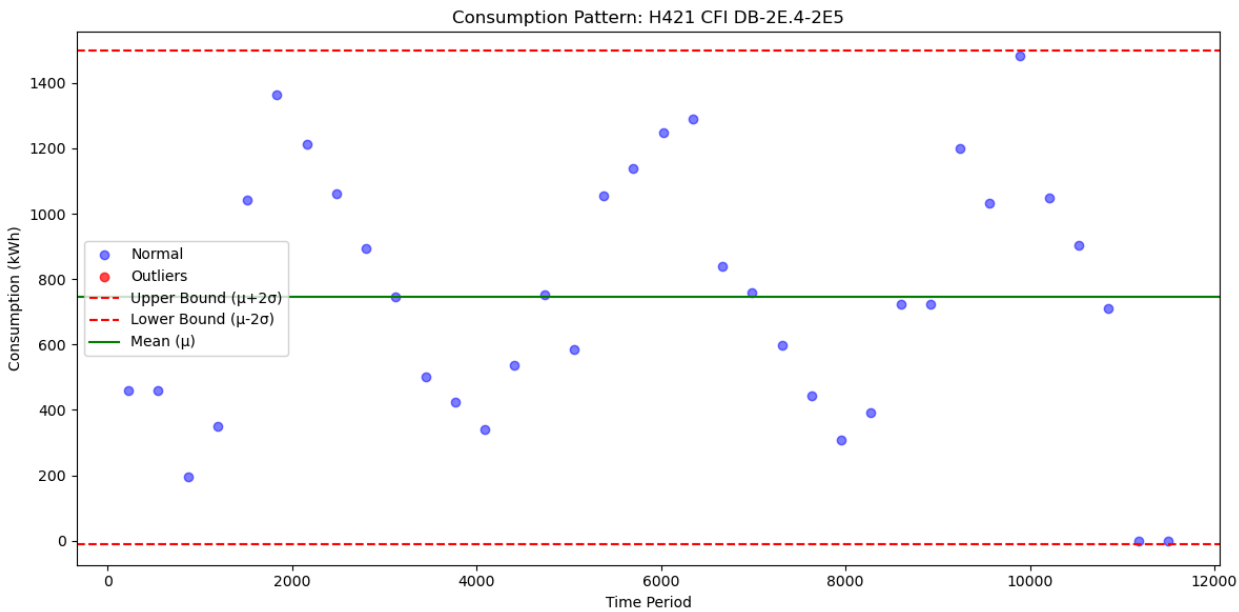
```
Statistics for H420 518 Castle Street:
Mean: 2292.56
Standard Deviation: 1421.80
Upper Bound: 5136.16
Lower Bound: -551.03
Number of outliers: 1
```



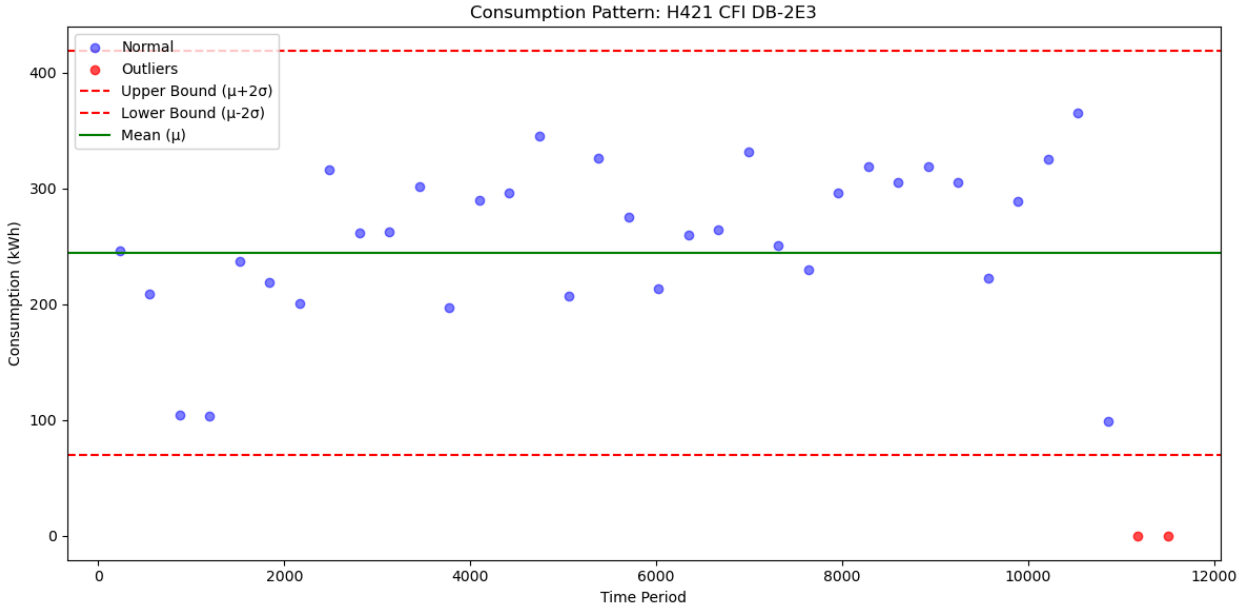

```
Statistics for H421 CFI DB-1E.3 first floor east unit 3:
Mean: 0.00
Standard Deviation: 0.00
Upper Bound: 0.00
Lower Bound: 0.00
Number of outliers: 0
```



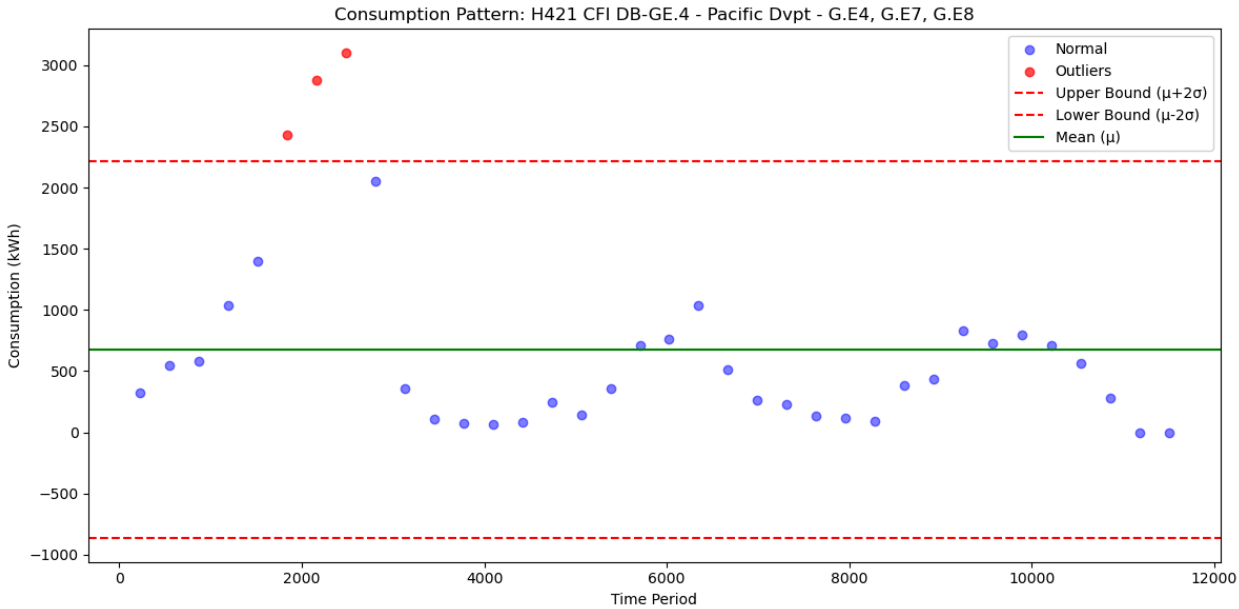

```
Statistics for H421 CFI DB-2E.4-2E5:
Mean: 745.14
Standard Deviation: 377.40
Upper Bound: 1499.95
Lower Bound: -9.67
Number of outliers: 0
```



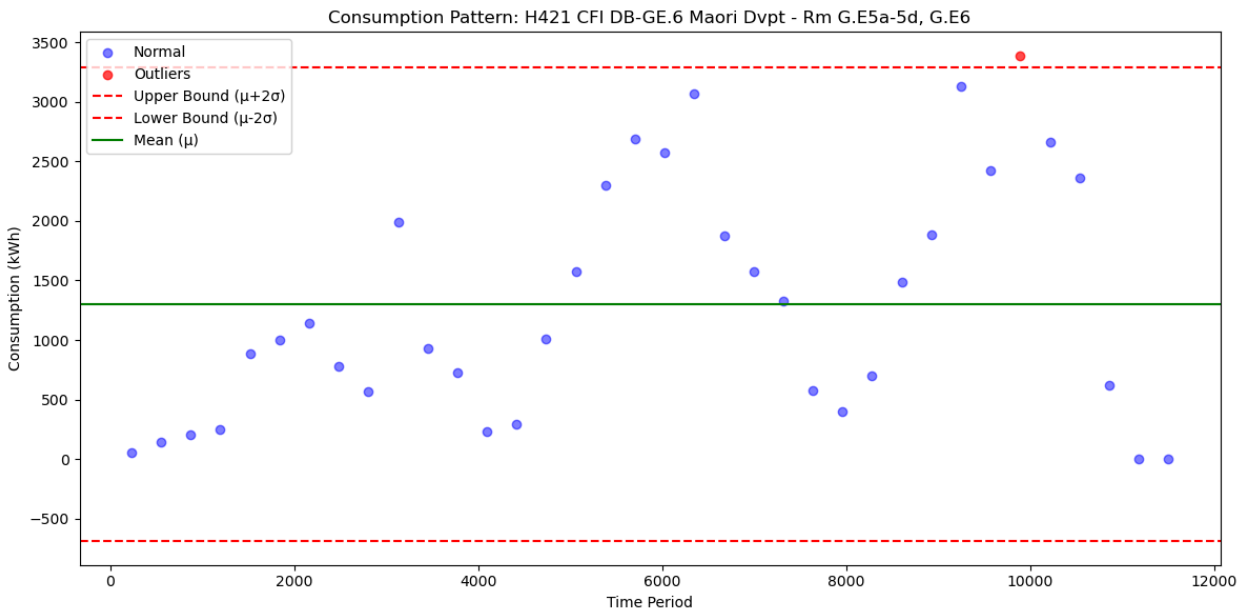

```
Statistics for H421 CFI DB-2E3:
Mean: 244.28
Standard Deviation: 87.24
Upper Bound: 418.76
Lower Bound: 69.80
Number of outliers: 2
```



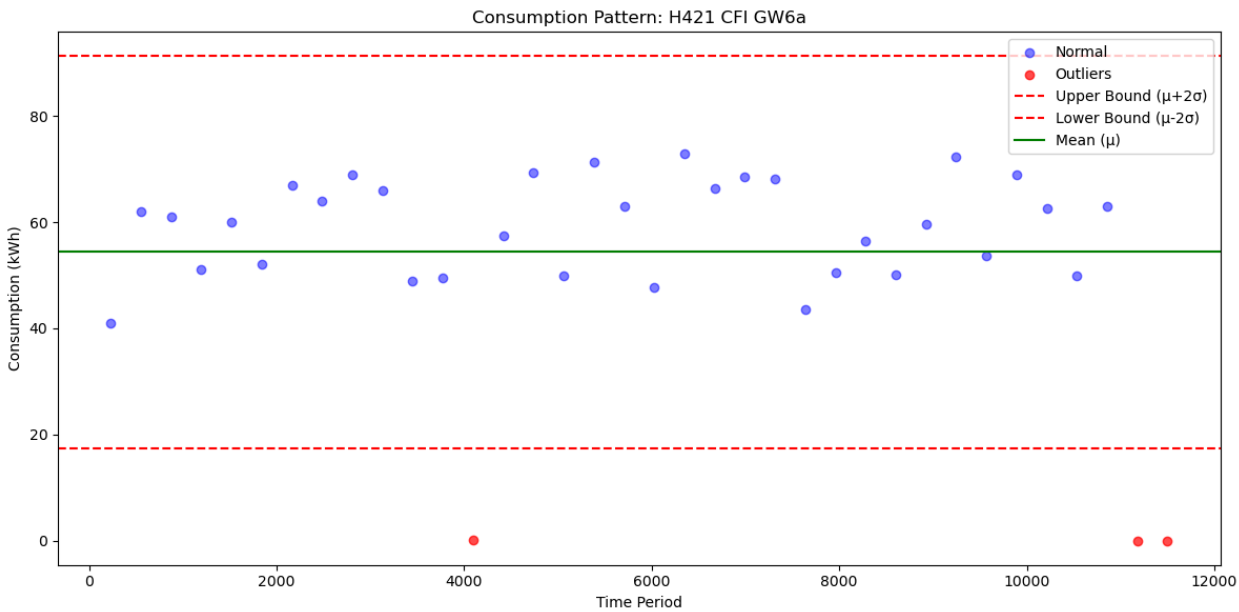

```
Statistics for H421 CFI DB-GE.4 - Pacific Dvpt - G.E4, G.E7, G.E8:
Mean: 676.91
Standard Deviation: 769.20
Upper Bound: 2215.32
Lower Bound: -861.49
Number of outliers: 3
```



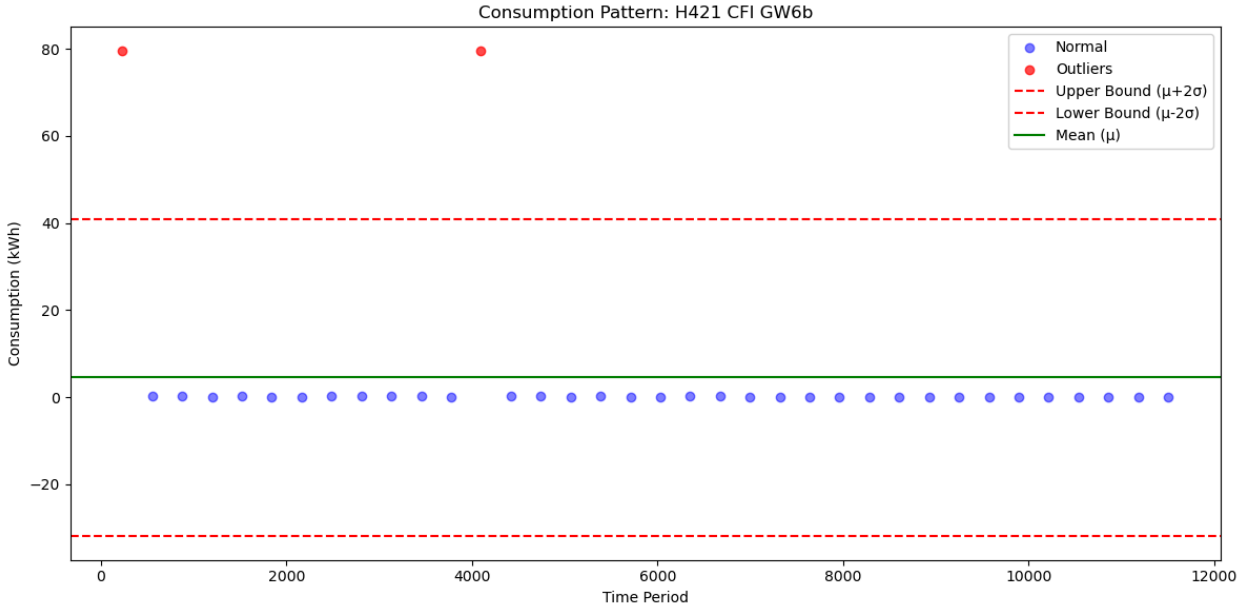

```
Statistics for H421 CFI DB-GE.6 Maori Dvpt - Rm G.E5a-5d, G.E6:
Mean: 1300.64
Standard Deviation: 992.24
Upper Bound: 3285.13
Lower Bound: -683.85
Number of outliers: 1
```



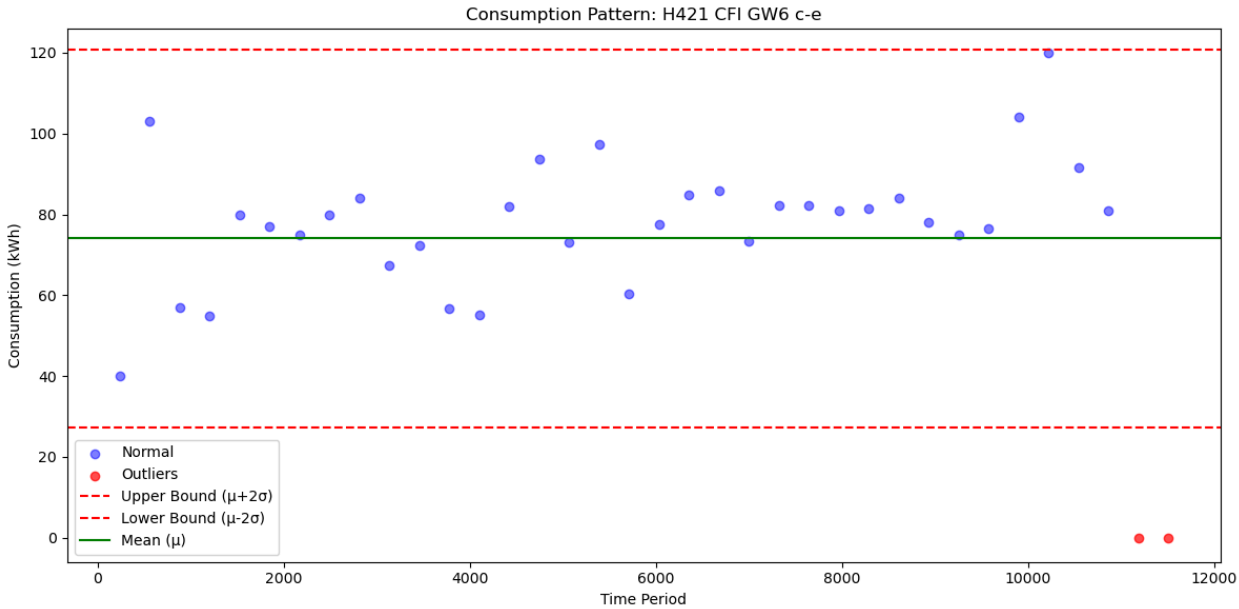

```
Statistics for H421 CFI GW6a:
Mean: 54.36
Standard Deviation: 18.45
Upper Bound: 91.26
Lower Bound: 17.47
Number of outliers: 3
```



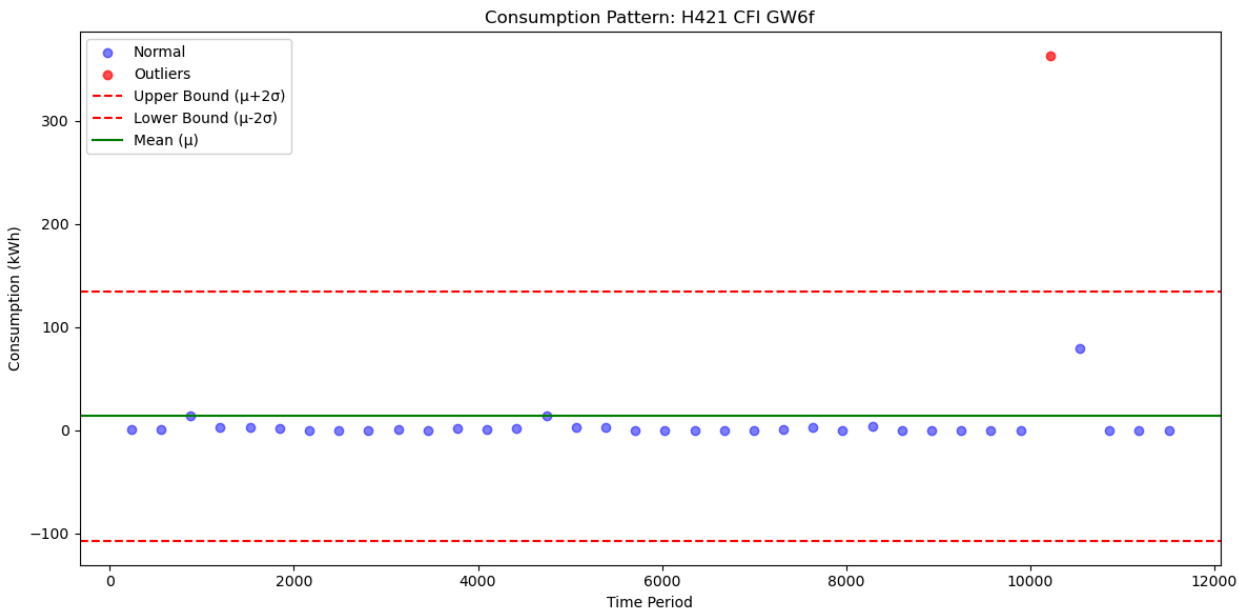

```
Statistics for H421 CFI GW6b:
Mean: 4.52
Standard Deviation: 18.17
Upper Bound: 40.86
Lower Bound: -31.82
Number of outliers: 2
```



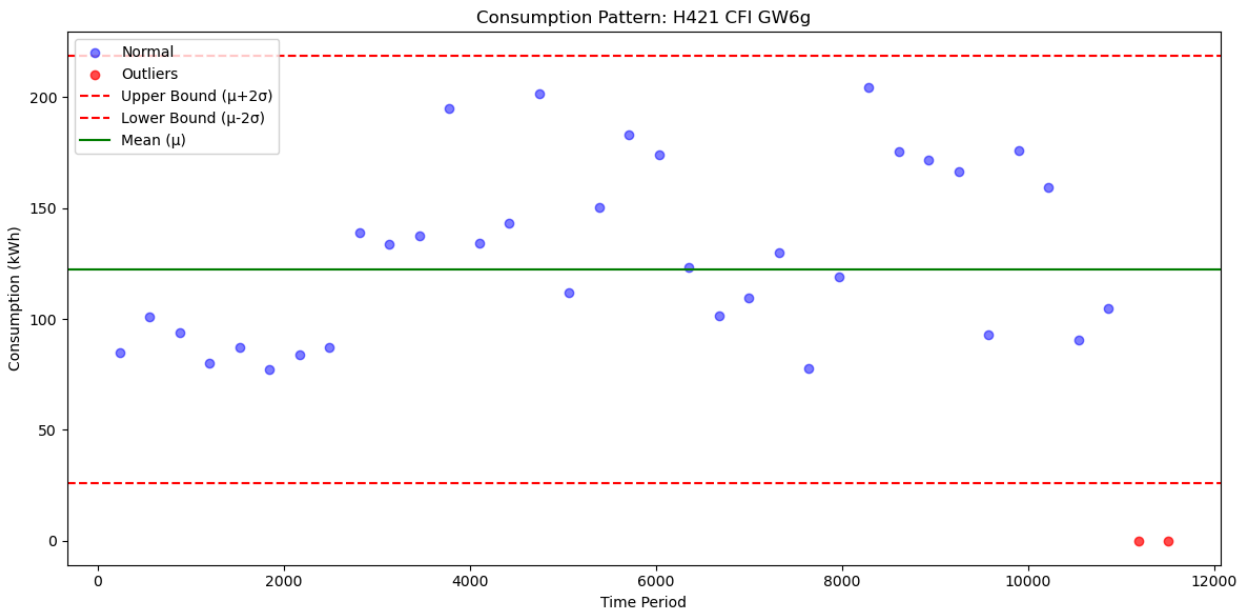

```
Statistics for H421 CFI GW6 c-e:
Mean: 74.10
Standard Deviation: 23.38
Upper Bound: 120.86
Lower Bound: 27.35
Number of outliers: 2
```



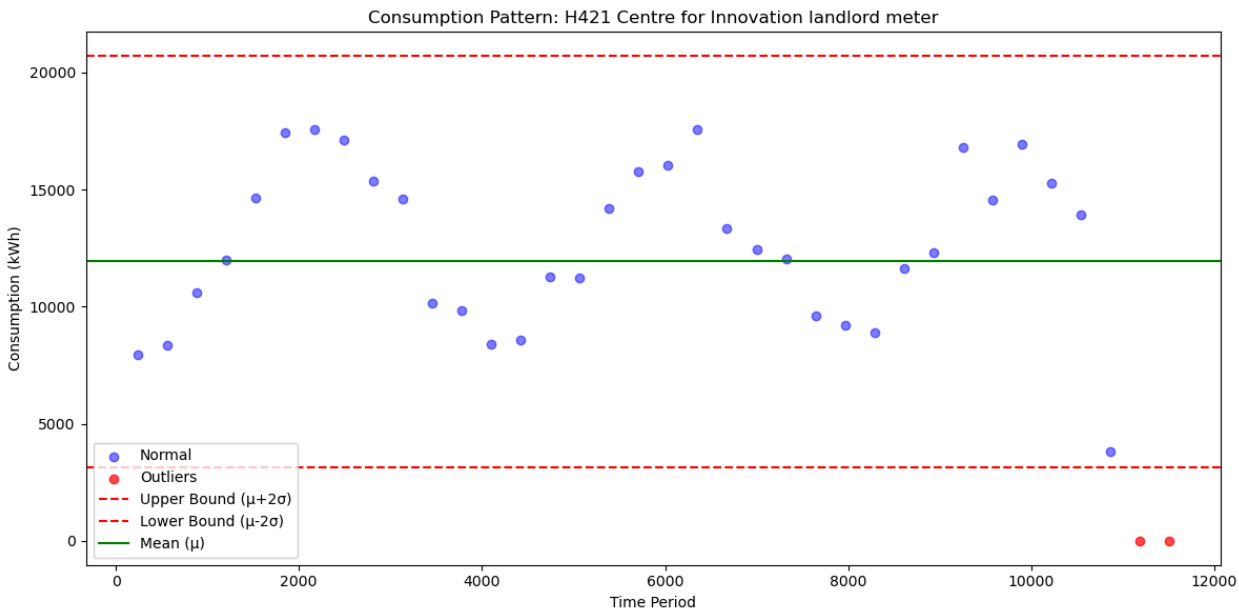

```
Statistics for H421 CFI GW6f:
Mean: 13.85
Standard Deviation: 60.37
Upper Bound: 134.59
Lower Bound: -106.88
Number of outliers: 1
```



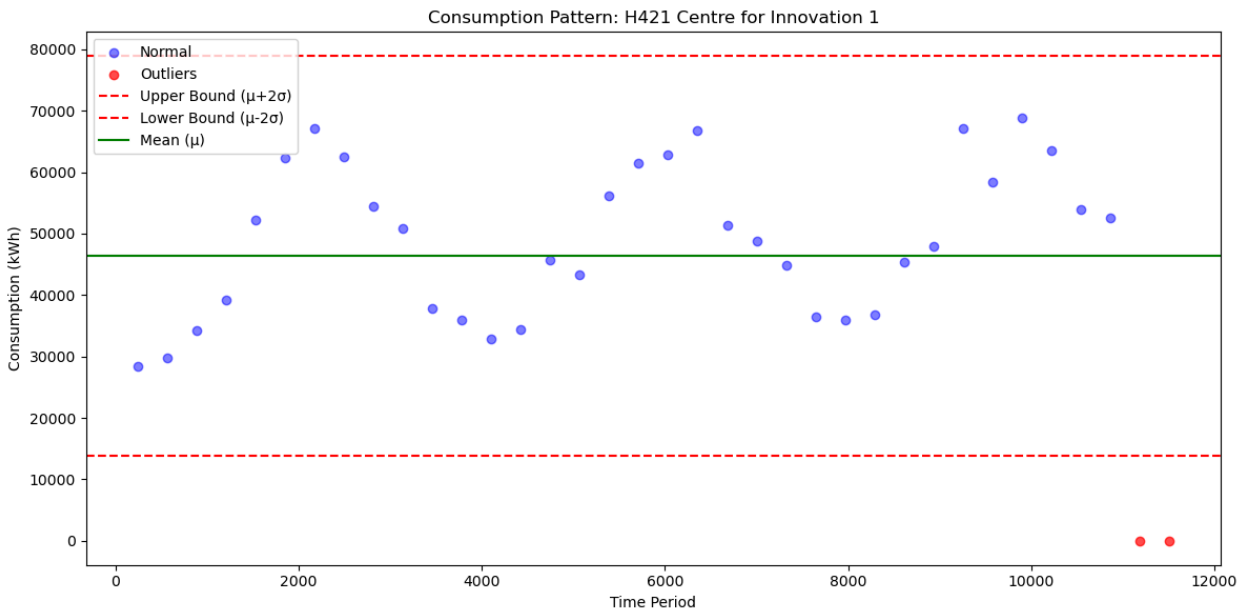

```
Statistics for H421 CFI GW6g:
Mean: 122.22
Standard Deviation: 48.15
Upper Bound: 218.52
Lower Bound: 25.93
Number of outliers: 2
```



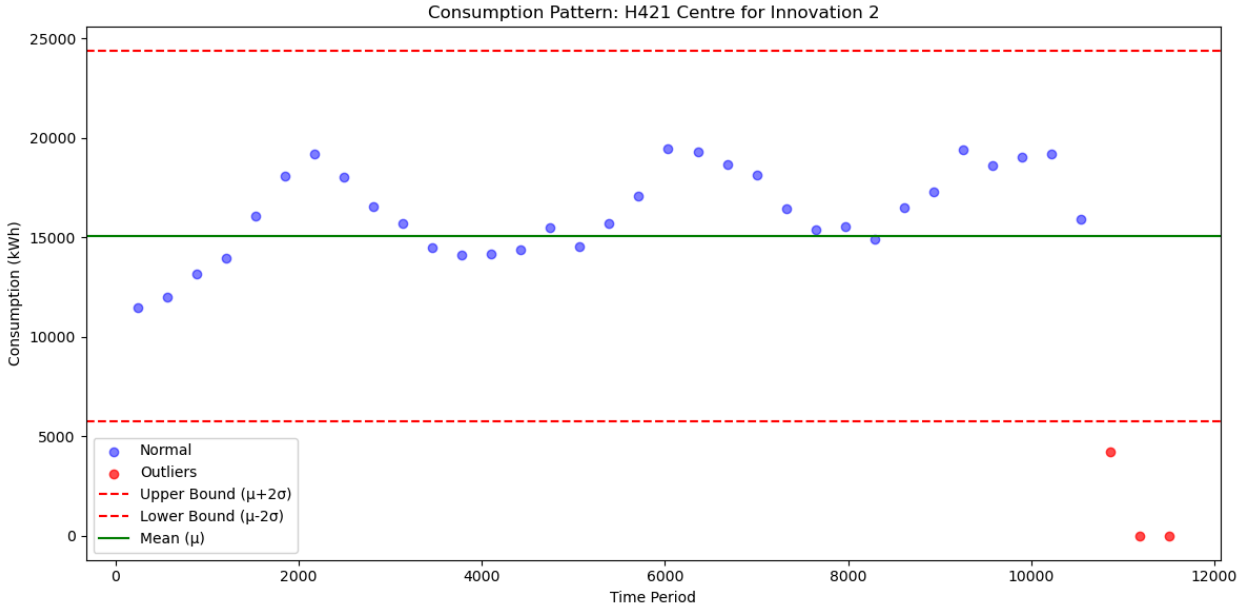

```
Statistics for H421 Centre for Innovation landlord meter:
Mean: 11928.46
Standard Deviation: 4386.01
Upper Bound: 20700.49
Lower Bound: 3156.43
Number of outliers: 2
```



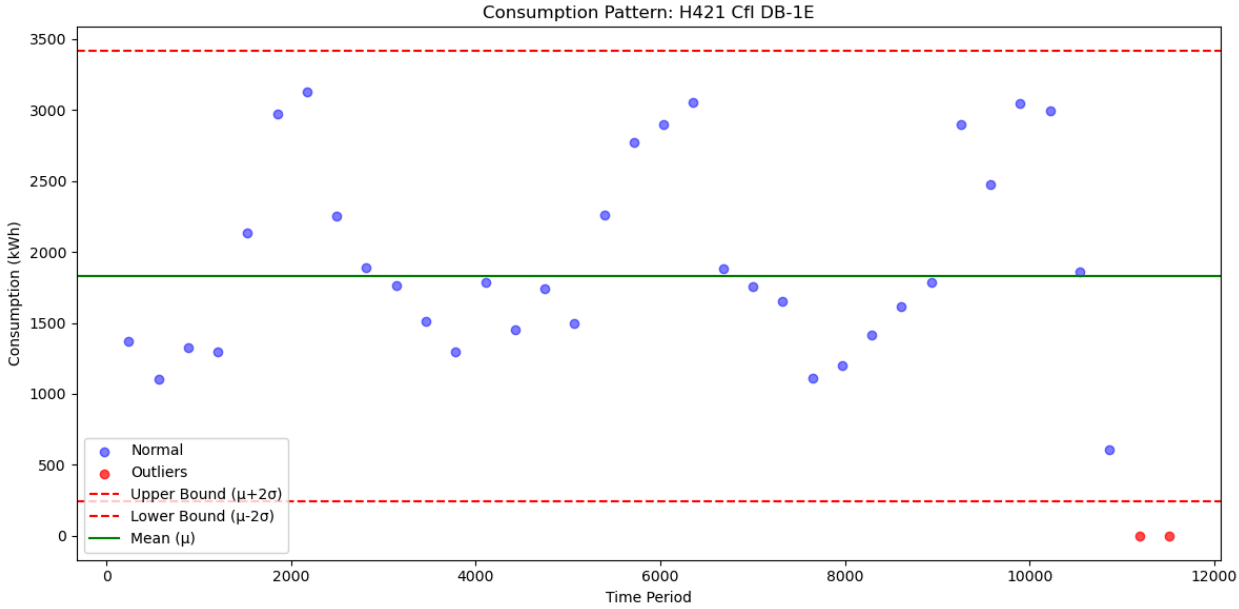

```
Statistics for H421 Centre for Innovation 1:
Mean: 46405.42
Standard Deviation: 16245.61
Upper Bound: 78896.65
Lower Bound: 13914.19
Number of outliers: 2
```



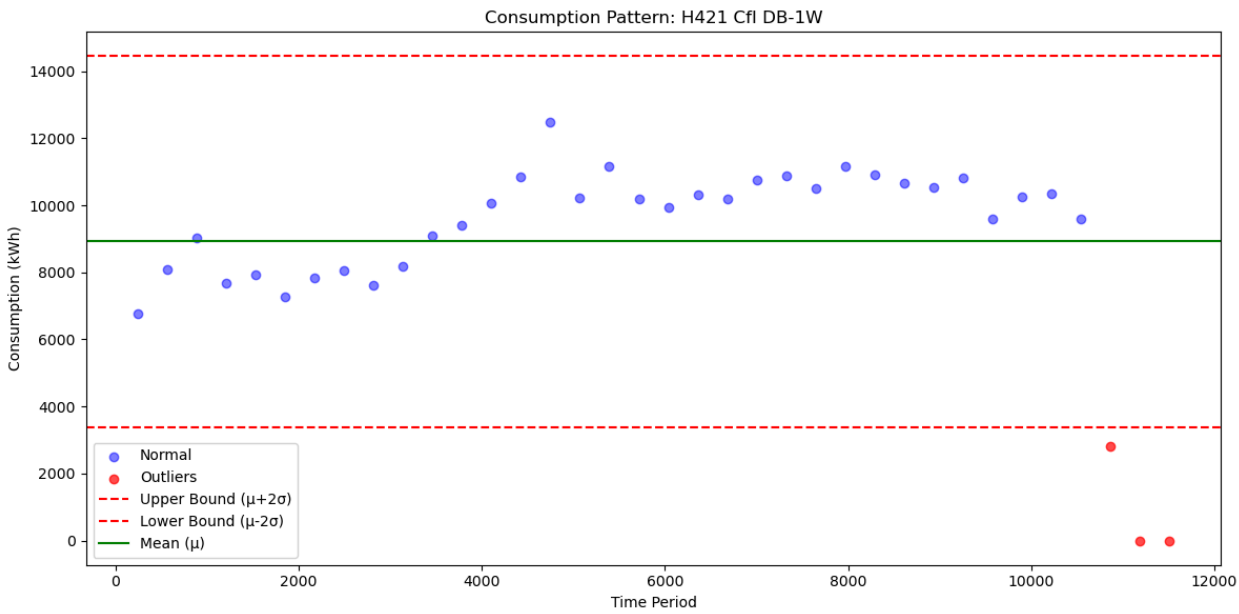

```
Statistics for H421 Centre for Innovation 2:
Mean: 15055.93
Standard Deviation: 4648.96
Upper Bound: 24353.84
Lower Bound: 5758.02
Number of outliers: 3
```



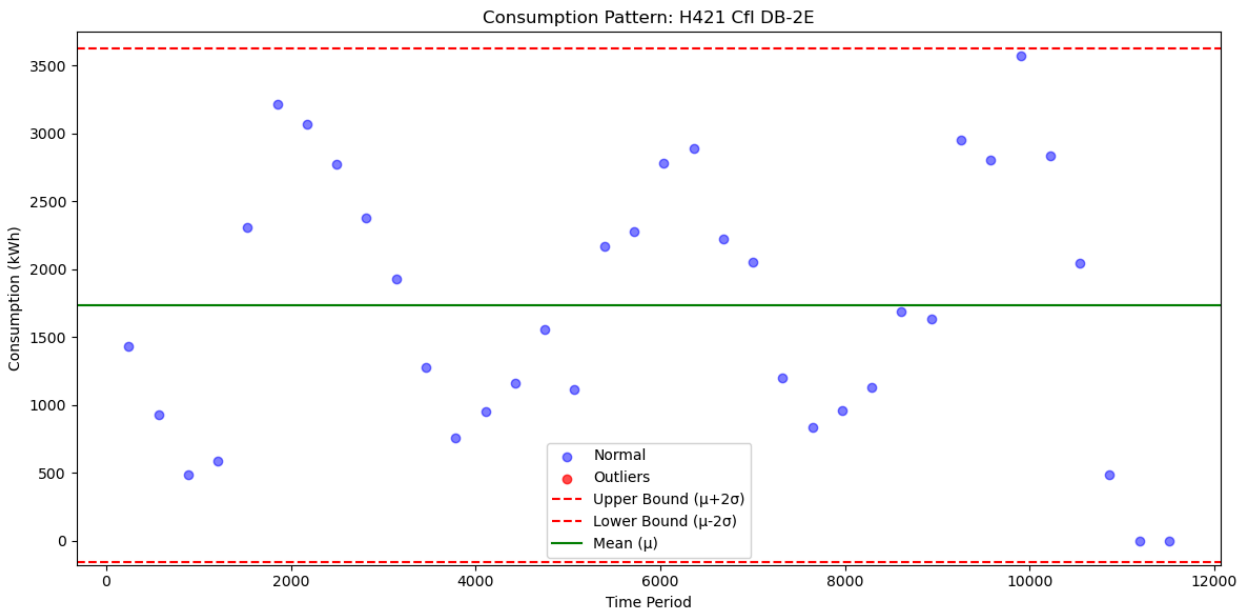

```
Statistics for H421 CfI DB-1E:
Mean: 1827.75
Standard Deviation: 793.74
Upper Bound: 3415.24
Lower Bound: 240.26
Number of outliers: 2
```



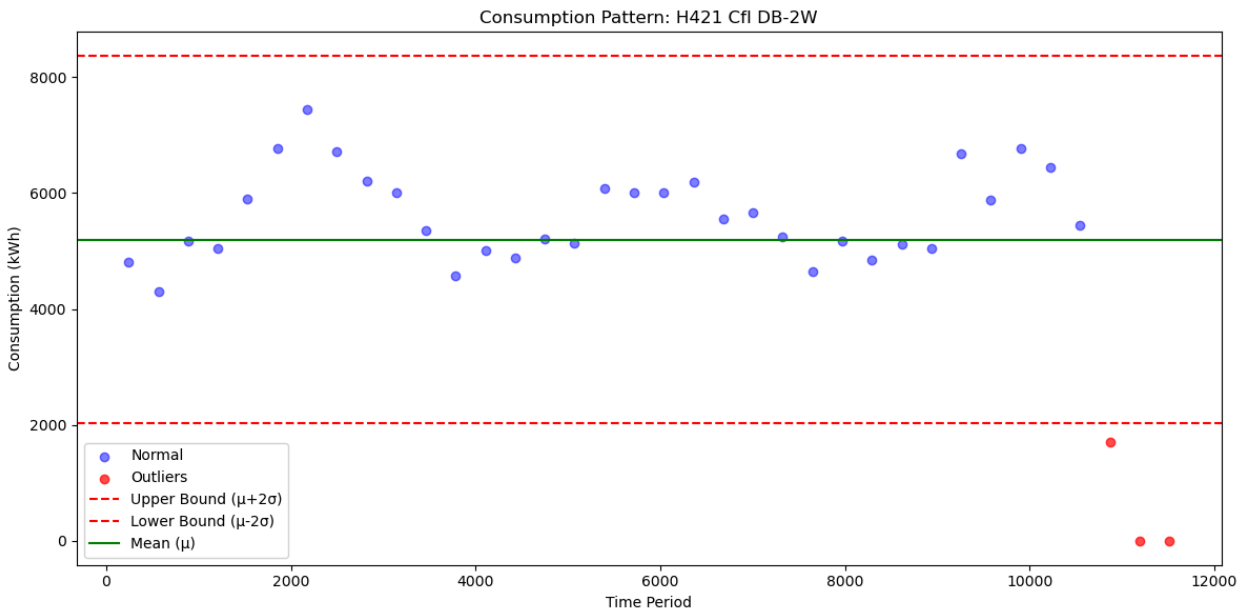

```
Statistics for H421 CfI DB-1W:
Mean: 8919.60
Standard Deviation: 2762.67
Upper Bound: 14444.93
Lower Bound: 3394.26
Number of outliers: 3
```



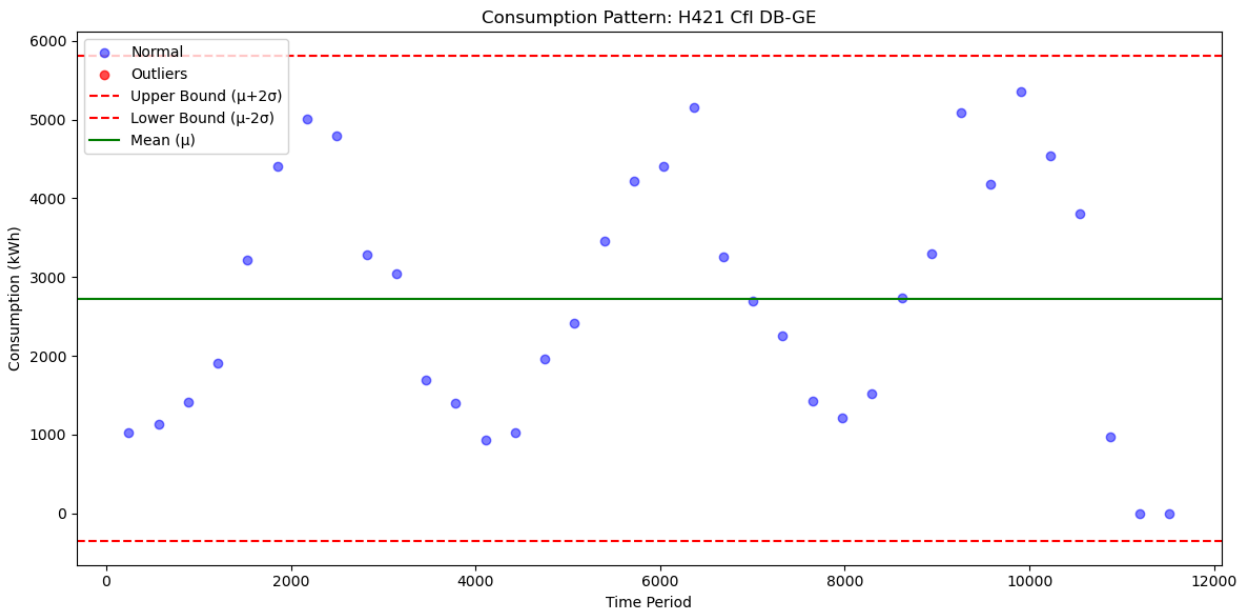

```
Statistics for H421 CfI DB-2E:
Mean: 1733.71
Standard Deviation: 944.61
Upper Bound: 3622.93
Lower Bound: -155.51
Number of outliers: 0
```



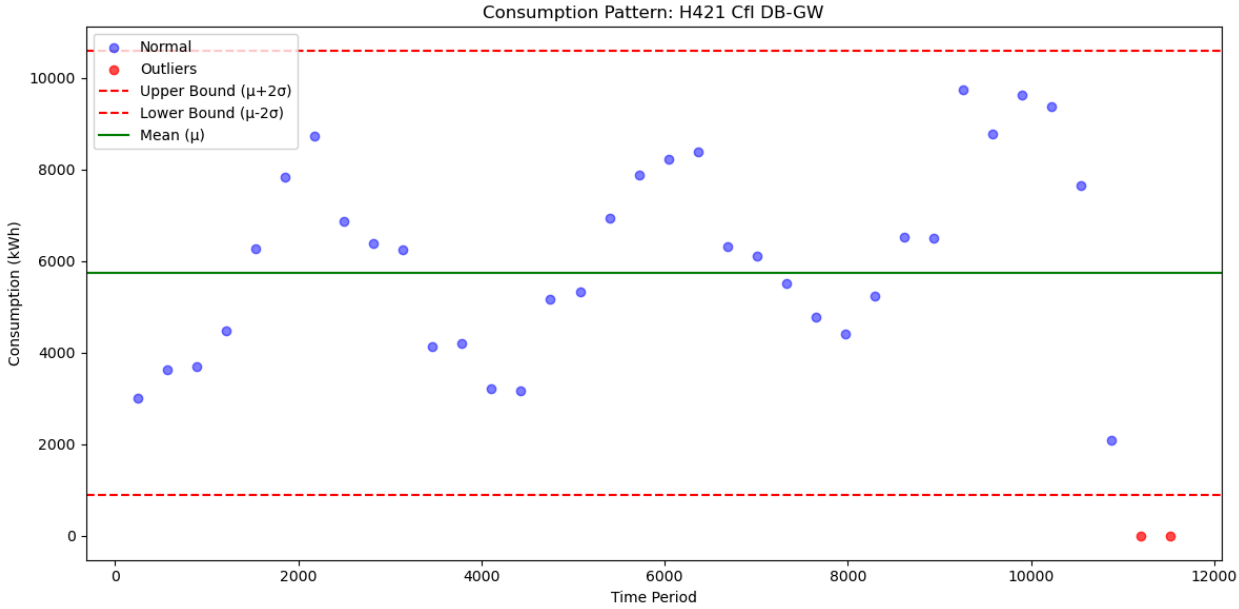

```
Statistics for H421 CfI DB-2W:
Mean: 5193.91
Standard Deviation: 1584.29
Upper Bound: 8362.48
Lower Bound: 2025.34
Number of outliers: 3
```



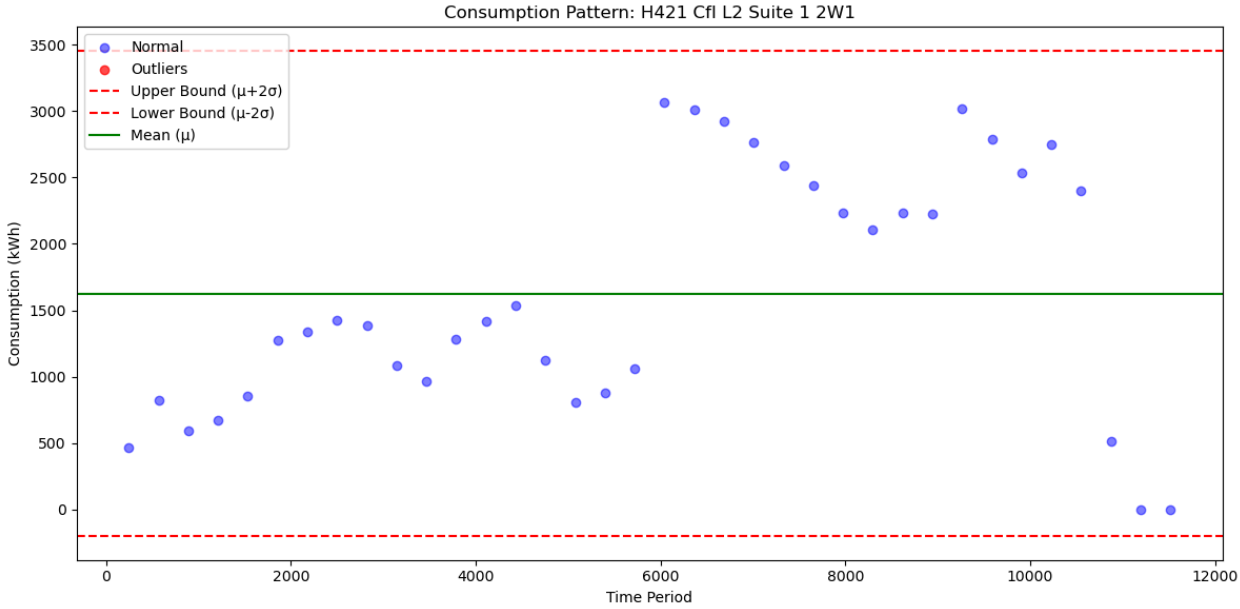

```
Statistics for H421 CfI DB-GE:
Mean: 2728.84
Standard Deviation: 1538.48
Upper Bound: 5805.80
Lower Bound: -348.13
Number of outliers: 0
```



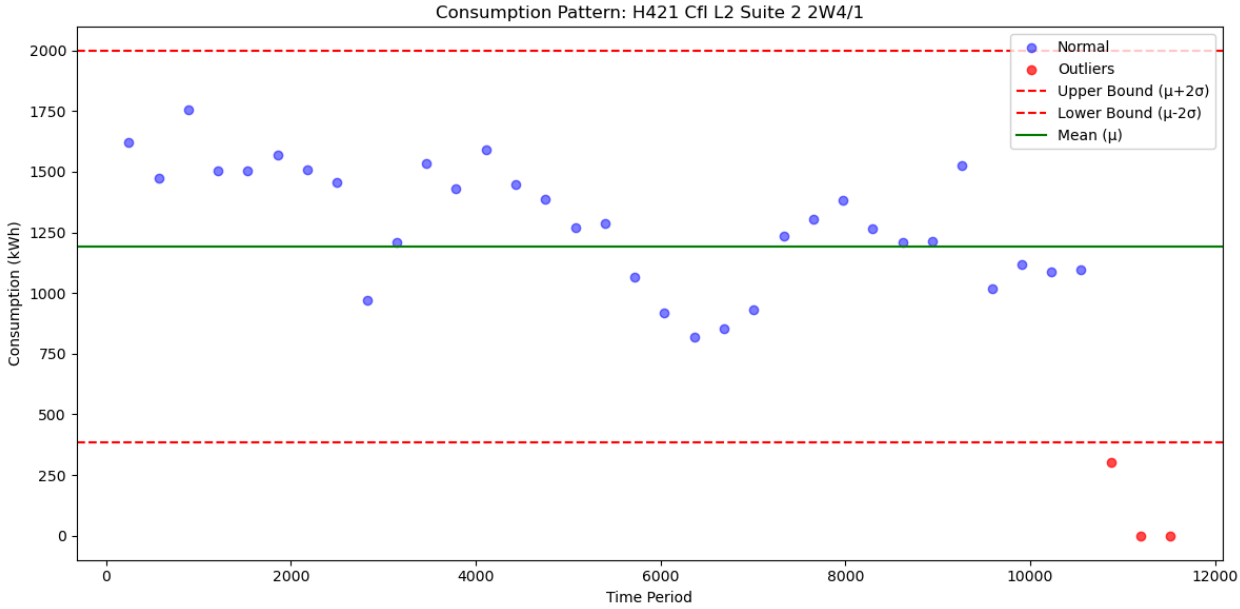

```
Statistics for H421 CfI DB-GW:
Mean: 5731.50
Standard Deviation: 2425.92
Upper Bound: 10583.34
Lower Bound: 879.65
Number of outliers: 2
```



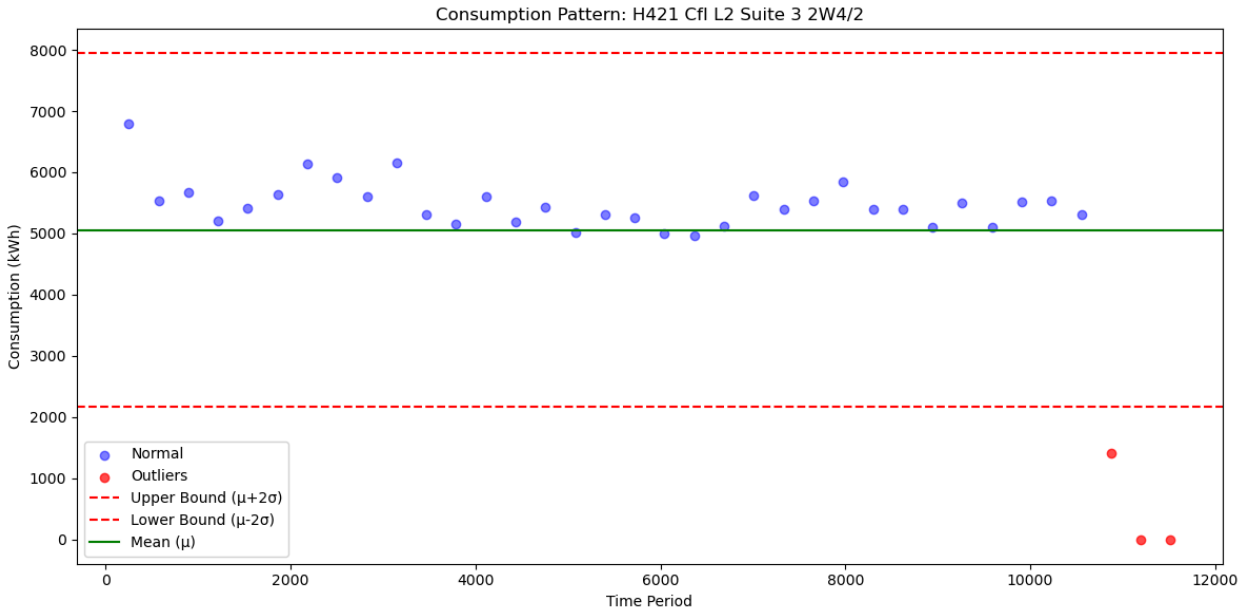

```
Statistics for H421 CfI L2 Suite 1 2W1:
Mean: 1626.91
Standard Deviation: 912.42
Upper Bound: 3451.75
Lower Bound: -197.93
Number of outliers: 0
```



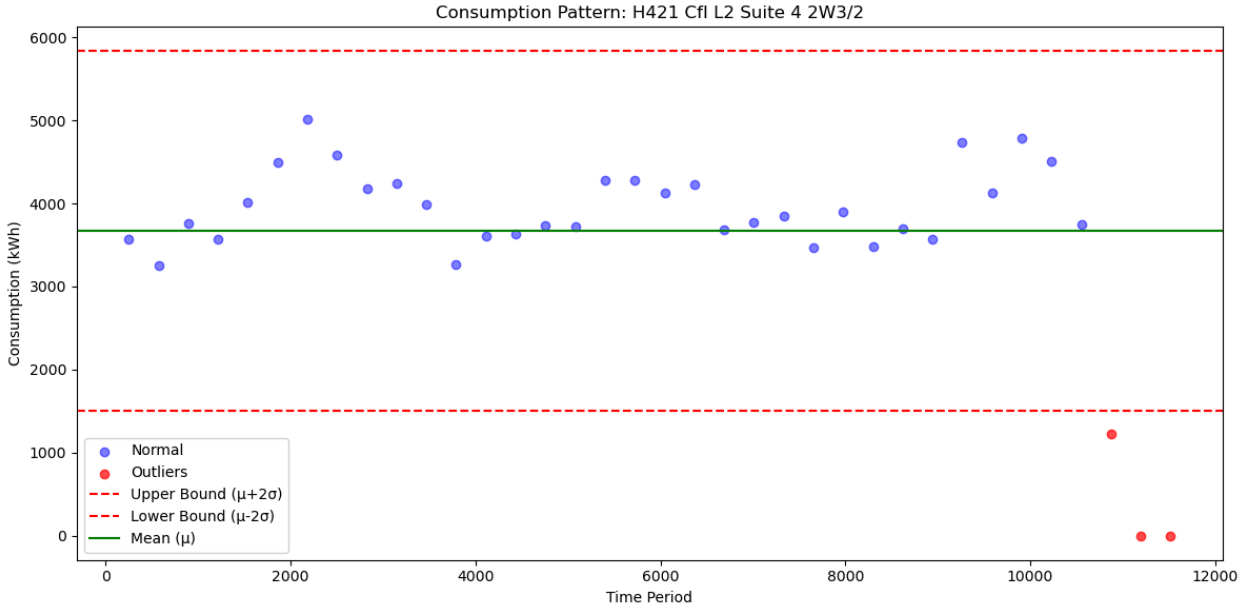

```
Statistics for H421 CfI L2 Suite 2 2W4/1:
Mean: 1190.46
Standard Deviation: 403.42
Upper Bound: 1997.30
Lower Bound: 383.61
Number of outliers: 3
```



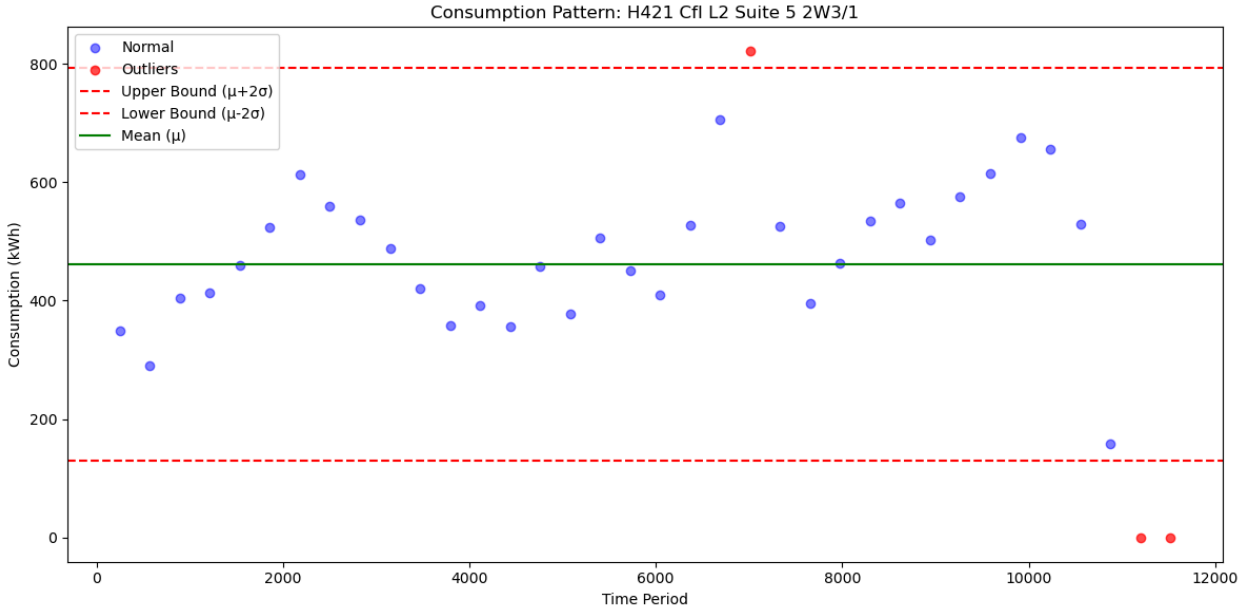

```
Statistics for H421 CfI L2 Suite 3 2W4/2:
Mean: 5058.53
Standard Deviation: 1442.24
Upper Bound: 7943.01
Lower Bound: 2174.05
Number of outliers: 3
```



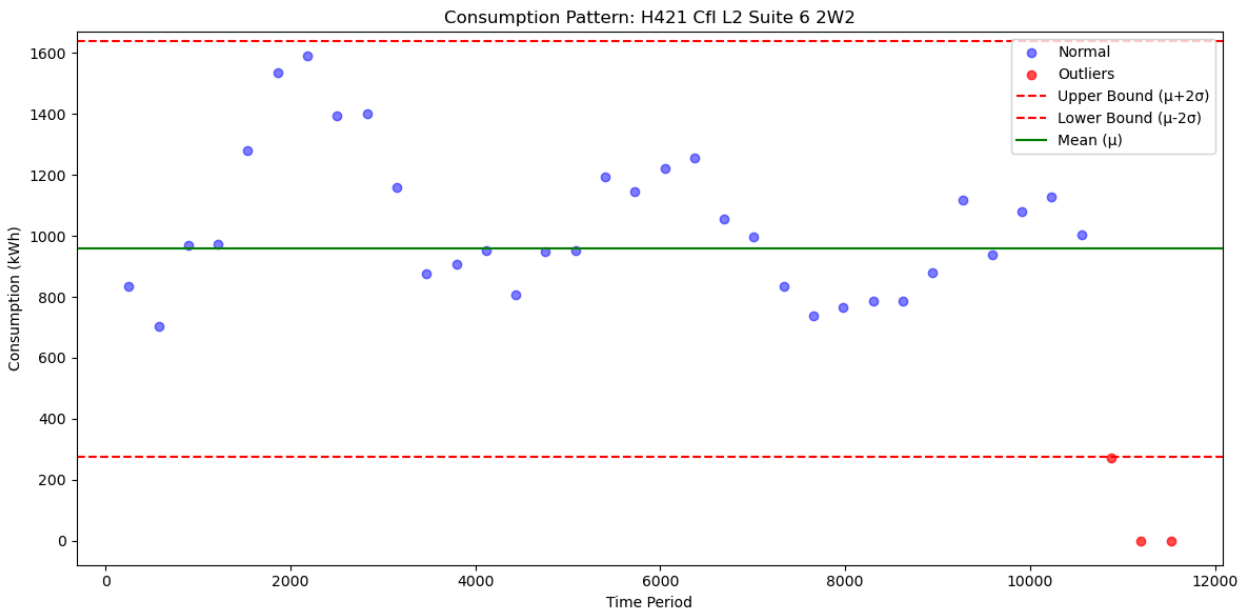

```
Statistics for H421 CfI L2 Suite 4 2W3/2:
Mean: 3668.92
Standard Deviation: 1082.68
Upper Bound: 5834.27
Lower Bound: 1503.57
Number of outliers: 3
```



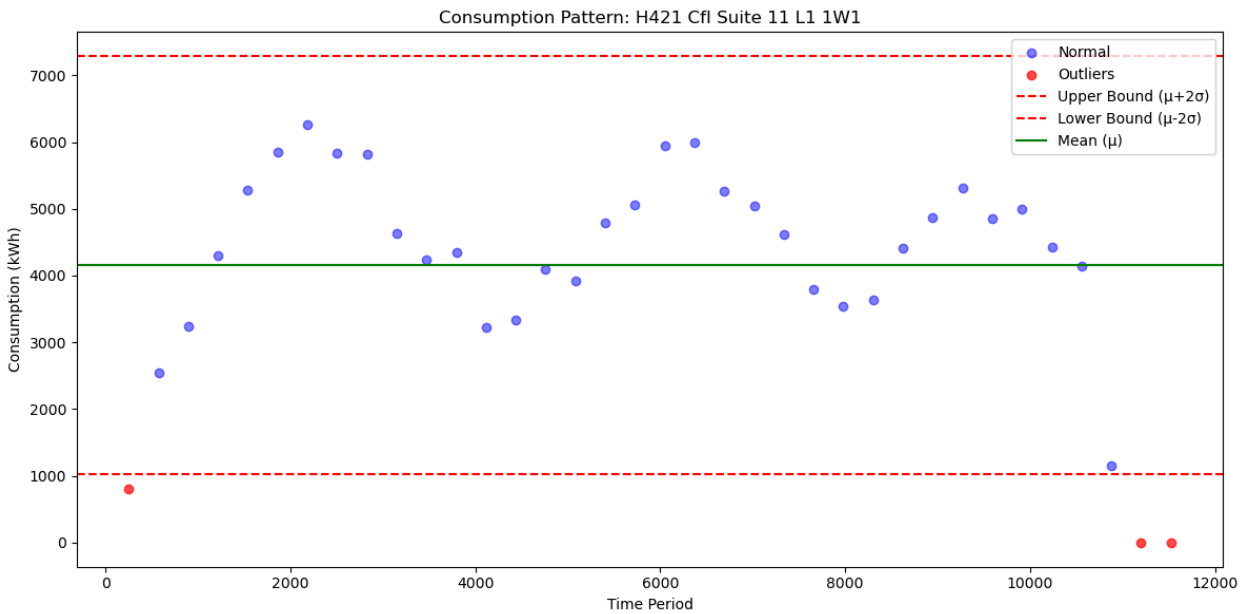

```
Statistics for H421 CfI L2 Suite 5 2W3/1:
Mean: 461.53
Standard Deviation: 166.12
Upper Bound: 793.77
Lower Bound: 129.28
Number of outliers: 3
```



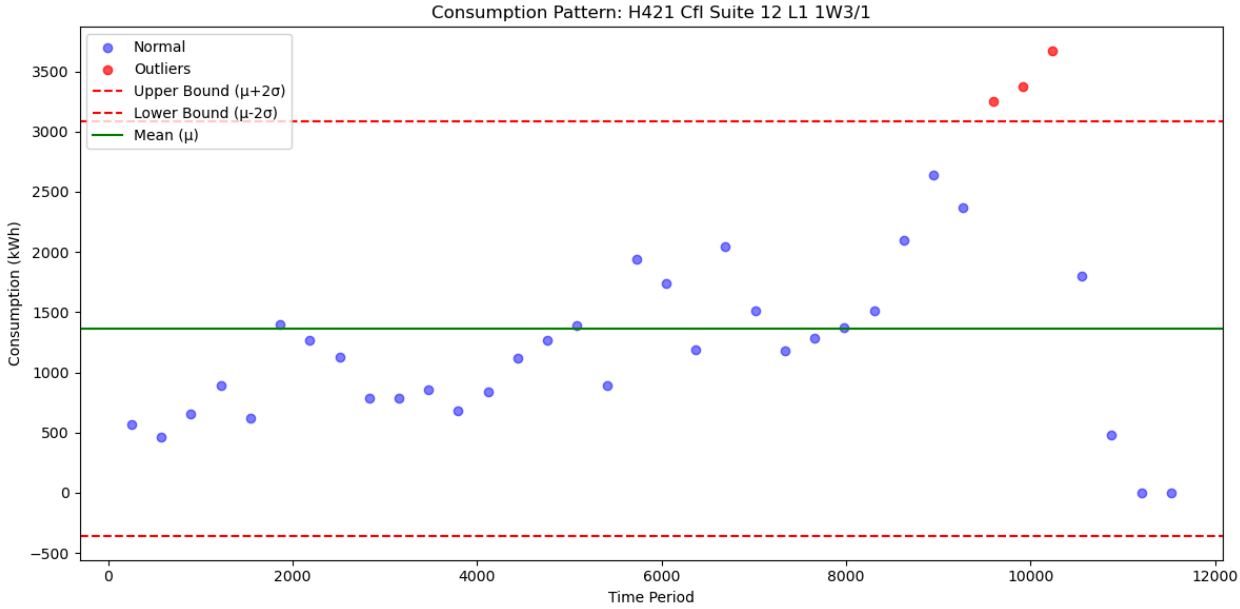

```
Statistics for H421 CfI L2 Suite 6 2W2:
Mean: 958.17
Standard Deviation: 340.87
Upper Bound: 1639.92
Lower Bound: 276.43
Number of outliers: 3
```



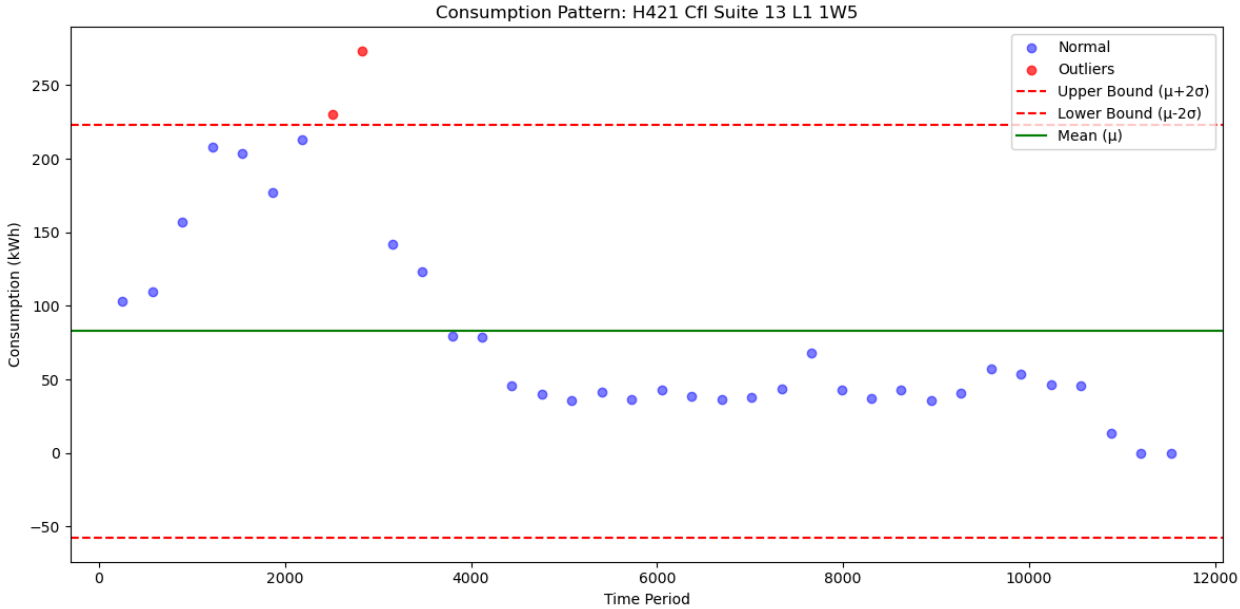

```
Statistics for H421 CfI Suite 11 L1 1W1:
Mean: 4154.69
Standard Deviation: 1565.98
Upper Bound: 7286.66
Lower Bound: 1022.73
Number of outliers: 3
```



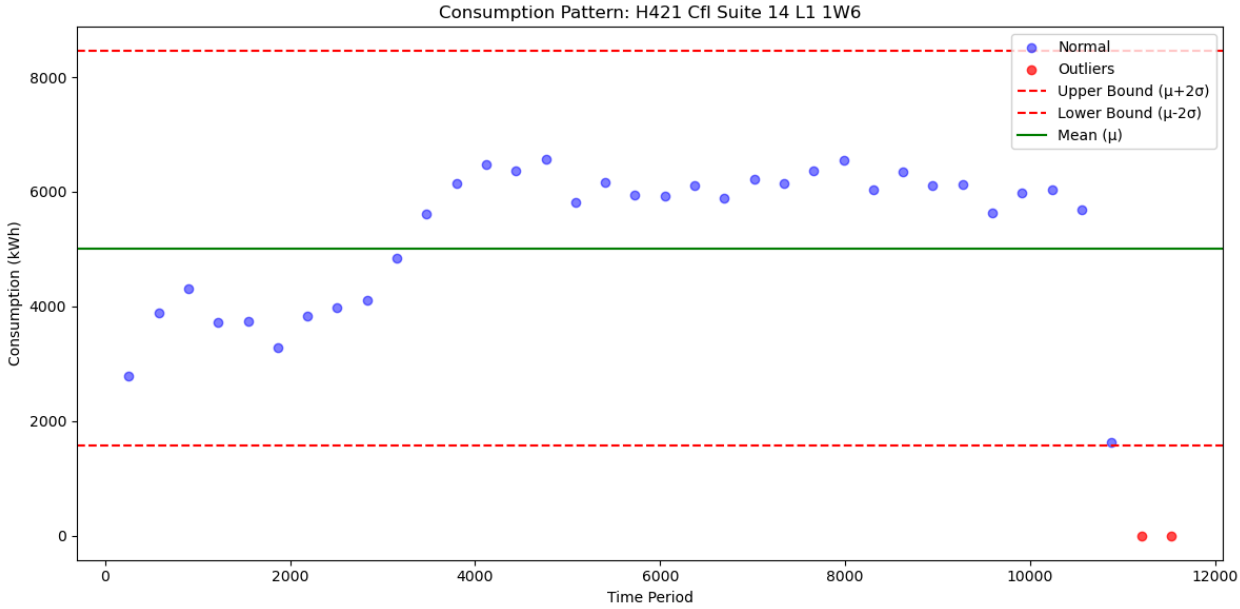

```
Statistics for H421 CfI Suite 12 L1 1W3/1:
Mean: 1363.17
Standard Deviation: 859.91
Upper Bound: 3083.00
Lower Bound: -356.66
Number of outliers: 3
```



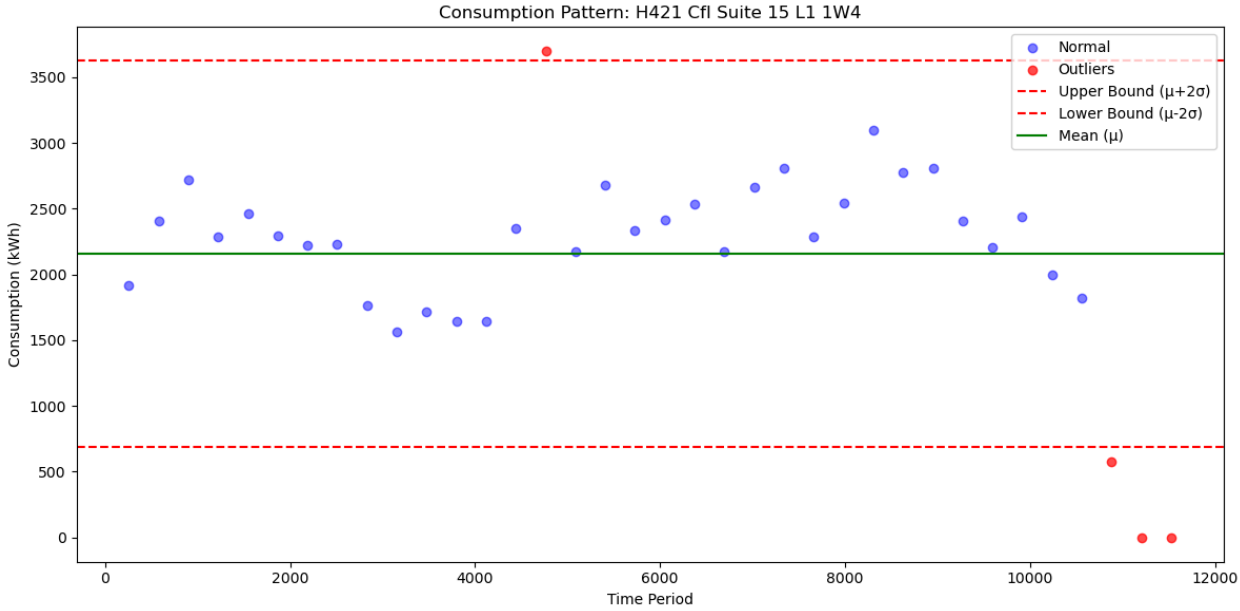

```
Statistics for H421 CfI Suite 13 L1 1W5:
Mean: 82.75
Standard Deviation: 70.05
Upper Bound: 222.85
Lower Bound: -57.34
Number of outliers: 2
```



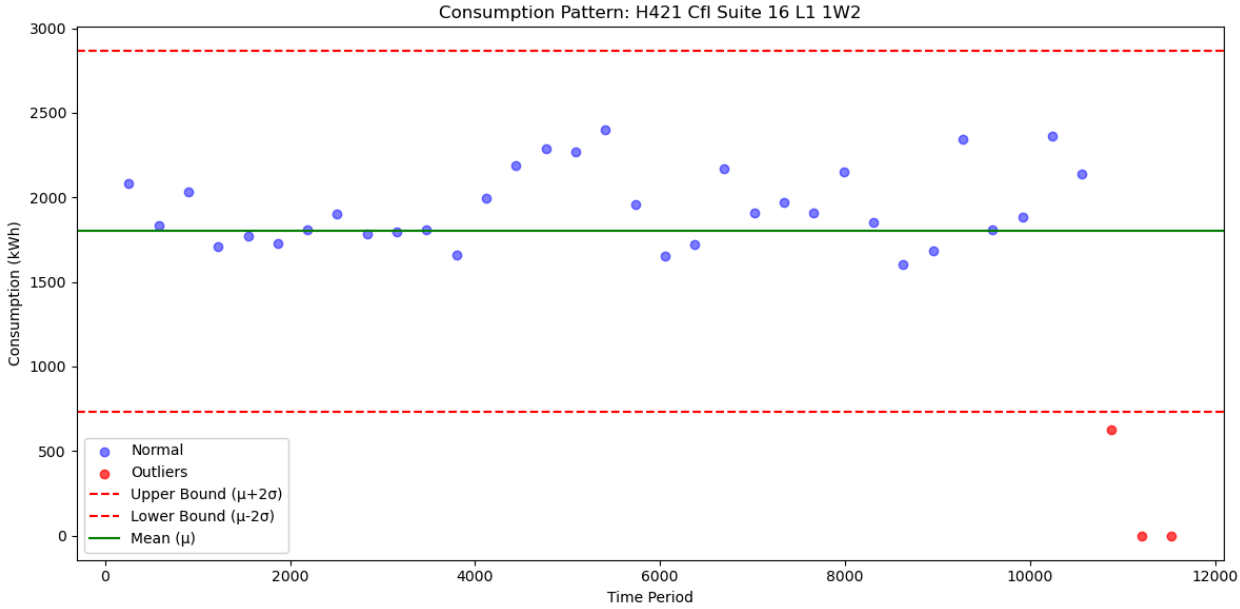

```
Statistics for H421 CfI Suite 14 L1 1W6:
Mean: 5009.02
Standard Deviation: 1721.79
Upper Bound: 8452.60
Lower Bound: 1565.43
Number of outliers: 2
```




```
Statistics for H421 CfI Suite 15 L1 1W4:
Mean: 2157.10
Standard Deviation: 733.28
Upper Bound: 3623.67
Lower Bound: 690.54
Number of outliers: 4
```

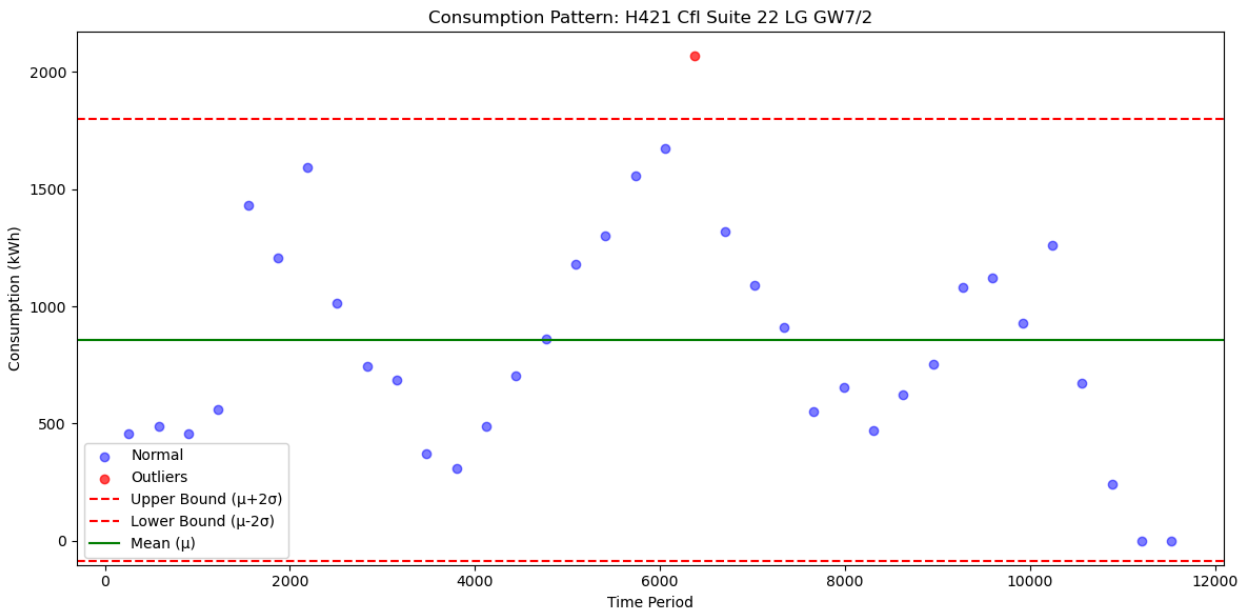



```
Statistics for H421 CfI Suite 16 L1 1W2:
Mean: 1799.92
Standard Deviation: 532.23
Upper Bound: 2864.37
Lower Bound: 735.46
Number of outliers: 3
```

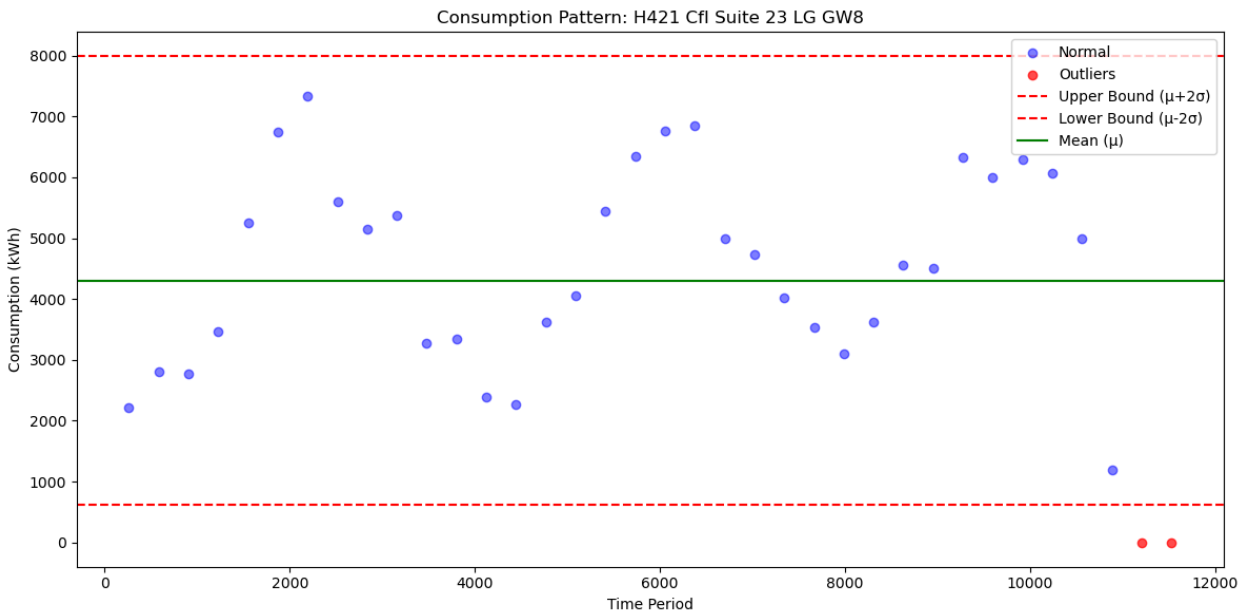



```
Statistics for H421 CfI Suite 21 LG GW7/1:
Mean: 805.84
Standard Deviation: 352.74
Upper Bound: 1511.31
Lower Bound: 100.36
Number of outliers: 2
```

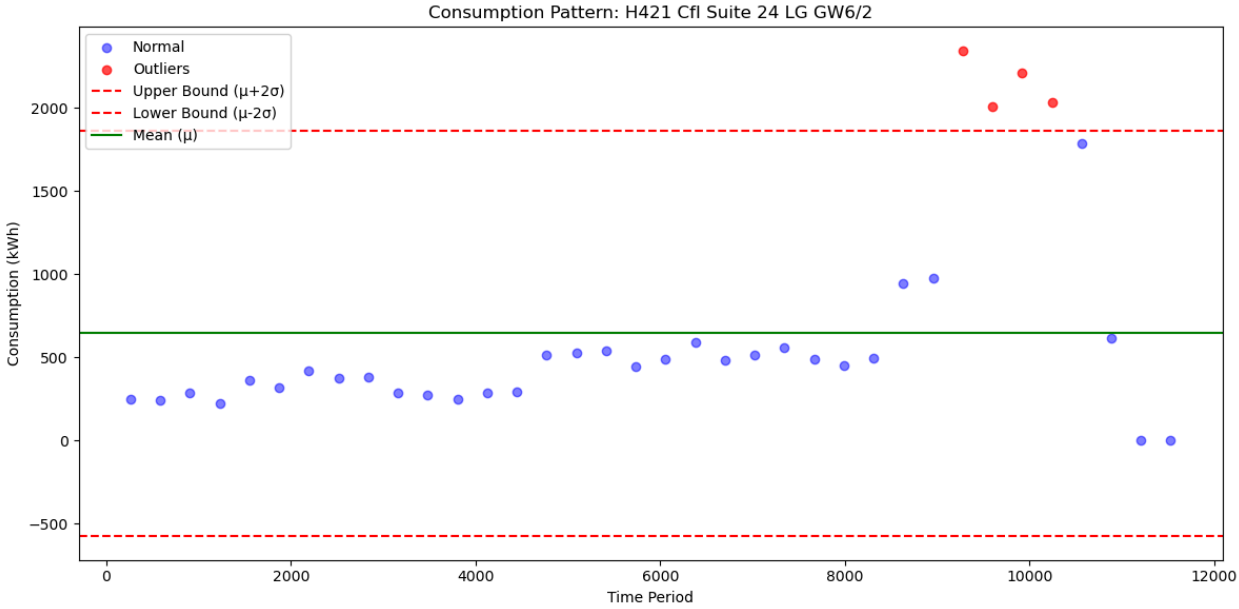
```
Statistics for H421 CfI Suite 22 LG GW7/2:
Mean: 856.10
Standard Deviation: 472.04
Upper Bound: 1800.17
Lower Bound: -87.98
Number of outliers: 1
```



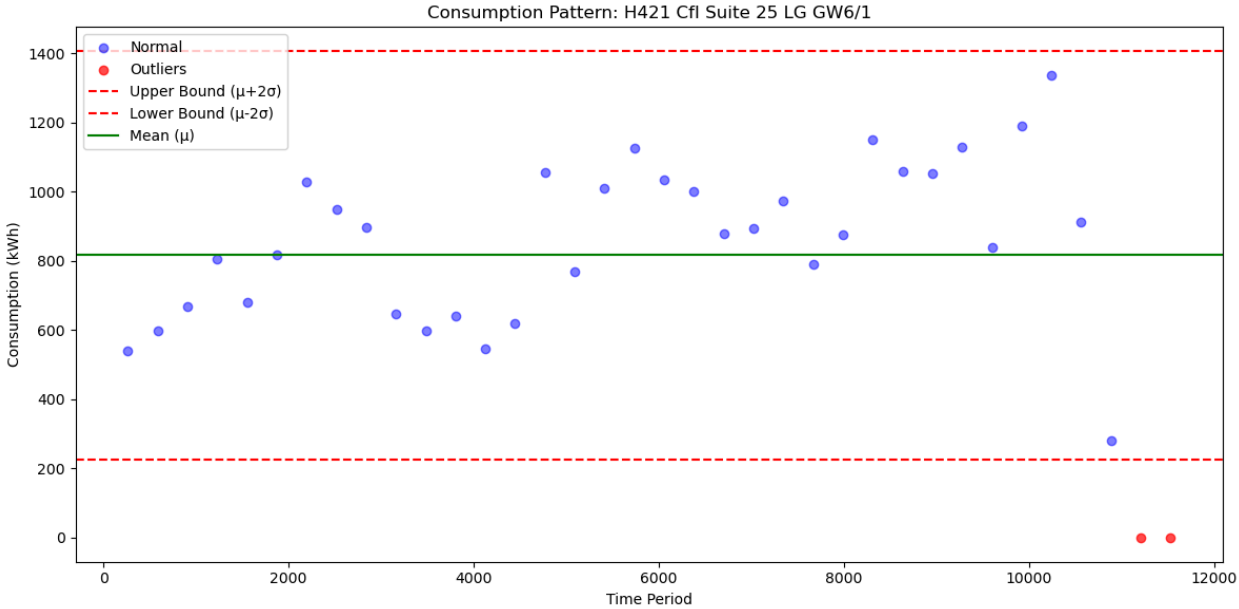

```
Statistics for H421 CfI Suite 23 LG GW8:
Mean: 4306.58
Standard Deviation: 1840.84
Upper Bound: 7988.26
Lower Bound: 624.90
Number of outliers: 2
```



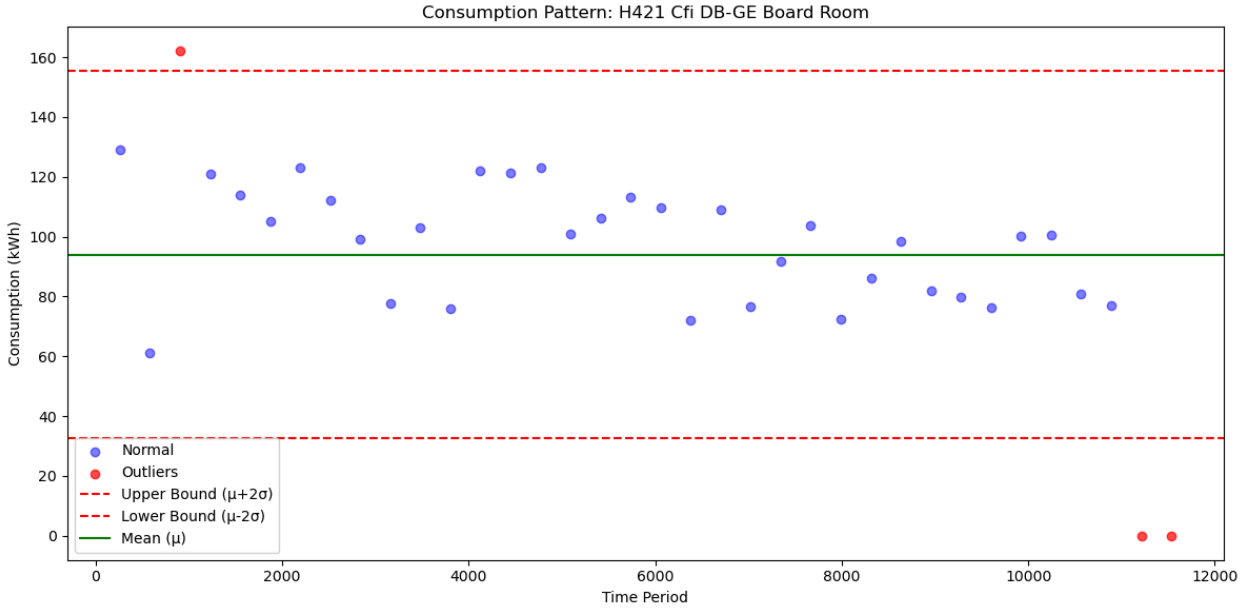

```
Statistics for H421 CfI Suite 24 LG GW6/2:
Mean: 645.21
Standard Deviation: 609.44
Upper Bound: 1864.09
Lower Bound: -573.67
Number of outliers: 4
```



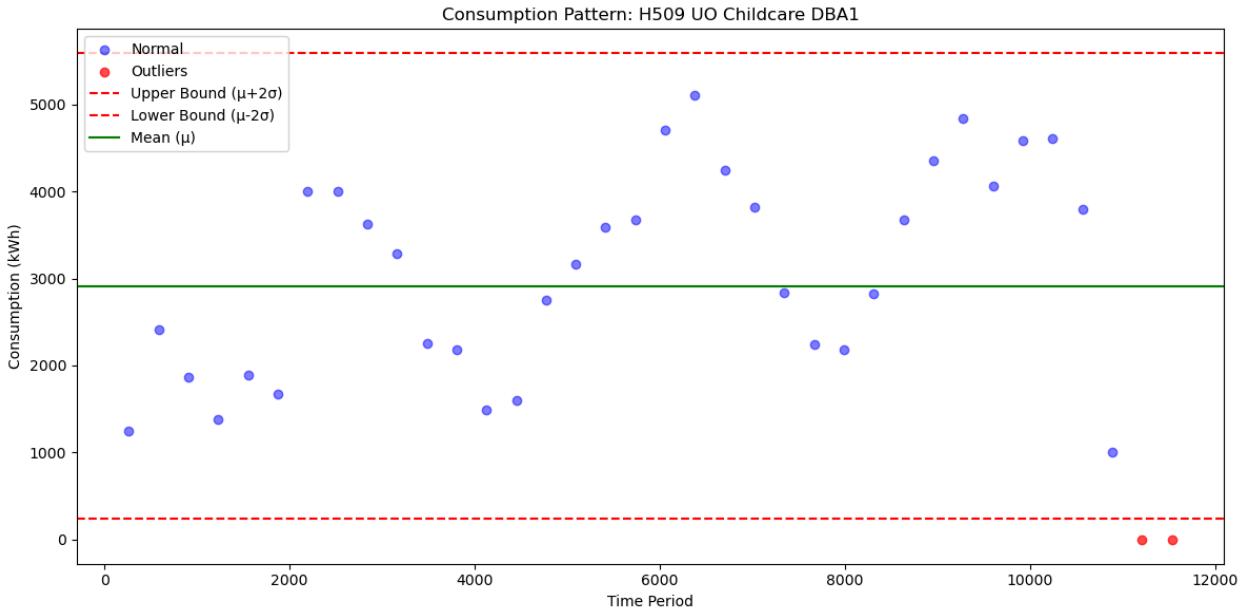

```
Statistics for H421 CfI Suite 25 LG GW6/1:
Mean: 815.68
Standard Deviation: 294.69
Upper Bound: 1405.06
Lower Bound: 226.31
Number of outliers: 2
```



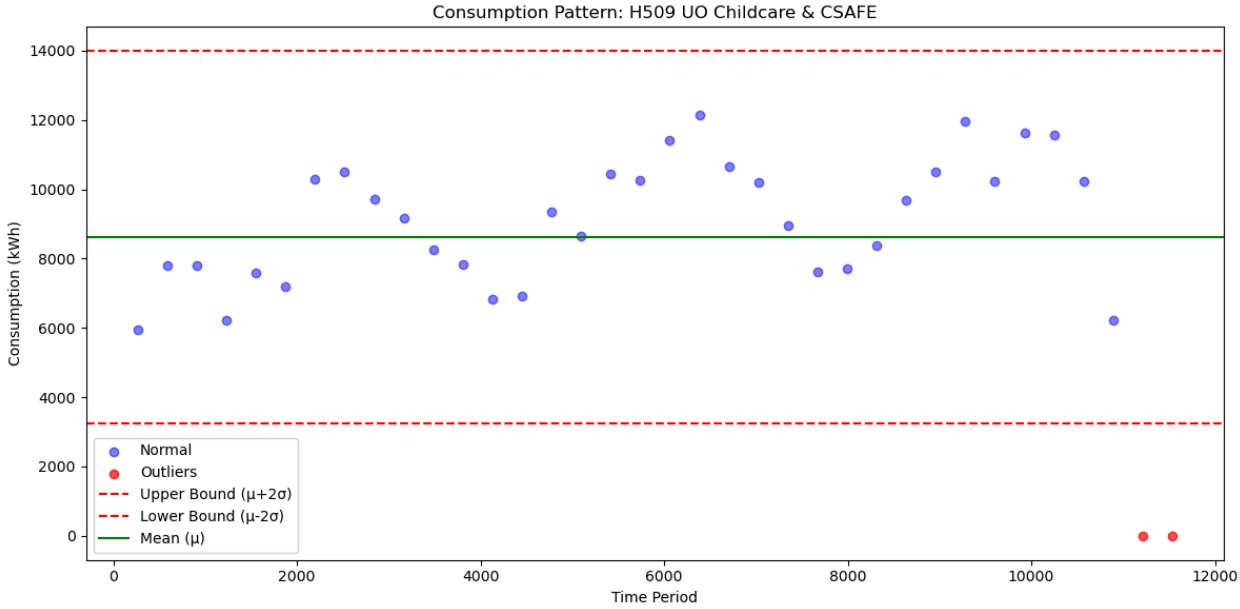

```
Statistics for H421 Cfi DB-GE Board Room:
Mean: 94.02
Standard Deviation: 30.63
Upper Bound: 155.29
Lower Bound: 32.76
Number of outliers: 3
```



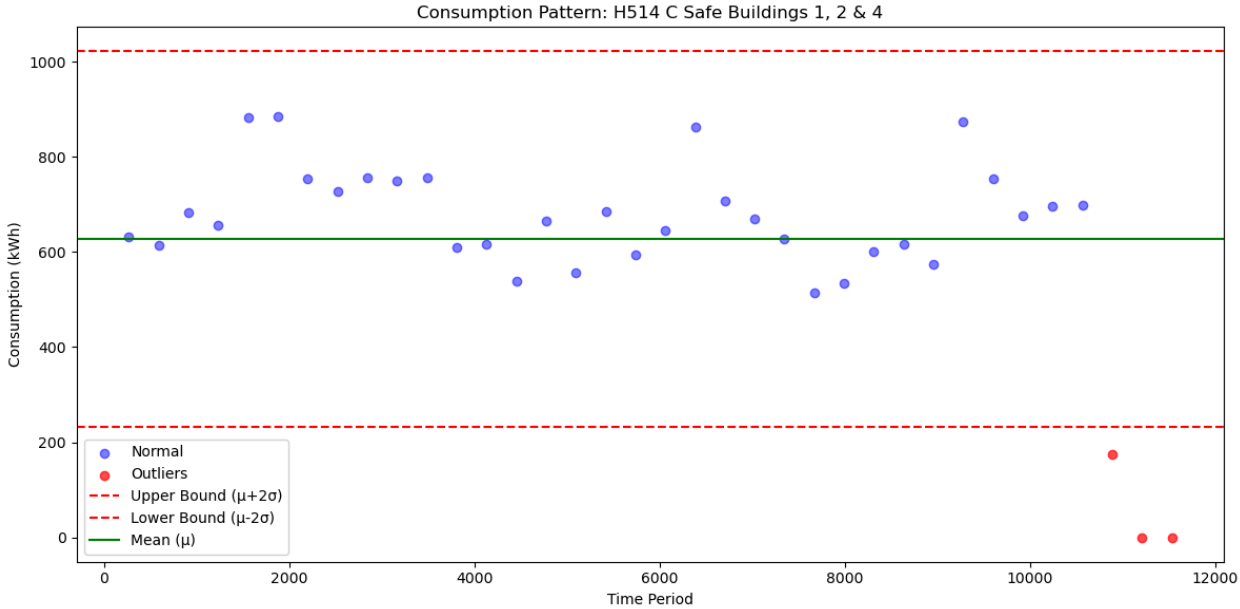

```
Statistics for H509 UO Childcare DBA1:
Mean: 2914.77
Standard Deviation: 1336.09
Upper Bound: 5586.95
Lower Bound: 242.59
Number of outliers: 2
```



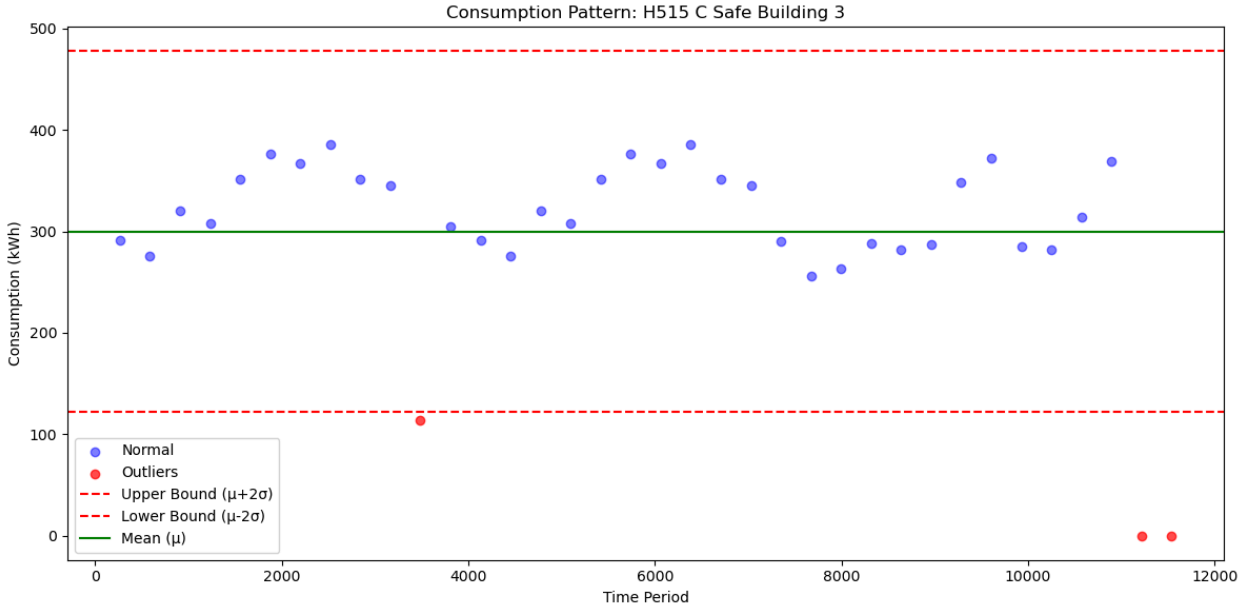

```
Statistics for H509 UO Childcare & CSAFE:
Mean: 8608.45
Standard Deviation: 2689.98
Upper Bound: 13988.41
Lower Bound: 3228.49
Number of outliers: 2
```



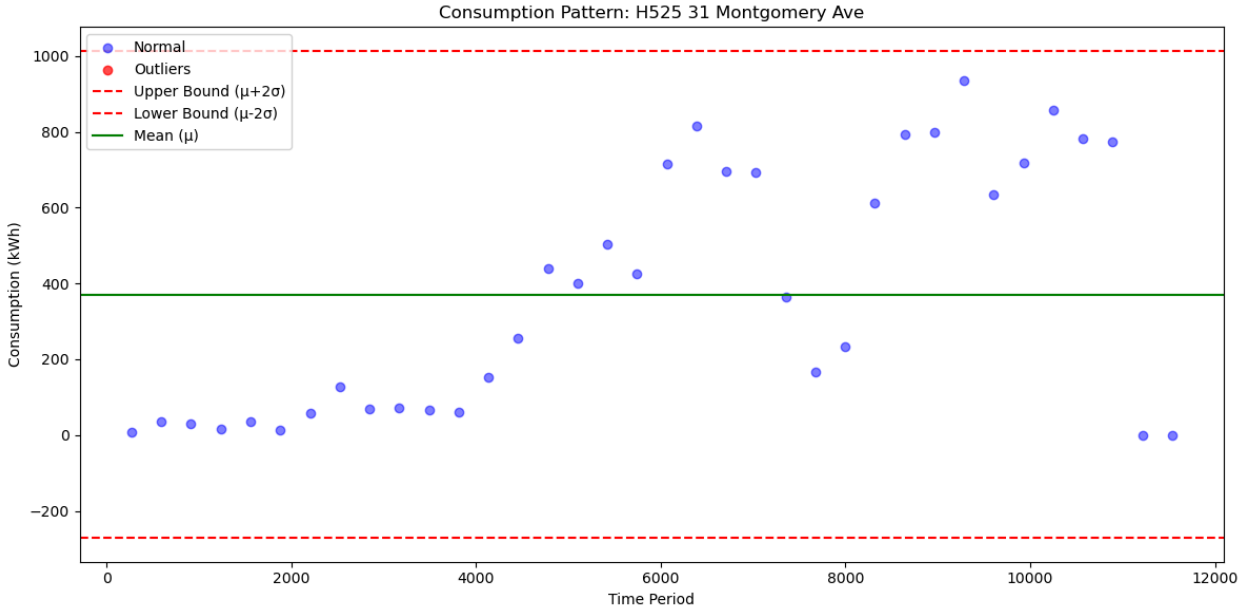

```
Statistics for H514 C Safe Buildings 1, 2 & 4:
Mean: 627.25
Standard Deviation: 197.09
Upper Bound: 1021.43
Lower Bound: 233.06
Number of outliers: 3
```



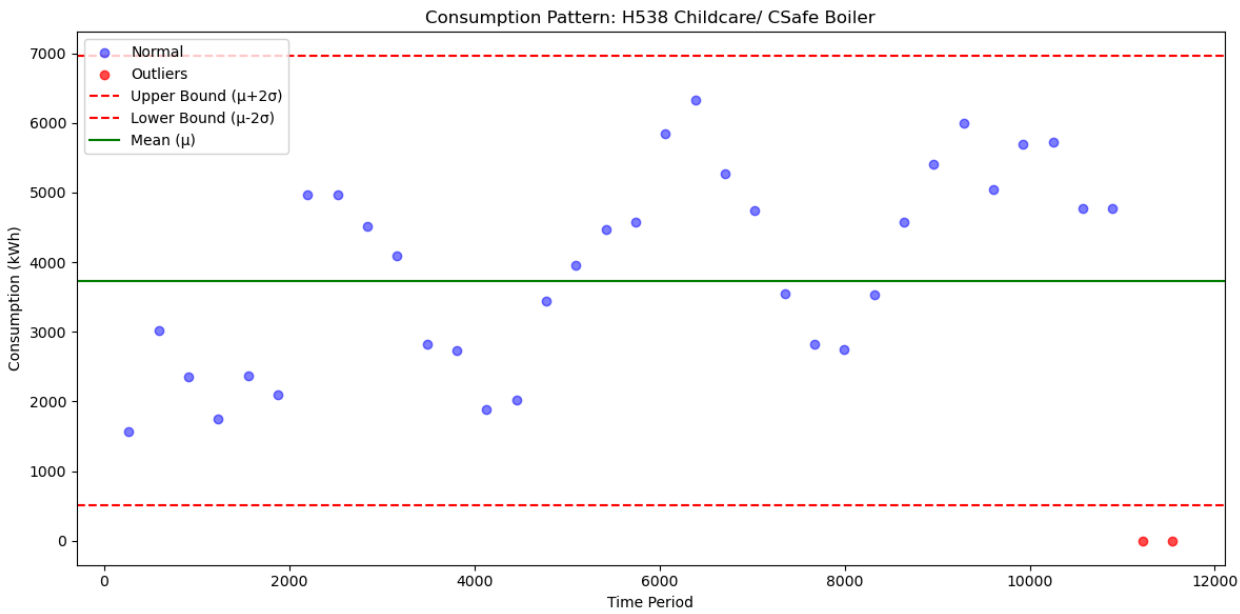

```
Statistics for H515 C Safe Building 3:
Mean: 299.96
Standard Deviation: 88.81
Upper Bound: 477.58
Lower Bound: 122.34
Number of outliers: 3
```



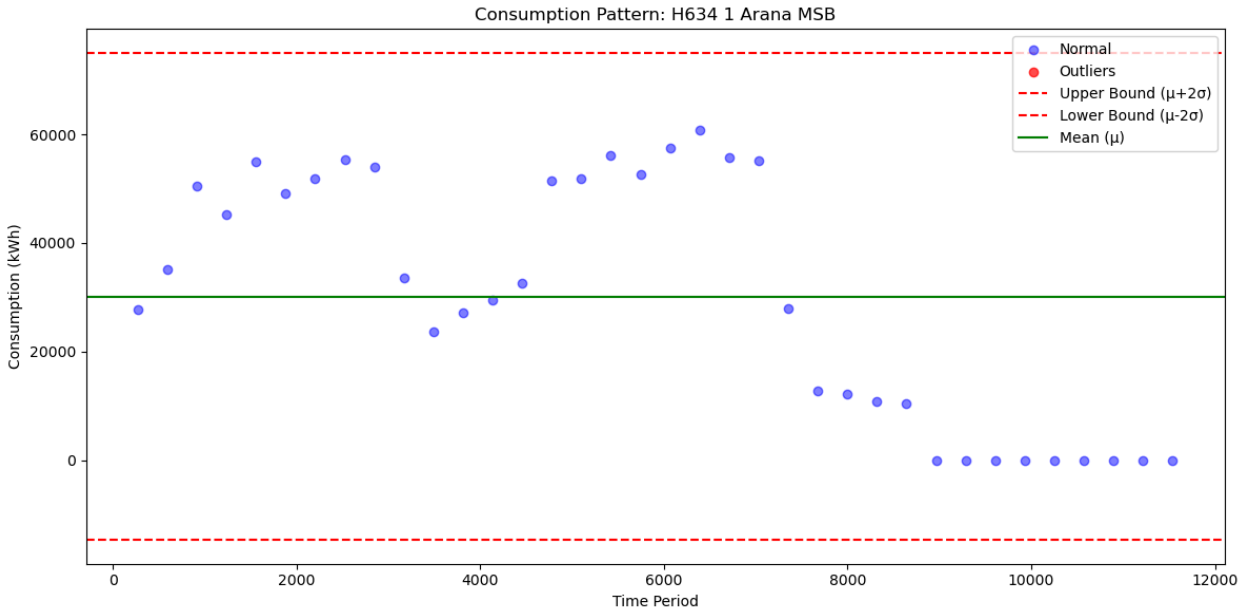

```
Statistics for H525 31 Montgomery Ave:
Mean: 371.19
Standard Deviation: 320.57
Upper Bound: 1012.33
Lower Bound: -269.94
Number of outliers: 0
```



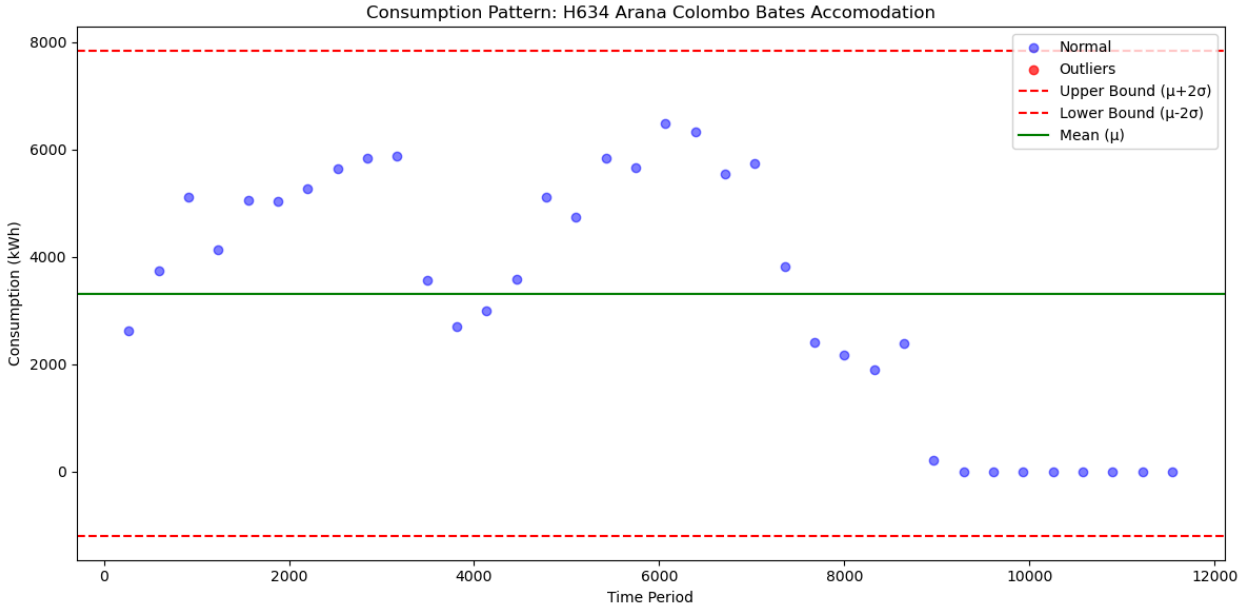

```
Statistics for H538 Childcare/ CSafe Boiler:
Mean: 3732.92
Standard Deviation: 1612.47
Upper Bound: 6957.86
Lower Bound: 507.98
Number of outliers: 2
```



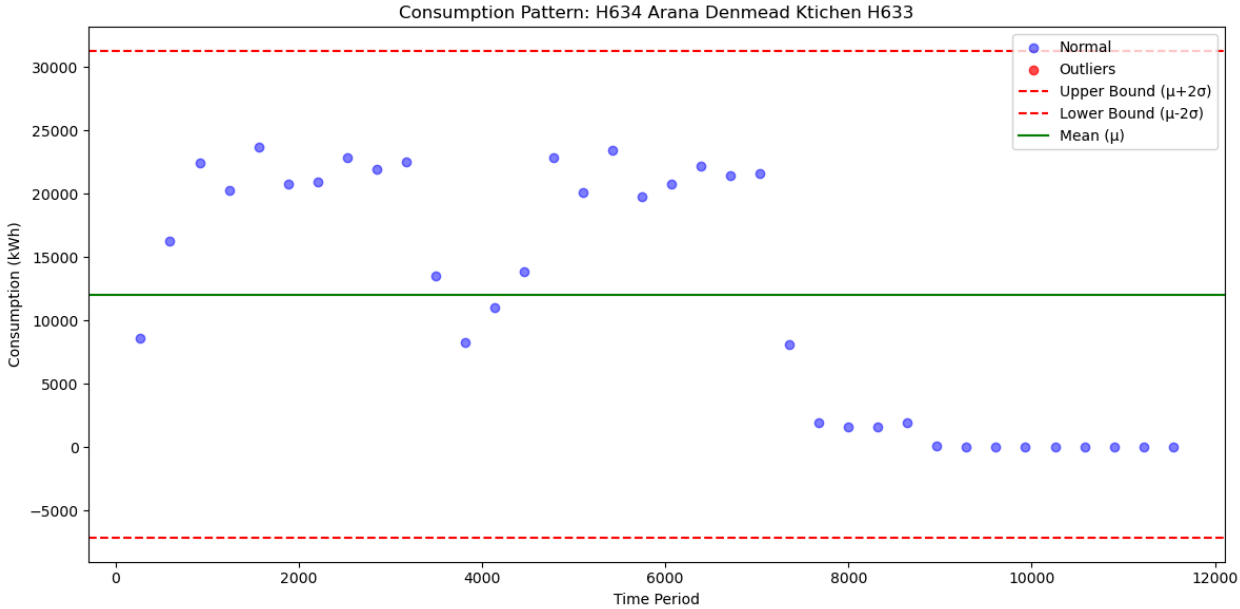

```
Statistics for H634 1 Arana MSB:
Mean: 30159.19
Standard Deviation: 22382.92
Upper Bound: 74925.04
Lower Bound: -14606.65
Number of outliers: 0
```



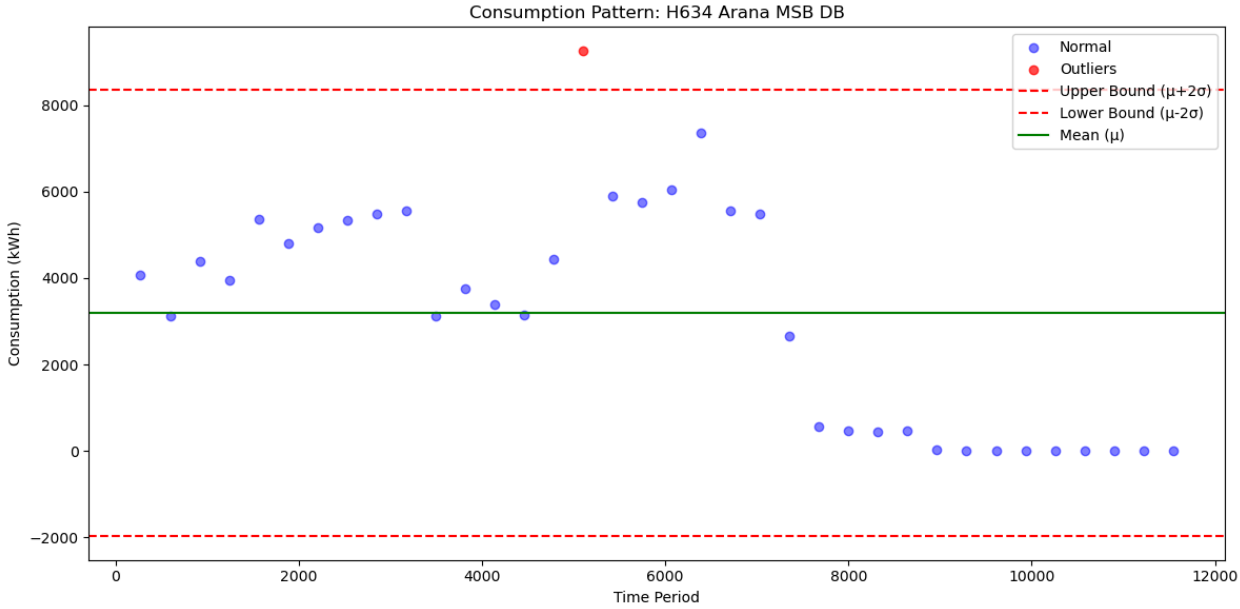

```
Statistics for H634 Arana Colombo Bates Accomodation:
Mean: 3321.29
Standard Deviation: 2255.76
Upper Bound: 7832.81
Lower Bound: -1190.22
Number of outliers: 0
```



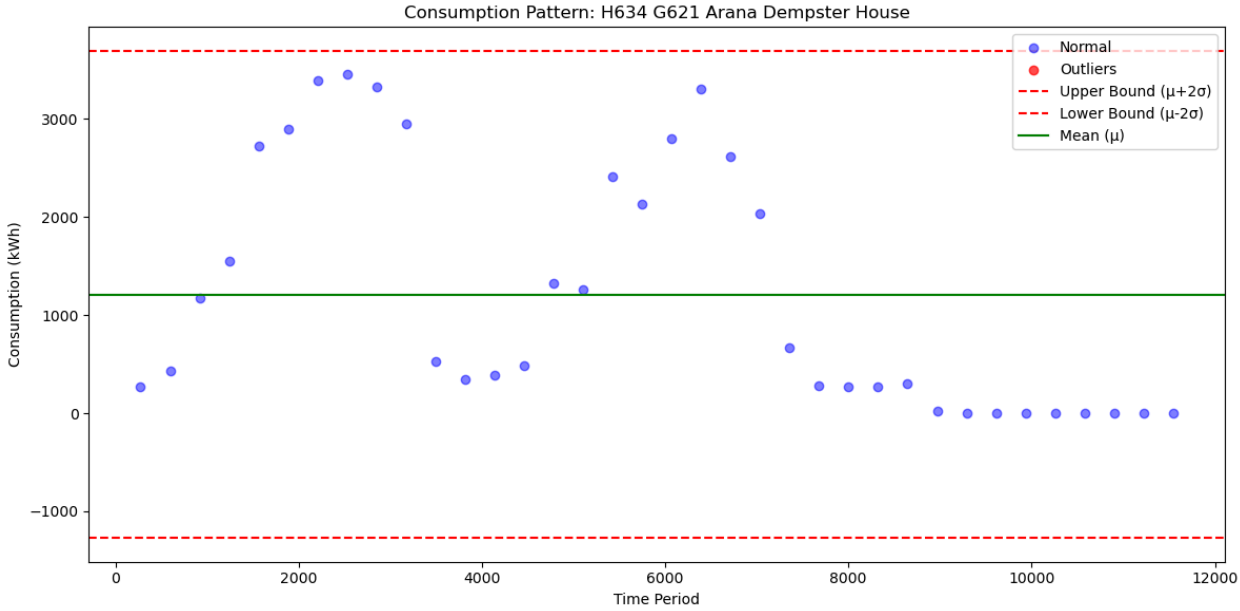

```
Statistics for H634 Arana Denmead Ktichen H633:
Mean: 12057.39
Standard Deviation: 9584.71
Upper Bound: 31226.82
Lower Bound: -7112.03
Number of outliers: 0
```



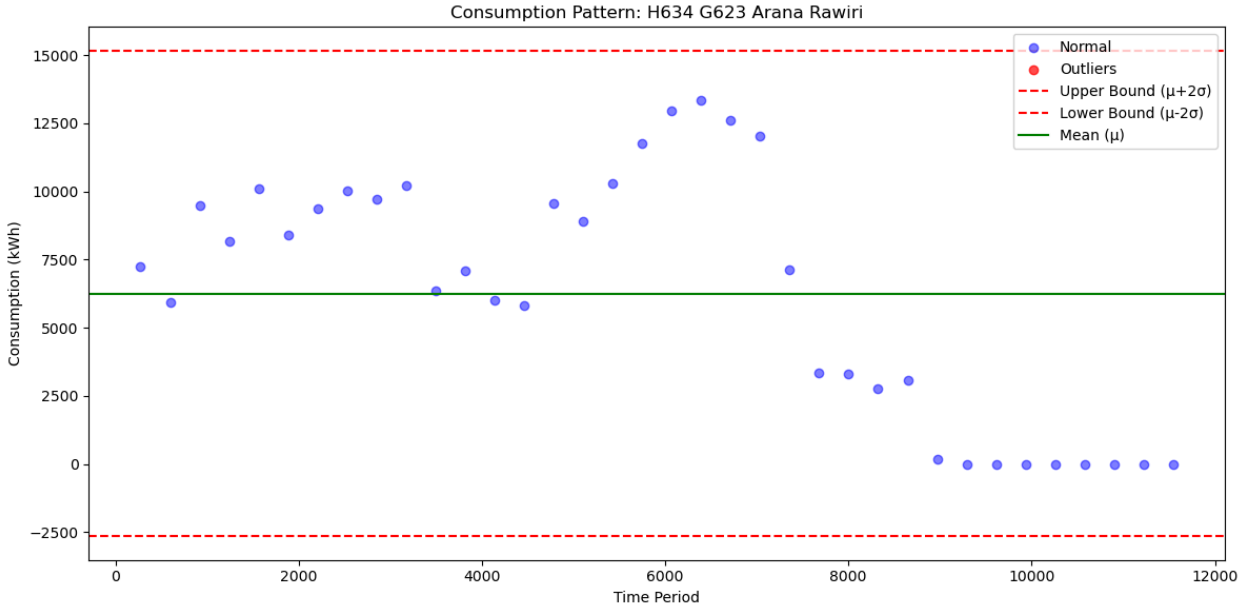

```
Statistics for H634 Arana MSB DB:
Mean: 3195.30
Standard Deviation: 2578.92
Upper Bound: 8353.14
Lower Bound: -1962.54
Number of outliers: 1
```



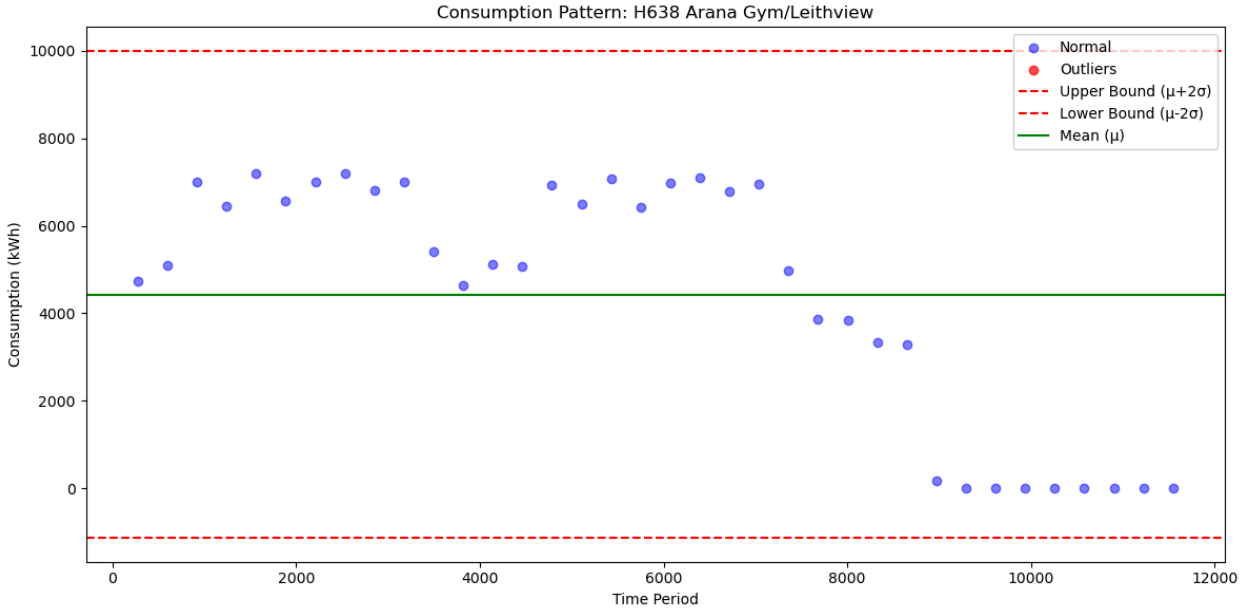

```
Statistics for H634 G621 Arana Dempster House:
Mean: 1211.28
Standard Deviation: 1238.84
Upper Bound: 3688.95
Lower Bound: -1266.40
Number of outliers: 0
```



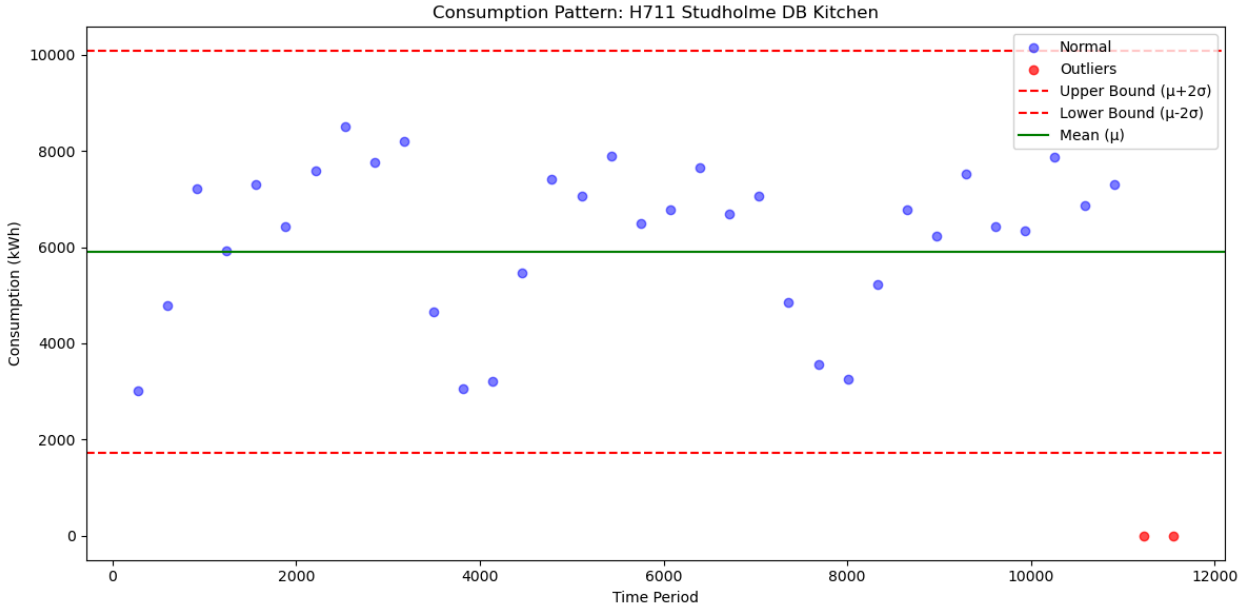

```
Statistics for H634 G623 Arana Rawiri:
Mean: 6255.73
Standard Deviation: 4444.30
Upper Bound: 15144.32
Lower Bound: -2632.87
Number of outliers: 0
```



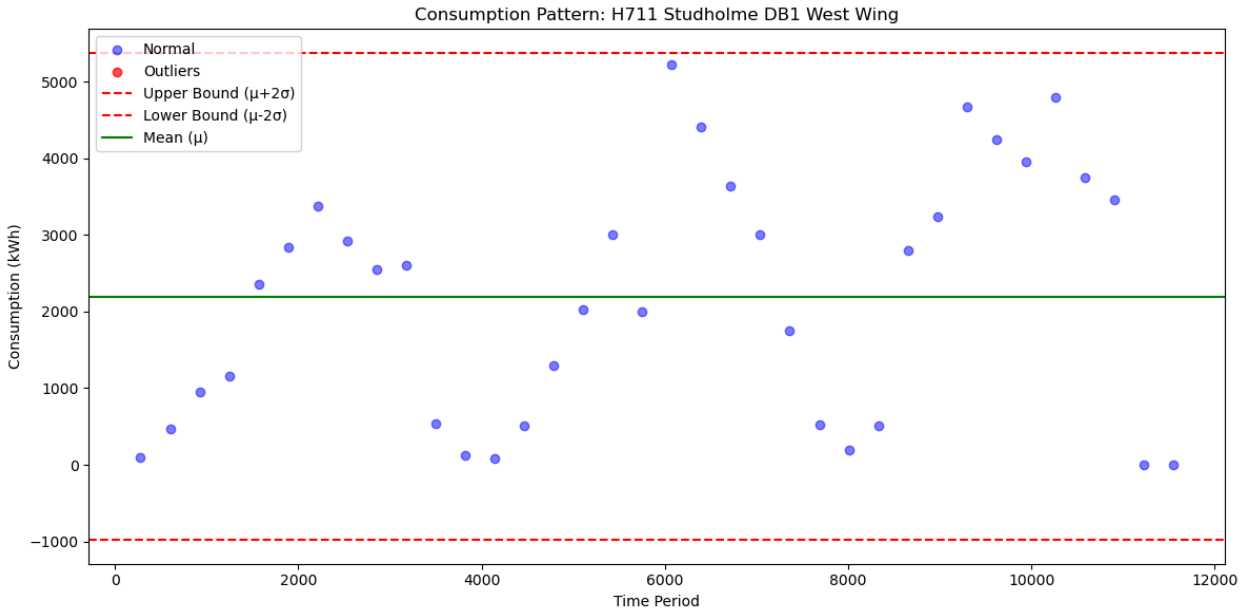

```
Statistics for H638 Arana Gym/Leithview:
Mean: 4431.35
Standard Deviation: 2777.50
Upper Bound: 9986.35
Lower Bound: -1123.65
Number of outliers: 0
```



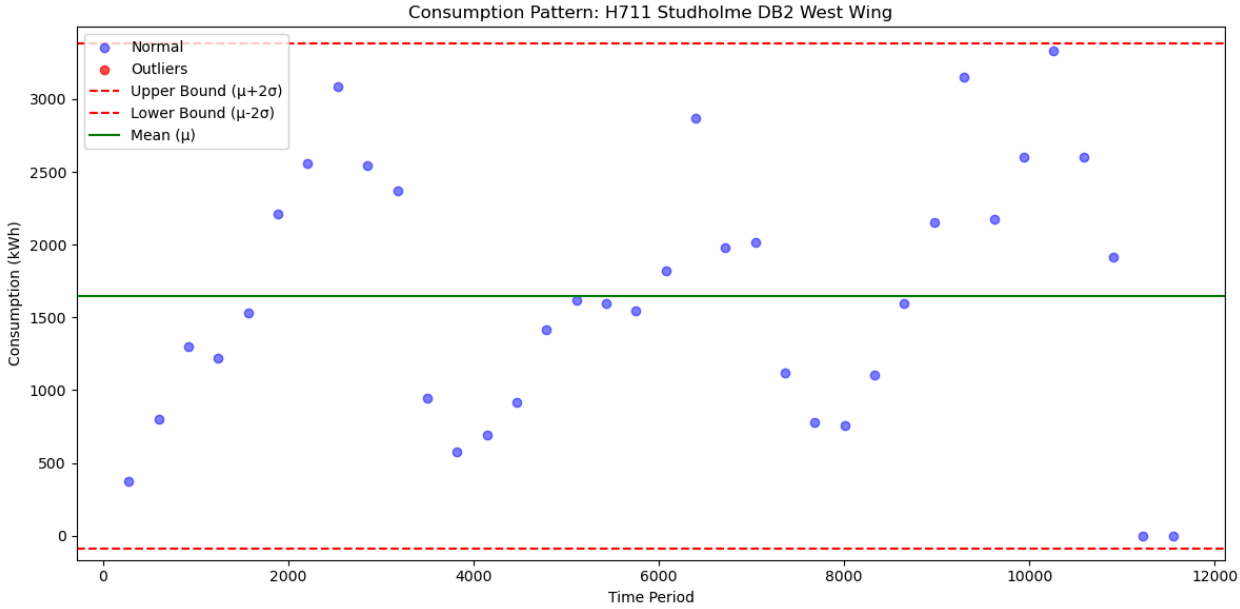

```
Statistics for H711 Studholme DB Kitchen:
Mean: 5897.78
Standard Deviation: 2087.96
Upper Bound: 10073.70
Lower Bound: 1721.86
Number of outliers: 2
```



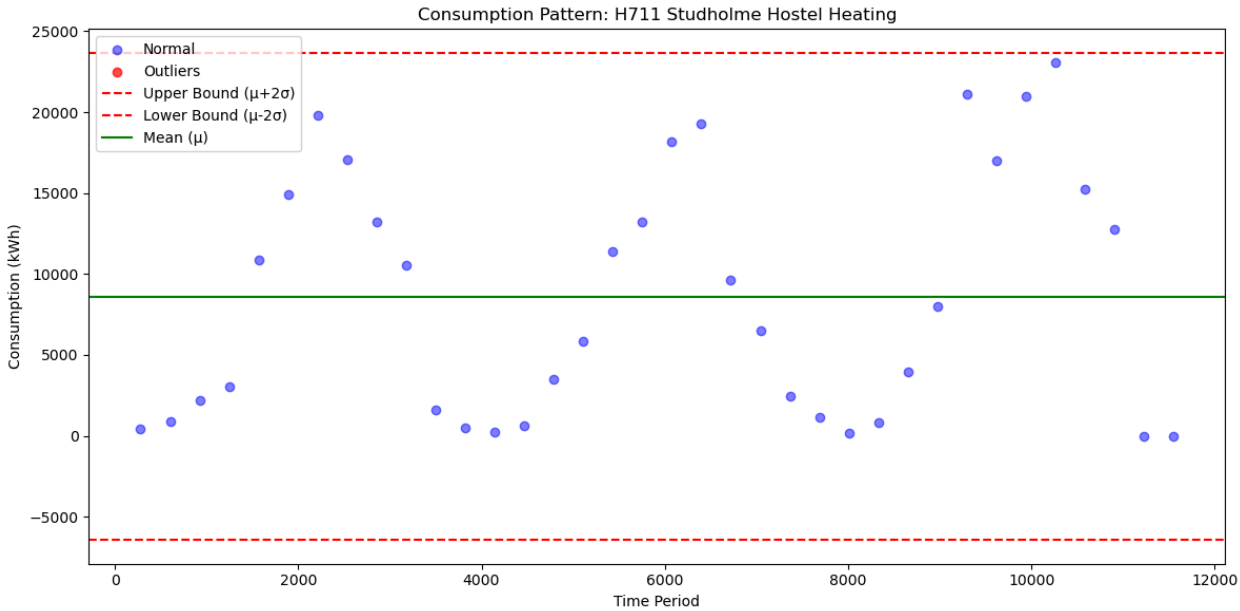

```
Statistics for H711 Studholme DB1 West Wing:
Mean: 2198.07
Standard Deviation: 1585.60
Upper Bound: 5369.27
Lower Bound: -973.12
Number of outliers: 0
```



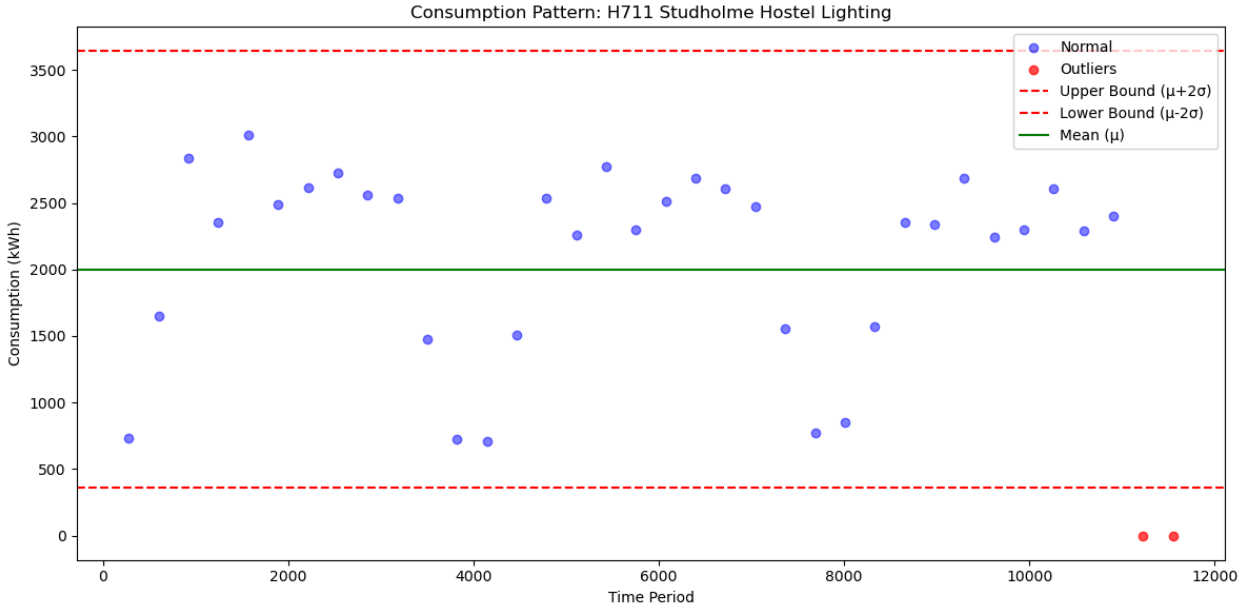

```
Statistics for H711 Studholme DB2 West Wing:
Mean: 1646.71
Standard Deviation: 868.93
Upper Bound: 3384.58
Lower Bound: -91.15
Number of outliers: 0
```



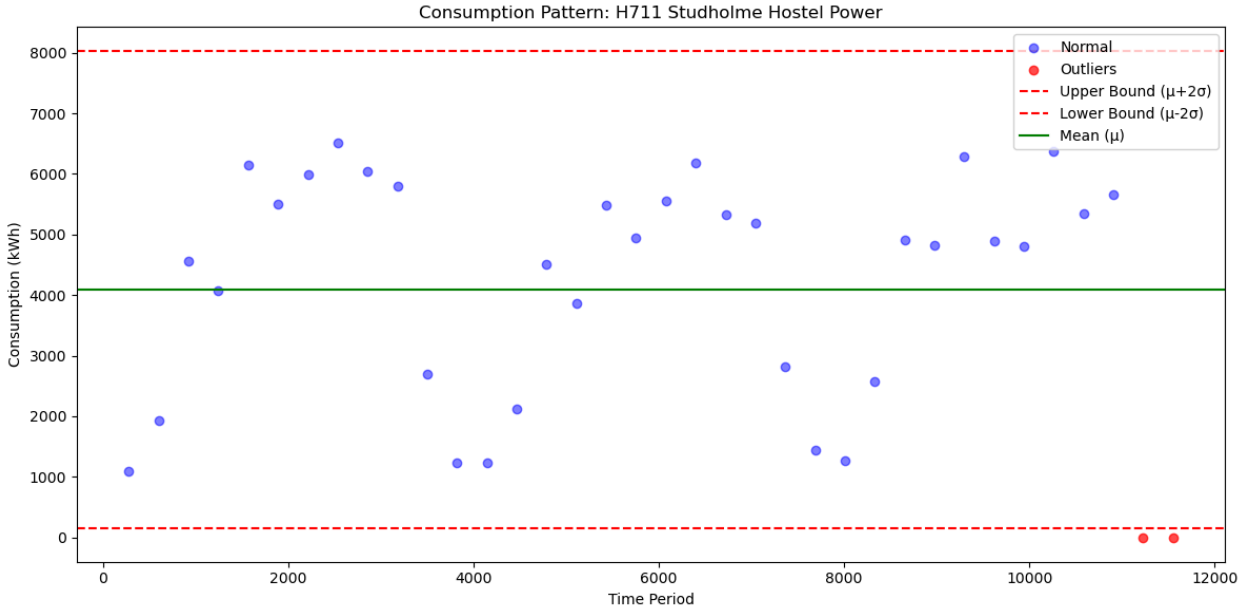

```
Statistics for H711 Studholme Hostel Heating:
Mean: 8618.16
Standard Deviation: 7507.46
Upper Bound: 23633.08
Lower Bound: -6396.75
Number of outliers: 0
```



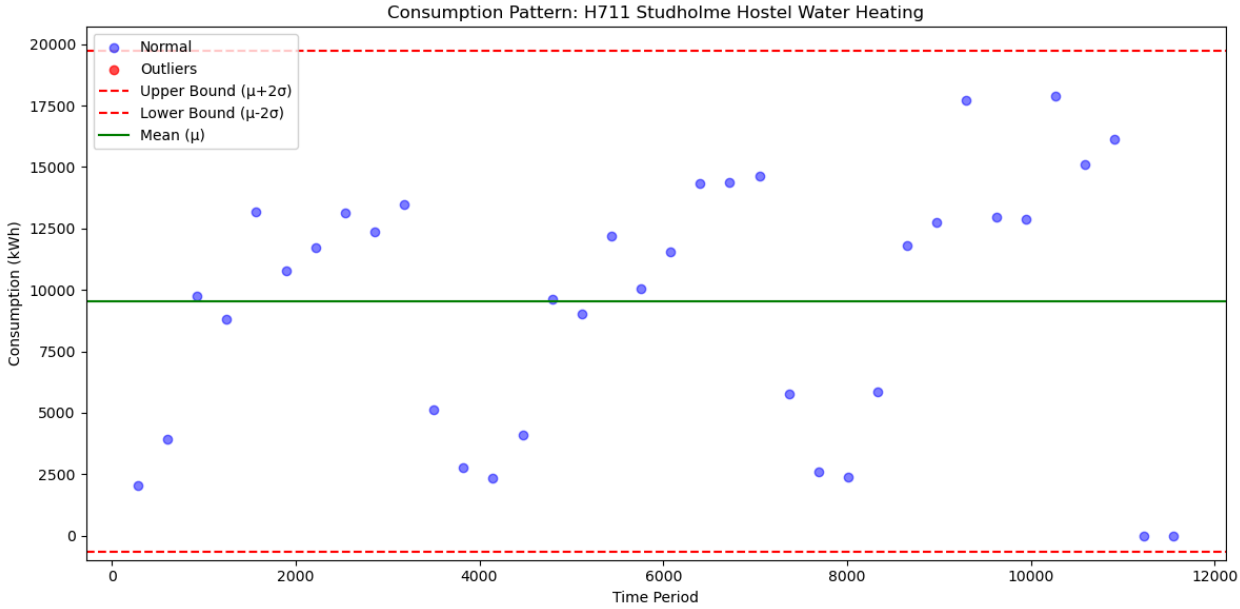

```
Statistics for H711 Studholme Hostel Lighting:
Mean: 2000.75
Standard Deviation: 819.48
Upper Bound: 3639.71
Lower Bound: 361.79
Number of outliers: 2
```



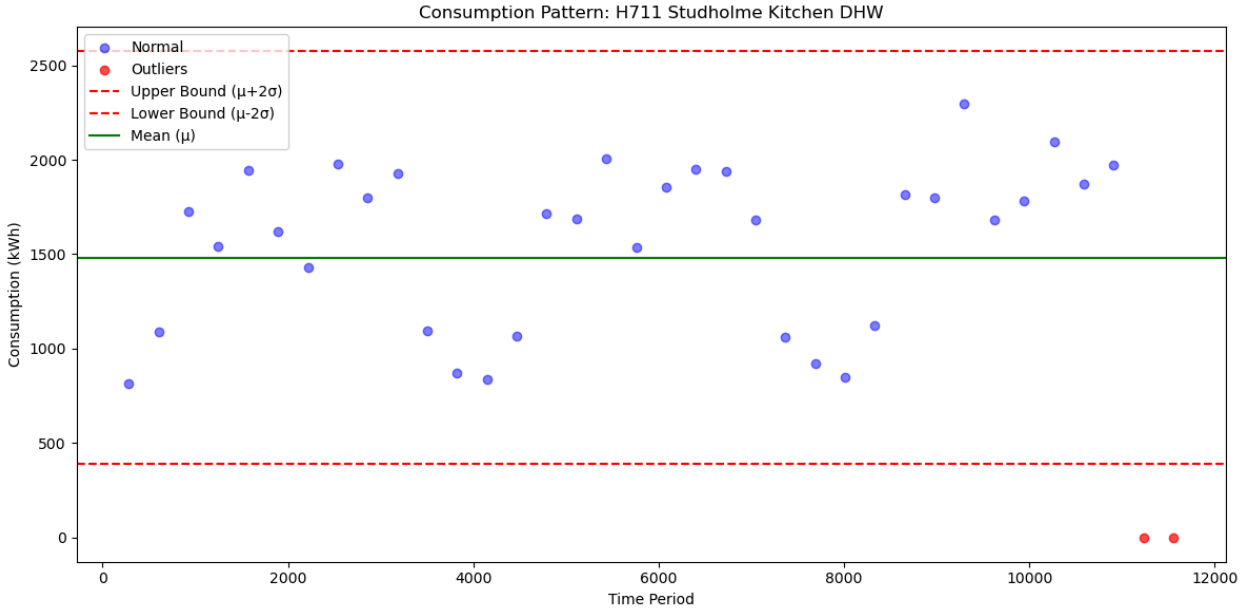

```
Statistics for H711 Studholme Hostel Power:
Mean: 4088.57
Standard Deviation: 1967.48
Upper Bound: 8023.53
Lower Bound: 153.61
Number of outliers: 2
```



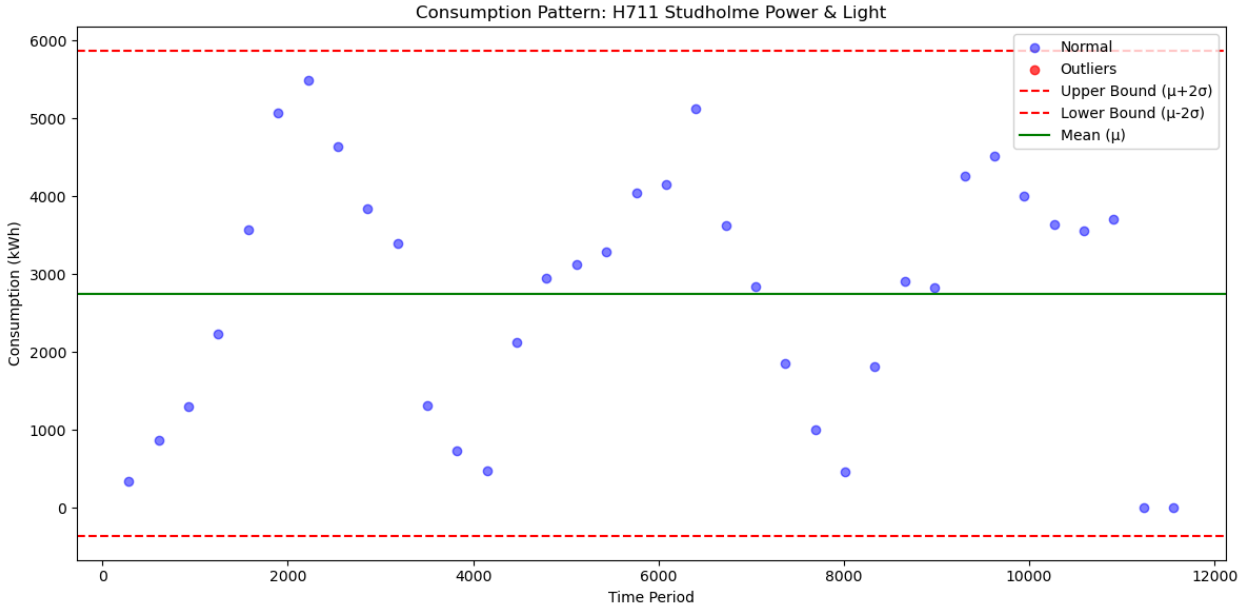

```
Statistics for H711 Studholme Hostel Water Heating:
Mean: 9534.70
Standard Deviation: 5089.89
Upper Bound: 19714.48
Lower Bound: -645.08
Number of outliers: 0
```



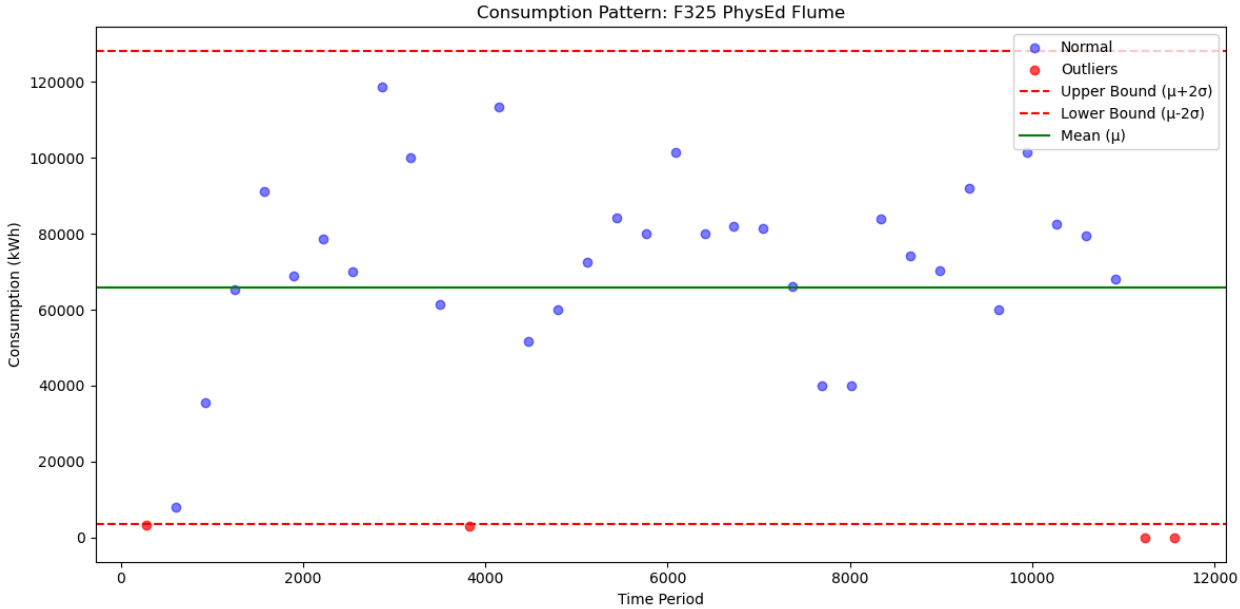

```
Statistics for H711 Studholme Kitchen DHW:
Mean: 1482.68
Standard Deviation: 545.87
Upper Bound: 2574.41
Lower Bound: 390.95
Number of outliers: 2
```



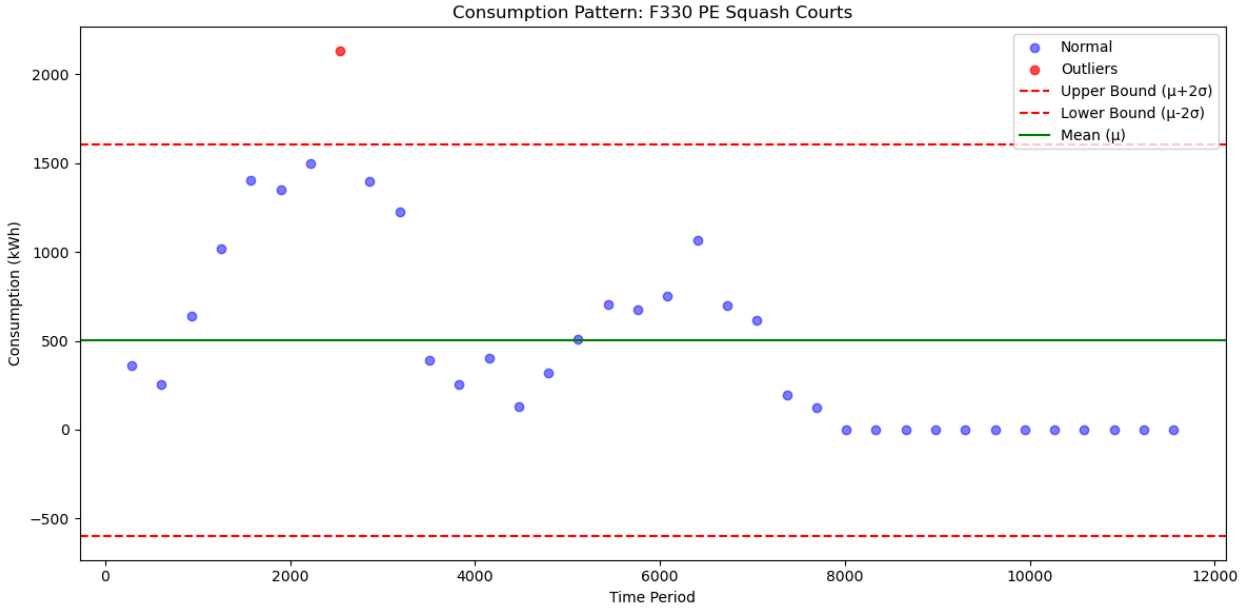

```
Statistics for H711 Studholme Power & Light:
Mean: 2753.18
Standard Deviation: 1556.64
Upper Bound: 5866.46
Lower Bound: -360.10
Number of outliers: 0
```



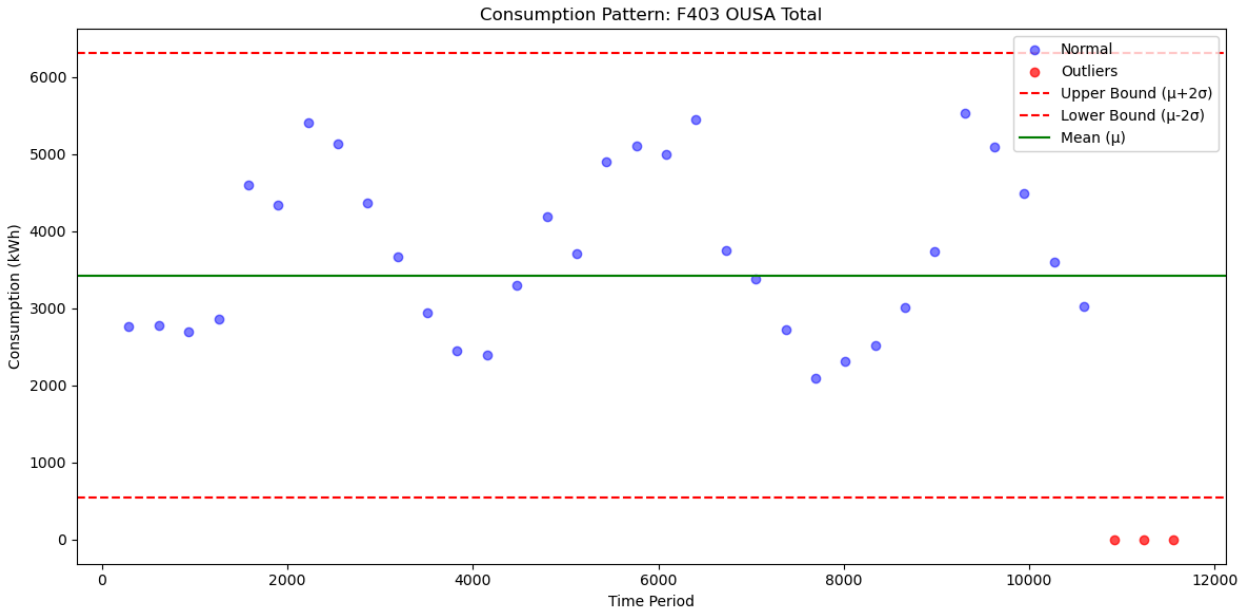

```
Statistics for F325 PhysEd Flume:
Mean: 65809.44
Standard Deviation: 31135.38
Upper Bound: 128080.20
Lower Bound: 3538.69
Number of outliers: 4
```




```
Statistics for F330 PE Squash Courts:
Mean: 503.62
Standard Deviation: 550.33
Upper Bound: 1604.29
Lower Bound: -597.04
Number of outliers: 1
```

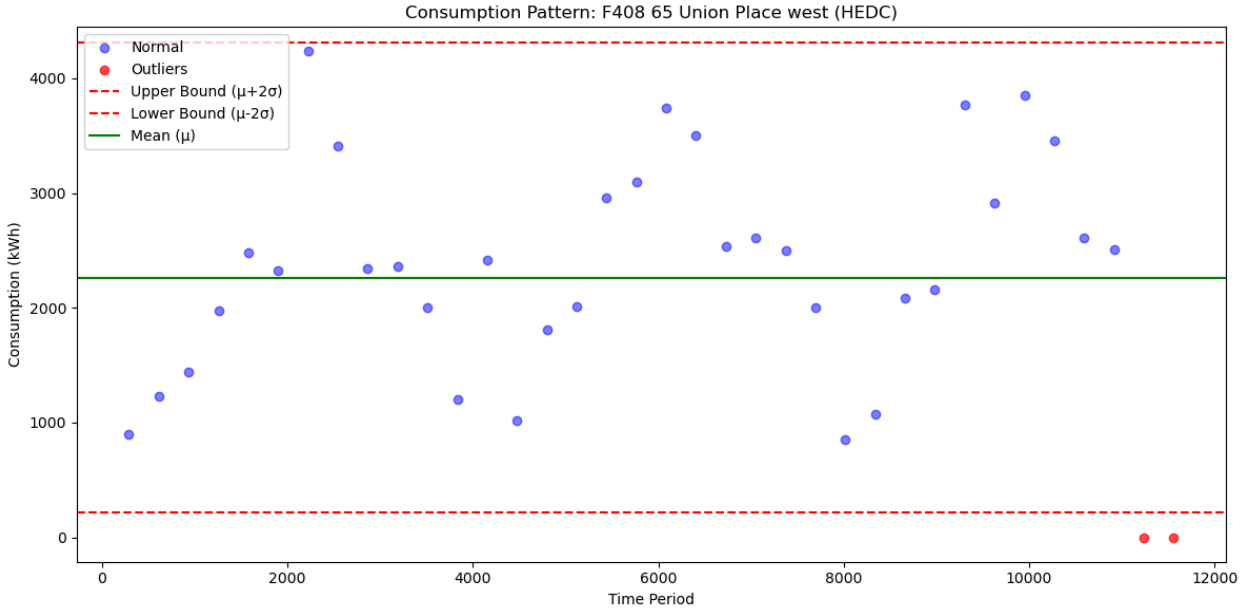



```
Statistics for F403 OUSA Total:
Mean: 3425.61
Standard Deviation: 1440.84
Upper Bound: 6307.30
Lower Bound: 543.92
Number of outliers: 3
```

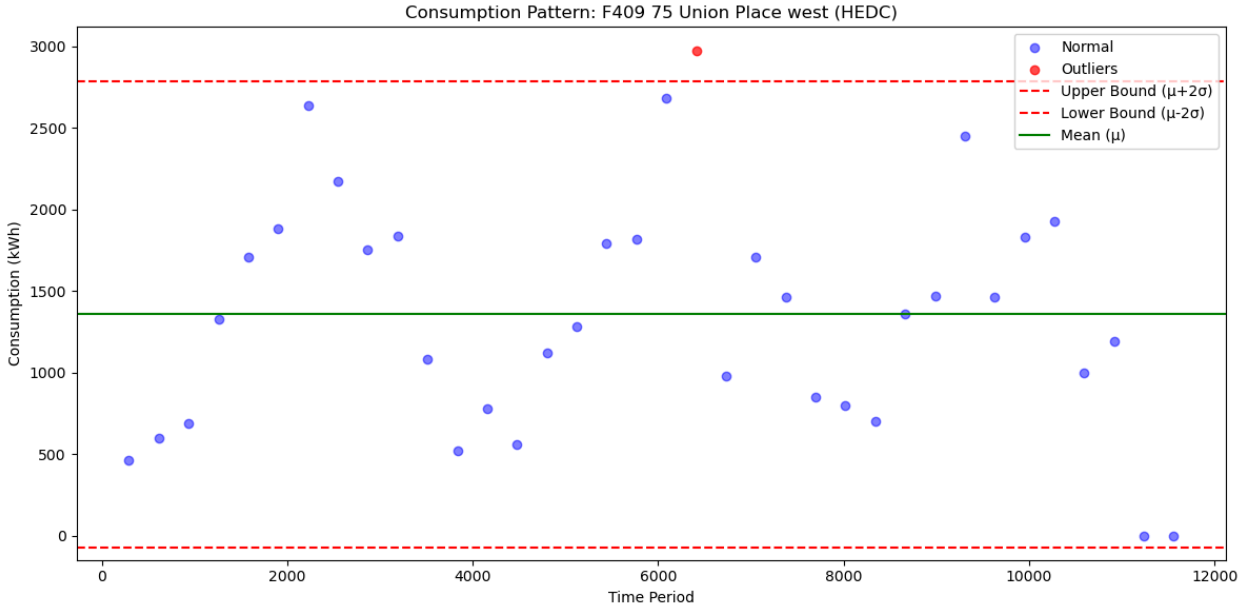



```
Statistics for F405 Smithells Gym:
Mean: 3762.84
Standard Deviation: 2134.51
Upper Bound: 8031.86
Lower Bound: -506.18
Number of outliers: 0
```

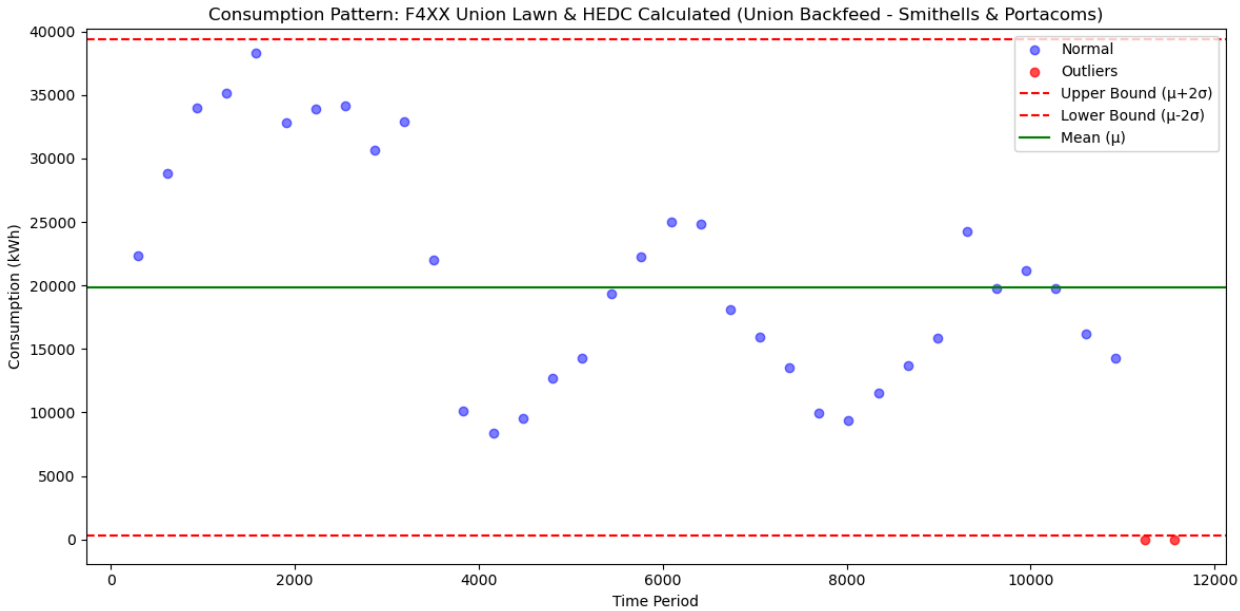
```
Statistics for F408 65 Union Place west (HEDC):
Mean: 2260.94
Standard Deviation: 1023.53
Upper Bound: 4308.01
Lower Bound: 213.88
Number of outliers: 2
```



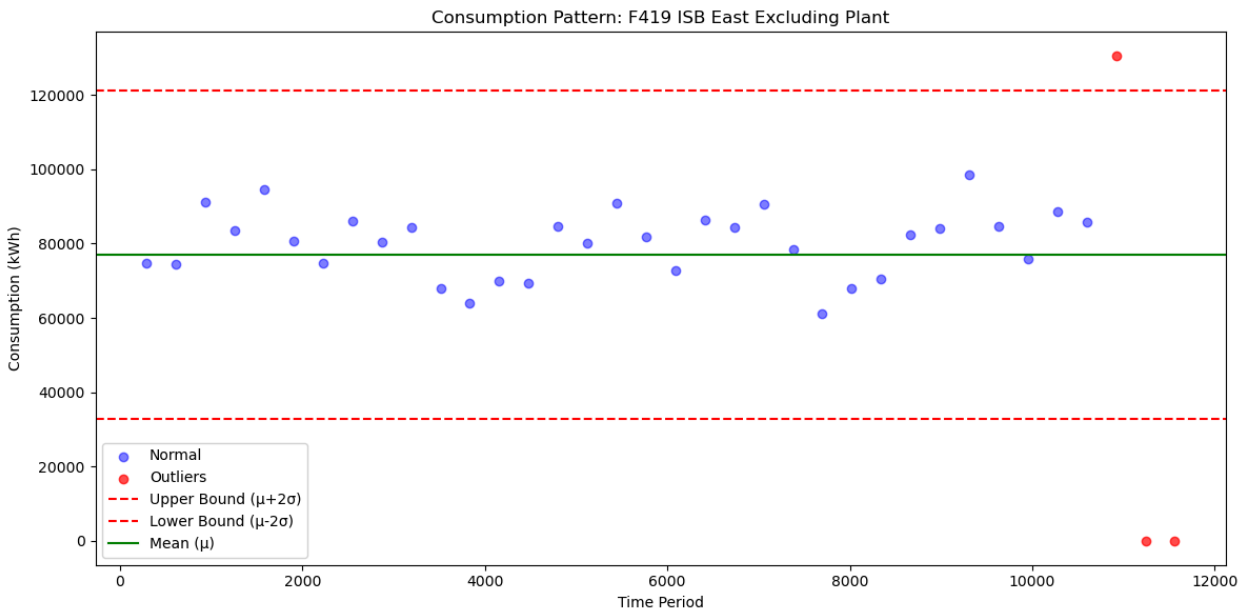

```
Statistics for F409 75 Union Place west (HEDC):
Mean: 1357.22
Standard Deviation: 714.19
Upper Bound: 2785.60
Lower Bound: -71.15
Number of outliers: 1
```



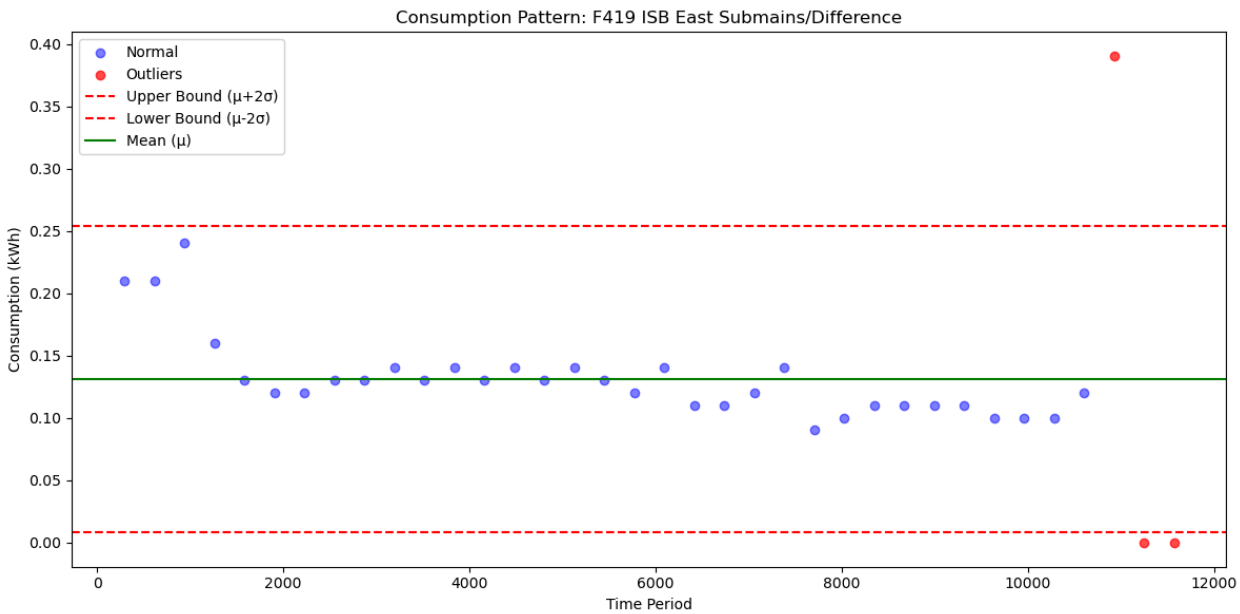

```
Statistics for F4XX Union Lawn & HEDC Calculated (Union Backfeed - Smithells & Portacoms):
Mean: 19859.20
Standard Deviation: 9768.87
Upper Bound: 39396.93
Lower Bound: 321.46
Number of outliers: 2
```



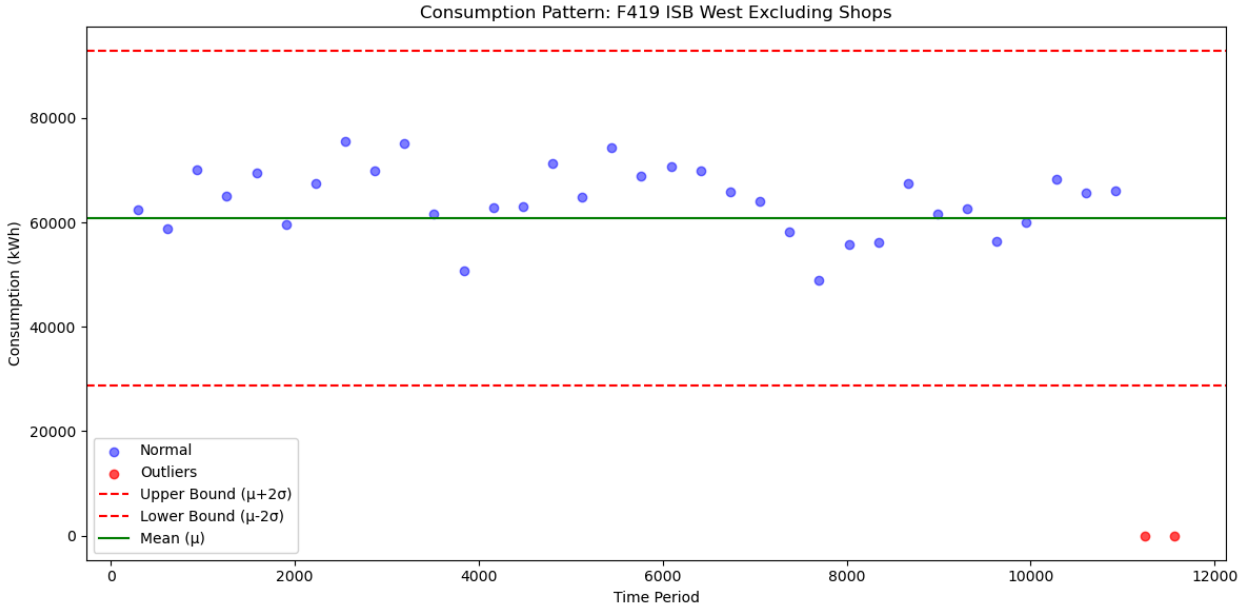

```
Statistics for F419 ISB East Excluding Plant:
Mean: 77097.91
Standard Deviation: 22125.59
Upper Bound: 121349.09
Lower Bound: 32846.73
Number of outliers: 3
```



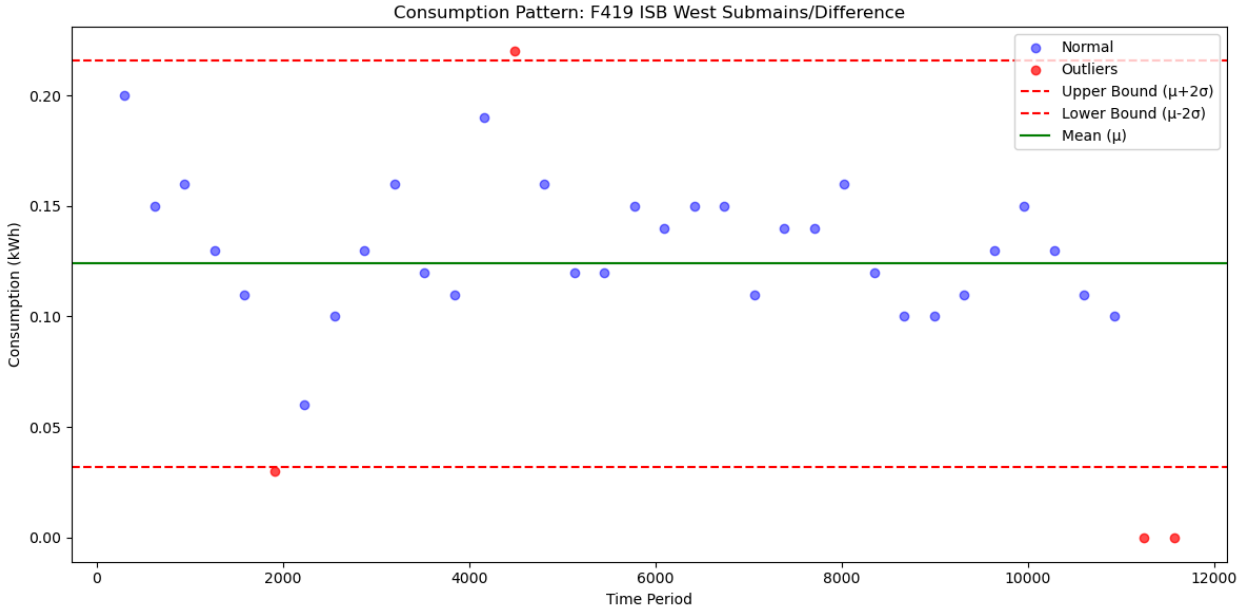

```
Statistics for F419 ISB East Submains/Difference:
Mean: 0.13
Standard Deviation: 0.06
Upper Bound: 0.25
Lower Bound: 0.01
Number of outliers: 3
```



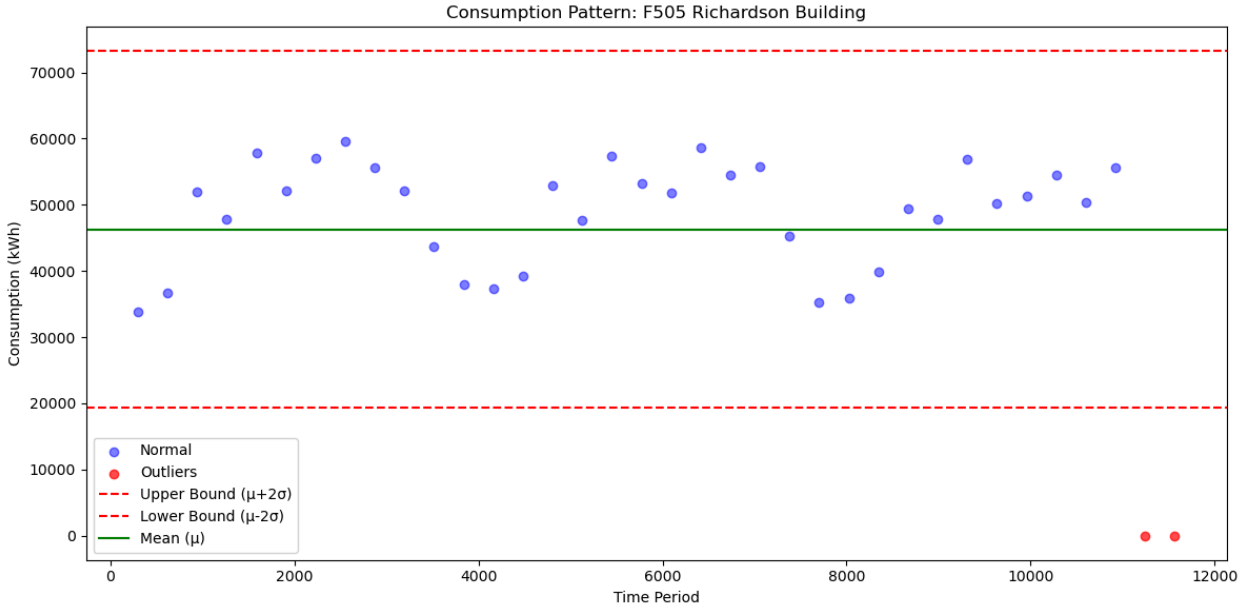

```
Statistics for F419 ISB West Excluding Shops:
Mean: 60806.96
Standard Deviation: 15992.98
Upper Bound: 92792.91
Lower Bound: 28821.01
Number of outliers: 2
```



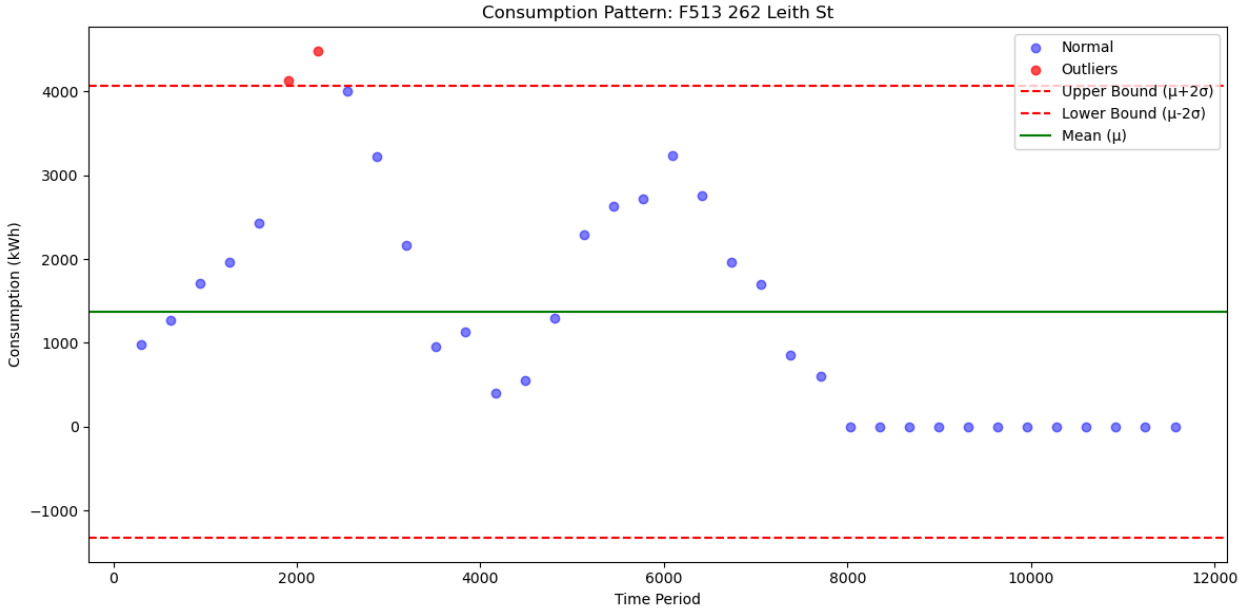

```
Statistics for F419 ISB West Submains/Difference:
Mean: 0.12
Standard Deviation: 0.05
Upper Bound: 0.22
Lower Bound: 0.03
Number of outliers: 4
```



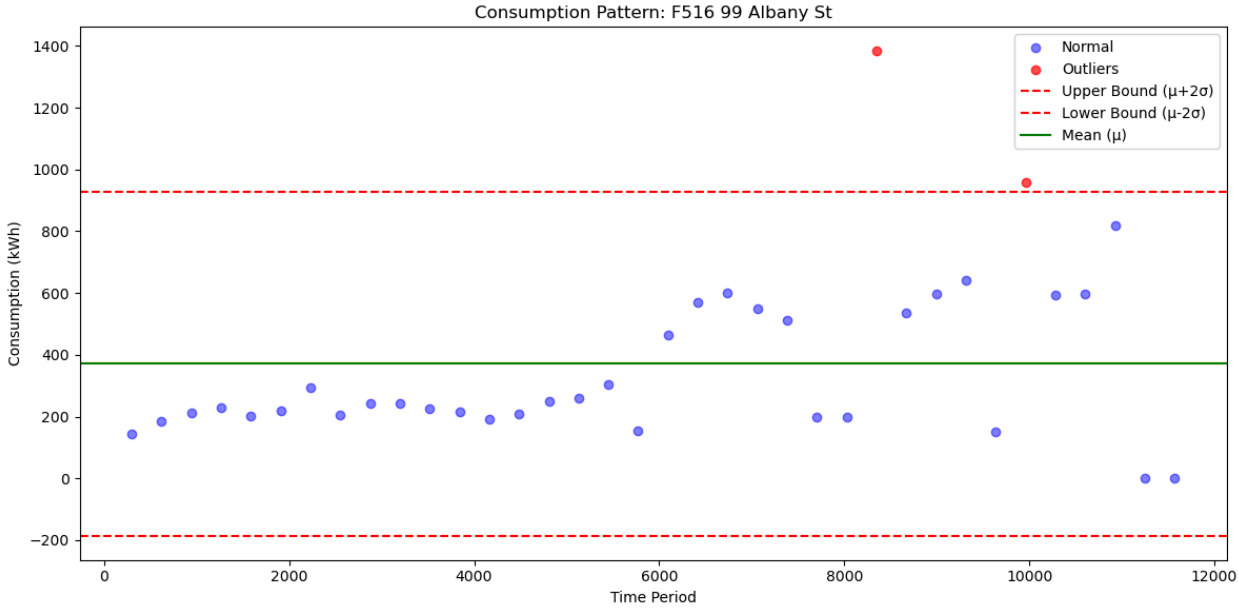

```
Statistics for F505 Richardson Building:
Mean: 46306.20
Standard Deviation: 13453.24
Upper Bound: 73212.68
Lower Bound: 19399.71
Number of outliers: 2
```



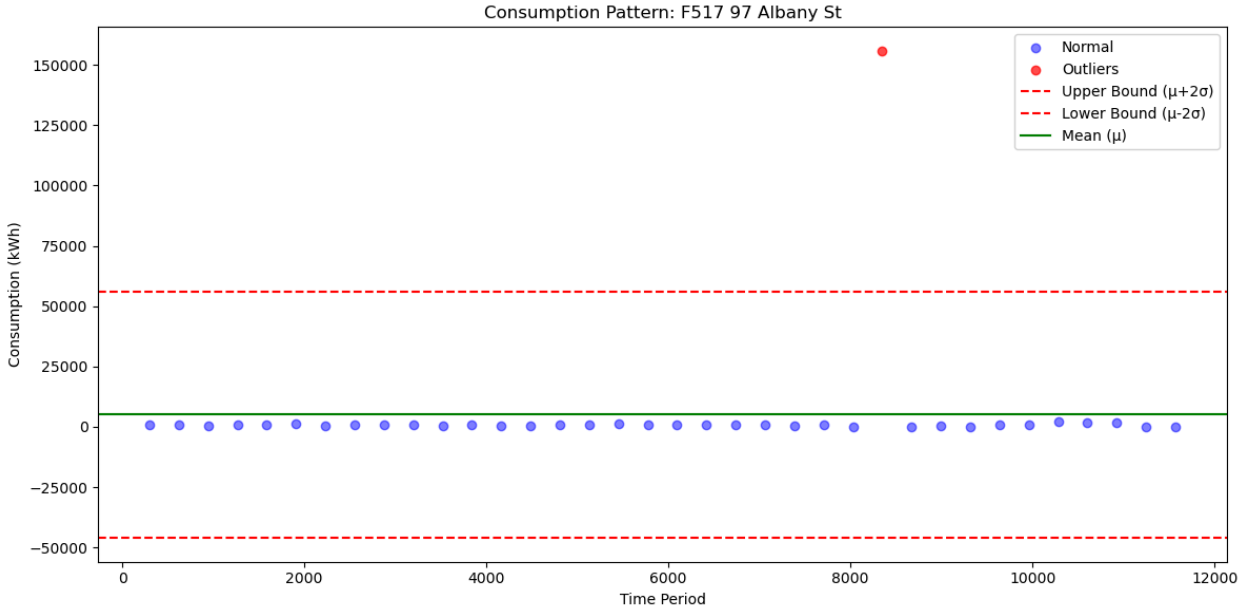

```
Statistics for F513 262 Leith St:
Mean: 1373.33
Standard Deviation: 1348.14
Upper Bound: 4069.61
Lower Bound: -1322.94
Number of outliers: 2
```



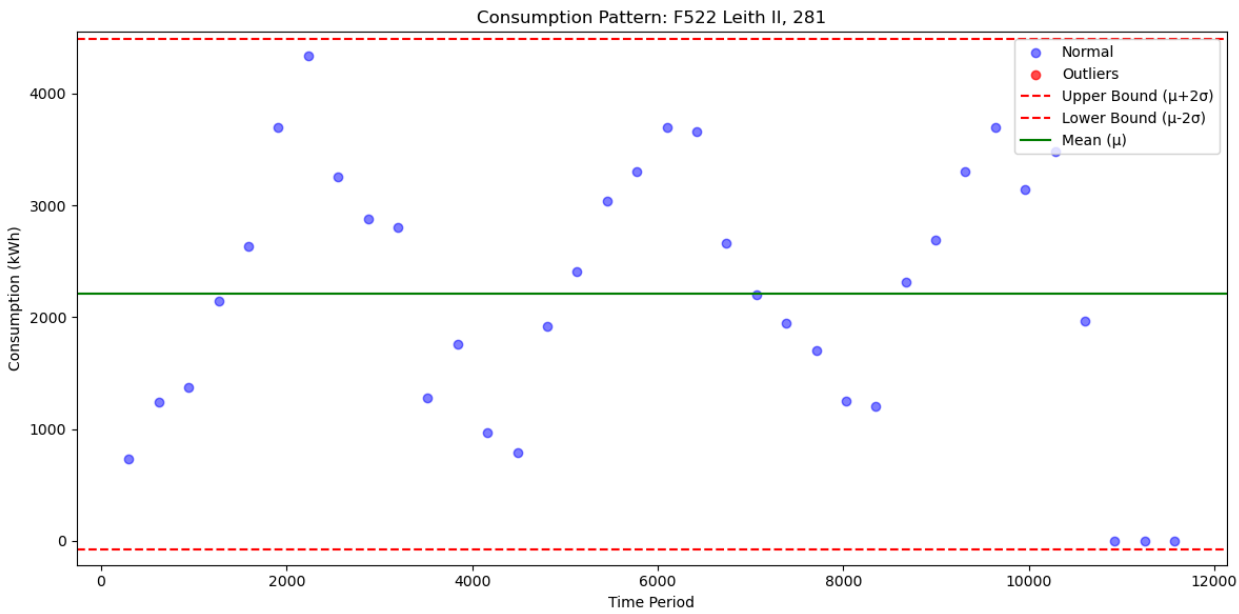

```
Statistics for F516 99 Albany St:
Mean: 371.03
Standard Deviation: 278.40
Upper Bound: 927.83
Lower Bound: -185.77
Number of outliers: 2
```



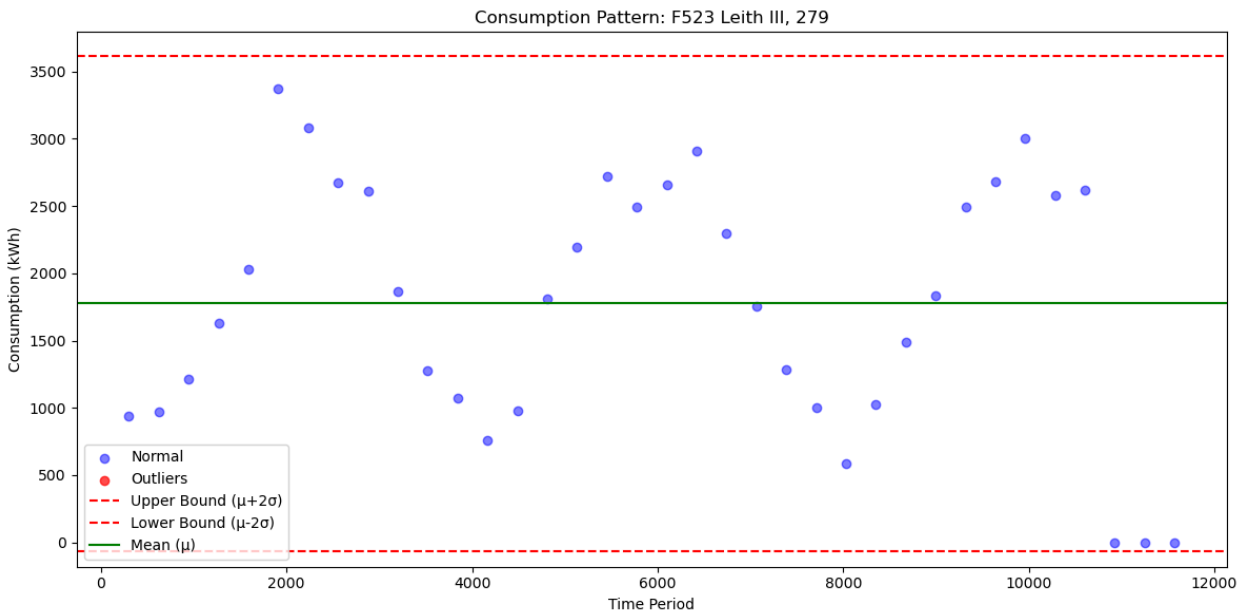

```
Statistics for F517 97 Albany St:
Mean: 5059.44
Standard Deviation: 25457.24
Upper Bound: 55973.92
Lower Bound: -45855.03
Number of outliers: 1
```



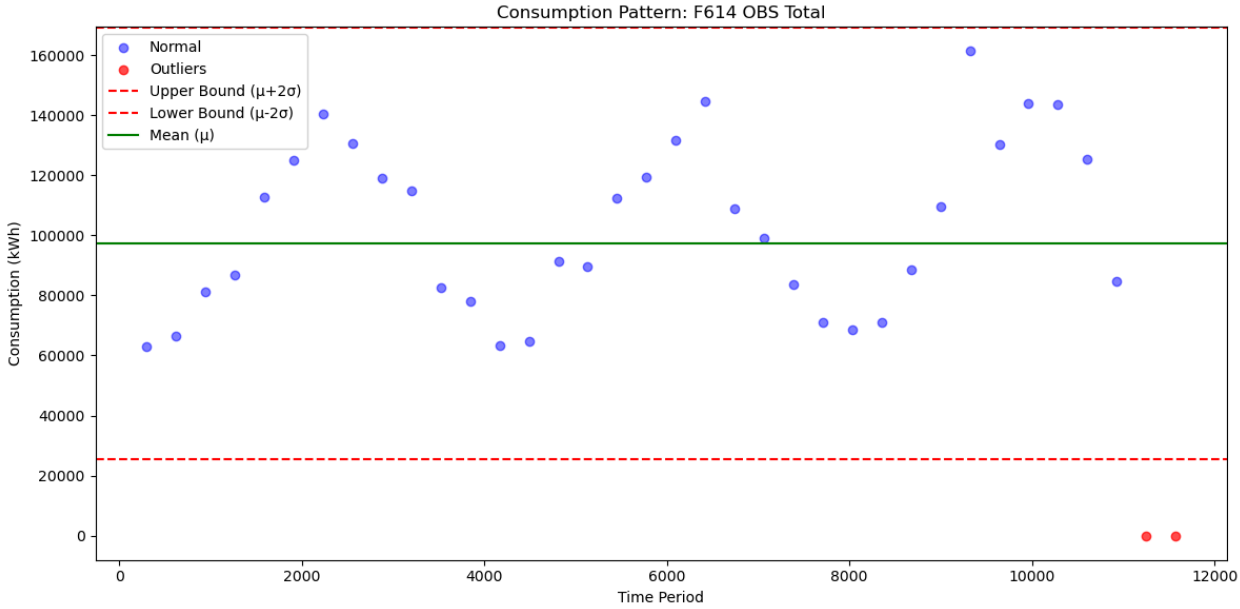

```
Statistics for F522 Leith II, 281:
Mean: 2207.89
Standard Deviation: 1141.52
Upper Bound: 4490.93
Lower Bound: -75.15
Number of outliers: 0
```



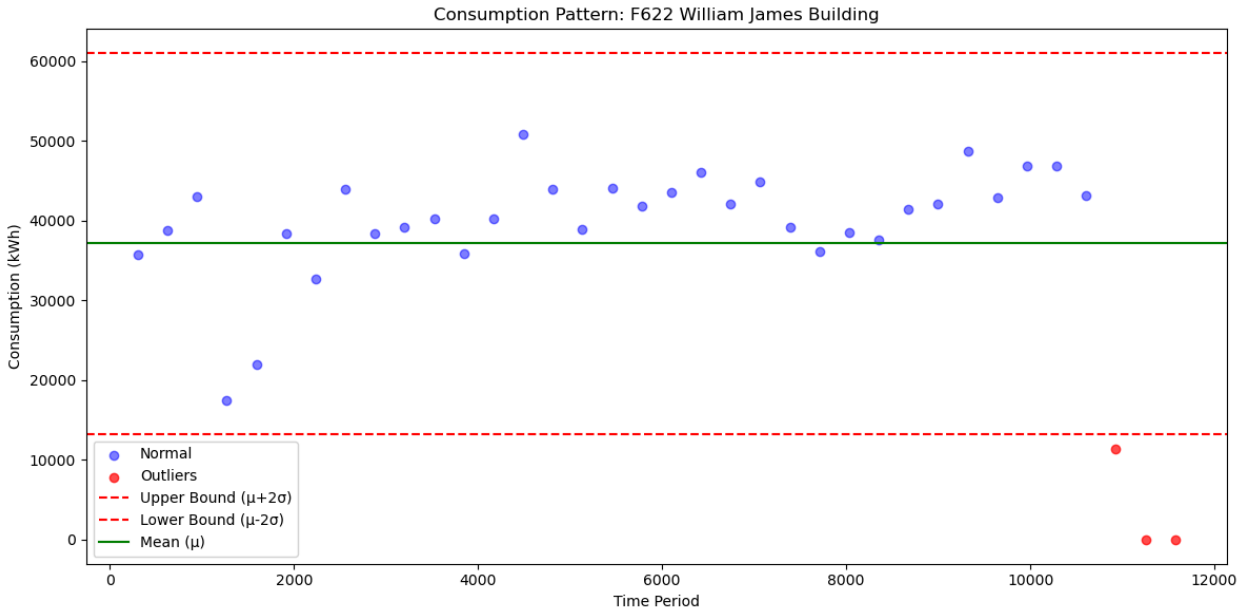

```
Statistics for F523 Leith III, 279:
Mean: 1775.19
Standard Deviation: 918.76
Upper Bound: 3612.71
Lower Bound: -62.33
Number of outliers: 0
```



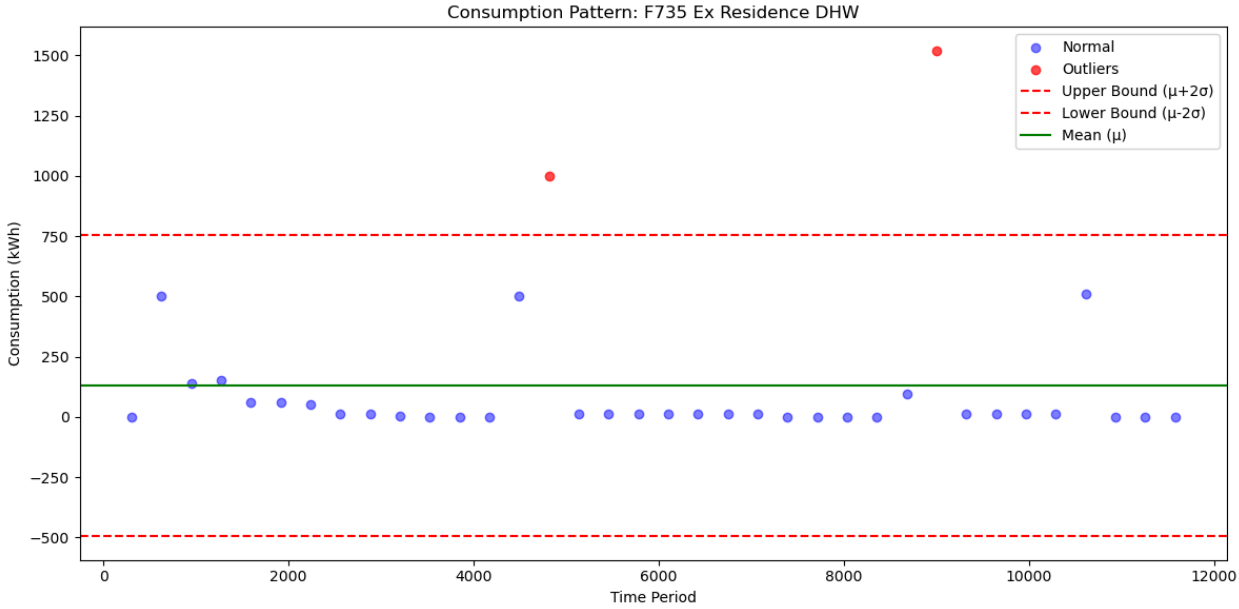

```
Statistics for F614 OBS Total:
Mean: 97421.63
Standard Deviation: 35898.37
Upper Bound: 169218.36
Lower Bound: 25624.89
Number of outliers: 2
```



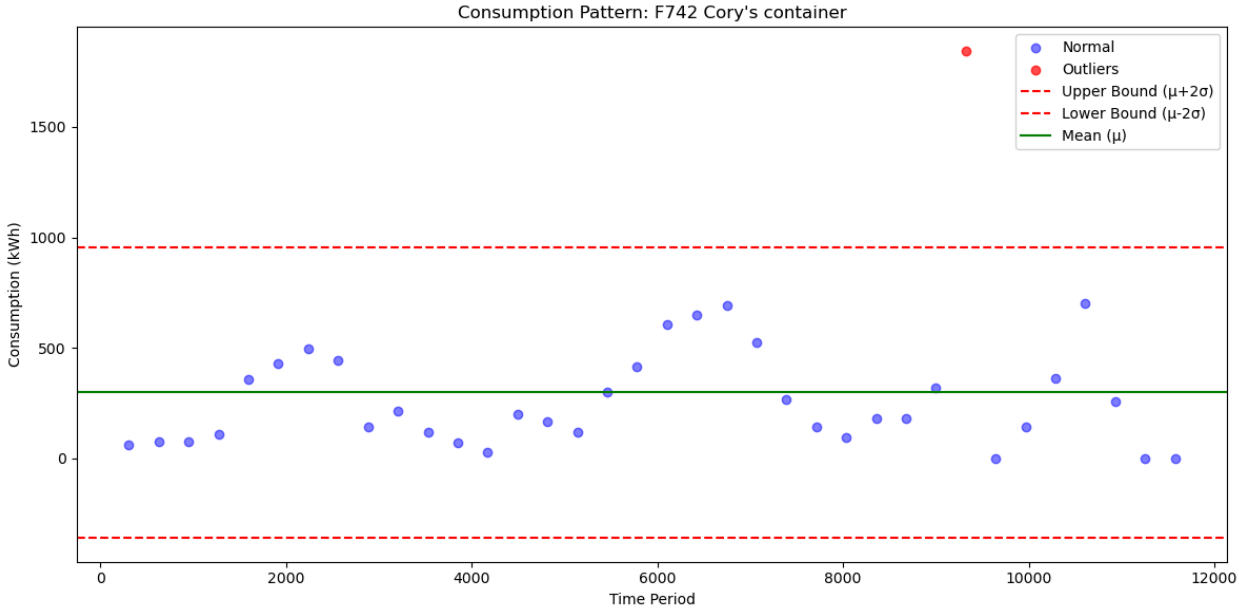

```
Statistics for F622 William James Building:
Mean: 37111.85
Standard Deviation: 11922.51
Upper Bound: 60956.87
Lower Bound: 13266.82
Number of outliers: 3
```



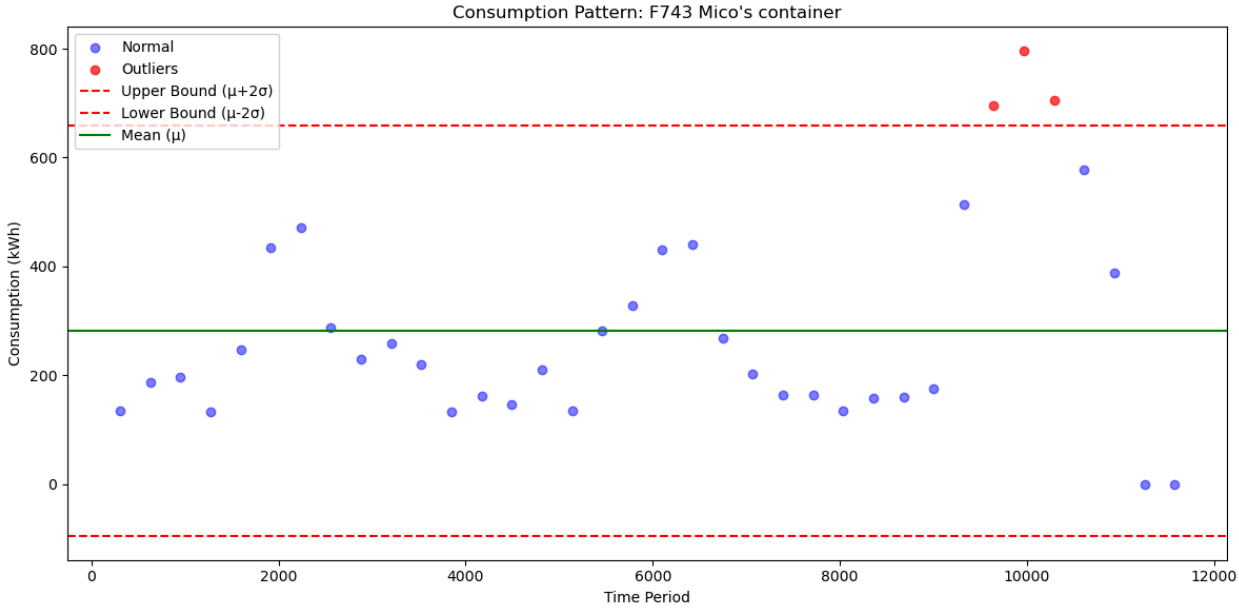

```
Statistics for F735 Ex Residence DHW:
Mean: 131.07
Standard Deviation: 312.07
Upper Bound: 755.21
Lower Bound: -493.07
Number of outliers: 2
```



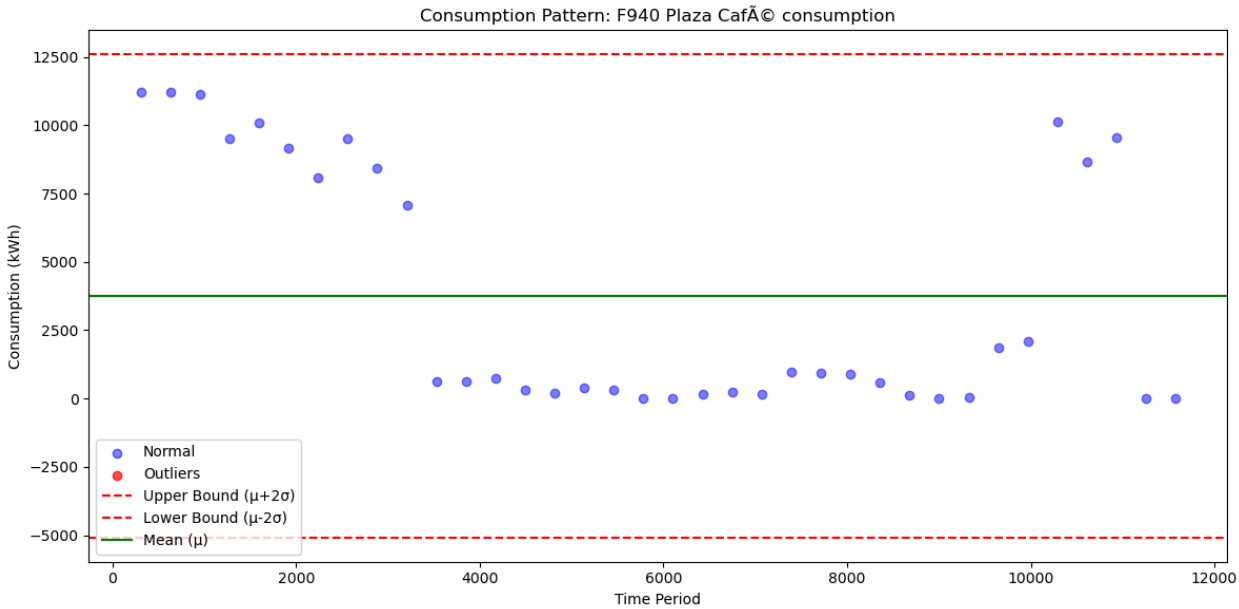

```
Statistics for F742 Cory's container:
Mean: 299.78
Standard Deviation: 328.60
Upper Bound: 956.97
Lower Bound: -357.41
Number of outliers: 1
```



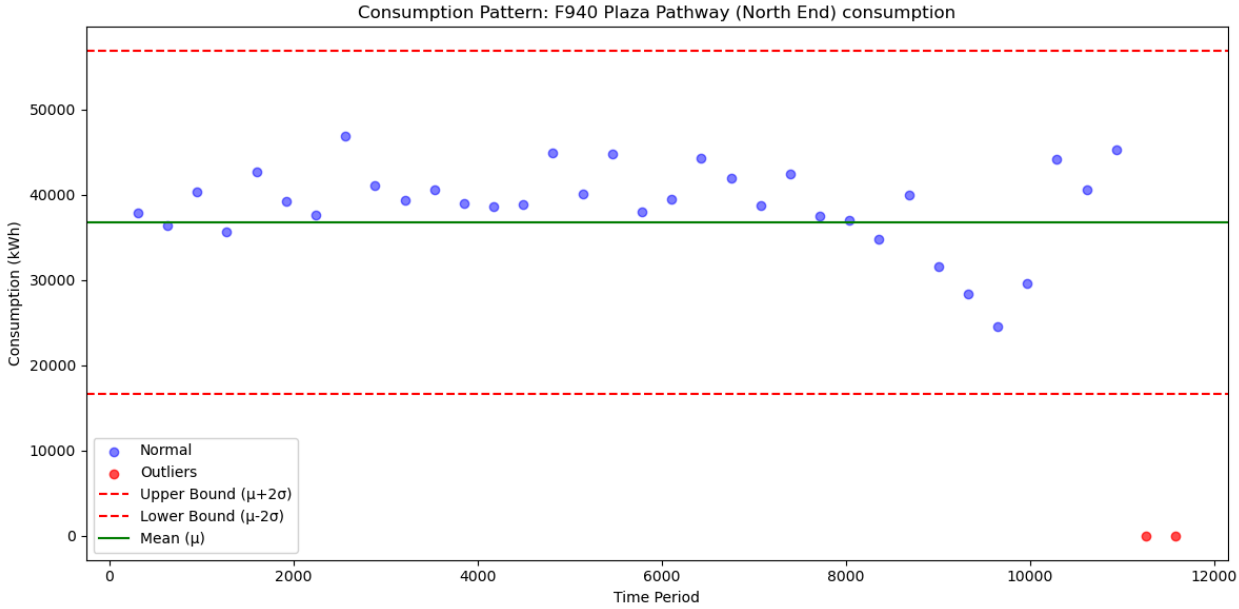

```
Statistics for F743 Mico's container:
Mean: 282.39
Standard Deviation: 188.65
Upper Bound: 659.68
Lower Bound: -94.90
Number of outliers: 3
```



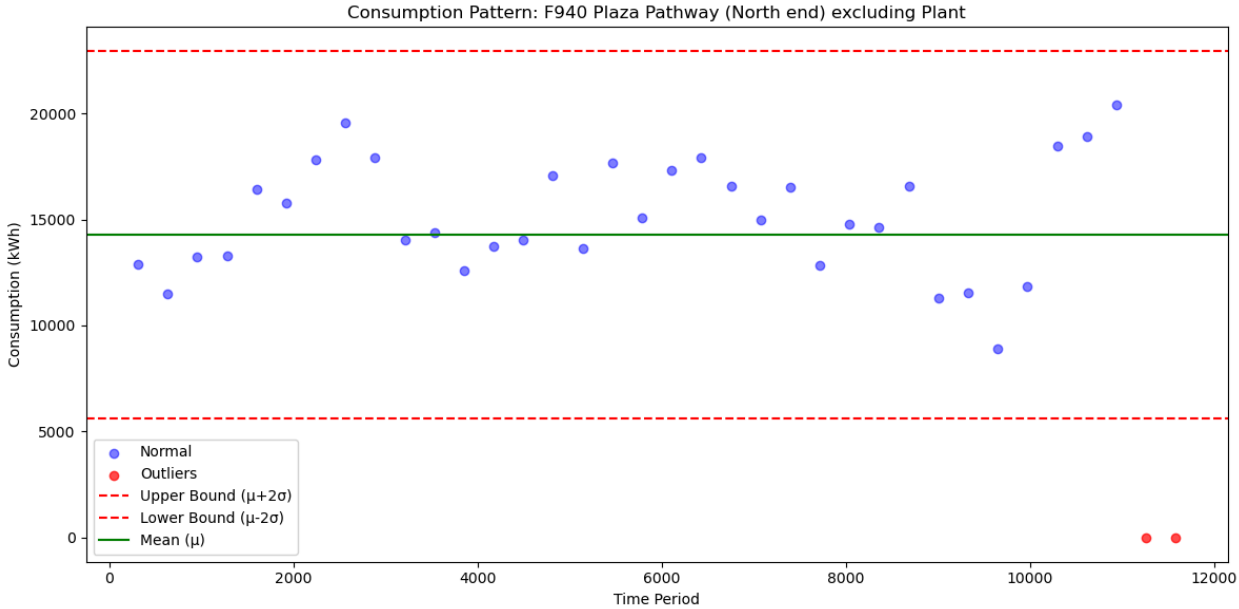

```
Statistics for F940 Plaza CafÃ© consumption:
Mean: 3753.44
Standard Deviation: 4421.05
Upper Bound: 12595.55
Lower Bound: -5088.66
Number of outliers: 0
```



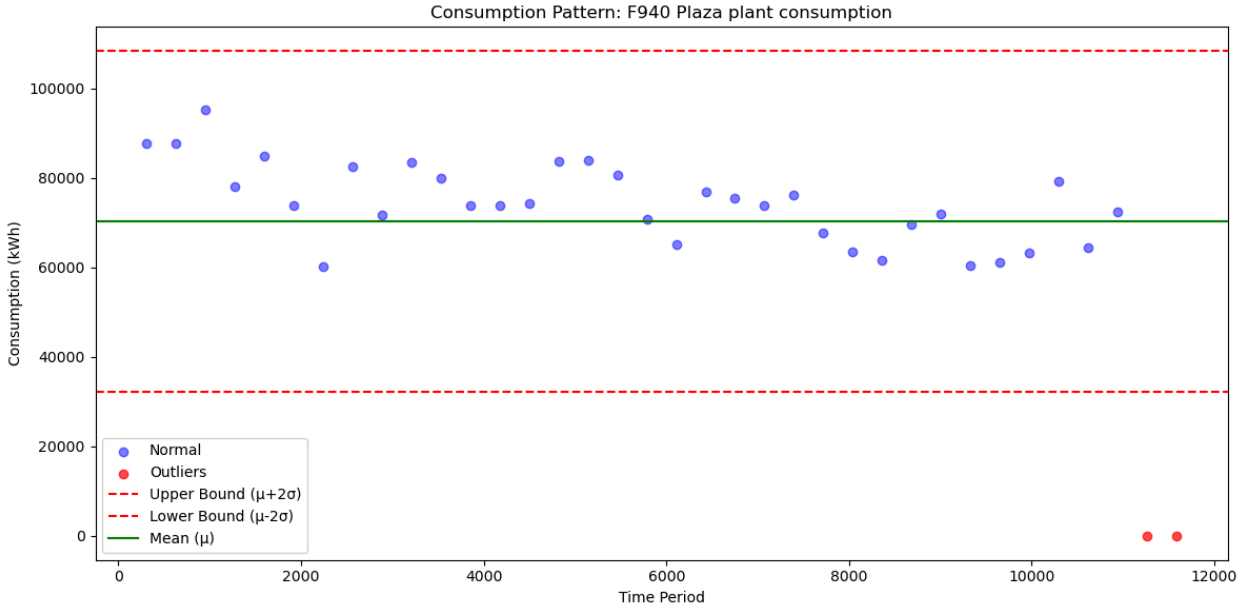

```
Statistics for F940 Plaza Pathway (North End) consumption:
Mean: 36740.69
Standard Deviation: 10057.22
Upper Bound: 56855.13
Lower Bound: 16626.25
Number of outliers: 2
```



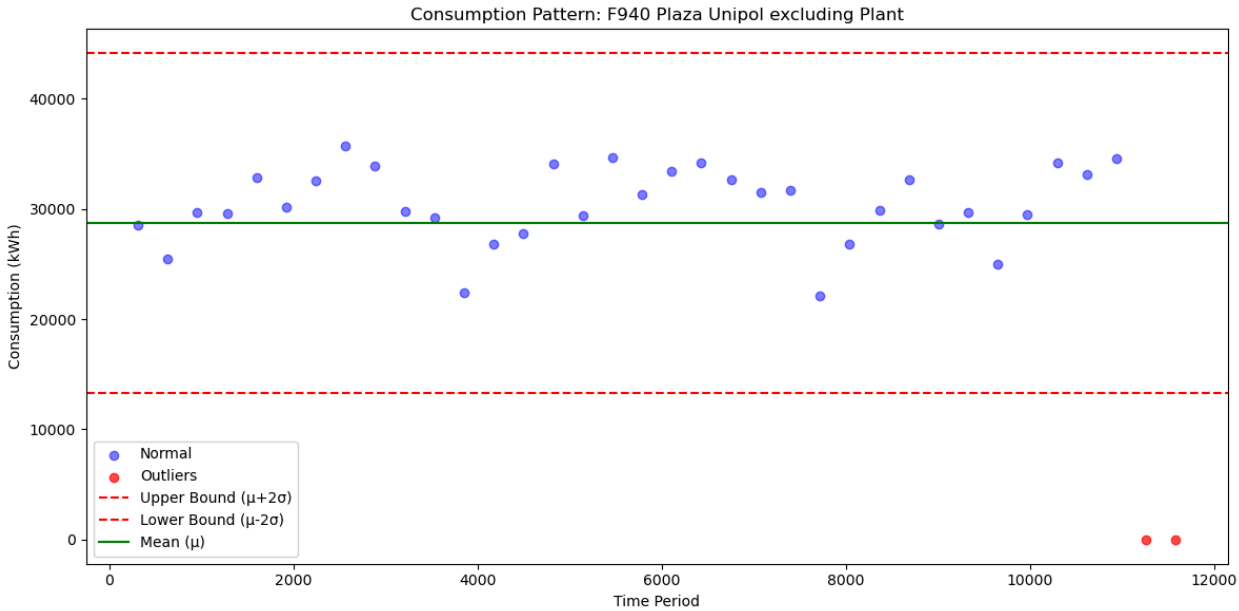

```
Statistics for F940 Plaza Pathway (North end) excluding Plant:
Mean: 14282.06
Standard Deviation: 4323.54
Upper Bound: 22929.13
Lower Bound: 5634.99
Number of outliers: 2
```



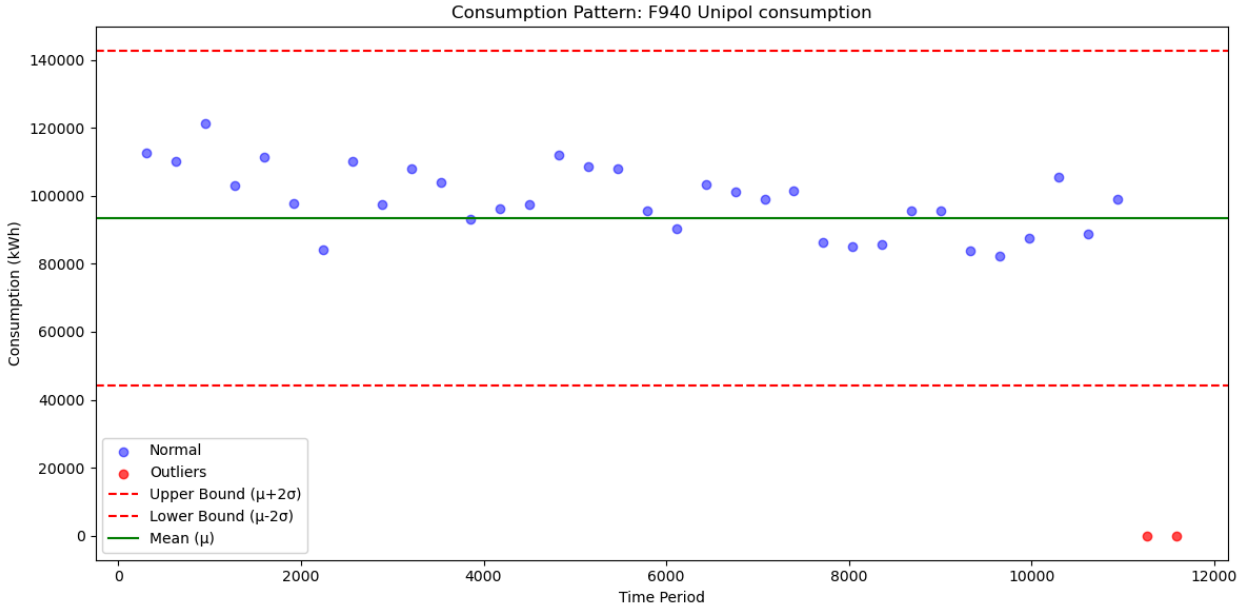

```
Statistics for F940 Plaza plant consumption:
Mean: 70250.02
Standard Deviation: 19059.68
Upper Bound: 108369.38
Lower Bound: 32130.65
Number of outliers: 2
```



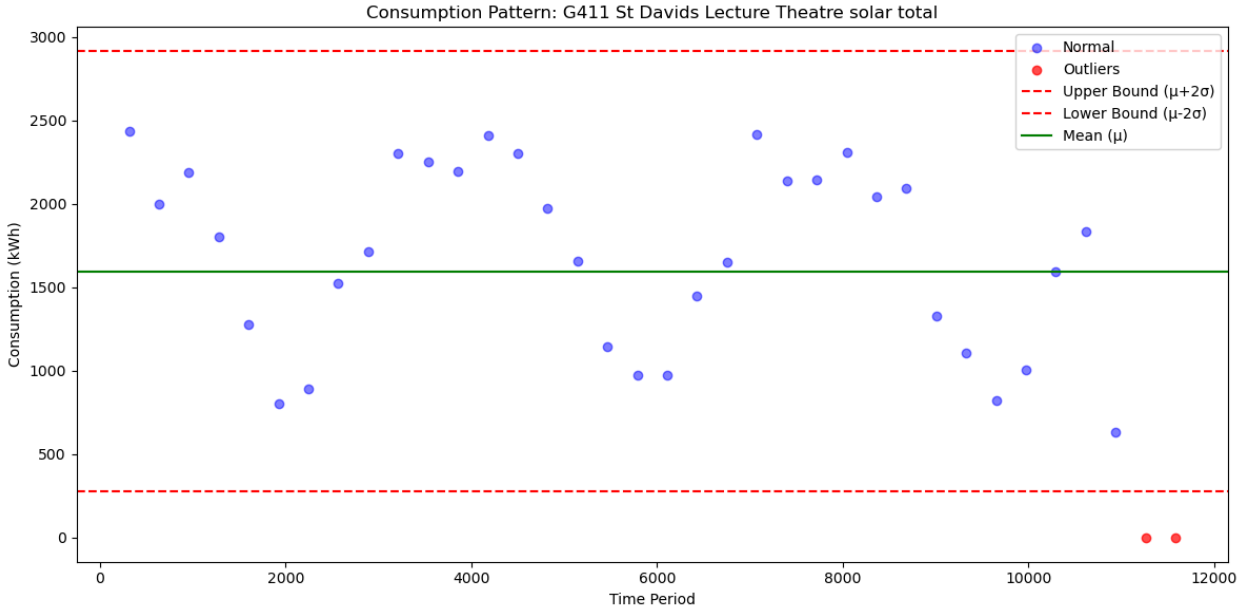

```
Statistics for F940 Plaza Unipol excluding Plant:
Mean: 28721.07
Standard Deviation: 7705.20
Upper Bound: 44131.47
Lower Bound: 13310.67
Number of outliers: 2
```



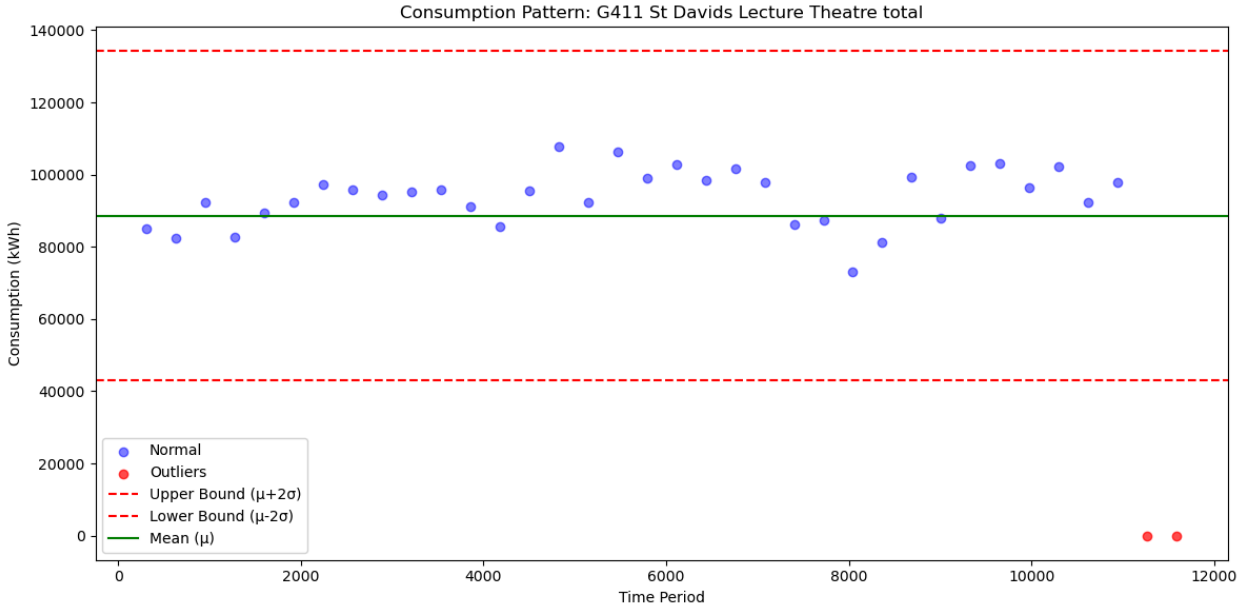

```
Statistics for F940 Unipol consumption:
Mean: 93381.85
Standard Deviation: 24579.15
Upper Bound: 142540.14
Lower Bound: 44223.55
Number of outliers: 2
```



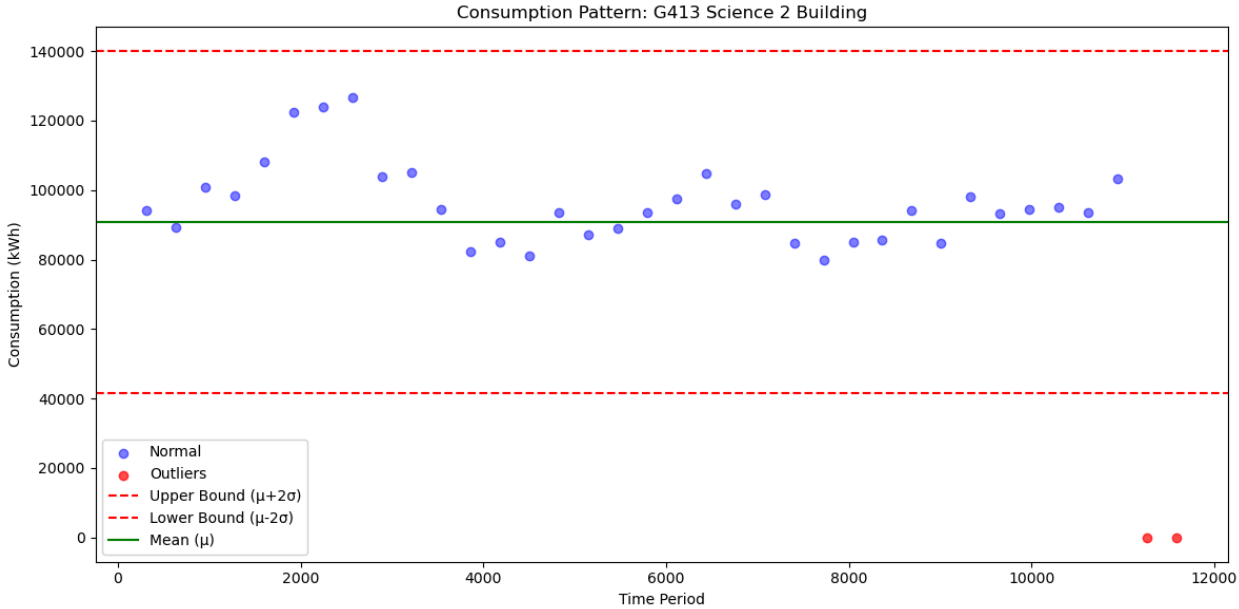

```
Statistics for G411 St Davids Lecture Theatre solar total:
Mean: 1593.83
Standard Deviation: 660.15
Upper Bound: 2914.14
Lower Bound: 273.52
Number of outliers: 2
```



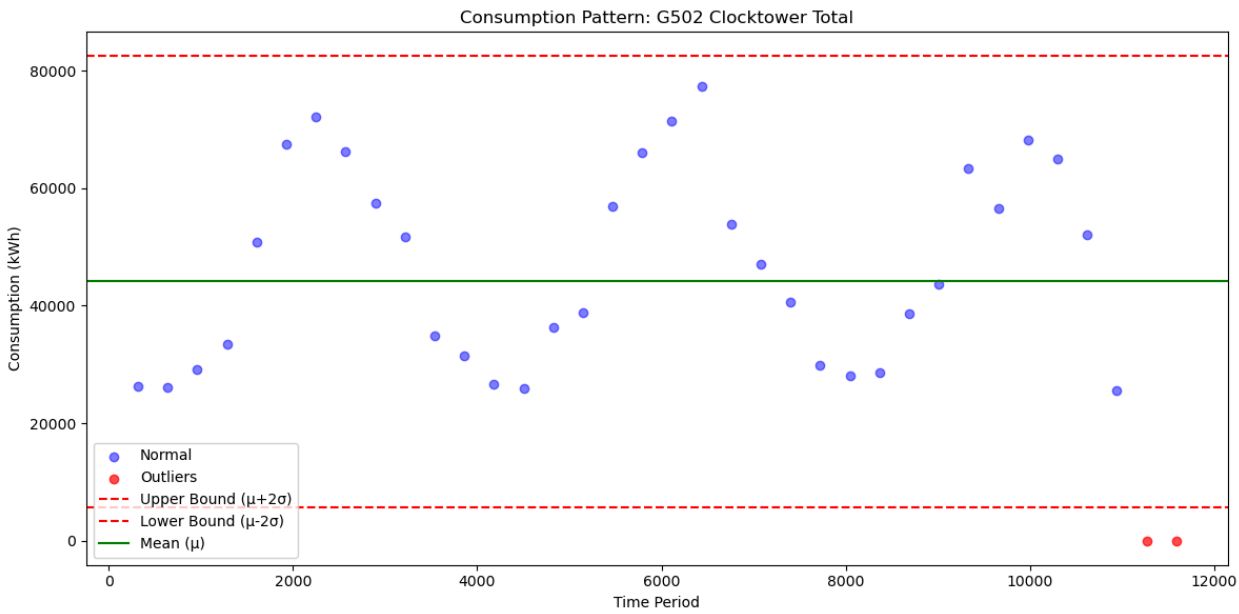

```
Statistics for G411 St Davids Lecture Theatre total:
Mean: 88651.26
Standard Deviation: 22773.42
Upper Bound: 134198.09
Lower Bound: 43104.43
Number of outliers: 2
```



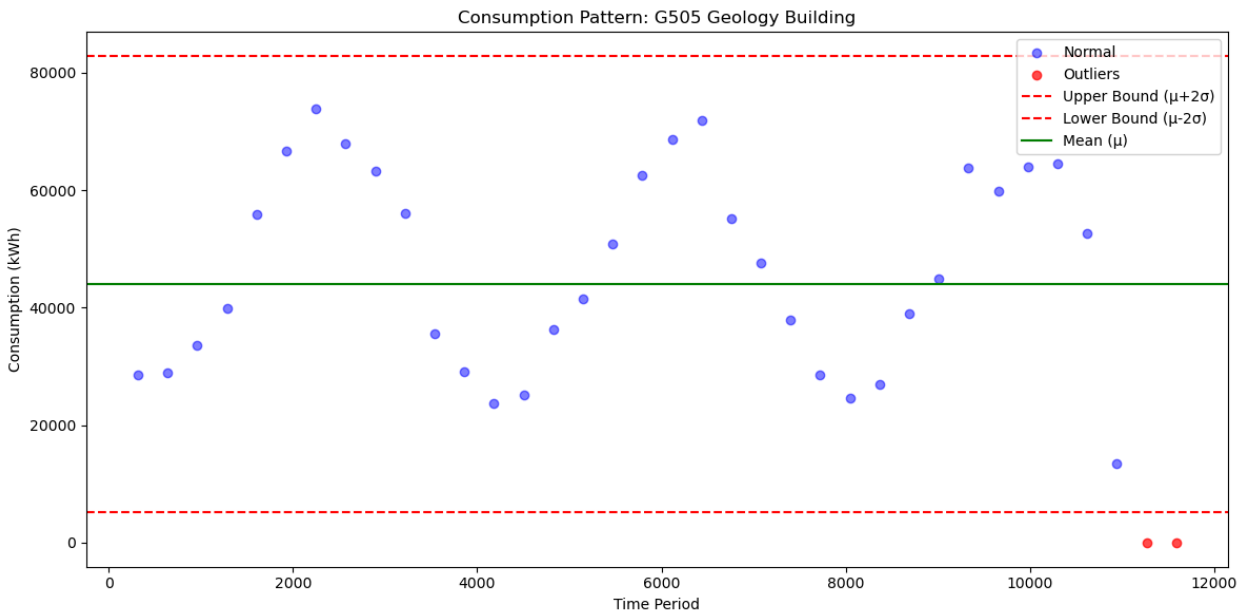

```
Statistics for G413 Science 2 Building:
Mean: 90733.26
Standard Deviation: 24602.84
Upper Bound: 139938.95
Lower Bound: 41527.57
Number of outliers: 2
```



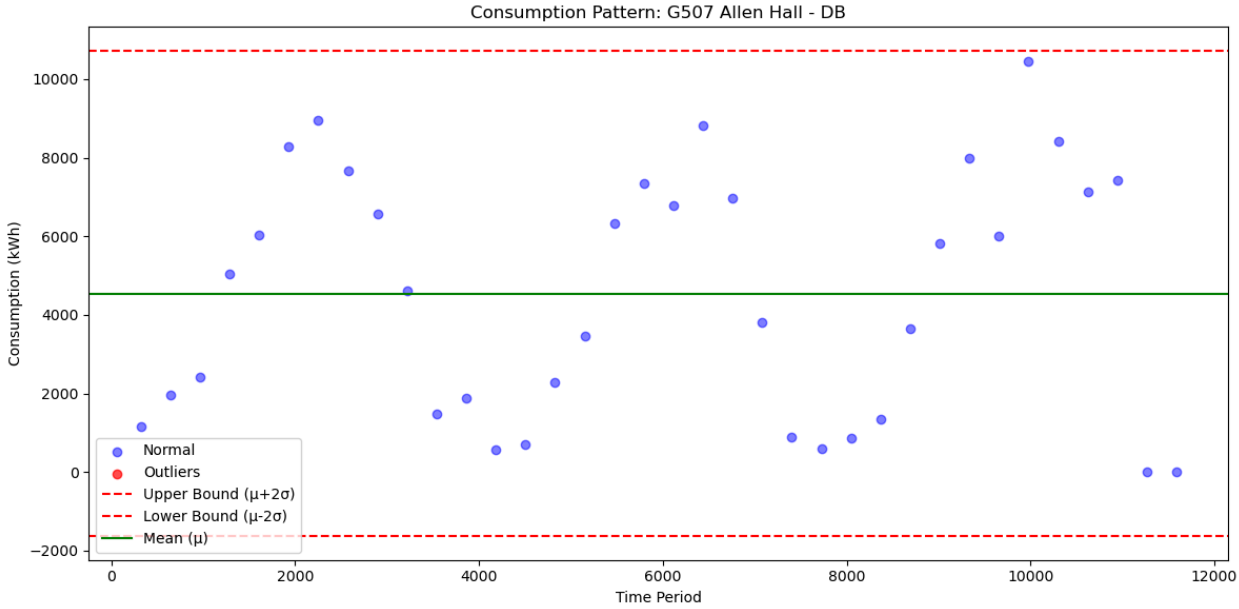

```
Statistics for G502 Clocktower Total:
Mean: 44111.88
Standard Deviation: 19176.33
Upper Bound: 82464.53
Lower Bound: 5759.22
Number of outliers: 2
```



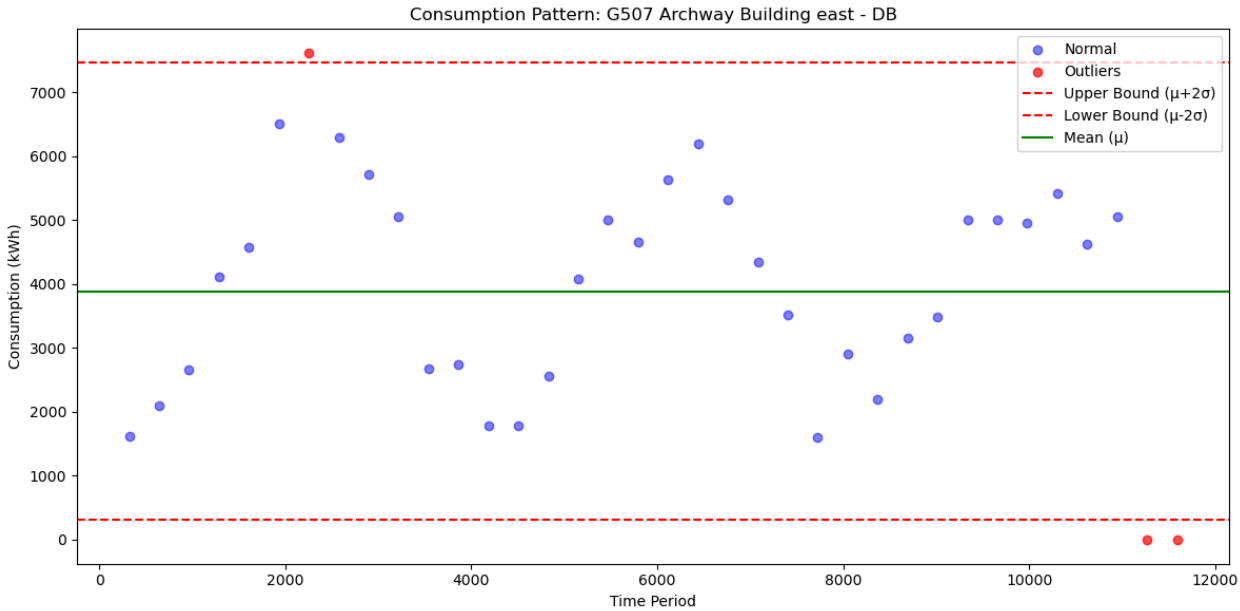

```
Statistics for G505 Geology Building:
Mean: 43969.96
Standard Deviation: 19413.98
Upper Bound: 82797.92
Lower Bound: 5142.00
Number of outliers: 2
```



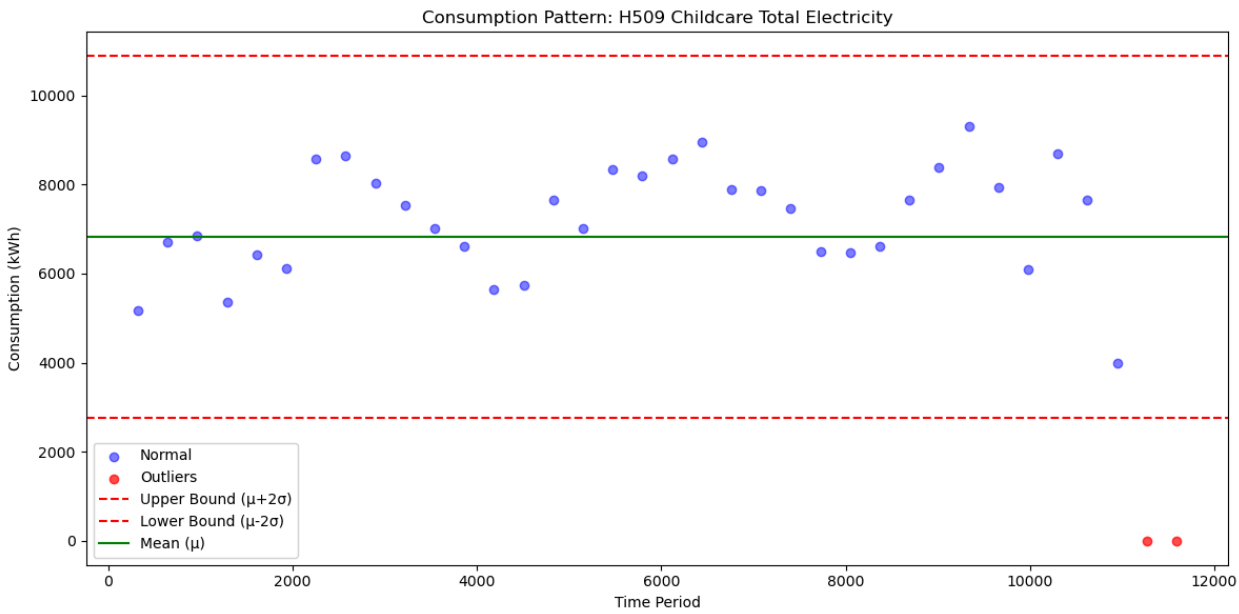

```
Statistics for G507 Allen Hall - DB:
Mean: 4545.28
Standard Deviation: 3083.36
Upper Bound: 10712.01
Lower Bound: -1621.45
Number of outliers: 0
```




```
Statistics for G507 Archway Building east - DB:
Mean: 3886.11
Standard Deviation: 1791.03
Upper Bound: 7468.17
Lower Bound: 304.05
Number of outliers: 3
```



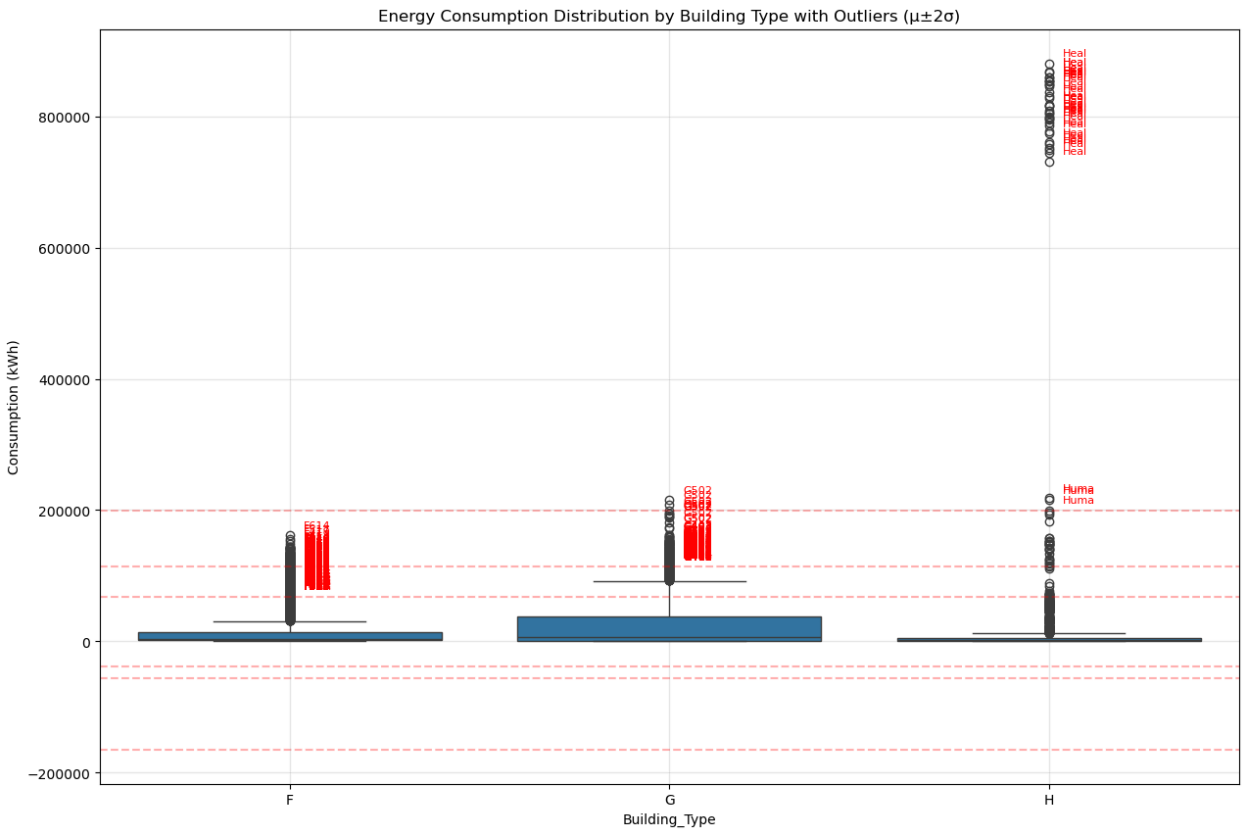

```
Statistics for H509 Childcare Total Electricity:
Mean: 6823.67
Standard Deviation: 2030.05
Upper Bound: 10883.77
Lower Bound: 2763.57
Number of outliers: 2
```



In [34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Read and prepare data
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Extract building type (F, G, or H) from Meter_Location
df['Building_Type'] = df['Meter_Location'].str.extract('^([FGH])')

# Melt the dataframe
df_melted = pd.melt(df[df['Building_Type'].isin(['F', 'G', 'H'])],
                    id_vars=['Meter_Location', 'Building_Type'],
                    var_name='Date',
                    value_name='Consumption')

# Create the plot
plt.figure(figsize=(15, 10))

# Create boxplot for each building type
sns.boxplot(x='Building_Type', y='Consumption', data=df_melted)

# Dictionary to store outliers for each building type
outliers_dict = {}

# Calculate and plot outlier thresholds for each building type
for building in ['F', 'G', 'H']:
    building_data = df_melted[df_melted['Building_Type'] == building]

    mean = np.mean(building_data['Consumption'])
    std = np.std(building_data['Consumption'])
    upper_bound = mean + (2 * std)
    lower_bound = mean - (2 * std)

    # Identify outliers
    outliers = building_data[
        (building_data['Consumption'] > upper_bound) |
        (building_data['Consumption'] < lower_bound)
    ]

    # Store outliers
    outliers_dict[building] = outliers

    # Add threshold lines
    plt.axhline(y=upper_bound, color='r', linestyle='--', alpha=0.3)
    plt.axhline(y=lower_bound, color='r', linestyle='--', alpha=0.3)

    # Label outliers
    for idx, row in outliers.iterrows():
        plt.annotate(
            row['Meter_Location'][:4],
            xy=(building_data['Building_Type'].iloc[0], row['Consumption']),
            xytext=(10, 5),
            textcoords='offset points',
            fontsize=8,
            color='red'
        )

    # Print statistics
    print(f"\nBuilding Type {building}:")
    print(f"Mean: {mean:.2f}")
```

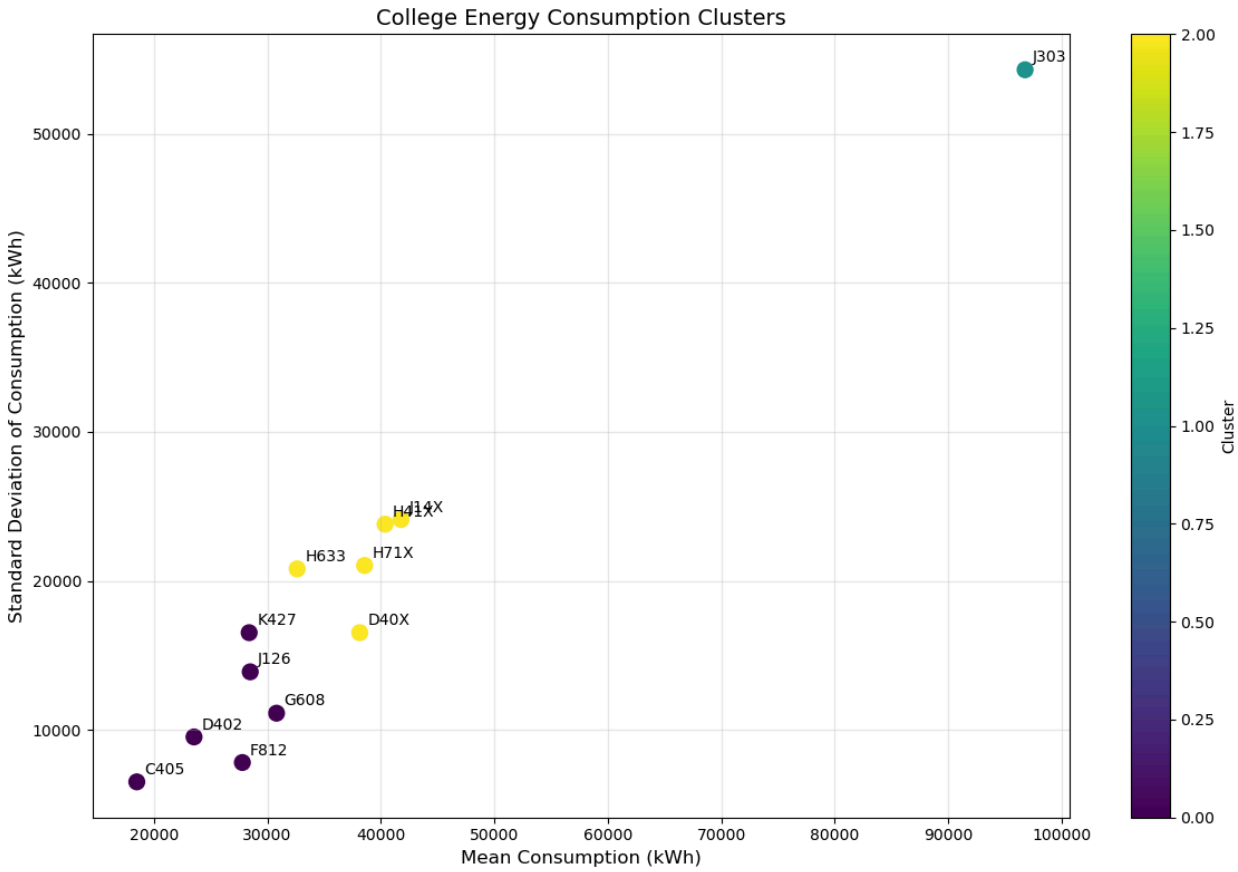
```
    print(f"Standard Deviation: {std:.2f}")
    print(f"Upper Bound (μ+2σ): {upper_bound:.2f}")
    print(f"Lower Bound (μ-2σ): {lower_bound:.2f}")
    print(f"Number of outliers: {len(outliers)}")
    print("\nOutlier Buildings:")
    for _, row in outliers.drop_duplicates('Meter_Location').iterrows():
        print(f"- {row['Meter_Location']}")

plt.title('Energy Consumption Distribution by Building Type with Outliers (μ±2σ)', fontsize=12)
plt.ylabel('Consumption (kWh)')
plt.grid(True, alpha=0.3)
plt.show()
```

```
Building Type F:
Mean: 15237.79
Standard Deviation: 26571.29
Upper Bound (μ+2σ): 68380.37
Lower Bound (μ-2σ): -37904.79
Number of outliers: 408

Outlier Buildings:
- F204 444 Great King Street - kWh
- F940 Plaza Building - kWh
- F419 ISB east 1 main
- F940 Plaza 1 Mains
- F419 ISB East Excluding Plant
- F940 Plaza plant consumption
- F940 Unipol consumption
- F419 ISB west (1) Mains
- F618 1 Psychology substation- Goddard
- F419 ISB West Excluding Shops
- F614 OBS Total
- F505 1 Richardson Mains
- F507 - F518 ITS Substation Mains F512
- F614 1 School of Business Incomer 1 (Lower floors)
- F325 PhysEd Flume
- F9XX College of Education main (Boiler room) - kWh
- F900 College of Education
- F614 2 School of Business Incomer 2 (Upper floors)
- F517 97 Albany St

Building Type G:
Mean: 28642.57
Standard Deviation: 42457.00
Upper Bound (μ+2σ): 113556.56
Lower Bound (μ-2σ): -56271.43
Number of outliers: 210

Outlier Buildings:
- G403 Biochemistry - kWh
- G413 Science 2 - kWh
- G403 Biochemistry
- G405 Science 3 - MSB
- G413 1 Science 2 Main Board
- G401 Mellor Laboratories - kWh
- G401 1 Mellor Laboratories Ground MSB
- G502 1 Main Block substation G508
- G413 Science 2 Building
- G404 Microbiology - kWh
- G404 Microbiology

Building Type H:
Mean: 17048.87
Standard Deviation: 90798.46
Upper Bound (μ+2σ): 198645.79
Lower Bound (μ-2σ): -164548.05
Number of outliers: 37

Outlier Buildings:
- Health Science Total -kWh
- Humanities Total - kWh
```



In [4]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')
df.set_index('Meter_Location', inplace=True)

# Define the Colleges group
colleges = [
    "C405 192 Castle College - kWh",
    "D402 Hayward College - kWh",
    "D40X Cumberland College - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G608 St Margarets College - kWh",
    "H41X Selwyn College - kWh E2",
    "H633 Arana College main - kWh",
    "H71X Studholm College - kWh E2",
    "J126 Carrington College (Kitchen/Dining) - kWh",
    "J14X Aquinas College - kWh",
    "J303 Caroline Freeman College - kWh",
    "K427 Abbey College - kWh"
]

# Filter and clean data
colleges_df = df.loc[df.index.isin(colleges)]
colleges_df = colleges_df.replace([np.inf, -np.inf], np.nan)
colleges_df = colleges_df.dropna(axis=1, how='all')

# Calculate features
features_dict = {
    'mean_consumption': colleges_df.mean(axis=1),
    'std_consumption': colleges_df.std(axis=1),
    'max_consumption': colleges_df.max(axis=1)
}

# Create features DataFrame
features_df = pd.DataFrame(features_dict)
features_df = features_df.dropna()

# Only proceed if we have data
if len(features_df) > 0:
    # Scale features
    scaler = StandardScaler()
    scaled_features = scaler.fit_transform(features_df)

    # Perform clustering
    n_clusters = min(3, len(features_df))  # Ensure we don't have more clusters than samples
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    features_df['cluster'] = kmeans.fit_predict(scaled_features)
```

```
    # Visualization
    plt.figure(figsize=(12, 8))
    scatter = plt.scatter(features_df['mean_consumption'],
                          features_df['std_consumption'],
                          c=features_df['cluster'],
                          cmap='viridis',
                          s=100)

    # Add labels
    plt.title('College Energy Consumption Clusters', fontsize=14)
    plt.xlabel('Mean Consumption (kWh)', fontsize=12)
    plt.ylabel('Standard Deviation of Consumption (kWh)', fontsize=12)

    # Add building labels
    for idx, row in features_df.iterrows():
        plt.annotate(idx[:4],
                     (row['mean_consumption'], row['std_consumption']),
                     xytext=(5, 5), textcoords='offset points')

    plt.colorbar(scatter, label='Cluster')
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    # Print cluster statistics
    print("\nCluster Statistics:")
    for cluster in range(n_clusters):
        cluster_buildings = features_df[features_df['cluster'] == cluster].index
        print(f"\nCluster {cluster}:")
        print(f"Buildings: {', '.join(b[:4] for b in cluster_buildings)}")
else:
    print("No valid data available for clustering")
```

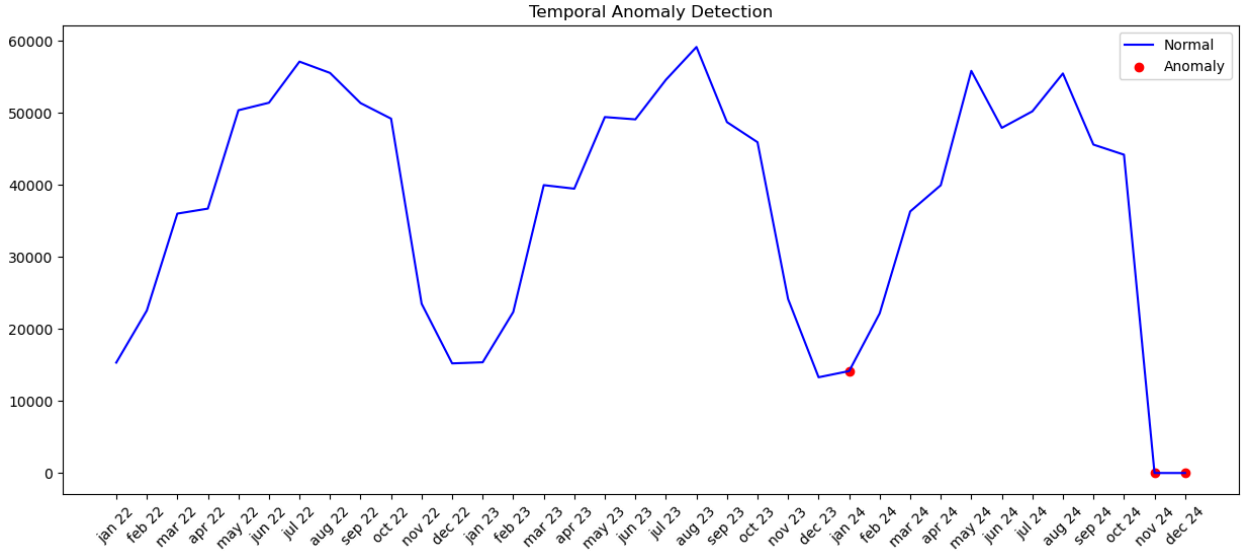



```
Cluster Statistics:

Cluster 0:
Buildings: C405, D402, F812, G608, J126, K427

Cluster 1:
Buildings: J303

Cluster 2:
Buildings: D40X, H41X, H633, H71X, J14X
```

In [5]:
```
from sklearn.ensemble import IsolationForest
import numpy as np

# Prepare data matrix
X = colleges_df.T.fillna(method='ffill').values
```

```
# Train model
clf = IsolationForest(contamination=0.08)
anomalies = clf.fit_predict(X)

# Visualize results
plt.figure(figsize=(15,6))
plt.plot(colleges_df.T.index, X.mean(axis=1),
        label='Normal', color='blue')
plt.scatter(colleges_df.T.index[anomalies==-1],
           X.mean(axis=1)[anomalies==-1],
           color='red', label='Anomaly')
plt.title('Temporal Anomaly Detection')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

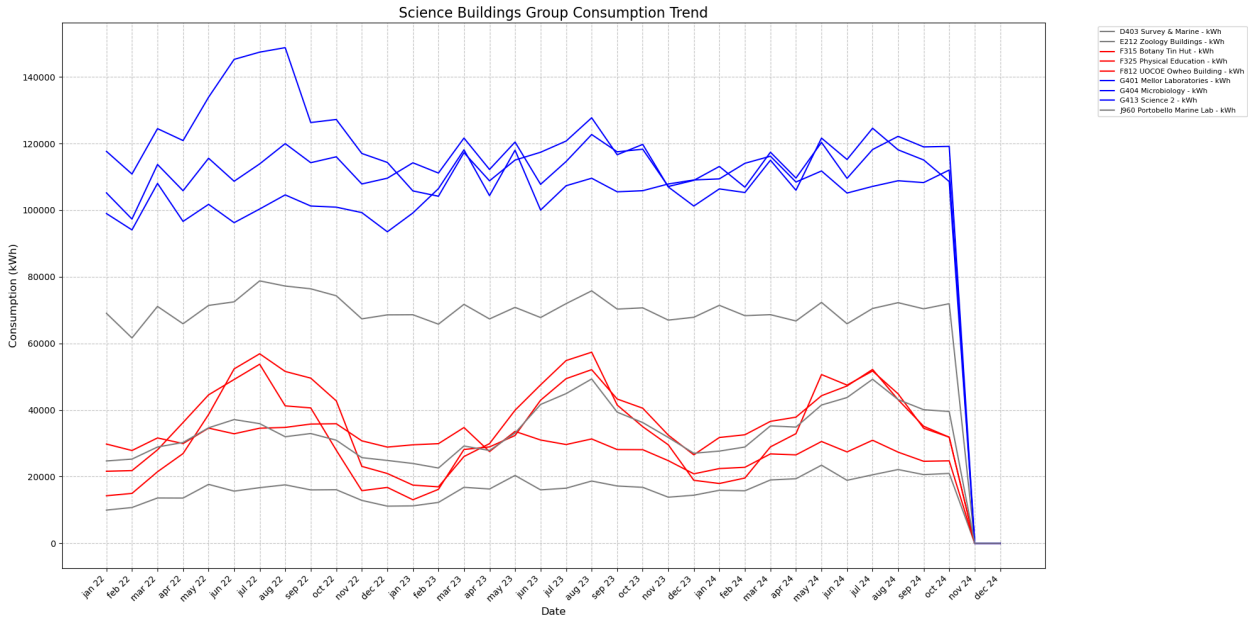
```
C:\Users\sugan\AppData\Local\Temp\ipykernel_8904\2229241827.py:5: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  X = colleges_df.T.fillna(method='ffill').values
```



In [6]:

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

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Set the 'Meter_Location' column as the index
df.set_index('Meter_Location', inplace=True)

# Define the Science group
science = [
    "D403 Survey & Marine - kWh",
    "E212 Zoology Buildings - kWh",
    "F315 Botany Tin Hut - kWh",
    "F325 Physical Education - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G401 Mellor Laboratories - kWh",
    "G404 Microbiology - kWh",
    "G413 Science 2 - kWh",
    "J960 Portobello Marine Lab - kWh",
    "G505 Geology north:",
    "G505 Geology south:"
]

# Filter the dataframe to include only the Science group
science_df = df.loc[df.index.isin(science)]

# Transpose the dataframe for easier plotting
science_df_t = science_df.T

# Create the plot
plt.figure(figsize=(20, 10))

# Color mapping based on the first letter of the building code
color_map = {
    'D': 'grey',
    'E': 'grey',
    'F': 'red',    # Highlighting F buildings
    'G': 'blue',   # Highlighting G buildings
    'J': 'grey'
}

# Plot each building's consumption
for building in science_df_t.columns:
    # Determine color based on the first letter of the building code
```

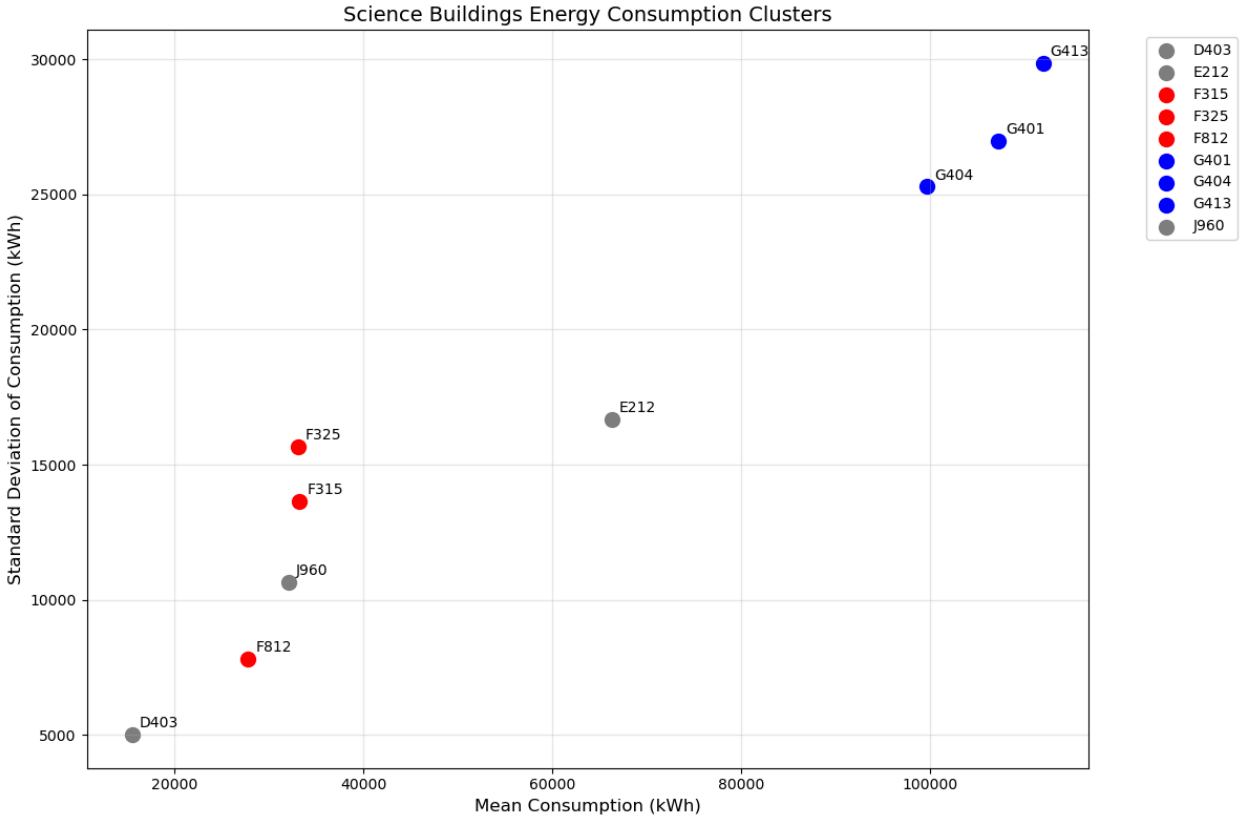
```
    color = color_map.get(building[0], 'grey')

    plt.plot(science_df_t.index, science_df_t[building], label=building, color=color)

# Customize the plot
plt.title('Science Buildings Group Consumption Trend', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Consumption (kWh)', fontsize=12)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=8)
plt.grid(True, linestyle='--', alpha=0.7)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Adjust layout to prevent cutting off labels
plt.tight_layout()

# Show the plot
plt.show()
```



In [7]:

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')
df.set_index('Meter_Location', inplace=True)

# Define the Science group
science = [
    "D403 Survey & Marine - kWh",
    "E212 Zoology Buildings - kWh",
    "F315 Botany Tin Hut - kWh",
    "F325 Physical Education - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G401 Mellor Laboratories - kWh",
    "G404 Microbiology - kWh",
    "G413 Science 2 - kWh",
    "J960 Portobello Marine Lab - kWh",
    "G505 Geology north:",
    "G505 Geology south:"
]

# Filter and clean data
science_df = df.loc[df.index.isin(science)]
science_df = science_df.replace([np.inf, -np.inf], np.nan)
science_df = science_df.dropna(axis=1, how='all')

# Calculate features
features_dict = {
    'mean_consumption': science_df.mean(axis=1),
    'std_consumption': science_df.std(axis=1),
    'max_consumption': science_df.max(axis=1)
}

# Create features DataFrame
features_df = pd.DataFrame(features_dict)
features_df = features_df.dropna()

# Only proceed if we have data
```

```python
if len(features_df) > 0:
    # Scale features
    scaler = StandardScaler()
    scaled_features = scaler.fit_transform(features_df)

    # Perform clustering
    n_clusters = min(3, len(features_df))  # Ensure we don't have more clusters than samples
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    features_df['cluster'] = kmeans.fit_predict(scaled_features)

    # Create color mapping for buildings
    def get_color(building_code):
        if building_code.startswith('F'):
            return 'red'
        elif building_code.startswith('G'):
            return 'blue'
        else:
            return 'grey'

    # Visualization
    plt.figure(figsize=(12, 8))

    # Create scatter plot with custom colors
    for idx, row in features_df.iterrows():
        plt.scatter(row['mean_consumption'],
                    row['std_consumption'],
                    c=get_color(idx),
                    s=100,
                    label=idx[:4])

    # Add labels
    plt.title('Science Buildings Energy Consumption Clusters', fontsize=14)
    plt.xlabel('Mean Consumption (kWh)', fontsize=12)
    plt.ylabel('Standard Deviation of Consumption (kWh)', fontsize=12)

    # Add building labels
    for idx, row in features_df.iterrows():
        plt.annotate(idx[:4],
                     (row['mean_consumption'], row['std_consumption']),
                     xytext=(5, 5), textcoords='offset points')

    # Add legend
    plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

    # Print cluster statistics with building types
    print("\nCluster Statistics:")
    for cluster in range(n_clusters):
        cluster_buildings = features_df[features_df['cluster'] == cluster].index
        print(f"\nCluster {cluster}:")
        print("Buildings:")
        for building in cluster_buildings:
            building_type = "F Building" if building.startswith('F') else \
                            "G Building" if building.startswith('G') else \
                            "Other Building"
            print(f"- {building[:30]} ({building_type})")

        # Calculate cluster metrics
        cluster_data = features_df[features_df['cluster'] == cluster]
        print(f"Average Consumption: {cluster_data['mean_consumption'].mean():.2f} kWh")
        print(f"Average Std Dev: {cluster_data['std_consumption'].mean():.2f} kWh")
else:
    print("No valid data available for clustering")
```

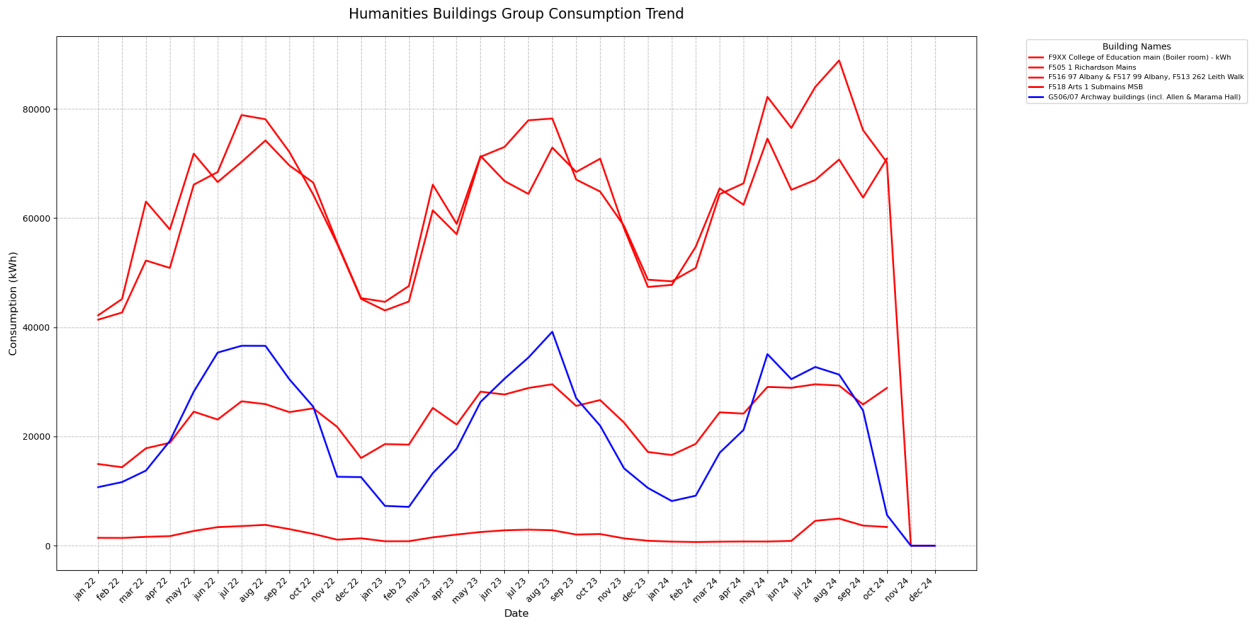



```
Cluster Statistics:

Cluster 0:
Buildings:
- E212 Zoology Buildings - kWh (Other Building)
- F315 Botany Tin Hut - kWh (F Building)
- F325 Physical Education - kWh (F Building)
- J960 Portobello Marine Lab - k (Other Building)
Average Consumption: 41161.21 kWh
Average Std Dev: 14150.34 kWh

Cluster 1:
Buildings:
- G401 Mellor Laboratories - kWh (G Building)
- G404 Microbiology - kWh (G Building)
- G413 Science 2 - kWh (G Building)
Average Consumption: 106286.26 kWh
Average Std Dev: 27374.60 kWh

Cluster 2:
Buildings:
- D403 Survey & Marine - kWh (Other Building)
- F812 UOCOE Owheo Building - kW (F Building)
Average Consumption: 21656.67 kWh
Average Std Dev: 6417.12 kWh
```

In [8]:

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Set the 'Meter_Location' column as the index
df.set_index('Meter_Location', inplace=True)

# Define the Humanities group
humanities = [
    "F9XX College of Education main (Boiler room) - kWh",
    "F518 Arts 1 Submains MSB",
    "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk",
    "F505 1 Richardson Mains",
    "G506/07 Archway buildings (incl. Allen & Marama Hall)"
]

# Filter the dataframe to include only the Humanities group
humanities_df = df.loc[df.index.isin(humanities)]

# Transpose the dataframe for easier plotting
humanities_df_t = humanities_df.T

# Create the plot
plt.figure(figsize=(20, 10))
```

```
# Color mapping based on the first letter of the building code
color_map = {
    'F': 'red',    # Highlighting F buildings
    'G': 'blue'    # Highlighting G buildings
}

# Plot each building's consumption
for building in humanities_df_t.columns:
    # Determine color based on the first letter of the building code
    color = color_map.get(building[0], 'grey')

    plt.plot(humanities_df_t.index, humanities_df_t[building],
             label=building,
             color=color,
             linewidth=2)

# Customize the plot
plt.title('Humanities Buildings Group Consumption Trend', fontsize=16, pad=20)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Consumption (kWh)', fontsize=12)
plt.legend(bbox_to_anchor=(1.05, 1),
           loc='upper left',
           fontsize=8,
           title='Building Names')
plt.grid(True, linestyle='--', alpha=0.7)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Adjust layout to prevent cutting off labels
plt.tight_layout()

# Show the plot
plt.show()
```

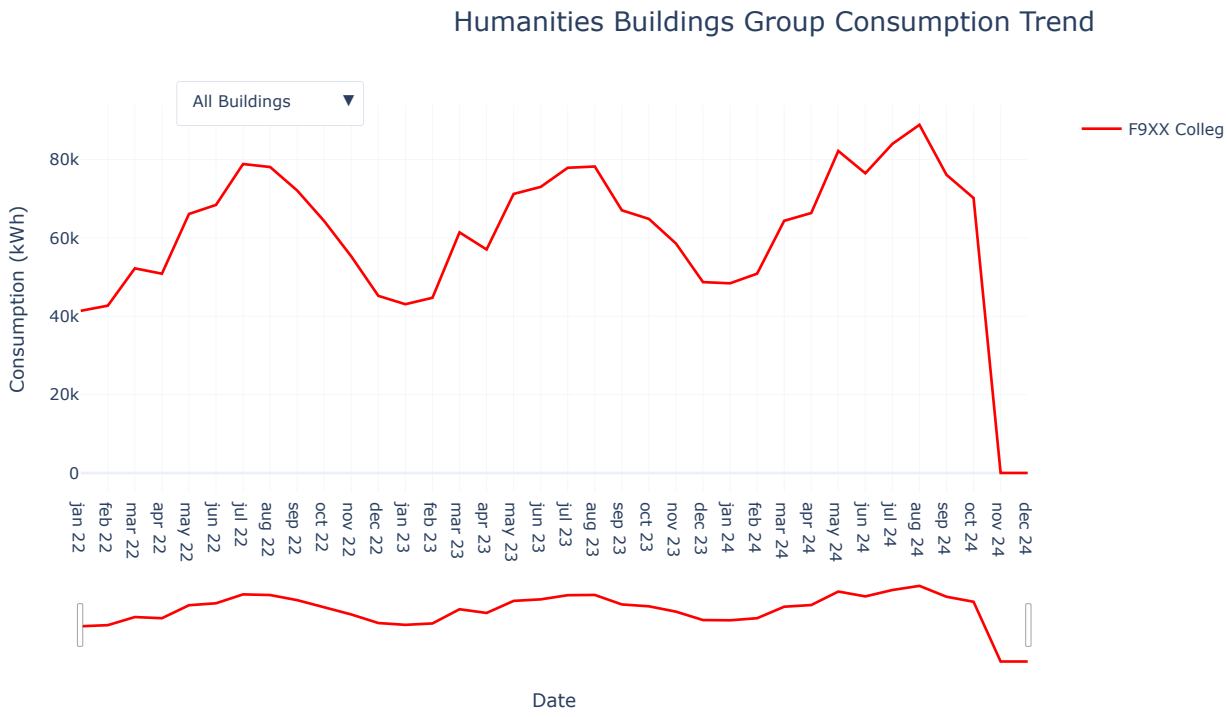


In [9]:

```
import pandas as pd
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Load the dataset
df = pd.read_excel('Stream_Janitza_Data.xlsx')
df.set_index('Meter_Location', inplace=True)

# Define the Humanities group
humanities = [
    "F9XX College of Education main (Boiler room) - kWh",
    "F518 Arts 1 Submains MSB - kWh",
    "F516 97 Albany & F517 99 Albany - kWh",
    "F505 1 Richardson Mains - kWh",
    "G506/07 Archway buildings (incl. Allen & Marama Hall) - kWh"
]

# Filter the dataframe to include only the Humanities group
humanities_df = df.loc[df.index.isin(humanities)]
humanities_df_t = humanities_df.T

# Create color mapping
color_map = {
    'F': 'red',    # F buildings in red
    'G': 'blue'    # G buildings in blue
}
```

```python
# Create the interactive plot
fig = go.Figure()

# Add traces for each building
for building in humanities_df_t.columns:
    color = color_map.get(building[0], 'grey')

    fig.add_trace(
        go.Scatter(
            x=humanities_df_t.index,
            y=humanities_df_t[building],
            name=building,
            line=dict(color=color),
            hovertemplate=
            '<b>%{x}</b><br>' +
            'Consumption: %{y:.2f} kWh<br>' +
            '<extra></extra>'
        )
    )

# Update layout
fig.update_layout(
    title={
        'text': 'Humanities Buildings Group Consumption Trend',
        'y':0.95,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top',
        'font': dict(size=20)
    },
    xaxis_title="Date",
    yaxis_title="Consumption (kWh)",
    hovermode='x unified',
    legend={
        'yanchor':"top",
        'y':0.99,
        'xanchor':"left",
        'x':1.05
    },
    showlegend=True,
    width=1200,
    height=600,
    template='plotly_white'
)

# Add range slider
fig.update_xaxes(rangeslider_visible=True)

# Add buttons for time range selection
fig.update_layout(
    updatemenus=[
        dict(
            buttons=list([
                dict(
                    args=[{"visible": [True] * len(humanities_df_t.columns)}],
                    label="All Buildings",
                    method="restyle"
                ),
                dict(
                    args=[{"visible": [building.startswith('F') for building in humanities_df_t.columns]}],
                    label="F Buildings Only",
                    method="restyle"
                ),
                dict(
                    args=[{"visible": [building.startswith('G') for building in humanities_df_t.columns]}],
                    label="G Buildings Only",
                    method="restyle"
                )
            ]),
            direction="down",
            pad={"r": 10, "t": 10},
            showactive=True,
            x=0.1,
            xanchor="left",
            y=1.1,
            yanchor="top"
        ),
    ]
)

# Show the plot
fig.show()

# Calculate and print statistics
print("\nBuilding Statistics:")
for building in humanities_df.index:
    stats = humanities_df.loc[building].describe()
    print(f"\n{building}")
    print(f"Mean consumption: {stats['mean']:.2f} kWh")
```

```
    print(f"Max consumption: {stats['max']:.2f} kWh")
    print(f"Standard deviation: {stats['std']:.2f} kWh")
```

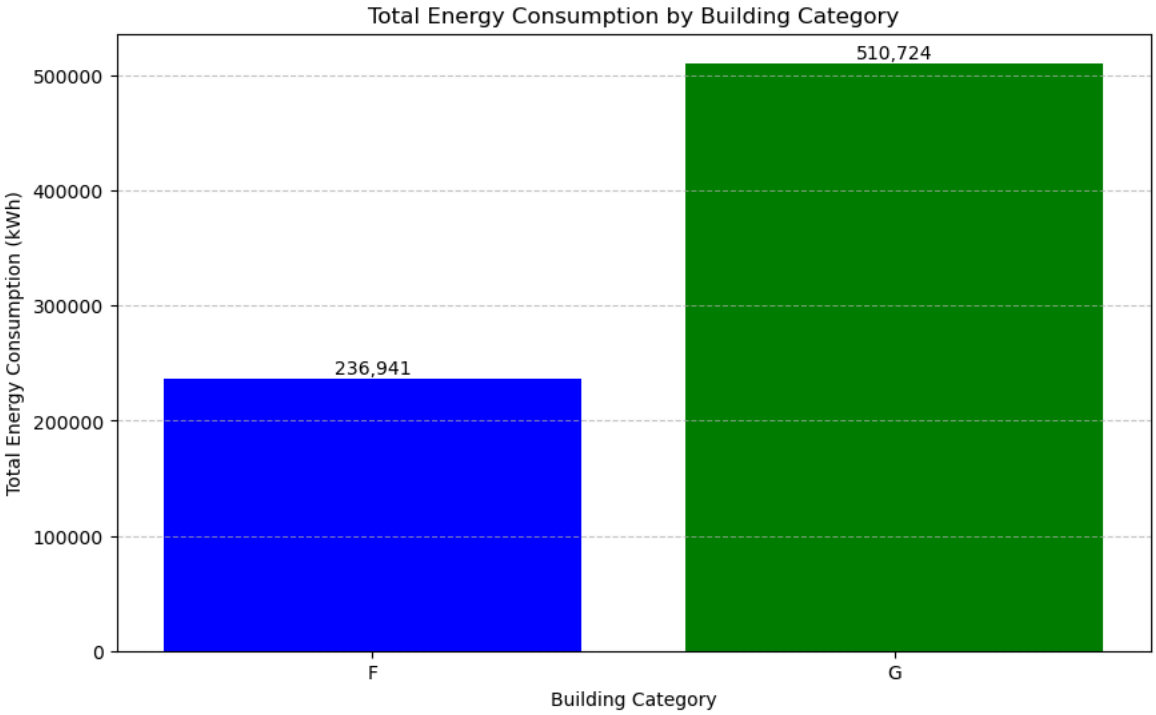



```
Building Statistics:

F9XX College of Education main (Boiler room) - kWh
Mean consumption: 60263.75 kWh
Max consumption: 88879.22 kWh
Standard deviation: 19772.67 kWh
```

In [10]:

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

# Data from the image
data = {
    'Building': [
        'F812 UOCOE Owheo Building', 'F813 UOCOE Robertson Library', 'F940 Plaza Building',
        'F9XX College of Education main', 'G401 Mellor Laboratories', 'G403 Biochemistry',
        'G404 Microbiology', 'G413 Science 2', 'G608 St Margarets College', 'G60X UNICOL'
    ],
    'kWh': [
        29788.49, 33022.12, 132734.41, 41395.89, 105178.75, 142364.04,
        98978.64, 117647.74, 15498.58, 31056.27
    ]
}

# Create DataFrame
df = pd.DataFrame(data)

# Categorize buildings by their prefixes
def categorize_building(building):
    if building.startswith('F'):
        return 'F'
    elif building.startswith('G'):
        return 'G'
    elif building.startswith('H'):
        return 'H'
    else:
        return 'Other'

# Add category column
df['Category'] = df['Building'].apply(categorize_building)

# Calculate sum by category
category_sums = df.groupby('Category')['kWh'].sum()

# Create bar plot
plt.figure(figsize=(10, 6))
colors = {'F': 'blue', 'G': 'green', 'H': 'red', 'Other': 'grey'}
bars = plt.bar(category_sums.index, category_sums.values,
               color=[colors.get(cat, 'grey') for cat in category_sums.index])

# Customize the plot
plt.xlabel('Building Category')
```

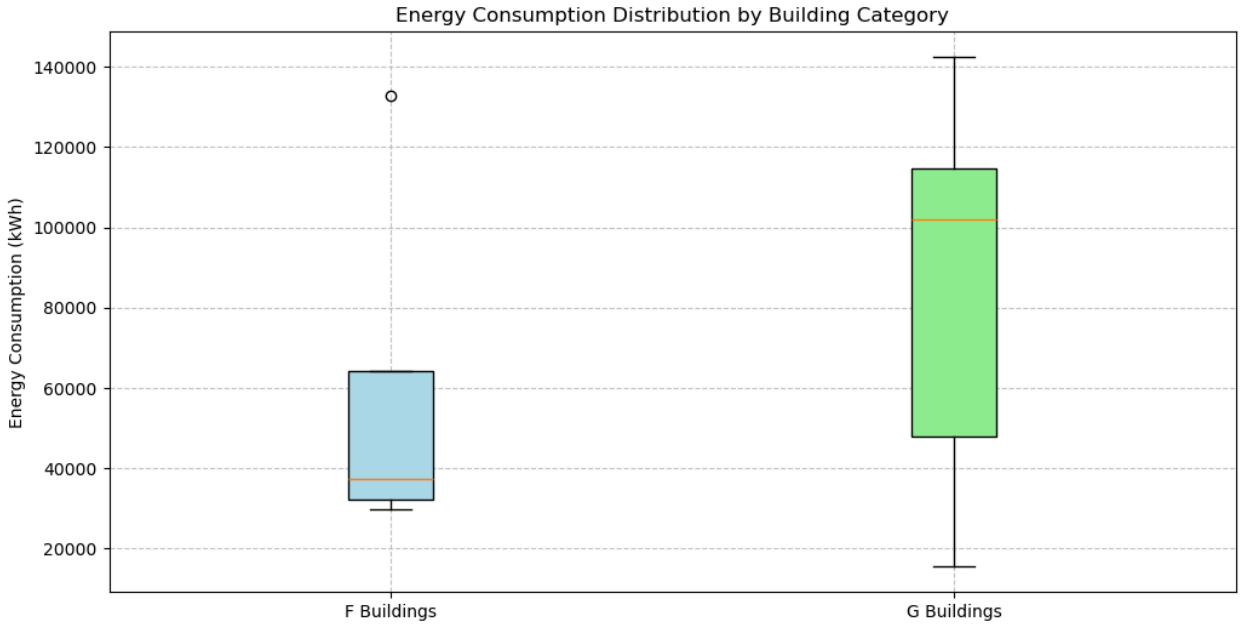
```
plt.ylabel('Total Energy Consumption (kWh)')
plt.title('Total Energy Consumption by Building Category')

# Add value labels on top of bars
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height,
             f'{height:,.0f}',
             ha='center', va='bottom')

plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



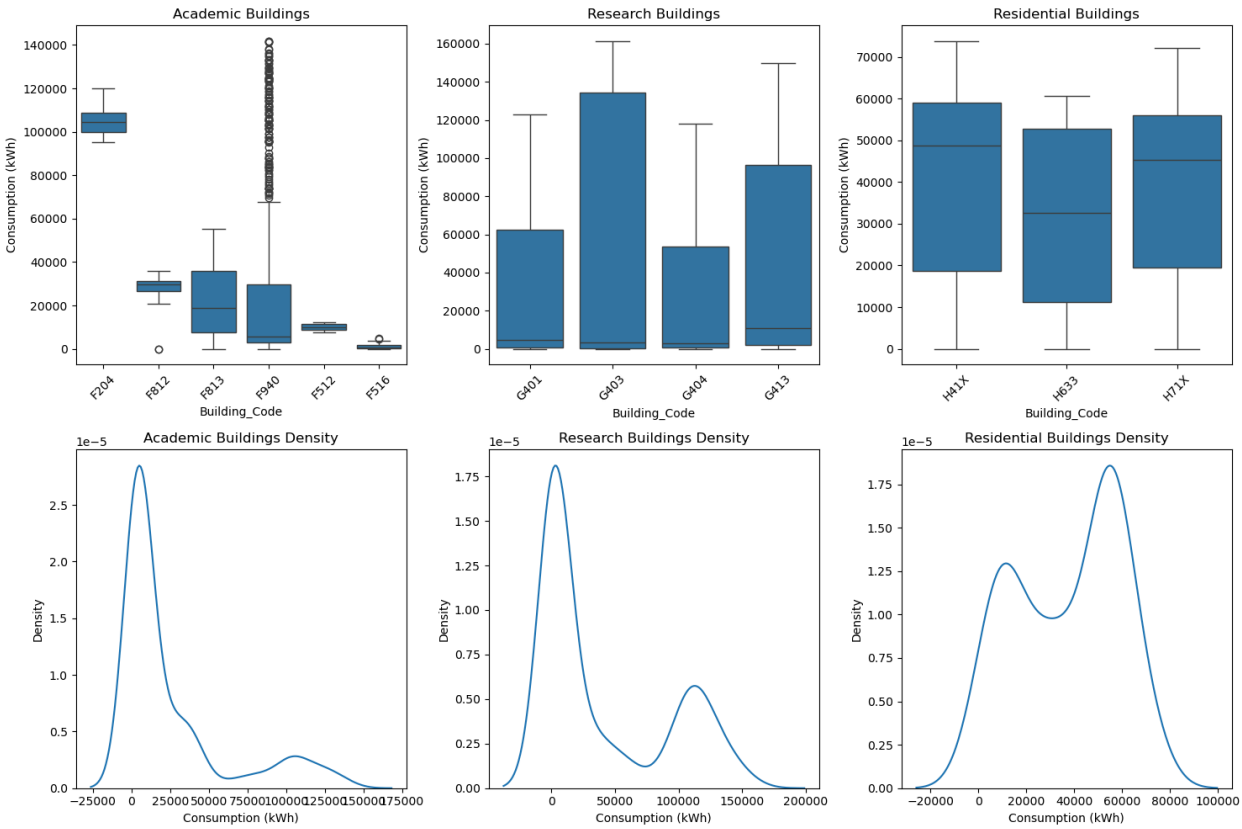
In [46]:

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

# Data preparation
data = {
    'Building': [
        'F812 UOCOE Owheo Building', 'F813 UOCOE Robertson Library', 'F940 Plaza Building',
        'F9XX College of Education main', 'G401 Mellor Laboratories', 'G403 Biochemistry',
        'G404 Microbiology', 'G413 Science 2', 'G608 St Margarets College', 'G60X UNICOL'
    ],
    'kWh': [
        29788.49, 33022.12, 132734.41, 41395.89, 105178.75, 142364.04,
        98978.64, 117647.74, 15498.58, 31056.27
    ]
}

df = pd.DataFrame(data)
df['Category'] = df['Building'].apply(lambda x: x[0])

# Create figure with box plot
plt.figure(figsize=(12, 6))

# Create box plot
box_plot = plt.boxplot([df[df['Category'] == cat]['kWh'] for cat in ['F', 'G']],
                       labels=['F Buildings', 'G Buildings'],
                       patch_artist=True)

# Customize box colors
colors = {'F': 'lightblue', 'G': 'lightgreen'}
for patch, color in zip(box_plot['boxes'], colors.values()):
    patch.set_facecolor(color)

plt.title('Energy Consumption Distribution by Building Category')
plt.ylabel('Energy Consumption (kWh)')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

```
C:\Users\sugan\AppData\Local\Temp\ipykernel_8904\1343443015.py:24: MatplotlibDeprecationWarning:

The 'labels' parameter of boxplot() has been renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be dropped
in 3.11.
```



In [44]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Read the data
df = pd.read_excel('Stream_Janitza_Data.xlsx')

# Melt the dataframe
df_melted = pd.melt(df, id_vars=['Meter_Location'],
                    var_name='Date',
                    value_name='Consumption')

# Extract building code from Meter_Location
df_melted['Building_Code'] = df_melted['Meter_Location'].str.extract(r'([A-Z]\d{3}|[A-Z]\d{2}X)')

# Define building groups
groups = {
    'Academic': ['F204', 'F512', 'F516', 'F812', 'F813', 'F940'],
    'Research': ['G401', 'G403', 'G404', 'G413'],
    'Residential': ['H41X', 'H633', 'H71X']
}

# Create figure
fig, axes = plt.subplots(2, 3, figsize=(15, 10))

# Plot for each group
for idx, (group_name, buildings) in enumerate(groups.items()):
    # Filter data for current group
    group_data = df_melted[df_melted['Building_Code'].isin(buildings)]

    # Box plot
    sns.boxplot(data=group_data, x='Building_Code', y='Consumption', ax=axes[0, idx])
    axes[0, idx].set_title(f'{group_name} Buildings')
    axes[0, idx].tick_params(axis='x', rotation=45)
    axes[0, idx].set_ylabel('Consumption (kWh)')

    # Density plot
    sns.kdeplot(data=group_data, x='Consumption', ax=axes[1, idx])
    axes[1, idx].set_title(f'{group_name} Buildings Density')
    axes[1, idx].set_xlabel('Consumption (kWh)')

plt.tight_layout()
plt.show()

# Print summary statistics for each group
for group_name, buildings in groups.items():
    group_data = df_melted[df_melted['Building_Code'].isin(buildings)]
    print(f"\n{group_name} Buildings Statistics:")
    print(f"Mean Consumption: {group_data['Consumption'].mean():.2f}")
    print(f"Median Consumption: {group_data['Consumption'].median():.2f}")
    print(f"Std Deviation: {group_data['Consumption'].std():.2f}")
```




```
Academic Buildings Statistics:
Mean Consumption: 24612.15
Median Consumption: 7078.89
Std Deviation: 35958.09

Research Buildings Statistics:
Mean Consumption: 40223.27
Median Consumption: 7546.44
Std Deviation: 50463.98

Residential Buildings Statistics:
Mean Consumption: 37188.52
Median Consumption: 42036.86
Std Deviation: 21971.21
```

In [3]:

```python
import pandas as pd
import numpy as np

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Calculate basic statistics
statistics = data.describe(percentiles=[0.25, 0.5, 0.75])

# Select only the numeric columns
numeric_columns = data.select_dtypes(include=[np.number]).columns

# Calculate additional statistics
additional_stats = pd.DataFrame({
    'Count': data[numeric_columns].count(),
    'Mean': data[numeric_columns].mean(),
    'Std Dev': data[numeric_columns].std(),
    'Min': data[numeric_columns].min(),
    '25%': statistics.loc['25%'],
    '50%': statistics.loc['50%'],
    '75%': statistics.loc['75%'],
    'Max': data[numeric_columns].max()
})

# Display the statistics
print(additional_stats.T)
```

	F204 444 Great King Street - kWh	F315 Botany Tin Hut - kWh \
Count	34.000000	34.000000
Mean	104984.115294	35180.897000
Std Dev	6268.750174	11238.201322
Min	95328.880000	13061.972000
25%	100041.962500	28075.883000
50%	104525.445000	35686.522000
75%	108787.095000	43253.159000
Max	119858.940000	53740.892000

	F325 Physical Education - kWh	F711 Executive Residence - kWh \
Count	34.000000	34.000000
Mean	34967.750559	10053.446118
Std Dev	13775.414136	8269.363494
Min	14269.856000	2385.090000
25%	21876.480000	3955.116000
50%	33722.024000	6402.114000
75%	47536.968000	16527.981000
Max	57357.600000	29937.318000

	F812 UOCOE Owheo Building - kWh	F813 UOCOE Robertson Library - kWh \
Count	34.000000	34.000000
Mean	29429.510000	38259.774706
Std Dev	3907.463887	5248.067608
Min	20844.600000	26873.480000
25%	27372.757500	35522.220000
50%	29713.740000	39622.065000
75%	31545.155000	42083.152500
Max	35880.810000	46953.250000

	F940 Plaza Building - kWh \
Count	34.000000
Mean	121209.342588
Std Dev	11951.230300
Min	95472.274000
25%	114178.711750
50%	123355.245000
75%	129360.228500
Max	141536.911000

	F9XX College of Education main (Boiler room) - kWh \
Count	34.000000
Mean	63808.682118
Std Dev	13475.951879
Min	41395.888000
25%	51221.812000
50%	65491.928000
75%	75340.688000
Max	88879.216000

	G401 Mellor Laboratories - kWh	G403 Biochemistry - kWh \
Count	34.000000	34.000000
Mean	113520.799941	137393.186824
Std Dev	5905.001152	9804.061099
Min	97322.720000	123717.024000
25%	109089.112000	131372.130000
50%	114046.608000	134675.724000
75%	118264.048000	142344.762000
Max	122692.288000	161403.528000

	G404 Microbiology - kWh	G413 Science 2 - kWh \
Count	34.000000	34.000000
Mean	105527.793647	118566.597647
Std Dev	6392.211757	11838.646743
Min	93505.920000	101231.704000
25%	100478.304000	109305.234000
50%	106147.092000	117134.760000
75%	108998.316000	123755.682000
Max	118099.872000	148776.216000

	G608 St Margarets College - kWh	G60X UNICOL - kWh \
Count	34.000000	34.000000
Mean	32617.197529	64788.225412
Std Dev	8391.687774	16843.159045
Min	12628.620000	31056.272000
25%	25992.573000	49984.092000
50%	35735.760000	70704.968000
75%	38539.654500	77734.248000
Max	42300.066000	84484.704000

	H41X Selwyn College - kWh E2	H633 Arana College main - kWh \
Count	34.000000	34.000000
Mean	42745.803176	34543.669647
Std Dev	22280.259141	19769.066869
Min	6726.402000	6576.912000
25%	20831.716500	11991.020000
50%	51861.000000	34058.912000
75%	59279.856000	53572.004000
Max	73830.822000	60545.264000

```
         H71X Studholm College - kWh E2
Count                        34.000000
Mean                      40838.764118
Std Dev                   19333.808156
Min                        9162.460000
25%                       21580.557500
50%                       46167.870000
75%                       56341.870000
Max                       72074.530000
```

In [19]:

```python
import pandas as pd
import numpy as np

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Define building groups
libraries = [
    "F813 UOCOE Robertson Library - kWh",
    "F813 Bill Robertson Library 1 MSB:",
    "F419 ISB West Excluding Shops",
    "F505 Richardson Library Block Rising Main:"
]

colleges = [
    "F711 Executive Residence - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G608 St Margarets College - kWh",
    "H41X Selwyn College - kWh E2",
    "H633 Arana College main - kWh",
    "H71X Studholm College - kWh E2"
]

science = [
    "F315 Botany Tin Hut - kWh",
    "F325 Physical Education - kWh",
    "F812 UOCOE Owheo Building - kWh",
    "G401 Mellor Laboratories - kWh",
    "G404 Microbiology - kWh",
    "G413 Science 2 - kWh",
    "G505 Geology north:",
    "G505 Geology south:"
]

humanities = [
    "F9XX College of Education main (Boiler room) - kWh",
    "F518 Arts 1 Submains MSB:",
    "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
    "F505 1 Richardson Mains:",
    "G506/07 Archway buildings (incl. Allen & Marama Hall):"
]

commerce = [
    "F614 1 School of Business Incomer 1 (Lower floors):",
    "F614 2 School of Business Incomer 2 (Upper floors):",
    "F618 1  Psychology substation- Goddard:"
]

its_servers = ["F204 444 Great King Street - kWh"]

# Function to calculate statistics for a group
def calculate_group_stats(data, group):
    group_data = data[group]
    stats = group_data.describe(percentiles=[0.25, 0.5, 0.75])
    return stats

# Calculate and print statistics for each group
groups = {
    "Libraries": libraries,
    "Colleges": colleges,
    "Science": science,
    "Humanities": humanities,
    "Commerce": commerce,
    "ITS Servers": its_servers
}

# Store results for each group
results = {}
for group_name, group_buildings in groups.items():
    results[group_name] = calculate_group_stats(data, group_buildings)

# Format the results into a matrix for each group
formatted_results = {}
for group_name, stats in results.items():
    formatted_results[group_name] = stats.T  # Transpose for better readability

# Display formatted results for each group
for group_name, stats in formatted_results.items():
    print(f"\n{group_name} Statistics:")
    print(stats)
```

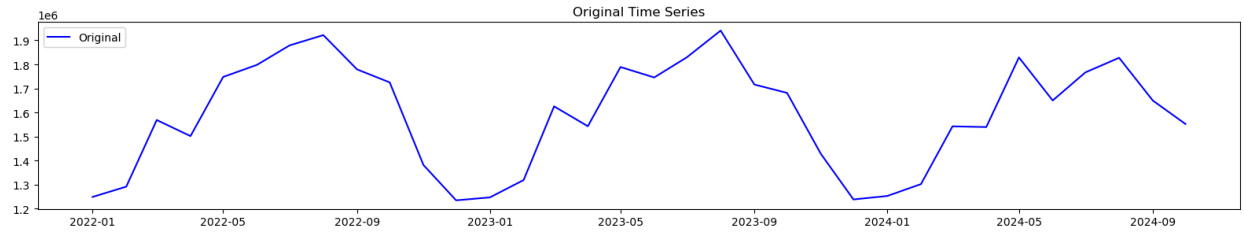
```
Libraries Statistics:
                                            count          mean          std  \
F813 UOCOE Robertson Library - kWh           34.0  38259.774706  5248.067608
F813 Bill Robertson Library 1 MSB:           34.0  37347.898235  7068.161567
F419 ISB West Excluding Shops                34.0  64383.837941  6461.866335
F505 Richardson Library Block Rising Main:   34.0  10948.797353  3601.776699

                                                 min         25%        50%  \
F813 UOCOE Robertson Library - kWh          26873.48  35522.2200  39622.065
F813 Bill Robertson Library 1 MSB:          23896.00  33454.0225  39765.395
F419 ISB West Excluding Shops               48941.78  60437.8600  64887.510
F505 Richardson Library Block Rising Main:    991.00   8150.0625  11784.720

                                                   75%       max
F813 UOCOE Robertson Library - kWh          42083.1525  46953.25
F813 Bill Robertson Library 1 MSB:          41478.7675  55406.17
F419 ISB West Excluding Shops               69313.9500  75562.00
F505 Richardson Library Block Rising Main:  13749.0000  15491.00

Colleges Statistics:
                                  count          mean           std        min  \
F711 Executive Residence - kWh     34.0  10053.446118   8269.363494   2385.090
F812 UOCOE Owheo Building - kWh    34.0  29429.510000   3907.463887  20844.600
G608 St Margarets College - kWh    34.0  32617.197529   8391.687774  12628.620
H41X Selwyn College - kWh E2       34.0  42745.803176  22280.259141   6726.402
H633 Arana College main - kWh      34.0  34543.669647  19769.066869   6576.912
H71X Studholm College - kWh E2     34.0  40838.764118  19333.808156   9162.460

                                        25%        50%         75%        max
F711 Executive Residence - kWh    3955.1160   6402.114  16527.9810  29937.318
F812 UOCOE Owheo Building - kWh  27372.7575  29713.740  31545.1550  35880.810
G608 St Margarets College - kWh  25992.5730  35735.760  38539.6545  42300.066
H41X Selwyn College - kWh E2     20831.7165  51861.000  59279.8560  73830.822
H633 Arana College main - kWh    11991.0200  34058.912  53572.0040  60545.264
H71X Studholm College - kWh E2   21580.5575  46167.870  56341.8700  72074.530

Science Statistics:
                                  count           mean           std  \
F315 Botany Tin Hut - kWh          34.0   35180.897000  11238.201322
F325 Physical Education - kWh      34.0   34967.750559  13775.414136
F812 UOCOE Owheo Building - kWh    34.0   29429.510000   3907.463887
G401 Mellor Laboratories - kWh     34.0  113520.799941   5905.001152
G404 Microbiology - kWh            34.0  105527.793647   6392.211757
G413 Science 2 - kWh               34.0  118566.597647  11838.646743
G505 Geology north:                34.0   35194.934416  12095.704919
G505 Geology south:                34.0   11361.495764   5013.986694

                                        min          25%         50%  \
F315 Botany Tin Hut - kWh          13061.972   28075.8830   35686.522
F325 Physical Education - kWh      14269.856   21876.4800   33722.024
F812 UOCOE Owheo Building - kWh    20844.600   27372.7575   29713.740
G401 Mellor Laboratories - kWh     97322.720  109089.1120  114046.608
G404 Microbiology - kWh            93505.920  100478.3040  106147.092
G413 Science 2 - kWh              101231.704  109305.2340  117134.760
G505 Geology north:                10240.510   23529.4200   36542.975
G505 Geology south:                 3243.260    7509.8800    9942.240

                                         75%         max
F315 Botany Tin Hut - kWh          43253.1590   53740.892
F325 Physical Education - kWh      47536.9680   57357.600
F812 UOCOE Owheo Building - kWh    31545.1550   35880.810
G401 Mellor Laboratories - kWh    118264.0480  122692.288
G404 Microbiology - kWh           108998.3160  118099.872
G413 Science 2 - kWh              123755.6820  148776.216
G505 Geology north:                46803.6075   52997.000
G505 Geology south:                15627.1050   20874.000

Humanities Statistics:
                                                     count          mean  \
F9XX College of Education main (Boiler room) - kWh   34.0  63808.682118
F518 Arts 1 Submains MSB:                            34.0  23527.792647
F516 97 Albany & F517 99 Albany, F513 262 Leith...   34.0   2099.598529
F505 1 Richardson Mains:                             34.0  62012.273529
G506/07 Archway buildings (incl. Allen & Marama...   34.0  21724.690000

                                                          std        min  \
F9XX College of Education main (Boiler room) - kWh  13475.951879  41395.888
F518 Arts 1 Submains MSB:                            4737.794359  14374.000
F516 97 Albany & F517 99 Albany, F513 262 Leith...   1202.747861    674.800
F505 1 Richardson Mains:                             9793.195525  42196.000
G506/07 Archway buildings (incl. Allen & Marama...  10455.484540   5636.610

                                                         25%        50%  \
F9XX College of Education main (Boiler room) - kWh  51221.8120  65491.928
F518 Arts 1 Submains MSB:                           18703.8050  24507.500
F516 97 Albany & F517 99 Albany, F513 262 Leith...    956.6675   1896.470
F505 1 Richardson Mains:                            56100.7600  65311.485
G506/07 Archway buildings (incl. Allen & Marama...  12577.5325  21597.185

                                                         75%        max
```

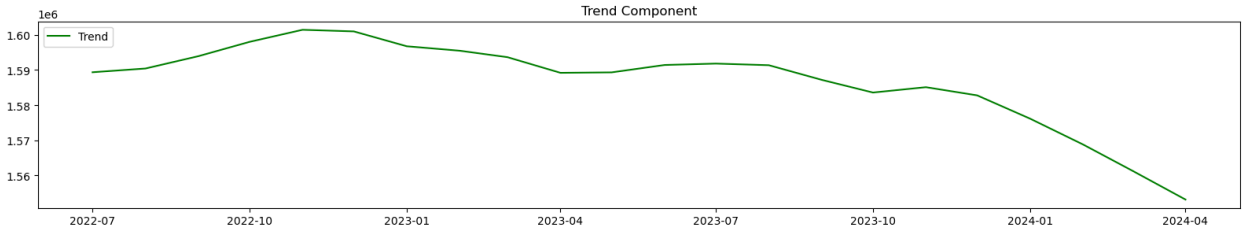
```
F9XX College of Education main (Boiler room) - kWh  75340.6880  88879.216
F518 Arts 1 Submains MSB:                           27440.1275  29567.900
F516 97 Albany & F517 99 Albany, F513 262 Leith...   2910.7275   4973.250
F505 1 Richardson Mains:                            70122.5000  74567.680
G506/07 Archway buildings (incl. Allen & Marama...  30559.4250  39190.020

Commerce Statistics:
                                                    count          mean  \
F614 1 School of Business Incomer 1 (Lower floo...   34.0  62236.191471
F614 2 School of Business Incomer 2 (Upper floo...   34.0  40916.118824
F618 1  Psychology substation- Goddard:              34.0  90395.376471

                                                          std       min  \
F614 1 School of Business Incomer 1 (Lower floo...  11420.561191  43467.27
F614 2 School of Business Incomer 2 (Upper floo...  18139.801865  14693.00
F618 1  Psychology substation- Goddard:             18838.860936  44891.00

                                                         25%        50%  \
F614 1 School of Business Incomer 1 (Lower floo...  53416.0625  63783.170
F614 2 School of Business Incomer 2 (Upper floo...  25007.4875  40988.055
F618 1  Psychology substation- Goddard:             80573.9550  95076.530

                                                         75%        max
F614 1 School of Business Incomer 1 (Lower floo...   70357.760   86793.22
F614 2 School of Business Incomer 2 (Upper floo...   54725.115   74491.91
F618 1  Psychology substation- Goddard:             101804.035  120805.38

ITS Servers Statistics:
                                   count           mean          std       min  \
F204 444 Great King Street - kWh    34.0  104984.115294  6268.750174  95328.88

                                          25%         50%         75%  \
F204 444 Great King Street - kWh  100041.9625  104525.445  108787.095

                                       max
F204 444 Great King Street - kWh  119858.94
```

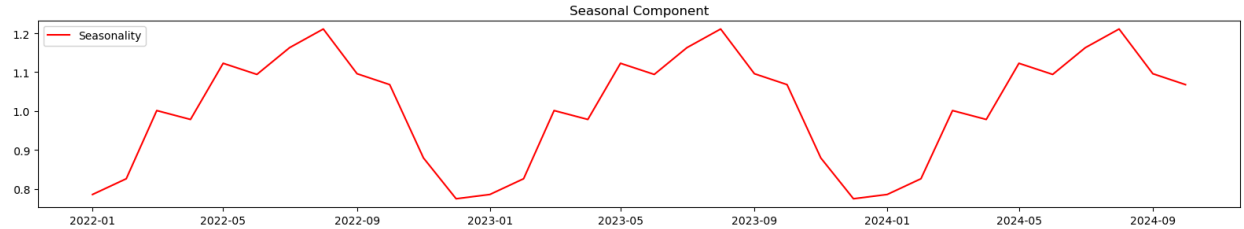
In [236…

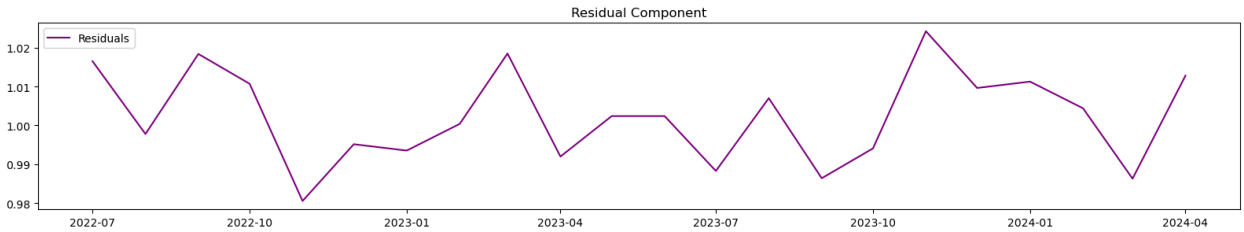
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal_decompose

# Load the dataset
df = pd.read_csv('stream_data_filtered - Copy.csv')

# Preprocessing
df['Month'] = pd.to_datetime(df['Month'], format='%b-%y')
df.set_index('Month', inplace=True)
df.sort_index(inplace=True)

# Ensure 'Total kwh' is numeric
df['Total kwh'] = pd.to_numeric(df['Total kwh'], errors='coerce')

# Handle missing values using linear interpolation
df['Total kwh'] = df['Total kwh'].interpolate(method='linear')

# Time Series Decomposition
# Using a multiplicative model
decomposition = seasonal_decompose(df['Total kwh'], model='multiplicative', period=12)

# Extract components
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

# Plotting the decomposition
plt.figure(figsize=(16, 12))

plt.subplot(411)
plt.plot(df['Total kwh'], label='Original', color='blue')
plt.legend(loc='upper left')
plt.title('Original Time Series')

plt.subplot(412)
plt.plot(trend, label='Trend', color='green')
plt.legend(loc='upper left')
plt.title('Trend Component')

plt.subplot(413)
plt.plot(seasonal, label='Seasonality', color='red')
plt.legend(loc='upper left')
plt.title('Seasonal Component')

plt.subplot(414)
plt.plot(residual, label='Residuals', color='purple')
plt.legend(loc='upper left')
plt.title('Residual Component')

plt.tight_layout()
plt.show()
```

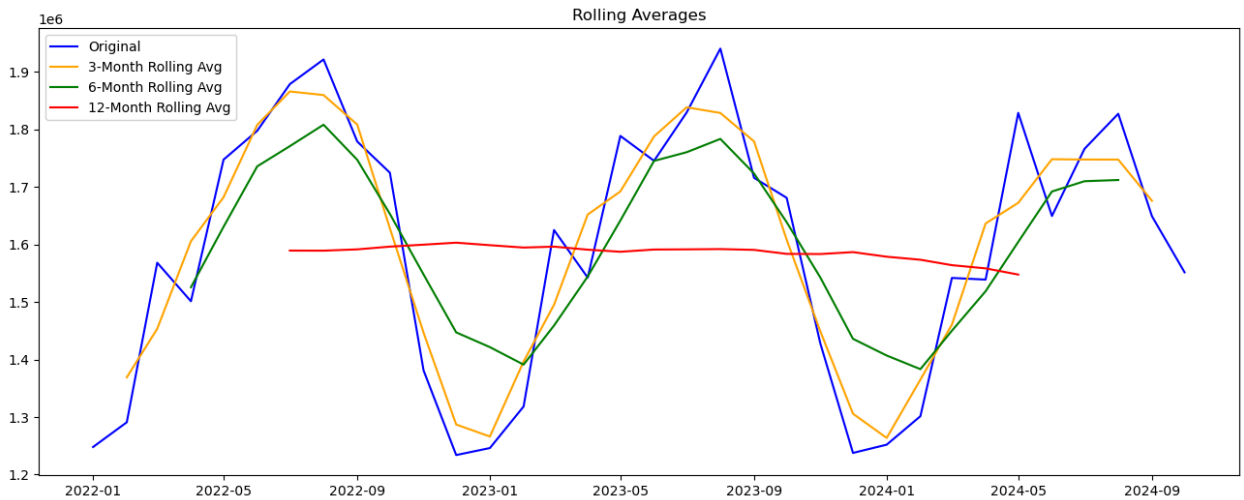
```
# Rolling Statistics
rolling_avg_3m = df['Total kwh'].rolling(window=3, center=True).mean()
rolling_avg_6m = df['Total kwh'].rolling(window=6, center=True).mean()
rolling_avg_12m = df['Total kwh'].rolling(window=12, center=True).mean()

# Plotting Rolling Statistics
plt.figure(figsize=(16, 6))
plt.plot(df['Total kwh'], label='Original', color='blue')
plt.plot(rolling_avg_3m, label='3-Month Rolling Avg', color='orange')
plt.plot(rolling_avg_6m, label='6-Month Rolling Avg', color='green')
plt.plot(rolling_avg_12m, label='12-Month Rolling Avg', color='red')
plt.legend(loc='upper left')
plt.title('Rolling Averages')
plt.show()

# Identifying Seasonal Peaks and Dips
seasonal_amplitude = seasonal.max() - seasonal.min()
seasonal_peaks = seasonal[seasonal > seasonal.mean()]
seasonal_dips = seasonal[seasonal < seasonal.mean()]

print("Seasonal Amplitude:", seasonal_amplitude)
print("\nSeasonal Peaks:\n", seasonal_peaks)
print("\nSeasonal Dips:\n", seasonal_dips)
```









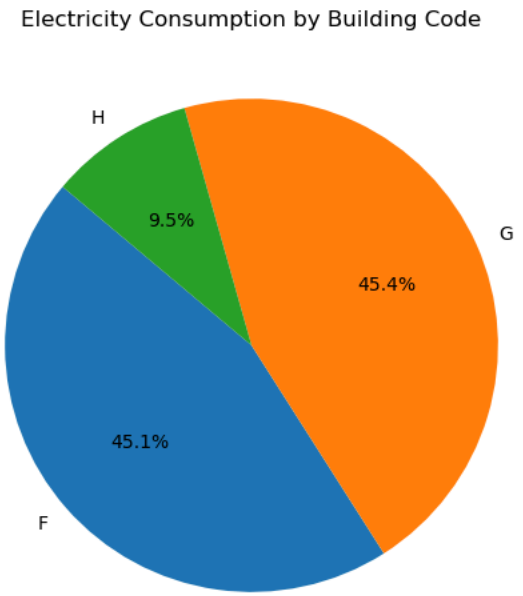


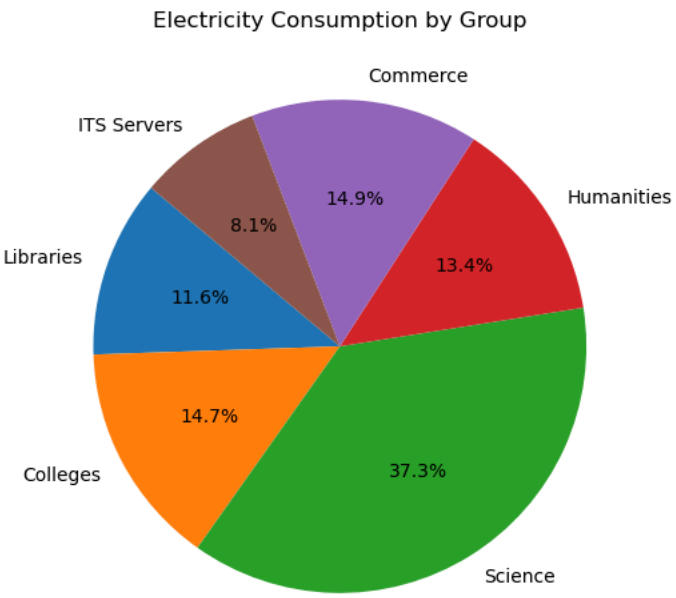
```
Seasonal Amplitude: 0.4364746965124353

Seasonal Peaks:
 Month
2022-05-01    1.122831
2022-06-01    1.094196
2022-07-01    1.162975
2022-08-01    1.210973
2022-09-01    1.095989
2022-10-01    1.067949
2023-05-01    1.122831
2023-06-01    1.094196
2023-07-01    1.162975
2023-08-01    1.210973
2023-09-01    1.095989
2023-10-01    1.067949
2024-05-01    1.122831
2024-06-01    1.094196
2024-07-01    1.162975
2024-08-01    1.210973
2024-09-01    1.095989
2024-10-01    1.067949
Name: seasonal, dtype: float64

Seasonal Dips:
 Month
2022-01-01    0.785498
2022-02-01    0.825974
2022-03-01    1.001283
2022-04-01    0.978392
2022-11-01    0.879441
2022-12-01    0.774498
2023-01-01    0.785498
2023-02-01    0.825974
2023-03-01    1.001283
2023-04-01    0.978392
2023-11-01    0.879441
2023-12-01    0.774498
2024-01-01    0.785498
2024-02-01    0.825974
2024-03-01    1.001283
2024-04-01    0.978392
Name: seasonal, dtype: float64
```

In [248…

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Sum total consumption by building codes (F, G, H)
building_codes = {
    'F': data.filter(like='F').sum().sum(),
    'G': data.filter(like='G').sum().sum(),
    'H': data.filter(like='H').sum().sum()
}

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
                   "F518 Arts 1 Submains MSB:",
                   "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                   "F505 1 Richardson Mains:",
                   "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Sum total consumption by group
group_totals = {group: data[buildings].sum().sum() for group, buildings in building_groups.items()}

# Plot Pie Chart for Building Codes
plt.figure(figsize=(10, 6))
plt.pie(building_codes.values(), labels=building_codes.keys(), autopct='%1.1f%%', startangle=140)
plt.title('Electricity Consumption by Building Code')
plt.show()


# Plot Pie Chart for Groups
```

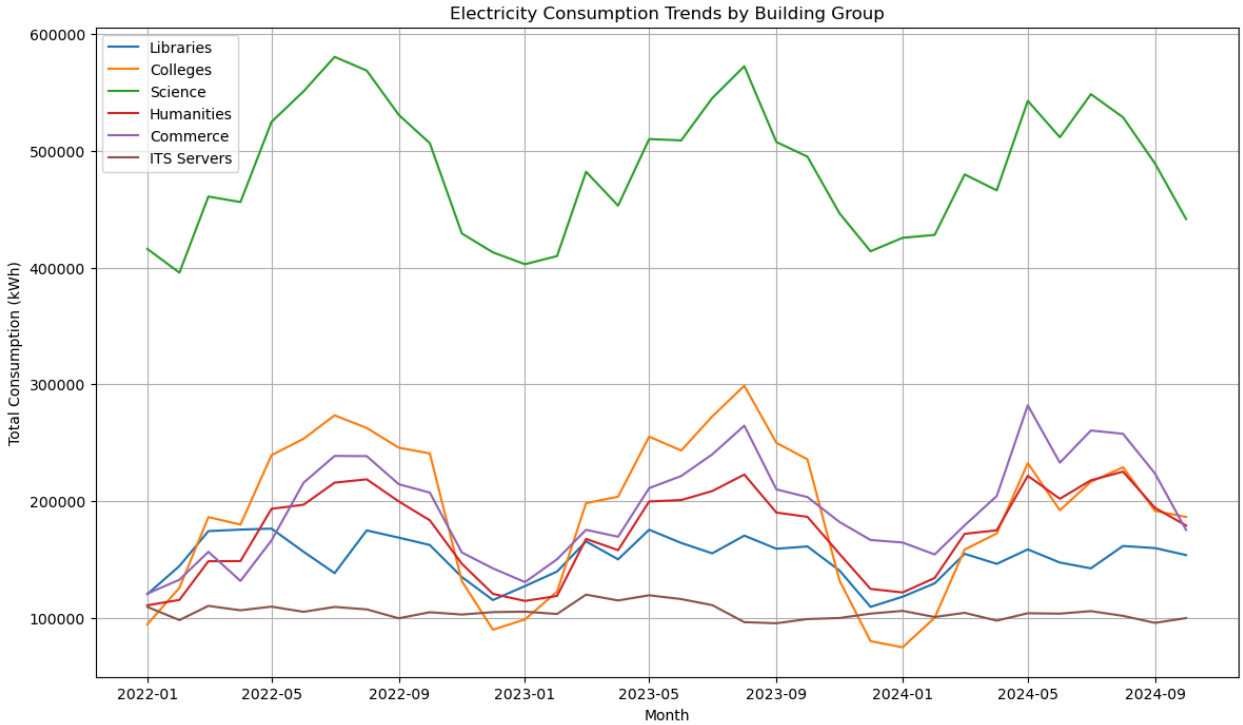
```
plt.figure(figsize=(10, 6))
plt.pie(group_totals.values(), labels=group_totals.keys(), autopct='%1.1f%%', startangle=140)
plt.title('Electricity Consumption by Group')
plt.show()
```





In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert the 'Month' column to datetime format
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')

# Set 'Month' as the index
data.set_index('Month', inplace=True)

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
                   "F518 Arts 1 Submains MSB:",
```

```
                    "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                    "F505 1 Richardson Mains:",
                    "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Calculate total consumption for each group
group_totals = {}
for group, buildings in building_groups.items():
    group_totals[group] = data[buildings].sum(axis=1)

# Create a DataFrame for group totals
group_totals_df = pd.DataFrame(group_totals)

# Plot the consumption trends for each group
plt.figure(figsize=(14, 8))

for group in group_totals_df.columns:
    plt.plot(group_totals_df.index, group_totals_df[group], label=group)

# Add labels, title, and legend
plt.xlabel('Month')
plt.ylabel('Total Consumption (kWh)')
plt.title('Electricity Consumption Trends by Building Group')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```

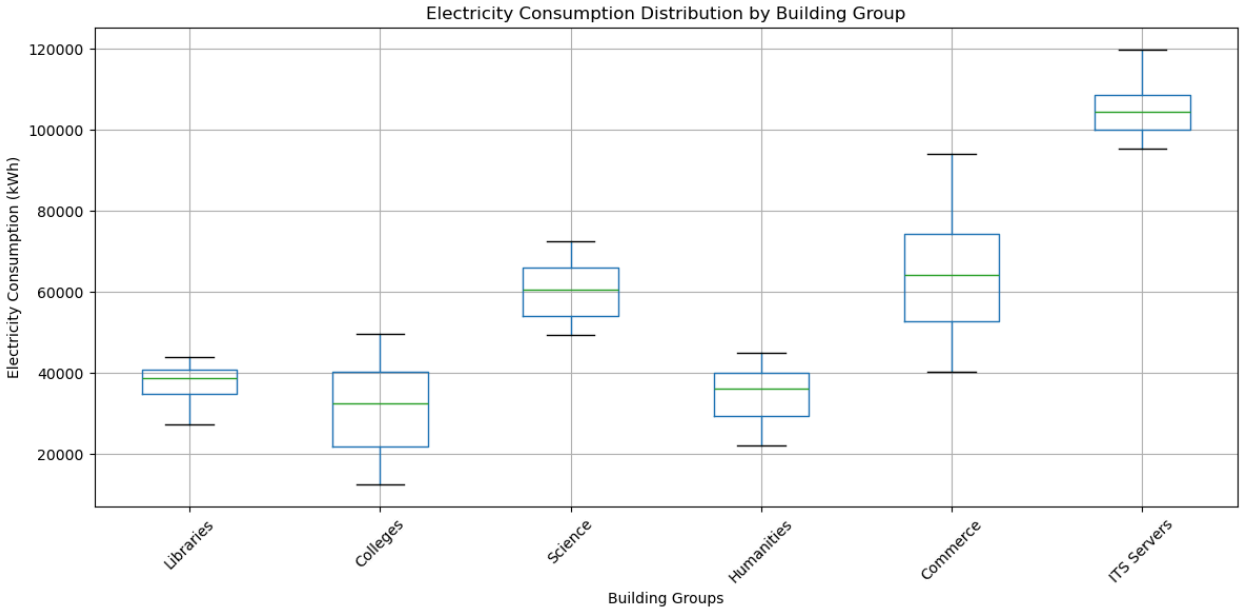


In [ ]:

In [27]:

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

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert the 'Month' column to datetime format
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')

# Set 'Month' as the index
data.set_index('Month', inplace=True)

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
```

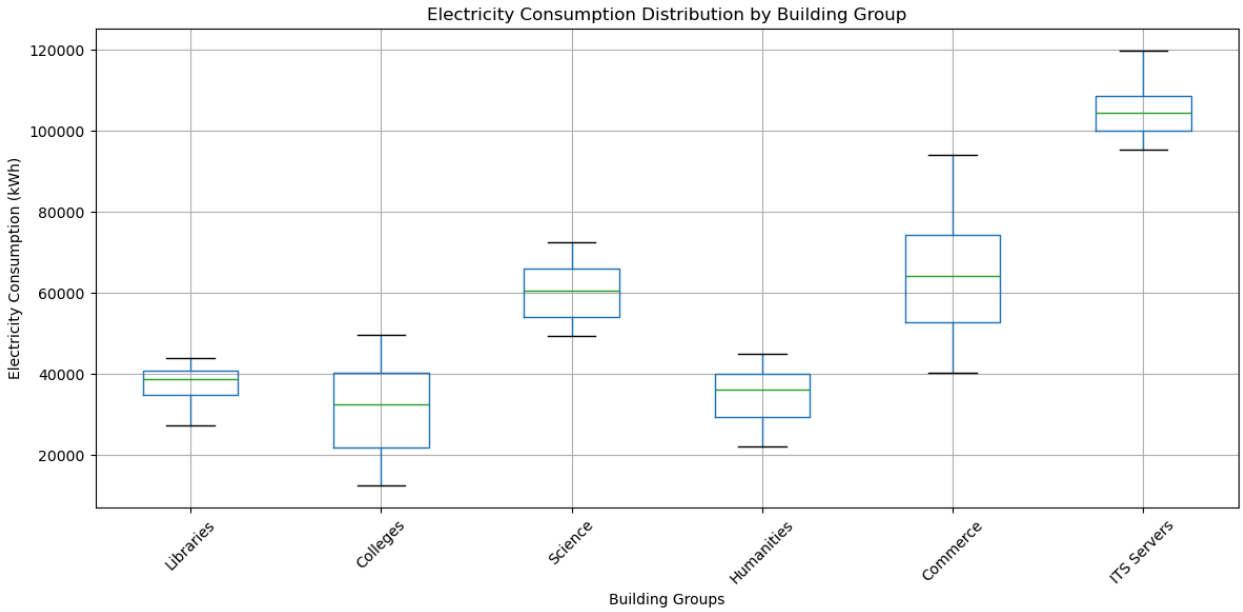
```
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
                   "F518 Arts 1 Submains MSB:",
                   "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                   "F505 1 Richardson Mains:",
                   "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Calculate average consumption for each group
group_averages = {}
for group, buildings in building_groups.items():
    group_averages[group] = data[buildings].mean(axis=1)

# Create a DataFrame for group averages
group_averages_df = pd.DataFrame(group_averages)

# Generate a boxplot for average consumption across groups
plt.figure(figsize=(12, 6))
group_averages_df.boxplot()
plt.title('Electricity Consumption Distribution by Building Group')
plt.ylabel('Electricity Consumption (kWh)')
plt.xlabel('Building Groups')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



In [31]:

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

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert the 'Month' column to datetime format
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')

# Set 'Month' as the index
data.set_index('Month', inplace=True)

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
                   "F518 Arts 1 Submains MSB:",
                   "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                   "F505 1 Richardson Mains:",
                   "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
```

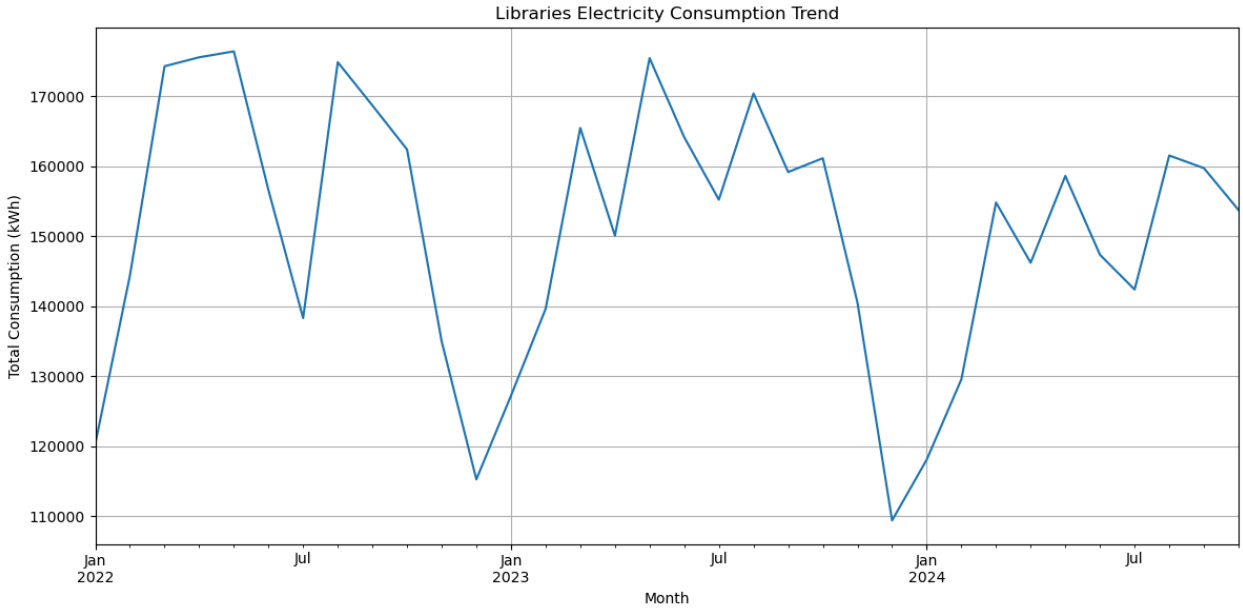
```
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Calculate average consumption for each group
group_averages = {}
for group, buildings in building_groups.items():
    group_averages[group] = data[buildings].mean(axis=1)

# Create a DataFrame for group averages
group_averages_df = pd.DataFrame(group_averages)

# Generate a boxplot for average consumption across groups
plt.figure(figsize=(12, 6))
group_averages_df.boxplot()
plt.title('Electricity Consumption Distribution by Building Group')
plt.ylabel('Electricity Consumption (kWh)')
plt.xlabel('Building Groups')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```

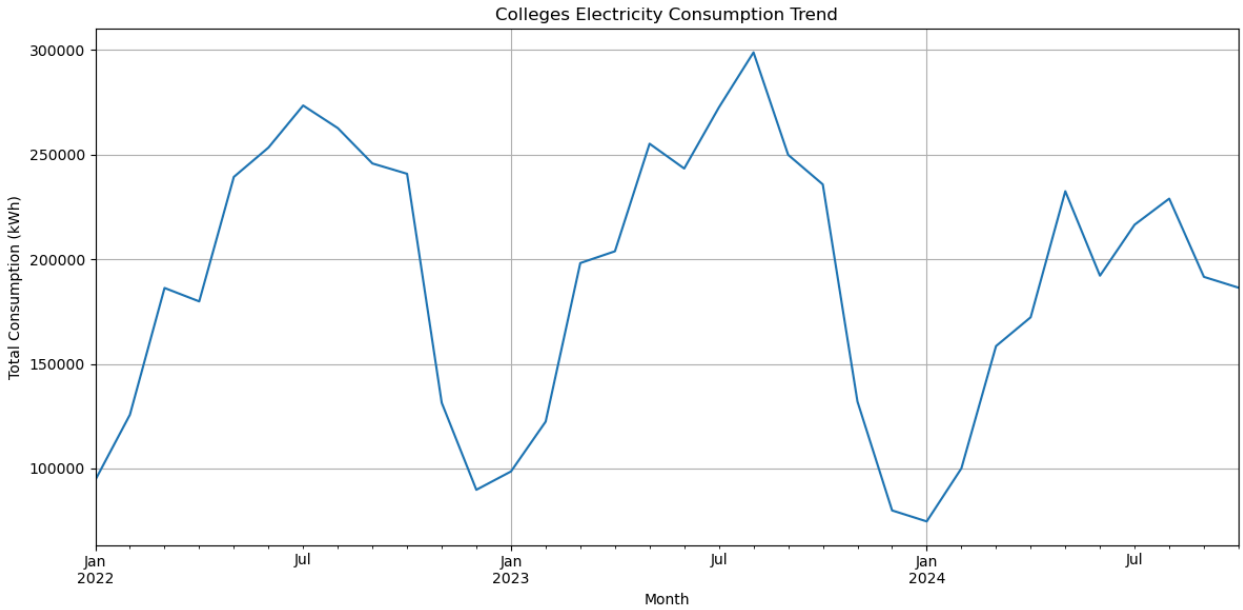


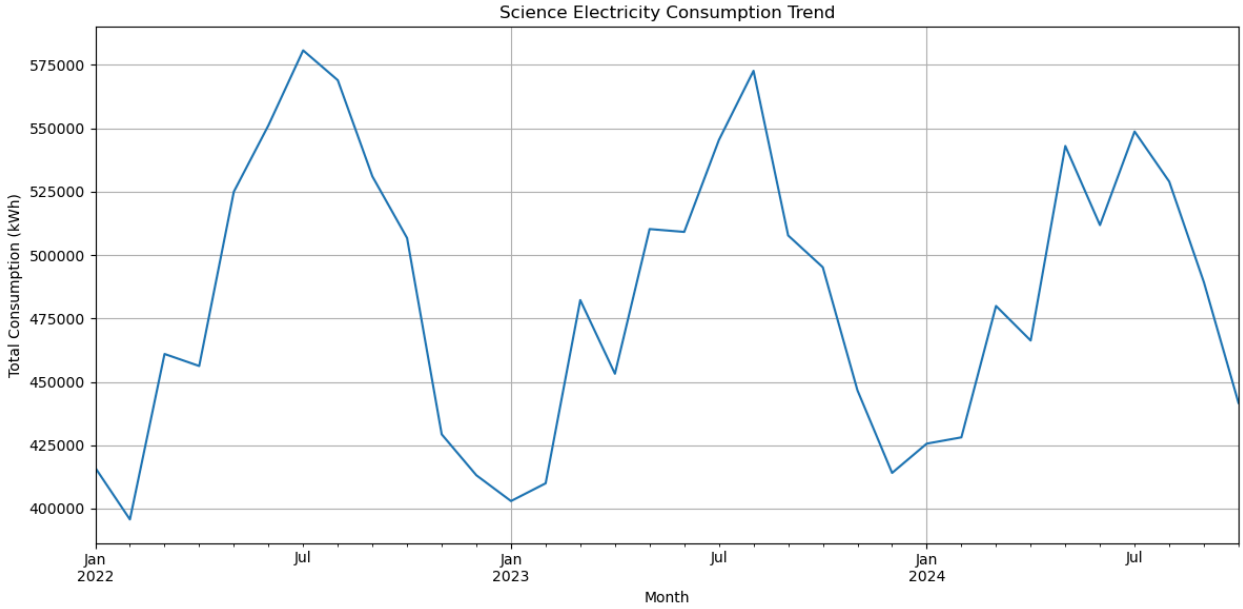
In [35]:

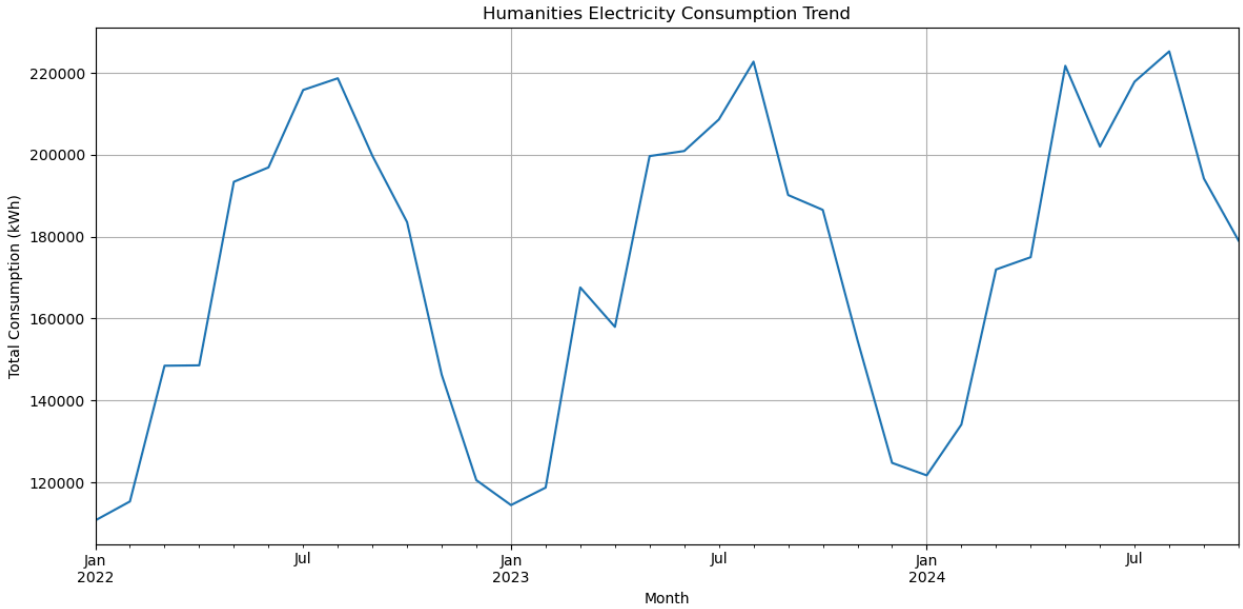
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert the 'Month' column to datetime format
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')

# Set 'Month' as the index
data.set_index('Month', inplace=True)

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
                   "F518 Arts 1 Submains MSB:",
                   "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                   "F505 1 Richardson Mains:",
                   "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Plot consumption trend for each group separately
for group_name, buildings in building_groups.items():
```

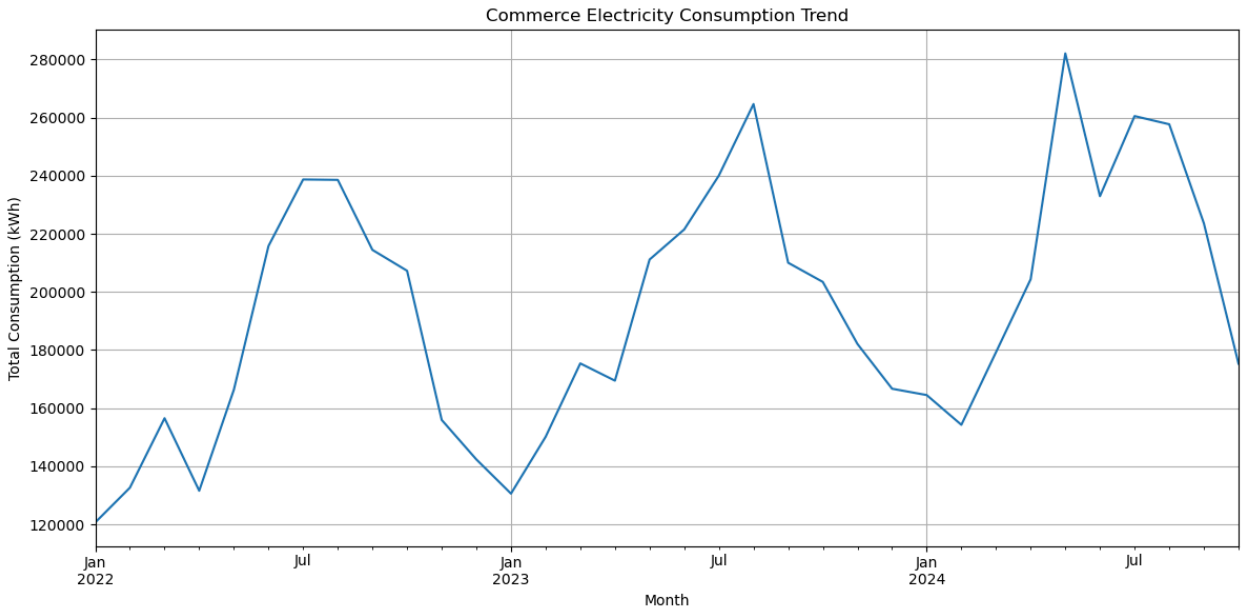
```
    plt.figure(figsize=(12, 6))
    data[buildings].sum(axis=1).plot()
    plt.title(f'{group_name} Electricity Consumption Trend')
    plt.ylabel('Total Consumption (kWh)')
    plt.xlabel('Month')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

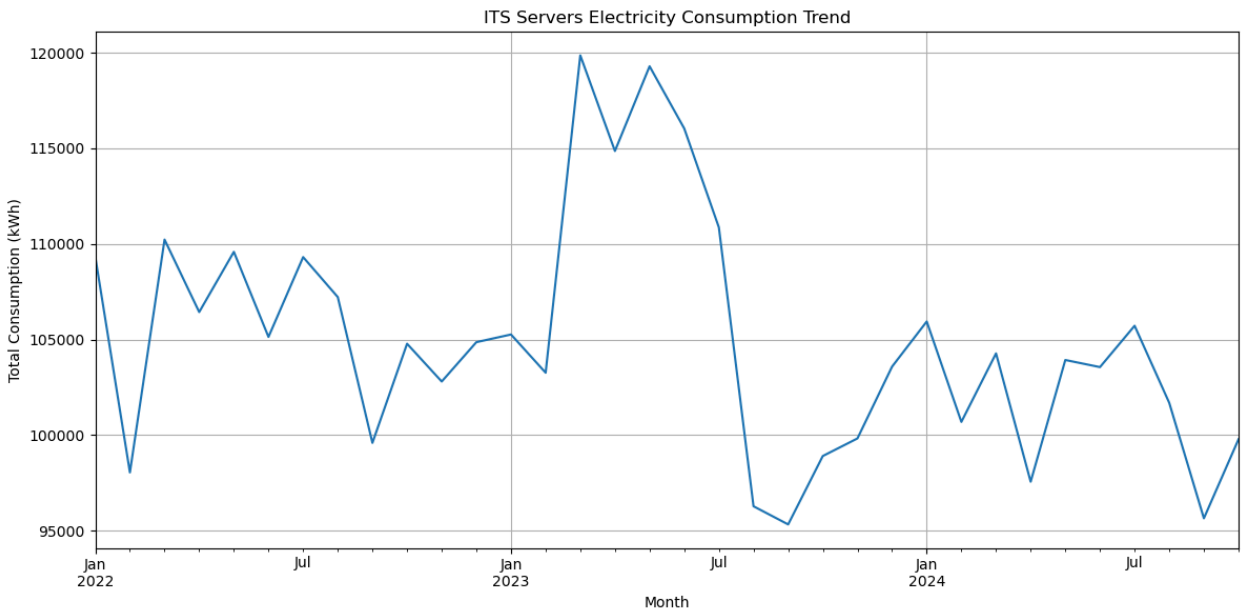





Science Electricity Consumption Trend
Total Consumption (kWh)
575000
550000
525000
500000
475000
450000
425000
400000
Jan 2022
Jul
Jan 2023
Jul
Jan 2024
Jul
Month

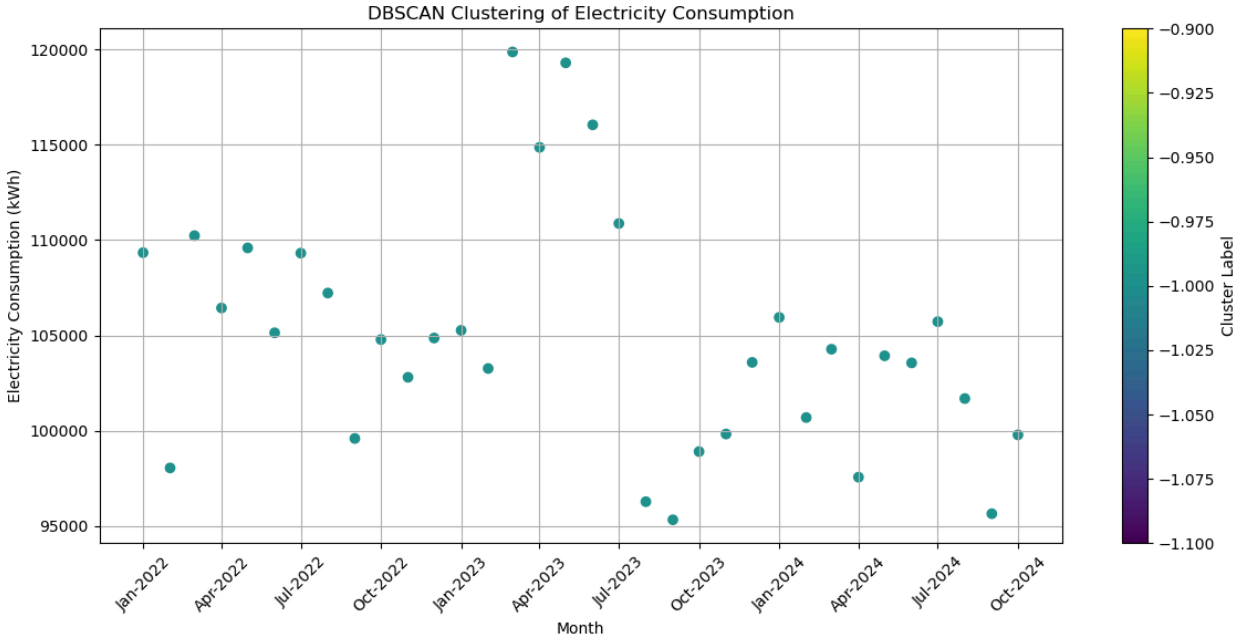

Humanities Electricity Consumption Trend
Total Consumption (kWh)
220000
200000
180000
160000
140000
120000
Jan 2022
Jul
Jan 2023
Jul
Jan 2024
Jul
Month


Commerce Electricity Consumption Trend
Total Consumption (kWh)
280000
260000
240000
220000
200000
180000
160000
140000
120000
Jan 2022
Jul
Jan 2023
Jul
Jan 2024
Jul
Month



In [45]:
```
from sklearn.cluster import DBSCAN
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert the 'Month' column to datetime and set it as the index
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')
data.set_index('Month', inplace=True)

# Select numeric columns (electricity consumption data)
consumption_data = data.select_dtypes(include=[np.number])

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(consumption_data)

# Apply DBSCAN
dbscan = DBSCAN(eps=1.5, min_samples=5)
labels = dbscan.fit_predict(scaled_data)

# Add cluster labels to the dataset
data['Cluster'] = labels

# Analyze clusters
print(data['Cluster'].value_counts())

# Visualize clusters with improved x-axis readability
plt.figure(figsize=(12, 6))
plt.scatter(data.index, data['F204 444 Great King Street - kWh'], c=data['Cluster'], cmap='viridis', label='Clusters')
plt.colorbar(label='Cluster Label')
plt.title('DBSCAN Clustering of Electricity Consumption')
plt.xlabel('Month')
plt.ylabel('Electricity Consumption (kWh)')

# Improve x-axis readability by formatting ticks
plt.xticks(ticks=data.index[::3], labels=data.index[::3].strftime('%b-%Y'), rotation=45)  # Show every 3rd month with rotation

plt.grid(True)
plt.tight_layout()
plt.show()
```

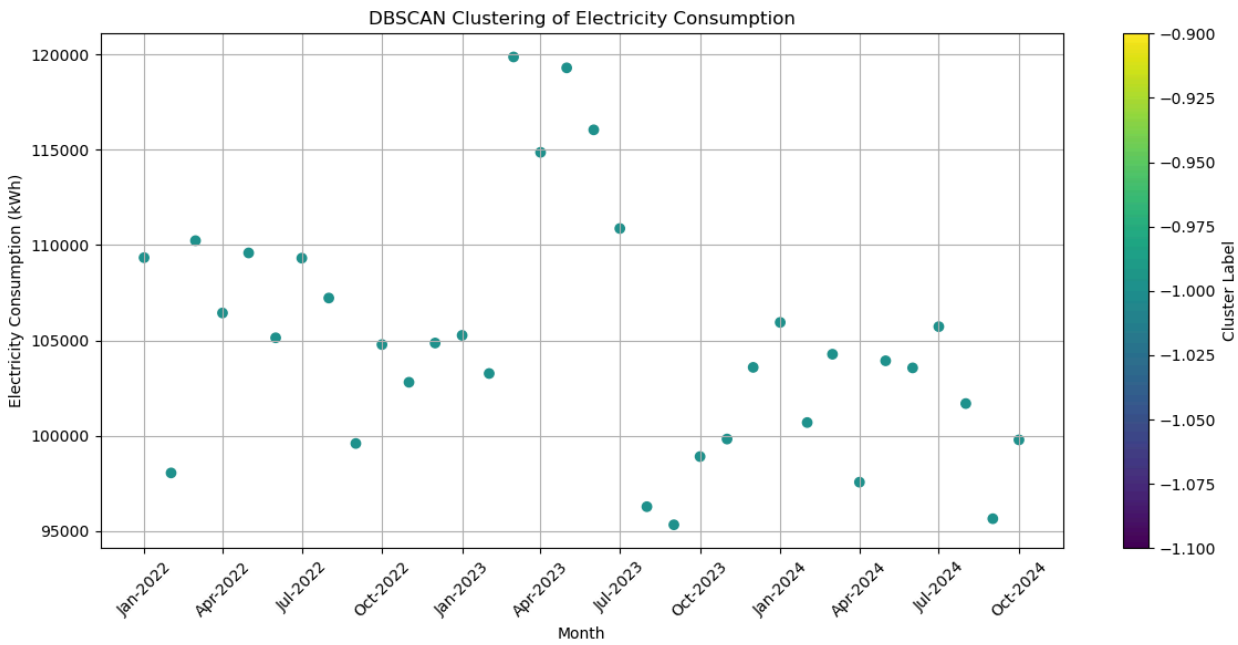
```
Cluster
-1    34
Name: count, dtype: int64
```



In [47]:

```
from sklearn.cluster import DBSCAN
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert the 'Month' column to datetime and set it as the index
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')
data.set_index('Month', inplace=True)

# Select numeric columns (electricity consumption data)
consumption_data = data.select_dtypes(include=[np.number])

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(consumption_data)

# Apply DBSCAN
dbscan = DBSCAN(eps=1.5, min_samples=5)
labels = dbscan.fit_predict(scaled_data)

# Add cluster labels to the dataset
data['Cluster'] = labels

# Analyze clusters
print(data['Cluster'].value_counts())

# Visualize clusters with improved x-axis readability
plt.figure(figsize=(12, 6))
plt.scatter(data.index, data['F204 444 Great King Street - kWh'], c=data['Cluster'], cmap='viridis', label='Clusters')
plt.colorbar(label='Cluster Label')
plt.title('DBSCAN Clustering of Electricity Consumption')
plt.xlabel('Month')
plt.ylabel('Electricity Consumption (kWh)')

# Improve x-axis readability by formatting ticks
plt.xticks(ticks=data.index[::3], labels=data.index[::3].strftime('%b-%Y'), rotation=45)  # Show every 3rd month with rotation

plt.grid(True)
plt.tight_layout()
plt.show()
```

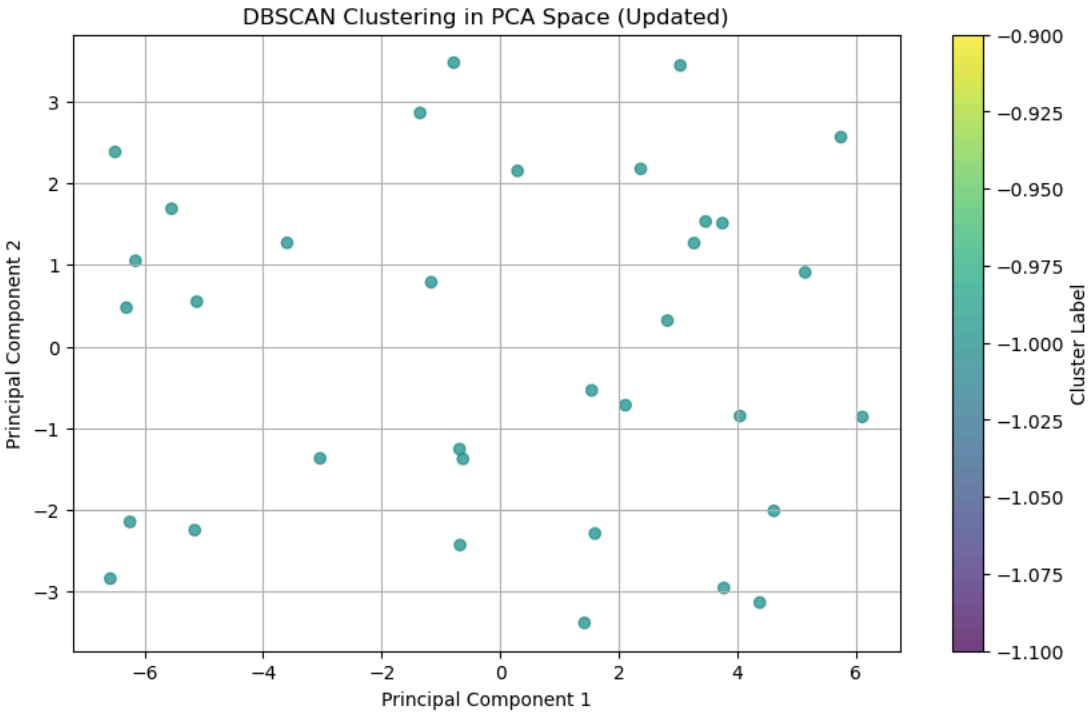
```
Cluster
-1    34
Name: count, dtype: int64
```



In [264…

```
# Adjust DBSCAN parameters
dbscan = DBSCAN(eps=2.5, min_samples=3)  # Increased eps for better clustering
labels = dbscan.fit_predict(scaled_data)

# Update cluster labels in the dataset
data['Cluster'] = labels

# Display cluster counts
print("Updated Cluster Distribution:")
print(data['Cluster'].value_counts())

# PCA for visualization
pca = PCA(n_components=2)
pca_transformed = pca.fit_transform(scaled_data)

# Scatter plot with fixed marker and clustering
plt.figure(figsize=(10, 6))
plt.scatter(pca_transformed[:, 0], pca_transformed[:, 1], c=data['Cluster'], cmap='viridis', alpha=0.75)
plt.colorbar(label='Cluster Label')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('DBSCAN Clustering in PCA Space (Updated)')
plt.grid(True)
plt.show()
```

```
Updated Cluster Distribution:
Cluster
-1    34
Name: count, dtype: int64
```

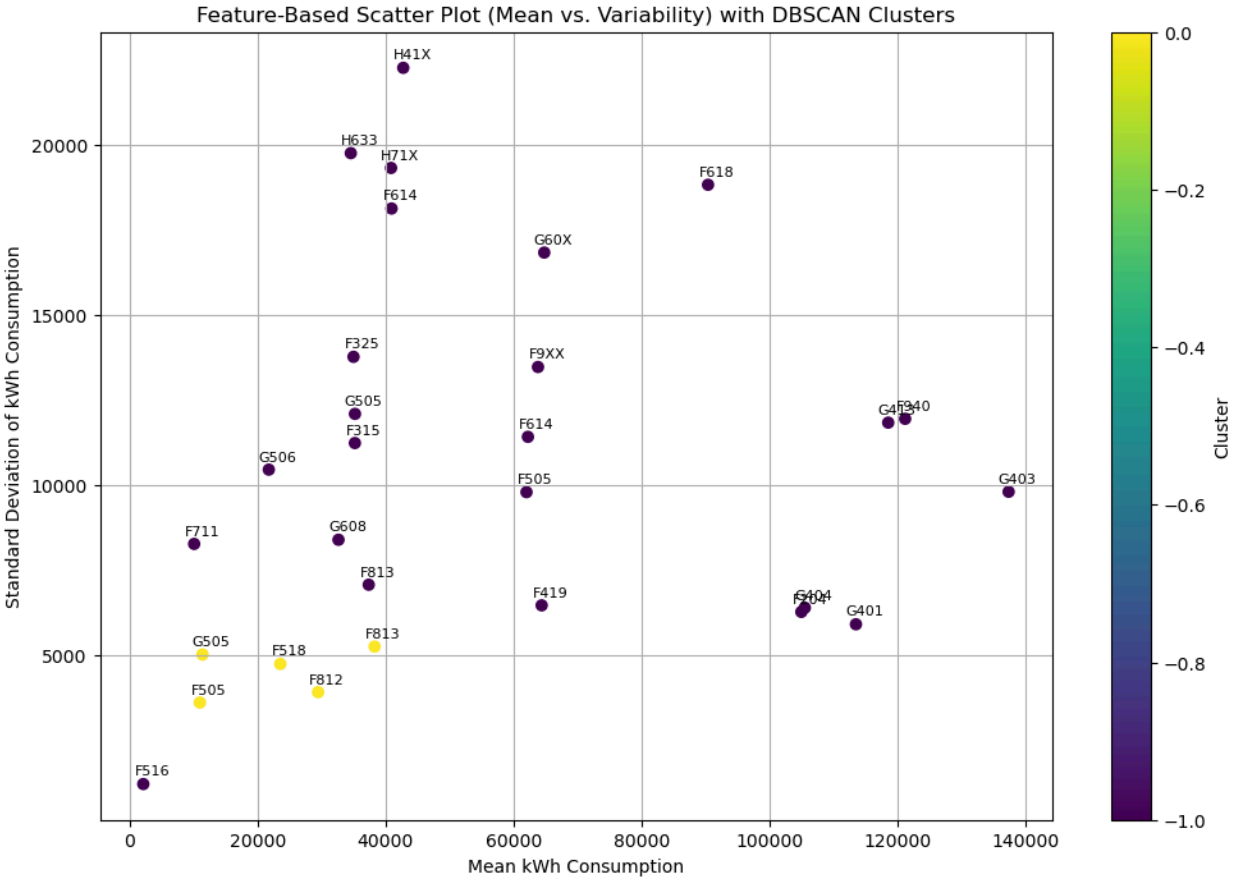


In [268…

```
import pandas as pd
import numpy as np
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

# Load the dataset
df = pd.read_csv("stream_data_filtered - Copy.csv")

# Data Cleaning and Preprocessing
# Handle missing values by filling with the mean of each column
df = df.fillna(df.mean(numeric_only=True))

# Identify and exclude non-numeric columns
non_numeric_cols = df.select_dtypes(exclude=['number']).columns
df_numeric = df.drop(columns=non_numeric_cols)

# Remove the 'Total kwh' column
if 'Total kwh' in df_numeric.columns:
    df_numeric = df_numeric.drop(columns=['Total kwh'])

# Calculate Mean and Variability (Standard Deviation) for each building
building_names = df_numeric.columns  # Building names are column names
means = df_numeric.mean()
variabilities = df_numeric.std()

# Prepare data for clustering
X = np.column_stack([means, variabilities])

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# DBSCAN Clustering
# You might need to adjust eps and min_samples based on your data
dbscan = DBSCAN(eps=0.5, min_samples=5)  # Adjust eps and min_samples
clusters = dbscan.fit_predict(X_scaled)

# Add cluster labels to the data
df_results = pd.DataFrame({'Building': building_names, 'Mean': means, 'Variability': variabilities, 'Cluster': clusters})

# Visualize the clusters
plt.figure(figsize=(12, 8))
scatter = plt.scatter(df_results['Mean'], df_results['Variability'], c=df_results['Cluster'], cmap='viridis')
plt.xlabel('Mean kWh Consumption')
plt.ylabel('Standard Deviation of kWh Consumption')
plt.title('Feature-Based Scatter Plot (Mean vs. Variability) with DBSCAN Clusters')

# Add annotations for each point, using only the first 4 digits of the building code
for i, building in enumerate(df_results['Building']):
    building_code = building[:4]  # Extract the first 4 characters
    plt.annotate(building_code, (df_results['Mean'][i], df_results['Variability'][i]), textcoords="offset points", xytext=(5,5),

plt.colorbar(scatter, label='Cluster')
plt.grid(True)
plt.show()
```

```python
# Print cluster assignments
print(df_results)

# Evaluate the clustering ( silhouette score is not good for DBScan)
from sklearn.metrics import davies_bouldin_score
# Filter out noise points (cluster -1) for evaluation
X_no_noise = X_scaled[clusters != -1]
clusters_no_noise = clusters[clusters != -1]

if len(set(clusters_no_noise)) > 1 and len(clusters_no_noise) > 0:
    db_index = davies_bouldin_score(X_no_noise, clusters_no_noise)
    print(f'Davies-Bouldin Index: {db_index}')
else:
    print("Davies-Bouldin Index cannot be computed for a single cluster or no clusters.")
```

```
C:\Users\sugan\AppData\Local\Temp\ipykernel_20752\618172478.py:52: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  plt.annotate(building_code, (df_results['Mean'][i], df_results['Variability'][i]), textcoords="offset points", xytext=(5,5), ha='center', fontsize=8)
```



	Building \
F204 444 Great King Street - kWh	F204 444 Great King Street - kWh
F813 Bill Robertson Library 1 MSB:	F813 Bill Robertson Library 1 MSB:
F419 ISB West Excluding Shops	F419 ISB West Excluding Shops
F505 Richardson Library Block Rising Main:	F505 Richardson Library Block Rising Main:
F518 Arts 1 Submains MSB:	F518 Arts 1 Submains MSB:
F516 97 Albany & F517 99 Albany, F513 262 Leith...	F516 97 Albany & F517 99 Albany, F513 262 Leit...
F614 1 School of Business Incomer 1 (Lower floo...	F614 1 School of Business Incomer 1 (Lower flo...
F614 2 School of Business Incomer 2 (Upper floo...	F614 2 School of Business Incomer 2 (Upper flo...
F618 1 Psychology substation- Goddard	F618 1 Psychology substation- Goddard
F505 1 Richardson Mains:	F505 1 Richardson Mains:
G506/07 Archway buildings (incl. Allen & Marama...	G506/07 Archway buildings (incl. Allen & Maram...
F315 Botany Tin Hut - kWh	F315 Botany Tin Hut - kWh
F325 Physical Education - kWh	F325 Physical Education - kWh
F711 Executive Residence - kWh	F711 Executive Residence - kWh
F812 UOCOE Owheo Building - kWh	F812 UOCOE Owheo Building - kWh
F813 UOCOE Robertson Library - kWh	F813 UOCOE Robertson Library - kWh
F940 Plaza Building - kWh	F940 Plaza Building - kWh
F9XX College of Education main (Boiler room) - kWh	F9XX College of Education main (Boiler room) -...
G401 Mellor Laboratories - kWh	G401 Mellor Laboratories - kWh
G505 Geology north:	G505 Geology north:
G505 Geology south:	G505 Geology south:
G403 Biochemistry - kWh	G403 Biochemistry - kWh
G404 Microbiology - kWh	G404 Microbiology - kWh
G413 Science 2 - kWh	G413 Science 2 - kWh
G608 St Margarets College - kWh	G608 St Margarets College - kWh
G60X UNICOL - kWh	G60X UNICOL - kWh
H41X Selwyn College - kWh E2	H41X Selwyn College - kWh E2
H633 Arana College main - kWh	H633 Arana College main - kWh
H71X Studholm College - kWh E2	H71X Studholm College - kWh E2

	Mean \
F204 444 Great King Street - kWh	104984.115294
F813 Bill Robertson Library 1 MSB:	37347.898235
F419 ISB West Excluding Shops	64383.837941
F505 Richardson Library Block Rising Main:	10948.797353
F518 Arts 1 Submains MSB:	23527.792647
F516 97 Albany & F517 99 Albany, F513 262 Leith...	2099.598529
F614 1 School of Business Incomer 1 (Lower floo...	62236.191471
F614 2 School of Business Incomer 2 (Upper floo...	40916.118824
F618 1 Psychology substation- Goddard	90395.376471
F505 1 Richardson Mains:	62012.273529
G506/07 Archway buildings (incl. Allen & Marama...	21724.690000
F315 Botany Tin Hut - kWh	35180.897000
F325 Physical Education - kWh	34967.750559
F711 Executive Residence - kWh	10053.446118
F812 UOCOE Owheo Building - kWh	29429.510000
F813 UOCOE Robertson Library - kWh	38259.774706
F940 Plaza Building - kWh	121209.342588
F9XX College of Education main (Boiler room) - kWh	63808.682118
G401 Mellor Laboratories - kWh	113520.799941
G505 Geology north:	35194.934416
G505 Geology south:	11361.495764
G403 Biochemistry - kWh	137393.186824
G404 Microbiology - kWh	105527.793647
G413 Science 2 - kWh	118566.597647
G608 St Margarets College - kWh	32617.197529
G60X UNICOL - kWh	64788.225412
H41X Selwyn College - kWh E2	42745.803176
H633 Arana College main - kWh	34543.669647
H71X Studholm College - kWh E2	40838.764118

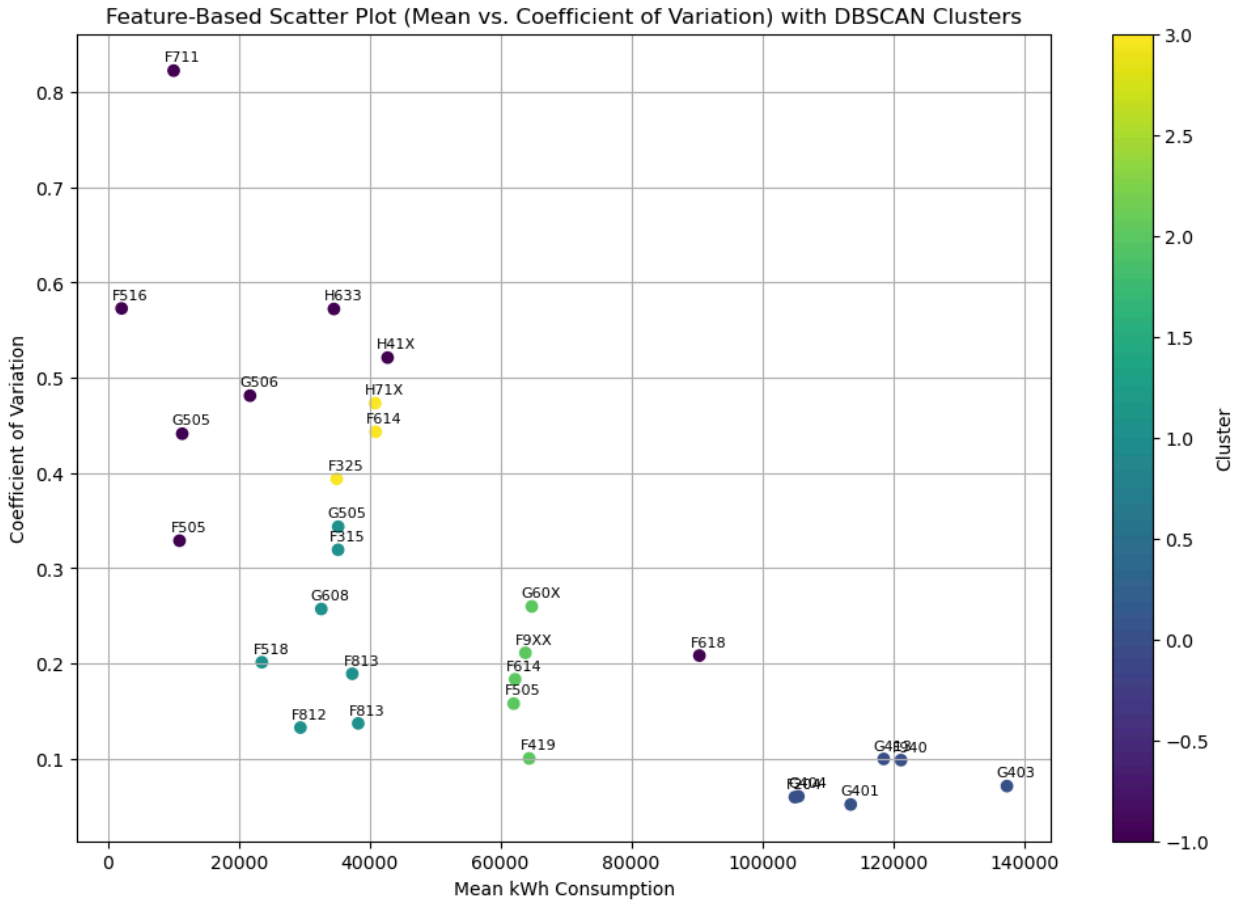
	Variability	Cluster
F204 444 Great King Street - kWh	6268.750174	-1
F813 Bill Robertson Library 1 MSB:	7068.161567	-1
F419 ISB West Excluding Shops	6461.866335	-1
F505 Richardson Library Block Rising Main:	3601.776699	0
F518 Arts 1 Submains MSB:	4737.794359	0
F516 97 Albany & F517 99 Albany, F513 262 Leith...	1202.747861	-1
F614 1 School of Business Incomer 1 (Lower floo...	11420.561191	-1
F614 2 School of Business Incomer 2 (Upper floo...	18139.801865	-1
F618 1 Psychology substation- Goddard	18838.860936	-1
F505 1 Richardson Mains:	9793.195525	-1
G506/07 Archway buildings (incl. Allen & Marama...	10455.484540	-1
F315 Botany Tin Hut - kWh	11238.201322	-1
F325 Physical Education - kWh	13775.414136	-1
F711 Executive Residence - kWh	8269.363494	-1
F812 UOCOE Owheo Building - kWh	3907.463887	0
F813 UOCOE Robertson Library - kWh	5248.067608	0
F940 Plaza Building - kWh	11951.230300	-1
F9XX College of Education main (Boiler room) - kWh	13475.951879	-1
G401 Mellor Laboratories - kWh	5905.001152	-1
G505 Geology north:	12095.704919	-1
G505 Geology south:	5013.986694	0
G403 Biochemistry - kWh	9804.061099	-1
G404 Microbiology - kWh	6392.211757	-1
G413 Science 2 - kWh	11838.646743	-1
G608 St Margarets College - kWh	8391.687774	-1
G60X UNICOL - kWh	16843.159045	-1
H41X Selwyn College - kWh E2	22280.259141	-1

```
H633 Arana College main - kWh                      19769.066869       -1
H71X Studholm College - kWh E2                     19333.808156       -1
Davies-Bouldin Index cannot be computed for a single cluster or no clusters.
```

In [270…

```python
import pandas as pd
import numpy as np
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

# Load the dataset
df = pd.read_csv("stream_data_filtered - Copy.csv")

# Data Cleaning and Preprocessing
# Handle missing values by filling with the mean of each column
df = df.fillna(df.mean(numeric_only=True))

# Identify and exclude non-numeric columns
non_numeric_cols = df.select_dtypes(exclude=['number']).columns
df_numeric = df.drop(columns=non_numeric_cols)

# Remove the 'Total kwh' column
if 'Total kwh' in df_numeric.columns:
    df_numeric = df_numeric.drop(columns=['Total kwh'])

# Calculate Mean and Coefficient of Variation (CV) for each building
building_names = df_numeric.columns  # Building names are column names
means = df_numeric.mean()
std_devs = df_numeric.std()
cvs = std_devs / means  # Coefficient of Variation

# Prepare data for clustering
X = np.column_stack([means, cvs])

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# DBSCAN Clustering
# You might need to adjust eps and min_samples based on your data
dbscan = DBSCAN(eps=0.5, min_samples=5)  # Adjust eps and min_samples
clusters = dbscan.fit_predict(X_scaled)

# Add cluster labels to the data
df_results = pd.DataFrame({'Building': building_names, 'Mean': means, 'Coefficient of Variation': cvs, 'Cluster': clusters})

# Visualize the clusters
plt.figure(figsize=(12, 8))
scatter = plt.scatter(df_results['Mean'], df_results['Coefficient of Variation'], c=df_results['Cluster'], cmap='viridis')
plt.xlabel('Mean kWh Consumption')
plt.ylabel('Coefficient of Variation')
plt.title('Feature-Based Scatter Plot (Mean vs. Coefficient of Variation) with DBSCAN Clusters')

# Add annotations for each point, using only the first 4 digits of the building code
for i, building in enumerate(df_results['Building']):
    building_code = building[:4]  # Extract the first 4 characters
    plt.annotate(building_code, (df_results['Mean'][i], df_results['Coefficient of Variation'][i]), textcoords="offset points", x

plt.colorbar(scatter, label='Cluster')
plt.grid(True)
plt.show()

# Print cluster assignments
print(df_results)

# Evaluate the clustering ( silhouette score is not good for DBScan)
from sklearn.metrics import davies_bouldin_score
# Filter out noise points (cluster -1) for evaluation
X_no_noise = X_scaled[clusters != -1]
clusters_no_noise = clusters[clusters != -1]

if len(set(clusters_no_noise)) > 1 and len(clusters_no_noise) > 0:
    db_index = davies_bouldin_score(X_no_noise, clusters_no_noise)
    print(f'Davies-Bouldin Index: {db_index}')
else:
    print("Davies-Bouldin Index cannot be computed for a single cluster or no clusters.")
```

```
C:\Users\sugan\AppData\Local\Temp\ipykernel_20752\4066143553.py:53: FutureWarning: Series.__getitem__ treating keys as positions i
s deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a
value by position, use `ser.iloc[pos]`
  plt.annotate(building_code, (df_results['Mean'][i], df_results['Coefficient of Variation'][i]), textcoords="offset points", xyte
xt=(5,5), ha='center', fontsize=8)
```


Feature-Based Scatter Plot (Mean vs. Coefficient of Variation) with DBSCAN Clusters
Coefficient of Variation
Mean kWh Consumption
Cluster
F711
F516
H633
H41X
G506
H71X
G505
F614
F325
G505
F505
F315
G608
G60X
F9XX
F618
F518
F813
F614
F505
F812
F813
F419
G403
G401
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20000
40000
60000
80000
100000
120000
140000
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0

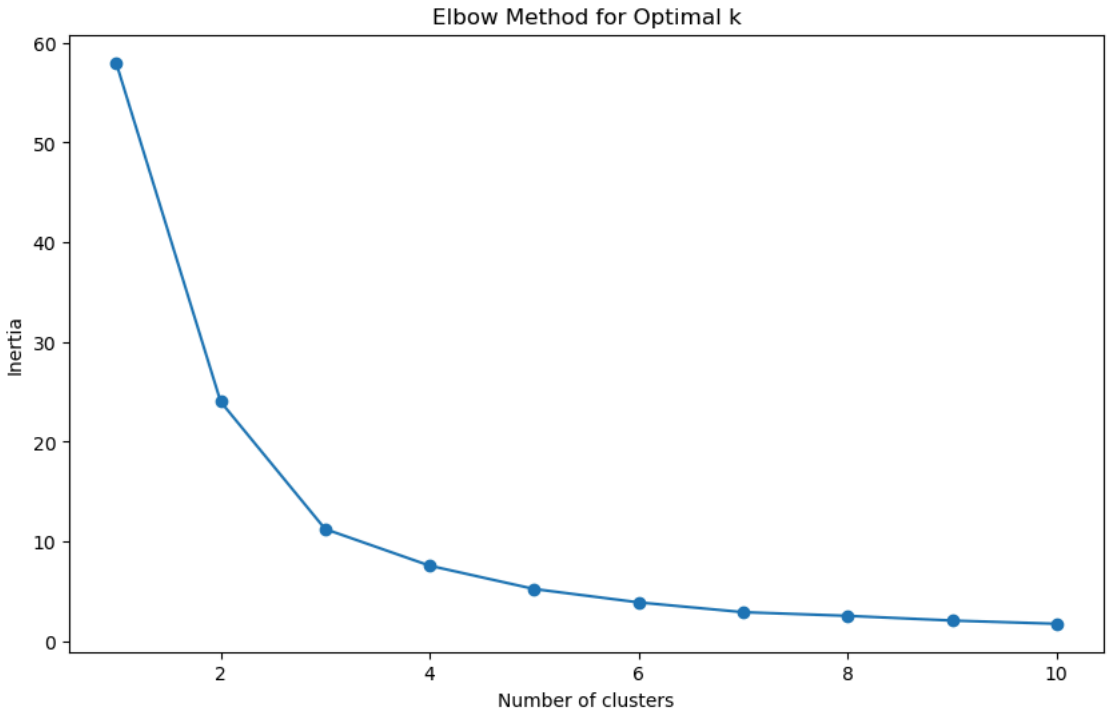
```
                                                                                                                Building  \
F204 444 Great King Street - kWh                                                   F204 444 Great King Street - kWh
F813 Bill Robertson Library 1 MSB:                                                F813 Bill Robertson Library 1 MSB:
F419 ISB West Excluding Shops                                                         F419 ISB West Excluding Shops
F505 Richardson Library Block Rising Main:                                F505 Richardson Library Block Rising Main:
F518 Arts 1 Submains MSB:                                                                 F518 Arts 1 Submains MSB:
F516 97 Albany & F517 99 Albany, F513 262 Leith...  F516 97 Albany & F517 99 Albany, F513 262 Leit...
F614 1 School of Business Incomer 1 (Lower floo...  F614 1 School of Business Incomer 1 (Lower flo...
F614 2 School of Business Incomer 2 (Upper floo...  F614 2 School of Business Incomer 2 (Upper flo...
F618 1 Psychology substation- Goddard                                   F618 1 Psychology substation- Goddard
F505 1 Richardson Mains:                                                                  F505 1 Richardson Mains:
G506/07 Archway buildings (incl. Allen & Marama...  G506/07 Archway buildings (incl. Allen & Maram...
F315 Botany Tin Hut - kWh                                                                 F315 Botany Tin Hut - kWh
F325 Physical Education - kWh                                                         F325 Physical Education - kWh
F711 Executive Residence - kWh                                                       F711 Executive Residence - kWh
F812 UOCOE Owheo Building - kWh                                                     F812 UOCOE Owheo Building - kWh
F813 UOCOE Robertson Library - kWh                                                F813 UOCOE Robertson Library - kWh
F940 Plaza Building - kWh                                                                 F940 Plaza Building - kWh
F9XX College of Education main (Boiler room) - kWh  F9XX College of Education main (Boiler room) -...
G401 Mellor Laboratories - kWh                                                       G401 Mellor Laboratories - kWh
G505 Geology north:                                                                           G505 Geology north:
G505 Geology south:                                                                           G505 Geology south:
G403 Biochemistry - kWh                                                                 G403 Biochemistry - kWh
G404 Microbiology - kWh                                                                 G404 Microbiology - kWh
G413 Science 2 - kWh                                                                       G413 Science 2 - kWh
G608 St Margarets College - kWh                                                     G608 St Margarets College - kWh
G60X UNICOL - kWh                                                                             G60X UNICOL - kWh
H41X Selwyn College - kWh E2                                                           H41X Selwyn College - kWh E2
H633 Arana College main - kWh                                                         H633 Arana College main - kWh
H71X Studholm College - kWh E2                                                       H71X Studholm College - kWh E2

                                                                 Mean  \
F204 444 Great King Street - kWh                     104984.115294
F813 Bill Robertson Library 1 MSB:                    37347.898235
F419 ISB West Excluding Shops                         64383.837941
F505 Richardson Library Block Rising Main:            10948.797353
F518 Arts 1 Submains MSB:                             23527.792647
F516 97 Albany & F517 99 Albany, F513 262 Leith...     2099.598529
F614 1 School of Business Incomer 1 (Lower floo...    62236.191471
F614 2 School of Business Incomer 2 (Upper floo...    40916.118824
F618 1 Psychology substation- Goddard                 90395.376471
F505 1 Richardson Mains:                              62012.273529
G506/07 Archway buildings (incl. Allen & Marama...    21724.690000
F315 Botany Tin Hut - kWh                             35180.897000
F325 Physical Education - kWh                         34967.750559
F711 Executive Residence - kWh                        10053.446118
F812 UOCOE Owheo Building - kWh                       29429.510000
F813 UOCOE Robertson Library - kWh                    38259.774706
F940 Plaza Building - kWh                            121209.342588
F9XX College of Education main (Boiler room) - kWh    63808.682118
G401 Mellor Laboratories - kWh                       113520.799941
G505 Geology north:                                   35194.934416
G505 Geology south:                                   11361.495764
G403 Biochemistry - kWh                              137393.186824
G404 Microbiology - kWh                              105527.793647
G413 Science 2 - kWh                                 118566.597647
G608 St Margarets College - kWh                       32617.197529
G60X UNICOL - kWh                                     64788.225412
H41X Selwyn College - kWh E2                          42745.803176
H633 Arana College main - kWh                         34543.669647
H71X Studholm College - kWh E2                        40838.764118

                                                     Coefficient of Variation  \
F204 444 Great King Street - kWh                                      0.059711
F813 Bill Robertson Library 1 MSB:                                    0.189252
F419 ISB West Excluding Shops                                         0.100365
F505 Richardson Library Block Rising Main:                            0.328966
F518 Arts 1 Submains MSB:                                             0.201370
F516 97 Albany & F517 99 Albany, F513 262 Leith...                    0.572847
F614 1 School of Business Incomer 1 (Lower floo...                    0.183504
F614 2 School of Business Incomer 2 (Upper floo...                    0.443341
F618 1 Psychology substation- Goddard                                 0.208405
F505 1 Richardson Mains:                                              0.157924
G506/07 Archway buildings (incl. Allen & Marama...                    0.481272
F315 Botany Tin Hut - kWh                                             0.319440
F325 Physical Education - kWh                                         0.393946
F711 Executive Residence - kWh                                        0.822540
F812 UOCOE Owheo Building - kWh                                       0.132774
F813 UOCOE Robertson Library - kWh                                    0.137169
F940 Plaza Building - kWh                                             0.098600
F9XX College of Education main (Boiler room) - kWh                    0.211193
G401 Mellor Laboratories - kWh                                        0.052017
G505 Geology north:                                                   0.343677
G505 Geology south:                                                   0.441314
G403 Biochemistry - kWh                                               0.071358
G404 Microbiology - kWh                                               0.060574
G413 Science 2 - kWh                                                  0.099848
G608 St Margarets College - kWh                                       0.257278
G60X UNICOL - kWh                                                     0.259973
H41X Selwyn College - kWh E2                                          0.521227
```

```
H633 Arana College main - kWh                                    0.572292
H71X Studholm College - kWh E2                                   0.473418

                                                        Cluster
F204 444 Great King Street - kWh                              0
F813 Bill Robertson Library 1 MSB:                            1
F419 ISB West Excluding Shops                                 2
F505 Richardson Library Block Rising Main:                   -1
F518 Arts 1 Submains MSB:                                     1
F516 97 Albany & F517 99 Albany, F513 262 Leith...           -1
F614 1 School of Business Incomer 1 (Lower floo...            2
F614 2 School of Business Incomer 2 (Upper floo...            3
F618 1 Psychology substation- Goddard                        -1
F505 1 Richardson Mains:                                      2
G506/07 Archway buildings (incl. Allen & Marama...           -1
F315 Botany Tin Hut - kWh                                     1
F325 Physical Education - kWh                                 3
F711 Executive Residence - kWh                               -1
F812 UOCOE Owheo Building - kWh                               1
F813 UOCOE Robertson Library - kWh                            1
F940 Plaza Building - kWh                                     0
F9XX College of Education main (Boiler room) - kWh            2
G401 Mellor Laboratories - kWh                                0
G505 Geology north:                                           1
G505 Geology south:                                          -1
G403 Biochemistry - kWh                                       0
G404 Microbiology - kWh                                       0
G413 Science 2 - kWh                                          0
G608 St Margarets College - kWh                               1
G60X UNICOL - kWh                                             2
H41X Selwyn College - kWh E2                                 -1
H633 Arana College main - kWh                                -1
H71X Studholm College - kWh E2                                3
Davies-Bouldin Index: 0.5783503529613474
```

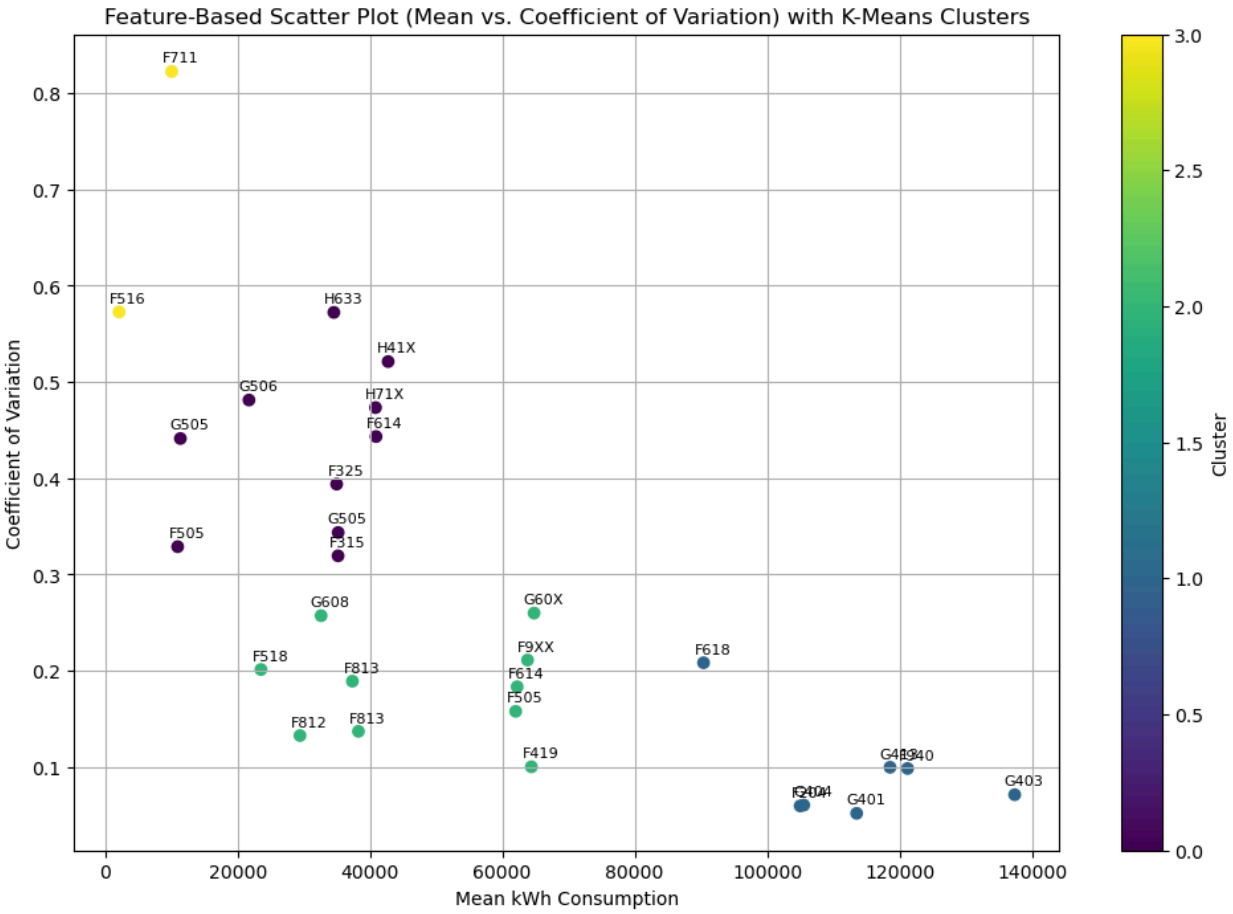
In [286…

```python
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import davies_bouldin_score

# Load the dataset
df = pd.read_csv("stream_data_filtered - Copy.csv")

# Data Cleaning and Preprocessing
# Handle missing values by filling with the mean of each column
df = df.fillna(df.mean(numeric_only=True))

# Identify and exclude non-numeric columns
non_numeric_cols = df.select_dtypes(exclude=['number']).columns
df_numeric = df.drop(columns=non_numeric_cols)

# Remove the 'Total kwh' column
if 'Total kwh' in df_numeric.columns:
    df_numeric = df_numeric.drop(columns=['Total kwh'])

# Calculate Mean and Coefficient of Variation (CV) for each building
building_names = df_numeric.columns  # Building names are column names
means = df_numeric.mean()
std_devs = df_numeric.std()
cvs = std_devs / means  # Coefficient of Variation

# Prepare data for clustering
X = np.column_stack([means, cvs])

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Determine optimal number of clusters using the Elbow Method
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42, n_init=10)  # Explicitly set n_init
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

# Plot the Elbow Method graph
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
# K-Means Clustering
n_clusters = 4 # Set the number of clusters based on the Elbow Method
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)  # Explicitly set n_init
clusters = kmeans.fit_predict(X_scaled)

# Add cluster labels to the data
```

```
df_results = pd.DataFrame({'Building': building_names, 'Mean': means, 'Coefficient of Variation': cvs, 'Cluster': clusters})

# Visualize the clusters
plt.figure(figsize=(12, 8))
scatter = plt.scatter(df_results['Mean'], df_results['Coefficient of Variation'], c=df_results['Cluster'], cmap='viridis')
plt.xlabel('Mean kWh Consumption')
plt.ylabel('Coefficient of Variation')
plt.title('Feature-Based Scatter Plot (Mean vs. Coefficient of Variation) with K-Means Clusters')

# Add annotations for each point, using only the first 4 digits of the building code
for i, building in enumerate(df_results['Building']):
    building_code = building[:4]  # Extract the first 4 characters
    plt.annotate(building_code, (df_results['Mean'][i], df_results['Coefficient of Variation'][i]), textcoords="offset points", x

plt.colorbar(scatter, label='Cluster')
plt.grid(True)
plt.show()


# Print cluster assignments
print(df_results)

# Evaluate the clustering
db_index = davies_bouldin_score(X_scaled, clusters)
print(f'Davies-Bouldin Index: {db_index}')
```




```
C:\Users\sugan\AppData\Local\Temp\ipykernel_20752\1661472713.py:68: FutureWarning: Series.__getitem__ treating keys as positions i
s deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a
value by position, use `ser.iloc[pos]`
  plt.annotate(building_code, (df_results['Mean'][i], df_results['Coefficient of Variation'][i]), textcoords="offset points", xyte
xt=(5,5), ha='center', fontsize=8)
```


Feature-Based Scatter Plot (Mean vs. Coefficient of Variation) with K-Means Clusters
Coefficient of Variation
Mean kWh Consumption
Cluster
F711
F516
H633
H41X
G506
H71X
G505
F614
F325
G505
F505
F315
G608
G60X
F9XX
F618
F518
F813
F614
F505
F812
F813
F419
G401
G403
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20000
40000
60000
80000
100000
120000
140000
3.0
2.5
2.0
1.5
1.0
0.5
0.0

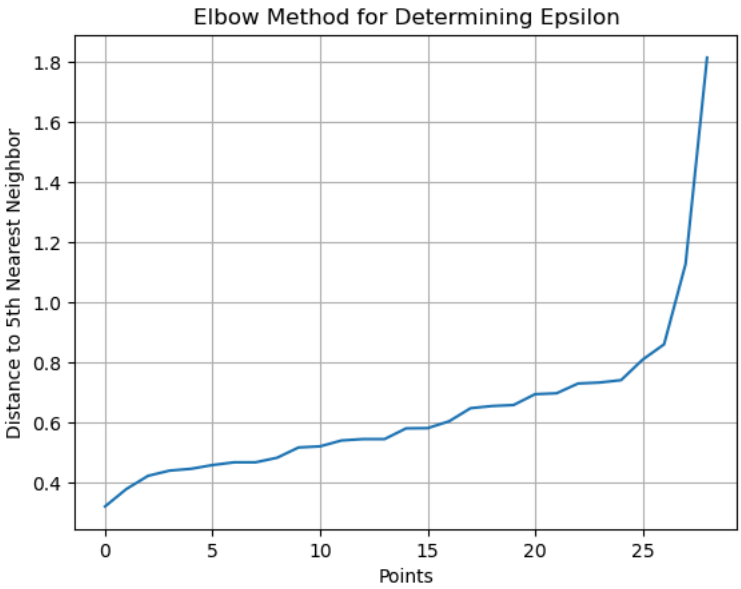
```
                                                                                                                Building  \
F204 444 Great King Street - kWh                                                  F204 444 Great King Street - kWh
F813 Bill Robertson Library 1 MSB:                                               F813 Bill Robertson Library 1 MSB:
F419 ISB West Excluding Shops                                                        F419 ISB West Excluding Shops
F505 Richardson Library Block Rising Main:                             F505 Richardson Library Block Rising Main:
F518 Arts 1 Submains MSB:                                                                 F518 Arts 1 Submains MSB:
F516 97 Albany & F517 99 Albany, F513 262 Leith...  F516 97 Albany & F517 99 Albany, F513 262 Leit...
F614 1 School of Business Incomer 1 (Lower floo...  F614 1 School of Business Incomer 1 (Lower flo...
F614 2 School of Business Incomer 2 (Upper floo...  F614 2 School of Business Incomer 2 (Upper flo...
F618 1 Psychology substation- Goddard                                  F618 1 Psychology substation- Goddard
F505 1 Richardson Mains:                                                                 F505 1 Richardson Mains:
G506/07 Archway buildings (incl. Allen & Marama...  G506/07 Archway buildings (incl. Allen & Maram...
F315 Botany Tin Hut - kWh                                                                 F315 Botany Tin Hut - kWh
F325 Physical Education - kWh                                                         F325 Physical Education - kWh
F711 Executive Residence - kWh                                                       F711 Executive Residence - kWh
F812 UOCOE Owheo Building - kWh                                                     F812 UOCOE Owheo Building - kWh
F813 UOCOE Robertson Library - kWh                                               F813 UOCOE Robertson Library - kWh
F940 Plaza Building - kWh                                                                 F940 Plaza Building - kWh
F9XX College of Education main (Boiler room) - kWh  F9XX College of Education main (Boiler room) -...
G401 Mellor Laboratories - kWh                                                       G401 Mellor Laboratories - kWh
G505 Geology north:                                                                       G505 Geology north:
G505 Geology south:                                                                       G505 Geology south:
G403 Biochemistry - kWh                                                                 G403 Biochemistry - kWh
G404 Microbiology - kWh                                                                 G404 Microbiology - kWh
G413 Science 2 - kWh                                                                       G413 Science 2 - kWh
G608 St Margarets College - kWh                                                     G608 St Margarets College - kWh
G60X UNICOL - kWh                                                                             G60X UNICOL - kWh
H41X Selwyn College - kWh E2                                                           H41X Selwyn College - kWh E2
H633 Arana College main - kWh                                                         H633 Arana College main - kWh
H71X Studholm College - kWh E2                                                       H71X Studholm College - kWh E2

                                                                 Mean  \
F204 444 Great King Street - kWh                       104984.115294
F813 Bill Robertson Library 1 MSB:                      37347.898235
F419 ISB West Excluding Shops                           64383.837941
F505 Richardson Library Block Rising Main:              10948.797353
F518 Arts 1 Submains MSB:                               23527.792647
F516 97 Albany & F517 99 Albany, F513 262 Leith...       2099.598529
F614 1 School of Business Incomer 1 (Lower floo...      62236.191471
F614 2 School of Business Incomer 2 (Upper floo...      40916.118824
F618 1 Psychology substation- Goddard                   90395.376471
F505 1 Richardson Mains:                                62012.273529
G506/07 Archway buildings (incl. Allen & Marama...      21724.690000
F315 Botany Tin Hut - kWh                               35180.897000
F325 Physical Education - kWh                           34967.750559
F711 Executive Residence - kWh                          10053.446118
F812 UOCOE Owheo Building - kWh                         29429.510000
F813 UOCOE Robertson Library - kWh                      38259.774706
F940 Plaza Building - kWh                              121209.342588
F9XX College of Education main (Boiler room) - kWh      63808.682118
G401 Mellor Laboratories - kWh                         113520.799941
G505 Geology north:                                     35194.934416
G505 Geology south:                                     11361.495764
G403 Biochemistry - kWh                                137393.186824
G404 Microbiology - kWh                                105527.793647
G413 Science 2 - kWh                                   118566.597647
G608 St Margarets College - kWh                         32617.197529
G60X UNICOL - kWh                                       64788.225412
H41X Selwyn College - kWh E2                            42745.803176
H633 Arana College main - kWh                           34543.669647
H71X Studholm College - kWh E2                          40838.764118

                                                   Coefficient of Variation  \
F204 444 Great King Street - kWh                                    0.059711
F813 Bill Robertson Library 1 MSB:                                  0.189252
F419 ISB West Excluding Shops                                       0.100365
F505 Richardson Library Block Rising Main:                          0.328966
F518 Arts 1 Submains MSB:                                           0.201370
F516 97 Albany & F517 99 Albany, F513 262 Leith...                  0.572847
F614 1 School of Business Incomer 1 (Lower floo...                  0.183504
F614 2 School of Business Incomer 2 (Upper floo...                  0.443341
F618 1 Psychology substation- Goddard                               0.208405
F505 1 Richardson Mains:                                            0.157924
G506/07 Archway buildings (incl. Allen & Marama...                  0.481272
F315 Botany Tin Hut - kWh                                           0.319440
F325 Physical Education - kWh                                       0.393946
F711 Executive Residence - kWh                                      0.822540
F812 UOCOE Owheo Building - kWh                                     0.132774
F813 UOCOE Robertson Library - kWh                                  0.137169
F940 Plaza Building - kWh                                           0.098600
F9XX College of Education main (Boiler room) - kWh                  0.211193
G401 Mellor Laboratories - kWh                                      0.052017
G505 Geology north:                                                 0.343677
G505 Geology south:                                                 0.441314
G403 Biochemistry - kWh                                             0.071358
G404 Microbiology - kWh                                             0.060574
G413 Science 2 - kWh                                                0.099848
G608 St Margarets College - kWh                                     0.257278
G60X UNICOL - kWh                                                   0.259973
H41X Selwyn College - kWh E2                                        0.521227
```

```
H633 Arana College main - kWh                                    0.572292
H71X Studholm College - kWh E2                                   0.473418

                                                        Cluster
F204 444 Great King Street - kWh                              1
F813 Bill Robertson Library 1 MSB:                            2
F419 ISB West Excluding Shops                                 2
F505 Richardson Library Block Rising Main:                    0
F518 Arts 1 Submains MSB:                                     2
F516 97 Albany & F517 99 Albany, F513 262 Leith...            3
F614 1 School of Business Incomer 1 (Lower floo...            2
F614 2 School of Business Incomer 2 (Upper floo...            0
F618 1 Psychology substation- Goddard                         1
F505 1 Richardson Mains:                                      2
G506/07 Archway buildings (incl. Allen & Marama...            0
F315 Botany Tin Hut - kWh                                     0
F325 Physical Education - kWh                                 0
F711 Executive Residence - kWh                                3
F812 UOCOE Owheo Building - kWh                               2
F813 UOCOE Robertson Library - kWh                            2
F940 Plaza Building - kWh                                     1
F9XX College of Education main (Boiler room) - kWh            2
G401 Mellor Laboratories - kWh                                1
G505 Geology north:                                           0
G505 Geology south:                                           0
G403 Biochemistry - kWh                                       1
G404 Microbiology - kWh                                       1
G413 Science 2 - kWh                                          1
G608 St Margarets College - kWh                               2
G60X UNICOL - kWh                                             2
H41X Selwyn College - kWh E2                                  0
H633 Arana College main - kWh                                 0
H71X Studholm College - kWh E2                                0
Davies-Bouldin Index: 0.669270300228924
```

In [272…

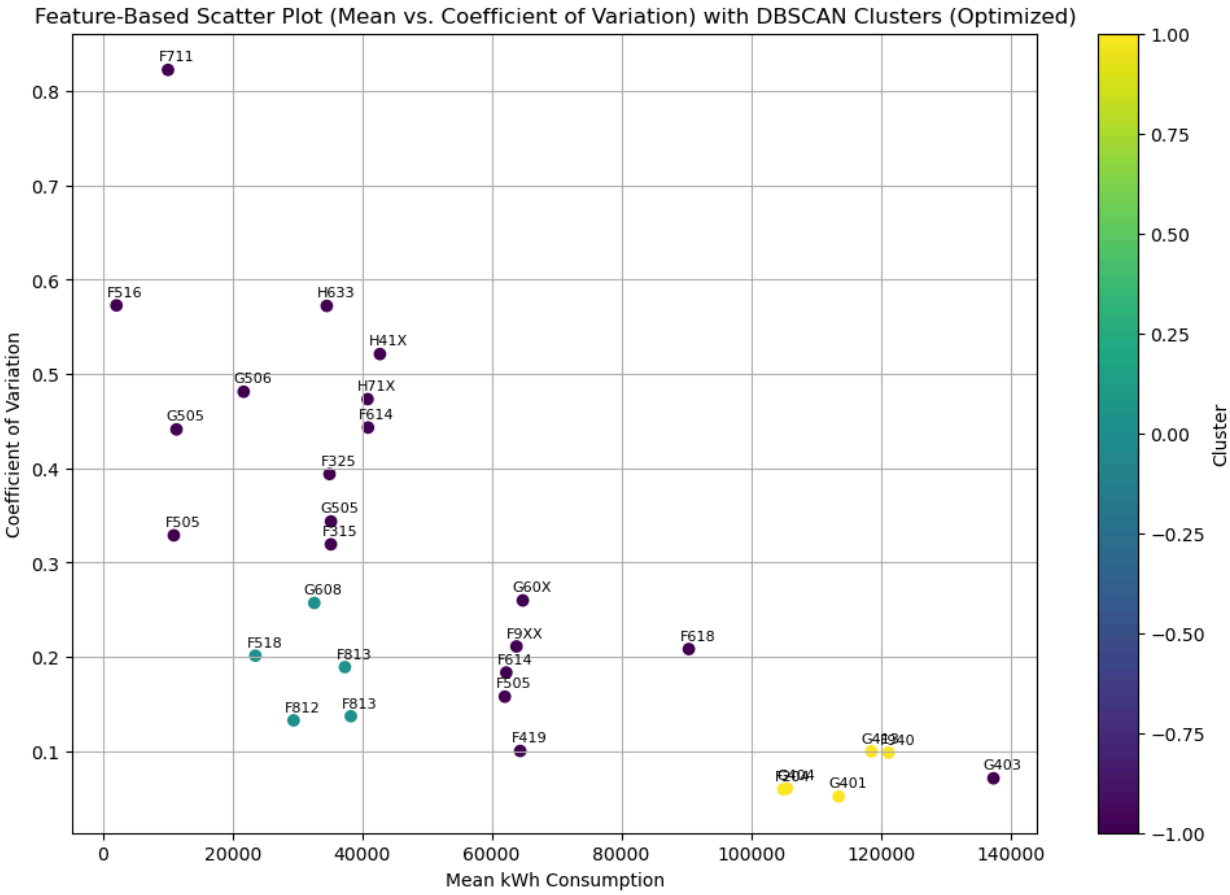
```python
import pandas as pd
import numpy as np
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors
from sklearn.metrics import davies_bouldin_score

# Load the dataset
df = pd.read_csv("stream_data_filtered-Copy.csv")

# Data Cleaning and Preprocessing
# Handle missing values by filling with the mean of each column
df = df.fillna(df.mean(numeric_only=True))

# Identify and exclude non-numeric columns
non_numeric_cols = df.select_dtypes(exclude=['number']).columns
df_numeric = df.drop(columns=non_numeric_cols)

# Remove the 'Total kwh' column
if 'Total kwh' in df_numeric.columns:
    df_numeric = df_numeric.drop(columns=['Total kwh'])

# Calculate Mean and Coefficient of Variation (CV) for each building
building_names = df_numeric.columns  # Building names are column names
means = df_numeric.mean()
std_devs = df_numeric.std()
cvs = std_devs / means  # Coefficient of Variation

# Prepare data for clustering
X = np.column_stack([means, cvs])

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# Elbow Method to find suitable eps
min_samples = 5  # You can adjust min_samples here, or tune it as well
neighbors = NearestNeighbors(n_neighbors=min_samples)
neighbors_fit = neighbors.fit(X_scaled)
distances, indices = neighbors_fit.kneighbors(X_scaled)

distances = np.sort(distances[:, min_samples-1], axis=0)
plt.plot(distances)
plt.xlabel("Points")
plt.ylabel(f"Distance to {min_samples}th Nearest Neighbor")
plt.title("Elbow Method for Determining Epsilon")
plt.grid(True)
plt.show()


# Parameter Tuning (Grid Search) -  Adjust eps and min_samples ranges based on Elbow method results
best_db_index = float('inf')  # Initialize with a high value (lower is better for DB index)
best_eps = None
```

```
best_min_samples = None
best_clusters = None


for eps in np.arange(0.2, 0.6, 0.1):  # Adjust the range and step for eps - based on elbow plot
    for min_samples in range(3, 8):  # Adjust the range for min_samples
        dbscan = DBSCAN(eps=eps, min_samples=min_samples)
        clusters = dbscan.fit_predict(X_scaled)

        # Evaluate (Davies-Bouldin), but only if there are at least 2 clusters and non-noise points
        X_no_noise = X_scaled[clusters != -1]
        clusters_no_noise = clusters[clusters != -1]

        if len(set(clusters_no_noise)) > 1 and len(clusters_no_noise) > 0:

            db_index = davies_bouldin_score(X_no_noise, clusters_no_noise)

            if db_index < best_db_index:
                best_db_index = db_index
                best_eps = eps
                best_min_samples = min_samples
                best_clusters = clusters


print(f"Best EPS: {best_eps}")
print(f"Best Min Samples: {best_min_samples}")
print(f"Best Davies-Bouldin Index: {best_db_index}")


# Use the best parameters to create the final DBSCAN model and visualization
if best_clusters is not None:
    # Add the best cluster labels to your results DataFrame
    df_results = pd.DataFrame({'Building': building_names, 'Mean': means, 'Coefficient of Variation': cvs, 'Cluster': best_cluste

    # Visualize the clusters with the best parameters (copy visualization code here)
    plt.figure(figsize=(12, 8))
    scatter = plt.scatter(df_results['Mean'], df_results['Coefficient of Variation'], c=df_results['Cluster'], cmap='viridis')
    plt.xlabel('Mean kWh Consumption')
    plt.ylabel('Coefficient of Variation')
    plt.title('Feature-Based Scatter Plot (Mean vs. Coefficient of Variation) with DBSCAN Clusters (Optimized)')

    # Add annotations for each point
    for i, building in enumerate(df_results['Building']):
        building_code = building[:4]  # Extract the first 4 characters
        plt.annotate(building_code, (df_results['Mean'][i], df_results['Coefficient of Variation'][i]), textcoords="offset points


    plt.colorbar(scatter, label='Cluster')
    plt.grid(True)
    plt.show()

    print(df_results)
    # Evaluate the clustering ( silhouette score is not good for DBScan)

    # Filter out noise points (cluster -1) for evaluation
    X_no_noise = X_scaled[best_clusters != -1]
    clusters_no_noise = best_clusters[best_clusters != -1]

    if len(set(clusters_no_noise)) > 1 and len(clusters_no_noise) > 0:
        db_index = davies_bouldin_score(X_no_noise, clusters_no_noise)
        print(f'Davies-Bouldin Index: {db_index}')
    else:
        print("Davies-Bouldin Index cannot be computed for a single cluster or no clusters.")


else:
    print("No valid clusters found during parameter tuning.")
```




```
Best EPS: 0.4000000000000001
Best Min Samples: 5
Best Davies-Bouldin Index: 0.21073172509686672
```

```
C:\Users\sugan\AppData\Local\Temp\ipykernel_20752\2729448121.py:100: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  plt.annotate(building_code, (df_results['Mean'][i], df_results['Coefficient of Variation'][i]), textcoords="offset points", xytext=(5,5), ha='center', fontsize=8)
```

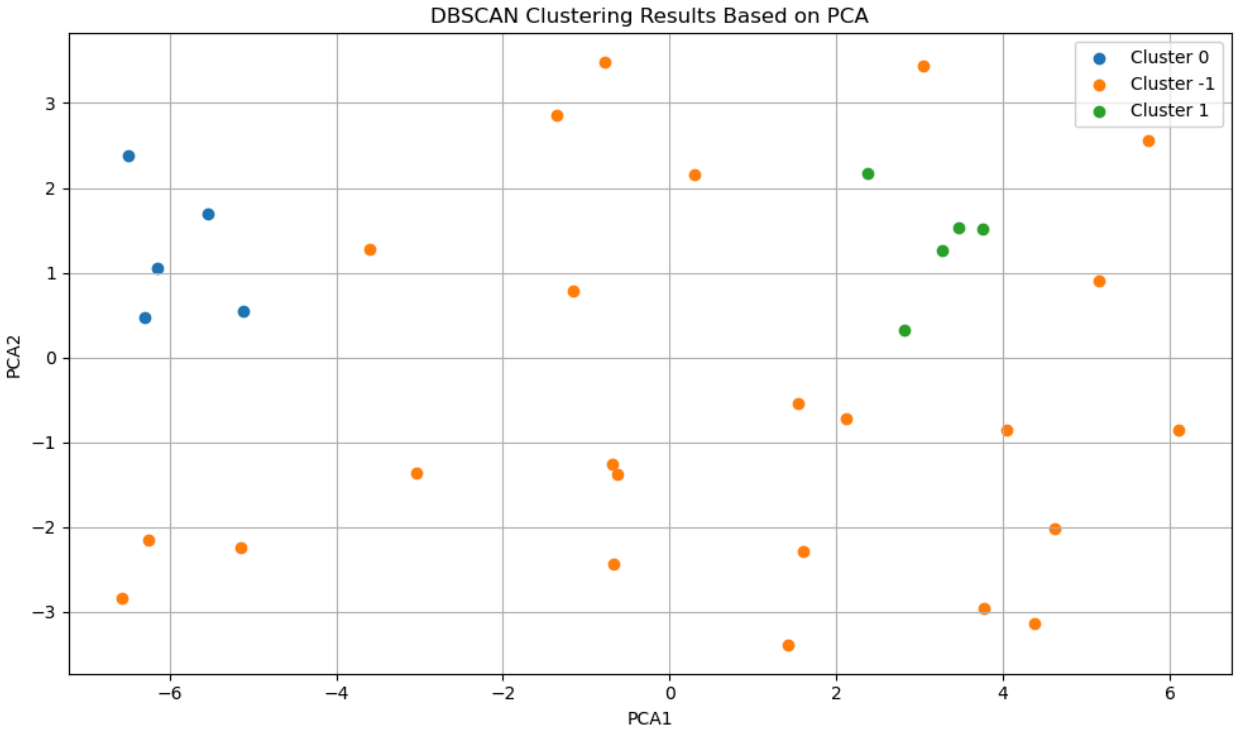


```
                                                                                                                Building  \
F204 444 Great King Street - kWh                                                   F204 444 Great King Street - kWh
F813 Bill Robertson Library 1 MSB:                                                F813 Bill Robertson Library 1 MSB:
F419 ISB West Excluding Shops                                                         F419 ISB West Excluding Shops
F505 Richardson Library Block Rising Main:                                F505 Richardson Library Block Rising Main:
F518 Arts 1 Submains MSB:                                                                 F518 Arts 1 Submains MSB:
F516 97 Albany & F517 99 Albany, F513 262 Leith...  F516 97 Albany & F517 99 Albany, F513 262 Leit...
F614 1 School of Business Incomer 1 (Lower floo...  F614 1 School of Business Incomer 1 (Lower flo...
F614 2 School of Business Incomer 2 (Upper floo...  F614 2 School of Business Incomer 2 (Upper flo...
F618 1 Psychology substation- Goddard                                  F618 1 Psychology substation- Goddard
F505 1 Richardson Mains:                                                                  F505 1 Richardson Mains:
G506/07 Archway buildings (incl. Allen & Marama...  G506/07 Archway buildings (incl. Allen & Maram...
F315 Botany Tin Hut - kWh                                                                 F315 Botany Tin Hut - kWh
F325 Physical Education - kWh                                                          F325 Physical Education - kWh
F711 Executive Residence - kWh                                                        F711 Executive Residence - kWh
F812 UOCOE Owheo Building - kWh                                                      F812 UOCOE Owheo Building - kWh
F813 UOCOE Robertson Library - kWh                                                F813 UOCOE Robertson Library - kWh
F940 Plaza Building - kWh                                                                 F940 Plaza Building - kWh
F9XX College of Education main (Boiler room) - kWh  F9XX College of Education main (Boiler room) -...
G401 Mellor Laboratories - kWh                                                        G401 Mellor Laboratories - kWh
G505 Geology north:                                                                         G505 Geology north:
G505 Geology south:                                                                         G505 Geology south:
G403 Biochemistry - kWh                                                                  G403 Biochemistry - kWh
G404 Microbiology - kWh                                                                  G404 Microbiology - kWh
G413 Science 2 - kWh                                                                        G413 Science 2 - kWh
G608 St Margarets College - kWh                                                      G608 St Margarets College - kWh
G60X UNICOL - kWh                                                                          G60X UNICOL - kWh
H41X Selwyn College - kWh E2                                                          H41X Selwyn College - kWh E2
H633 Arana College main - kWh                                                         H633 Arana College main - kWh
H71X Studholm College - kWh E2                                                        H71X Studholm College - kWh E2

                                                                 Mean  \
F204 444 Great King Street - kWh                       104984.115294
F813 Bill Robertson Library 1 MSB:                      37347.898235
F419 ISB West Excluding Shops                           64383.837941
F505 Richardson Library Block Rising Main:              10948.797353
F518 Arts 1 Submains MSB:                               23527.792647
F516 97 Albany & F517 99 Albany, F513 262 Leith...       2099.598529
F614 1 School of Business Incomer 1 (Lower floo...      62236.191471
F614 2 School of Business Incomer 2 (Upper floo...      40916.118824
F618 1 Psychology substation- Goddard                   90395.376471
F505 1 Richardson Mains:                                62012.273529
G506/07 Archway buildings (incl. Allen & Marama...      21724.690000
F315 Botany Tin Hut - kWh                               35180.897000
F325 Physical Education - kWh                           34967.750559
F711 Executive Residence - kWh                          10053.446118
F812 UOCOE Owheo Building - kWh                         29429.510000
F813 UOCOE Robertson Library - kWh                      38259.774706
F940 Plaza Building - kWh                              121209.342588
F9XX College of Education main (Boiler room) - kWh      63808.682118
G401 Mellor Laboratories - kWh                         113520.799941
G505 Geology north:                                     35194.934416
G505 Geology south:                                     11361.495764
G403 Biochemistry - kWh                                137393.186824
G404 Microbiology - kWh                                105527.793647
G413 Science 2 - kWh                                   118566.597647
G608 St Margarets College - kWh                         32617.197529
G60X UNICOL - kWh                                       64788.225412
H41X Selwyn College - kWh E2                            42745.803176
H633 Arana College main - kWh                           34543.669647
H71X Studholm College - kWh E2                          40838.764118

                                                       Coefficient of Variation  \
F204 444 Great King Street - kWh                                        0.059711
F813 Bill Robertson Library 1 MSB:                                      0.189252
F419 ISB West Excluding Shops                                           0.100365
F505 Richardson Library Block Rising Main:                              0.328966
F518 Arts 1 Submains MSB:                                               0.201370
F516 97 Albany & F517 99 Albany, F513 262 Leith...                      0.572847
F614 1 School of Business Incomer 1 (Lower floo...                      0.183504
F614 2 School of Business Incomer 2 (Upper floo...                      0.443341
F618 1 Psychology substation- Goddard                                   0.208405
F505 1 Richardson Mains:                                                0.157924
G506/07 Archway buildings (incl. Allen & Marama...                      0.481272
F315 Botany Tin Hut - kWh                                               0.319440
F325 Physical Education - kWh                                           0.393946
F711 Executive Residence - kWh                                          0.822540
F812 UOCOE Owheo Building - kWh                                         0.132774
F813 UOCOE Robertson Library - kWh                                      0.137169
F940 Plaza Building - kWh                                               0.098600
F9XX College of Education main (Boiler room) - kWh                      0.211193
G401 Mellor Laboratories - kWh                                          0.052017
G505 Geology north:                                                     0.343677
G505 Geology south:                                                     0.441314
G403 Biochemistry - kWh                                                 0.071358
G404 Microbiology - kWh                                                 0.060574
G413 Science 2 - kWh                                                    0.099848
G608 St Margarets College - kWh                                         0.257278
G60X UNICOL - kWh                                                       0.259973
H41X Selwyn College - kWh E2                                            0.521227
```

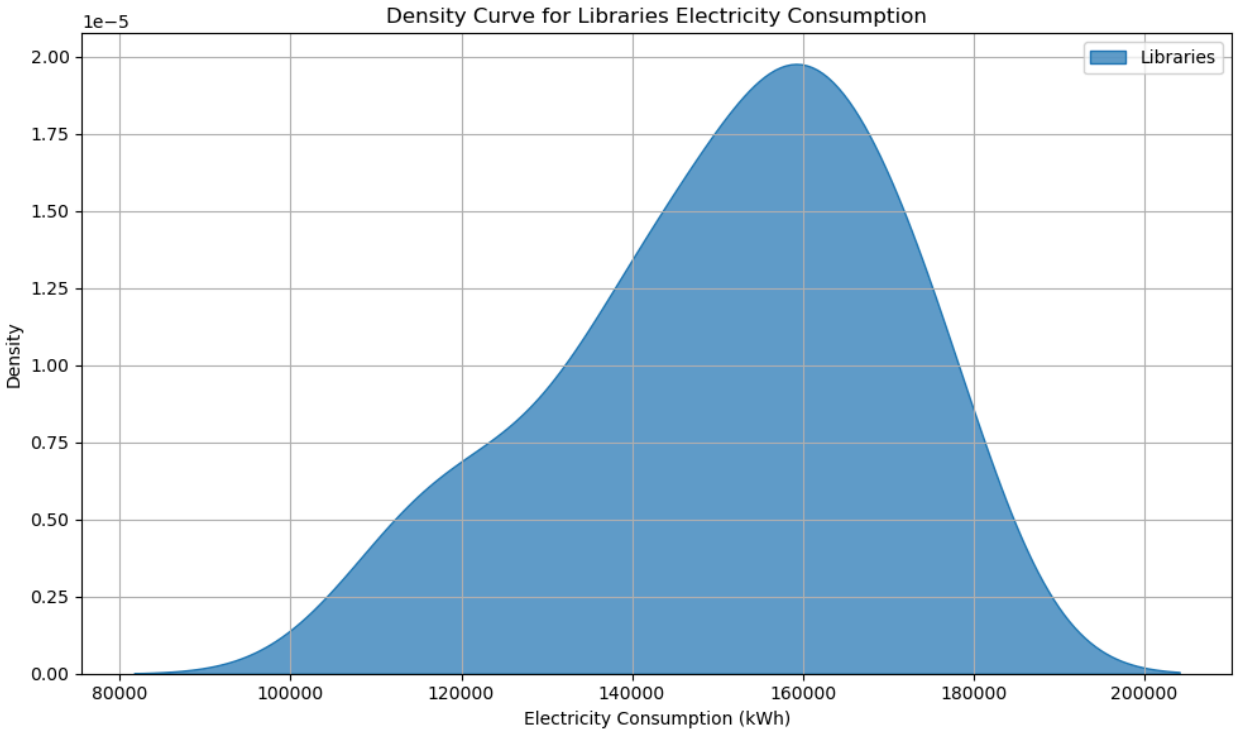
```
H633 Arana College main - kWh                                    0.572292
H71X Studholm College - kWh E2                                   0.473418

                                                        Cluster
F204 444 Great King Street - kWh                              1
F813 Bill Robertson Library 1 MSB:                            0
F419 ISB West Excluding Shops                                -1
F505 Richardson Library Block Rising Main:                   -1
F518 Arts 1 Submains MSB:                                     0
F516 97 Albany & F517 99 Albany, F513 262 Leith...           -1
F614 1 School of Business Incomer 1 (Lower floo...           -1
F614 2 School of Business Incomer 2 (Upper floo...           -1
F618 1 Psychology substation- Goddard                        -1
F505 1 Richardson Mains:                                     -1
G506/07 Archway buildings (incl. Allen & Marama...           -1
F315 Botany Tin Hut - kWh                                    -1
F325 Physical Education - kWh                                -1
F711 Executive Residence - kWh                               -1
F812 UOCOE Owheo Building - kWh                               0
F813 UOCOE Robertson Library - kWh                            0
F940 Plaza Building - kWh                                     1
F9XX College of Education main (Boiler room) - kWh           -1
G401 Mellor Laboratories - kWh                                1
G505 Geology north:                                          -1
G505 Geology south:                                          -1
G403 Biochemistry - kWh                                      -1
G404 Microbiology - kWh                                       1
G413 Science 2 - kWh                                          1
G608 St Margarets College - kWh                               0
G60X UNICOL - kWh                                            -1
H41X Selwyn College - kWh E2                                 -1
H633 Arana College main - kWh                                -1
H71X Studholm College - kWh E2                               -1
Davies-Bouldin Index: 0.21073172509686672
```

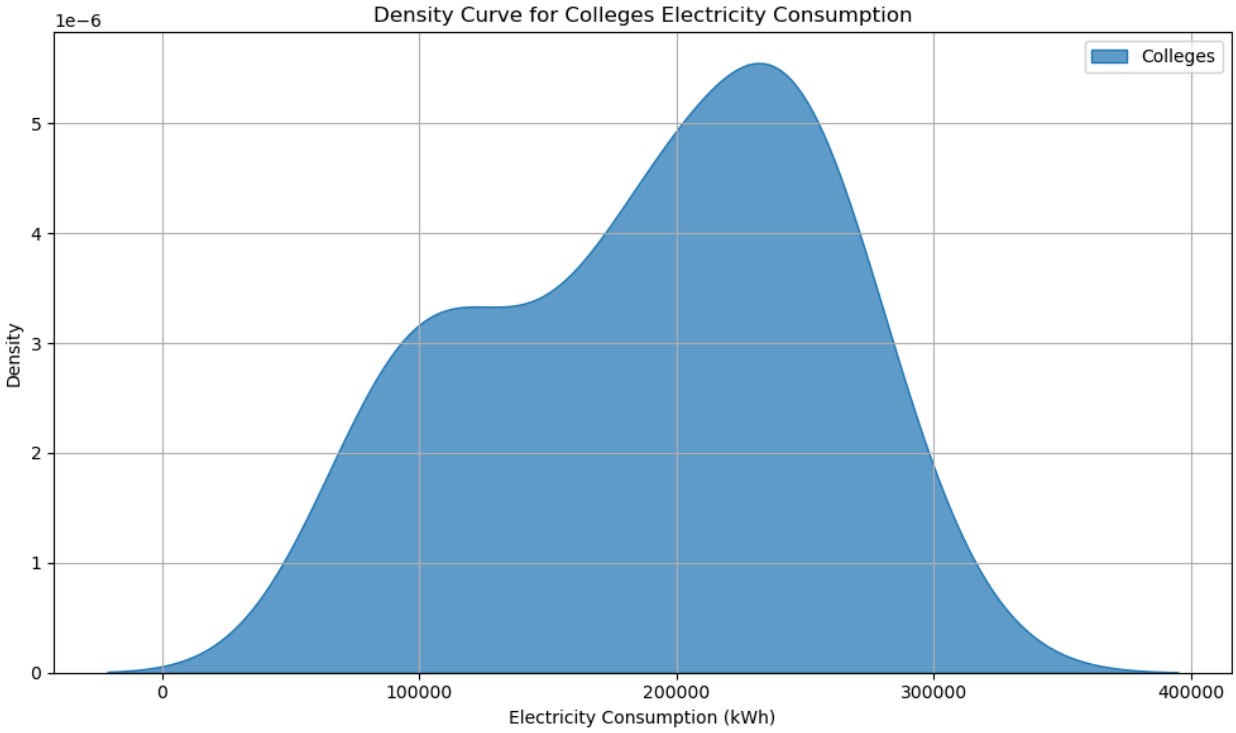
In [50]:

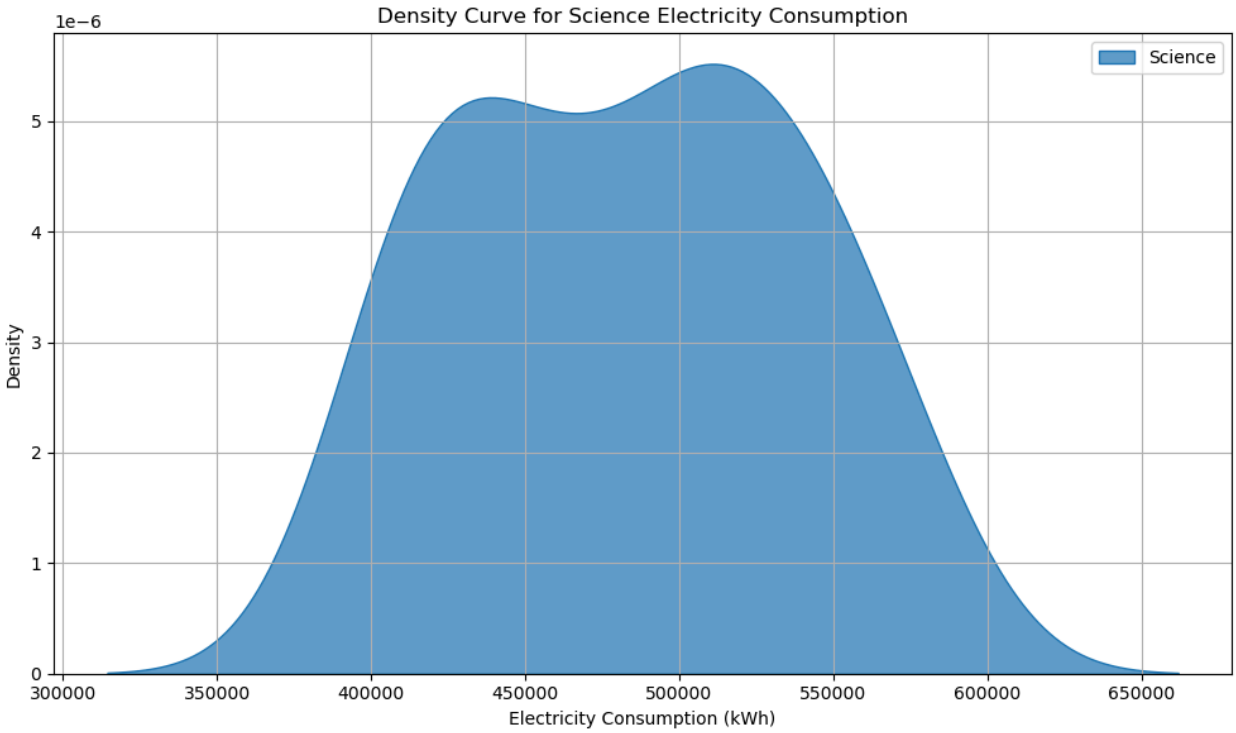
```python
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert 'Month' column to datetime and set as index
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')
data.set_index('Month', inplace=True)

# Select numeric columns (electricity consumption data)
consumption_data = data.select_dtypes(include=['number'])

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(consumption_data)

# Apply PCA to reduce dimensions to 2 components
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

# Create a DataFrame for PCA-transformed data
pca_df = pd.DataFrame(pca_data, columns=['PCA1', 'PCA2'])

# Apply DBSCAN clustering on the PCA-transformed data
dbscan = DBSCAN(eps=1.5, min_samples=5)
labels = dbscan.fit_predict(pca_df)

# Add cluster labels to the PCA DataFrame
pca_df['Cluster'] = labels

# Visualize the clustering results based on PCA
plt.figure(figsize=(10, 6))

# Plot each cluster with a different color
for cluster in pca_df['Cluster'].unique():
    cluster_data = pca_df[pca_df['Cluster'] == cluster]
    plt.scatter(cluster_data['PCA1'], cluster_data['PCA2'], label=f'Cluster {cluster}')

# Add plot details
plt.title('DBSCAN Clustering Results Based on PCA')
plt.xlabel('PCA1')
plt.ylabel('PCA2')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

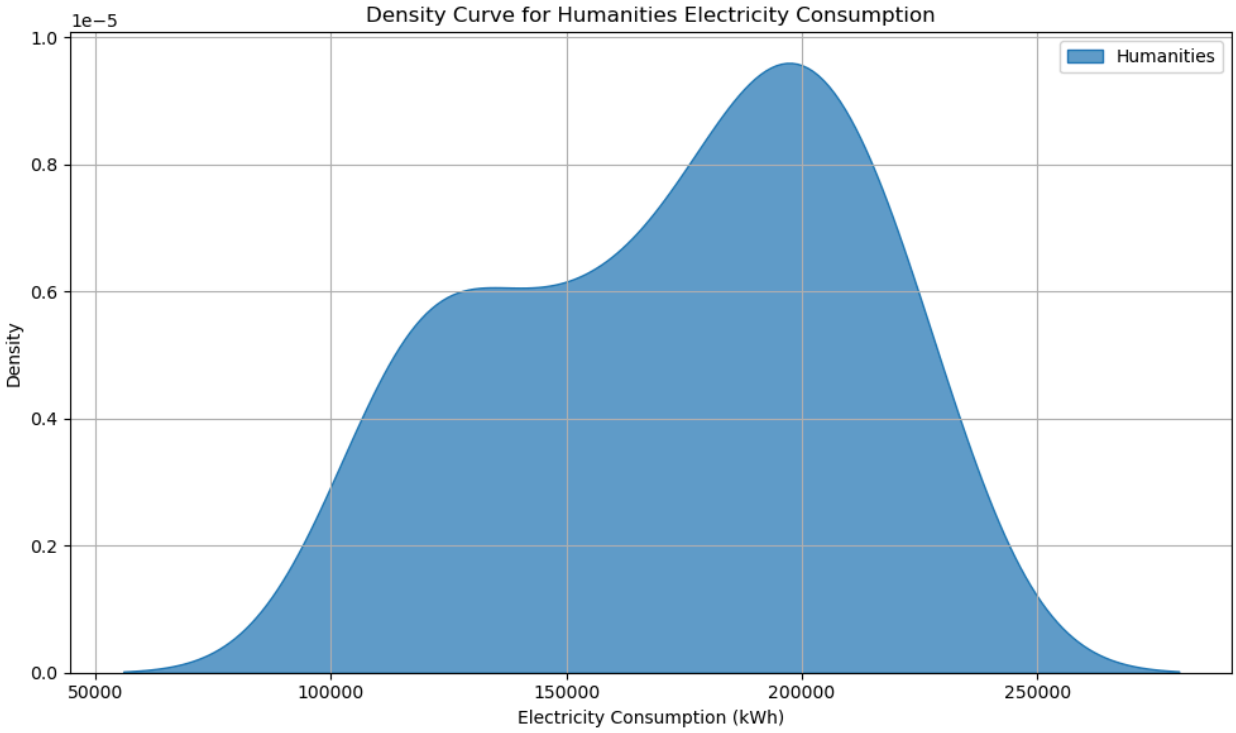


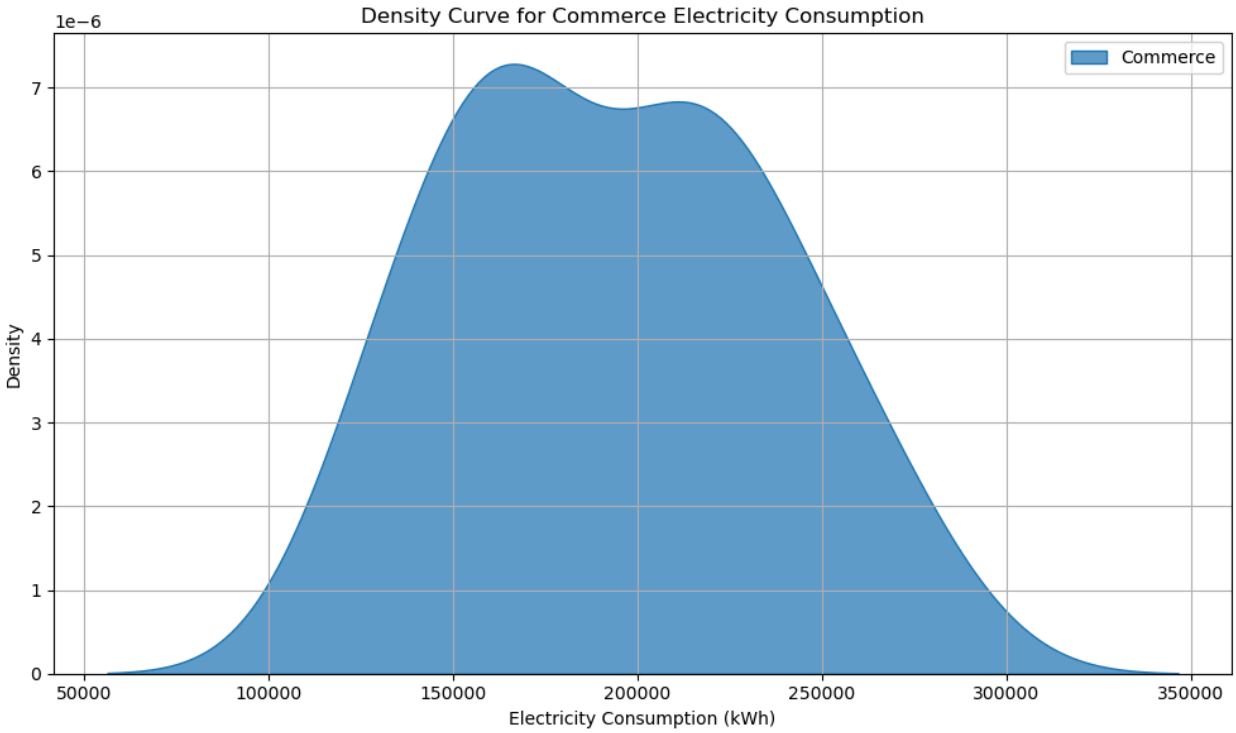
In [54]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert 'Month' column to datetime format
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')
data.set_index('Month', inplace=True)

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
                   "F518 Arts 1 Submains MSB:",
                   "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                   "F505 1 Richardson Mains:",
                   "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Plot density curves for each group
for group_name, buildings in building_groups.items():
    plt.figure(figsize=(10, 6))

    # Combine data from all buildings in the group
    group_data = data[buildings].sum(axis=1)

    # Plot density curve
    sns.kdeplot(group_data, fill=True, label=group_name, alpha=0.7)

    # Add plot details
    plt.title(f'Density Curve for {group_name} Electricity Consumption')
    plt.xlabel('Electricity Consumption (kWh)')
    plt.ylabel('Density')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

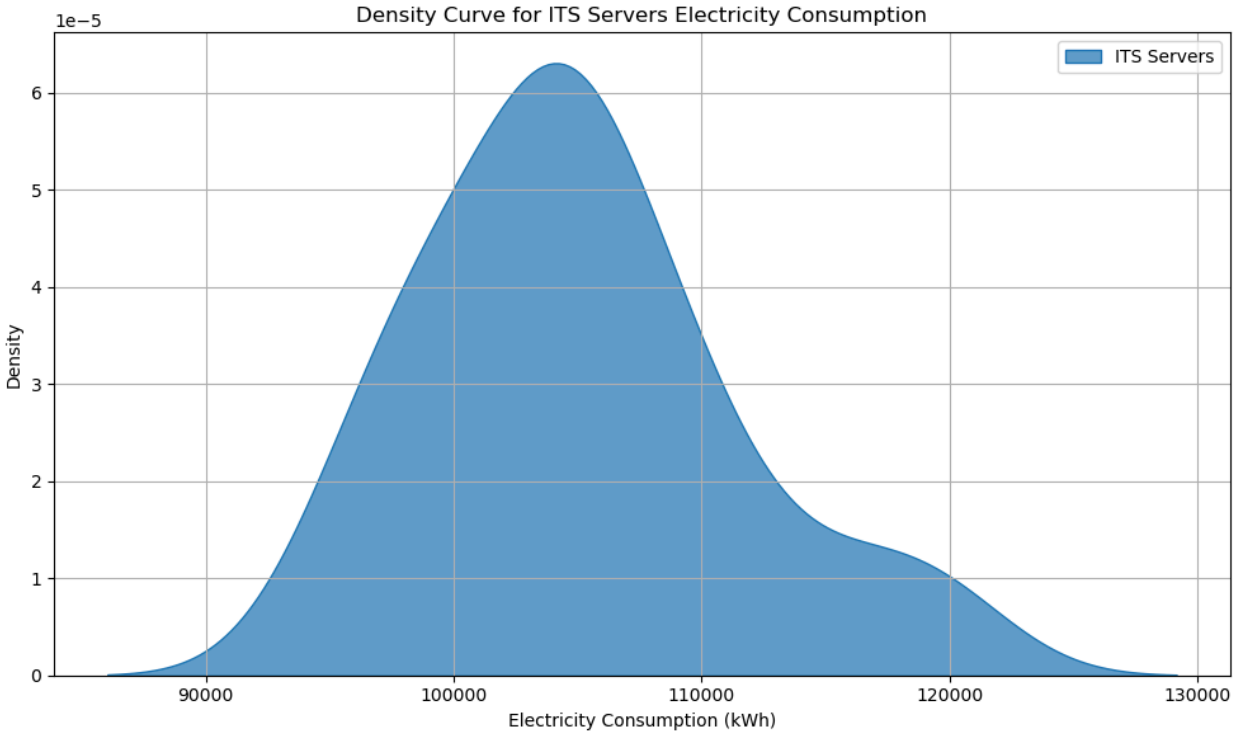

Density Curve for Libraries Electricity Consumption
1e−5
Libraries
Density
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
80000
100000
120000
140000
160000
180000
200000
Electricity Consumption (kWh)


Density Curve for Colleges Electricity Consumption
1e−6
Colleges
Density
5
4
3
2
1
0
0
100000
200000
300000
400000
Electricity Consumption (kWh)


Density Curve for Science Electricity Consumption
1e−6
Science
Density
0
1
2
3
4
5
300000
350000
400000
450000
500000
550000
600000
650000
Electricity Consumption (kWh)

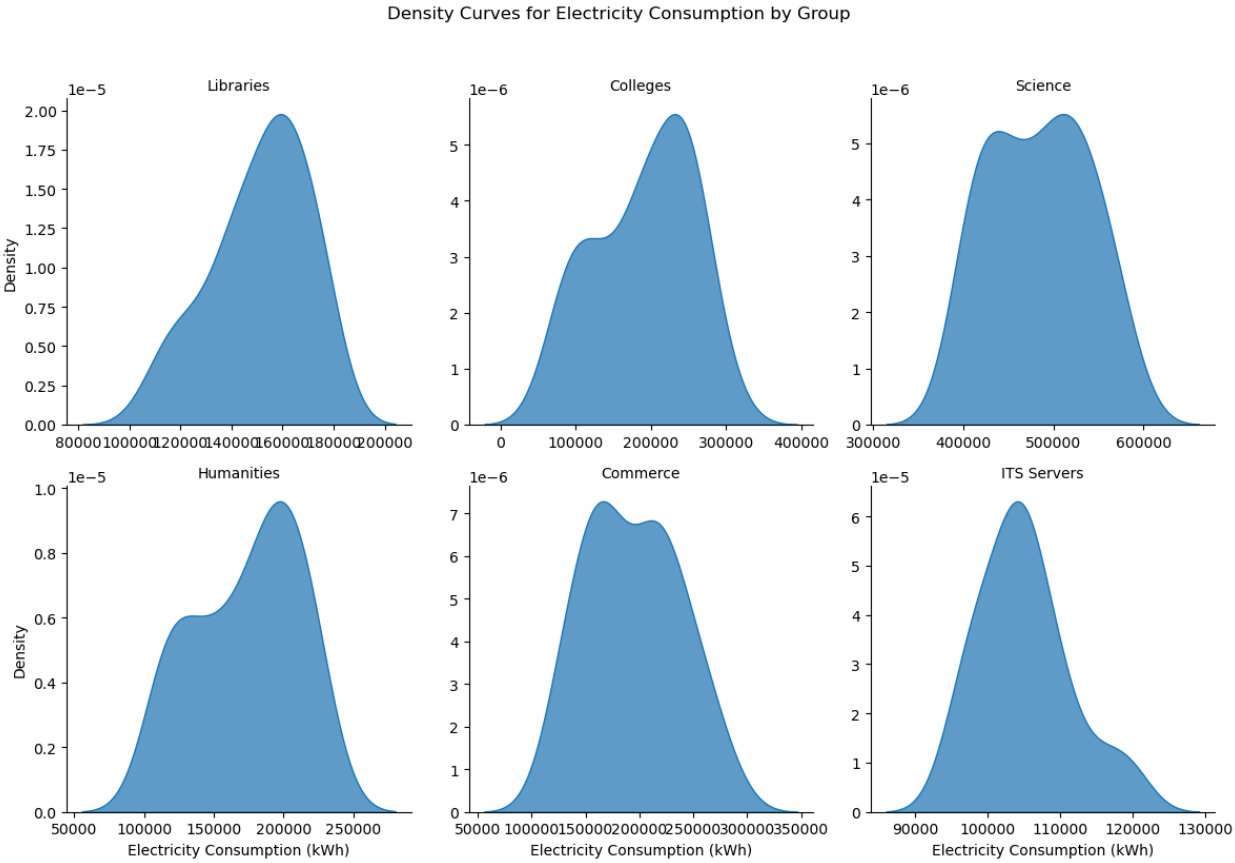

Density Curve for Humanities Electricity Consumption
1e−5
Humanities
Density
0.0
0.2
0.4
0.6
0.8
1.0
50000
100000
150000
200000
250000
Electricity Consumption (kWh)





In [60]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert 'Month' column to datetime format and set it as the index
data['Month'] = pd.to_datetime(data['Month'], format='%b-%y')
data.set_index('Month', inplace=True)

# Define building groups
building_groups = {
    'Libraries': ["F813 UOCOE Robertson Library - kWh", "F813 Bill Robertson Library 1 MSB:",
                  "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"],
    'Colleges': ["F711 Executive Residence - kWh", "F812 UOCOE Owheo Building - kWh",
                 "G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2",
                 "H633 Arana College main - kWh", "H71X Studholm College - kWh E2"],
    'Science': ["F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
                "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
                "G404 Microbiology - kWh", "G413 Science 2 - kWh",
                "G505 Geology north:", "G505 Geology south:"],
    'Humanities': ["F9XX College of Education main (Boiler room) - kWh",
```

```
                    "F518 Arts 1 Submains MSB:",
                    "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
                    "F505 1 Richardson Mains:",
                    "G506/07 Archway buildings (incl. Allen & Marama Hall):"],
    'Commerce': ["F614 1 School of Business Incomer 1 (Lower floors):",
                 "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1  Psychology substation- Goddard:"],
    'ITS Servers': ["F204 444 Great King Street - kWh"]
}

# Combine data for each group and prepare for plotting
group_data = []
for group_name, buildings in building_groups.items():
    total_consumption = data[buildings].sum(axis=1)
    group_data.append(pd.DataFrame({'Consumption': total_consumption, 'Group': group_name}))

# Concatenate all group data into a single DataFrame
all_groups_data = pd.concat(group_data)

# Create a FacetGrid for density plots
g = sns.FacetGrid(all_groups_data, col="Group", col_wrap=3, height=4, sharex=False, sharey=False)
g.map(sns.kdeplot, 'Consumption', fill=True, alpha=0.7)

# Add plot details
g.set_titles("{col_name}")
g.set_axis_labels("Electricity Consumption (kWh)", "Density")
g.fig.suptitle('Density Curves for Electricity Consumption by Group', y=1.03)
plt.tight_layout()

# Show the plot
plt.show()
```



In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('stream_data_filtered - Copy.csv')

# Target column for anomaly detection
target_column = 'F204 444 Great King Street - kWh'

# Calculate mean and standard deviation for the target column
mean = df[target_column].mean()
std = df[target_column].std()

# Define the threshold for anomalies (Mean + 2 Sigma)
threshold = mean + 2 * std

# Identify anomalies
df['is_anomaly'] = df[target_column] > threshold
```

```python
# Print summary of anomalies
print(f"Total data points: {len(df)}")
print(f"Number of anomalies: {df['is_anomaly'].sum()}")
print(f"Percentage of anomalies: {(df['is_anomaly'].sum() / len(df)) * 100:.2f}%")

# Print the anomalies
anomalies = df[df['is_anomaly']]
print("\nAnomalies Detected:")
print(anomalies[['Month', target_column]])
```

```
Total data points: 34
Number of anomalies: 2
Percentage of anomalies: 5.88%

Anomalies Detected:
     Month  F204 444 Great King Street - kWh
14  Mar-23                         119858.94
16  May-23                         119288.30
```

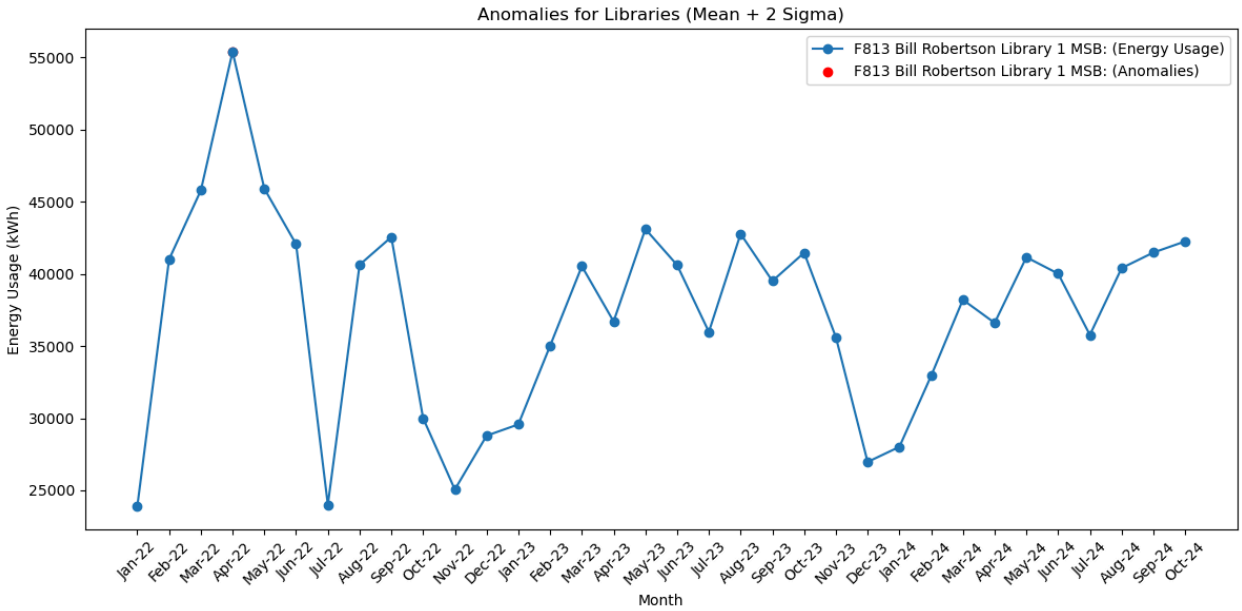
In [ ]:

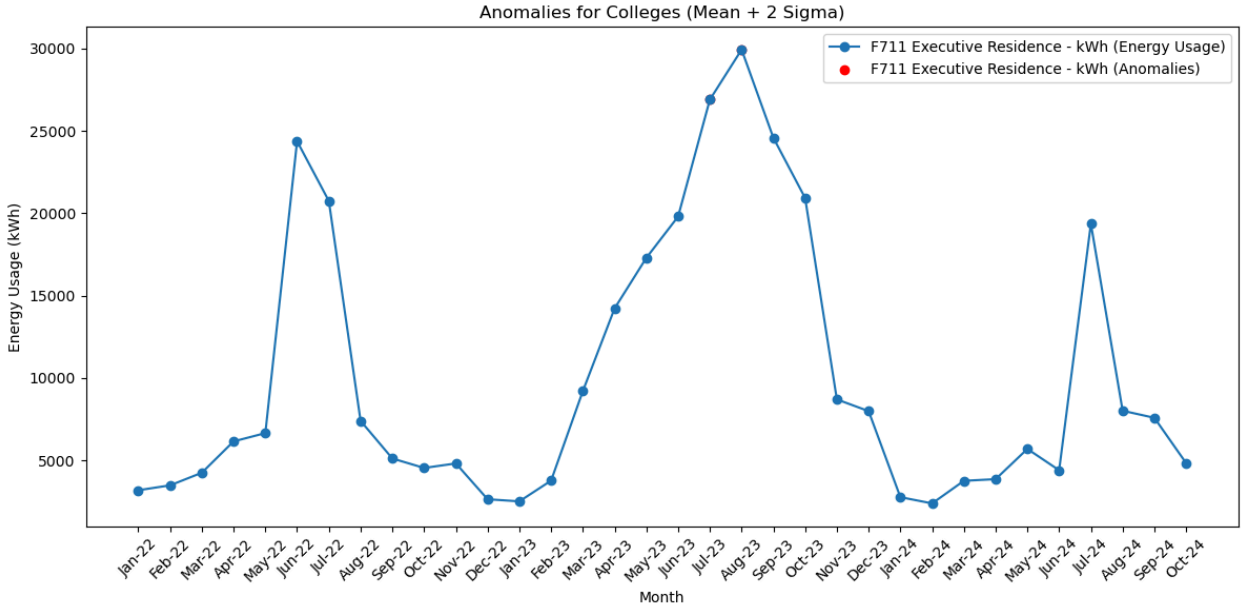
```python
# Visualize the data and anomalies
plt.figure(figsize=(12, 6))
plt.plot(df['Month'], df[target_column], label='Energy Usage', marker='o')
plt.axhline(y=threshold, color='r', linestyle='--', label='Threshold (Mean + 2 Sigma)')
plt.scatter(anomalies['Month'], anomalies[target_column], color='red', label='Anomalies', zorder=5)
plt.xlabel('Month')
plt.ylabel('Energy Usage (kWh)')
plt.title('Energy Usage Anomalies Detection (Mean + 2 Sigma Method)')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()

# Save anomalies to a new CSV file for further analysis
anomalies.to_csv('anomalies_detected.csv', index=False)
```

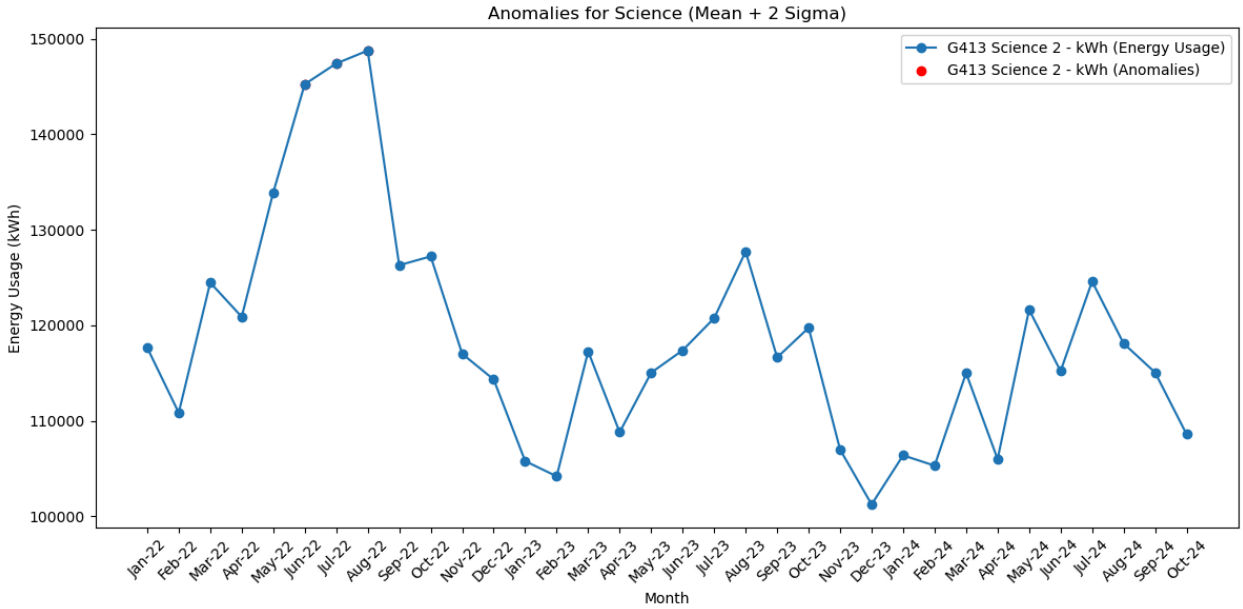
In [82]:

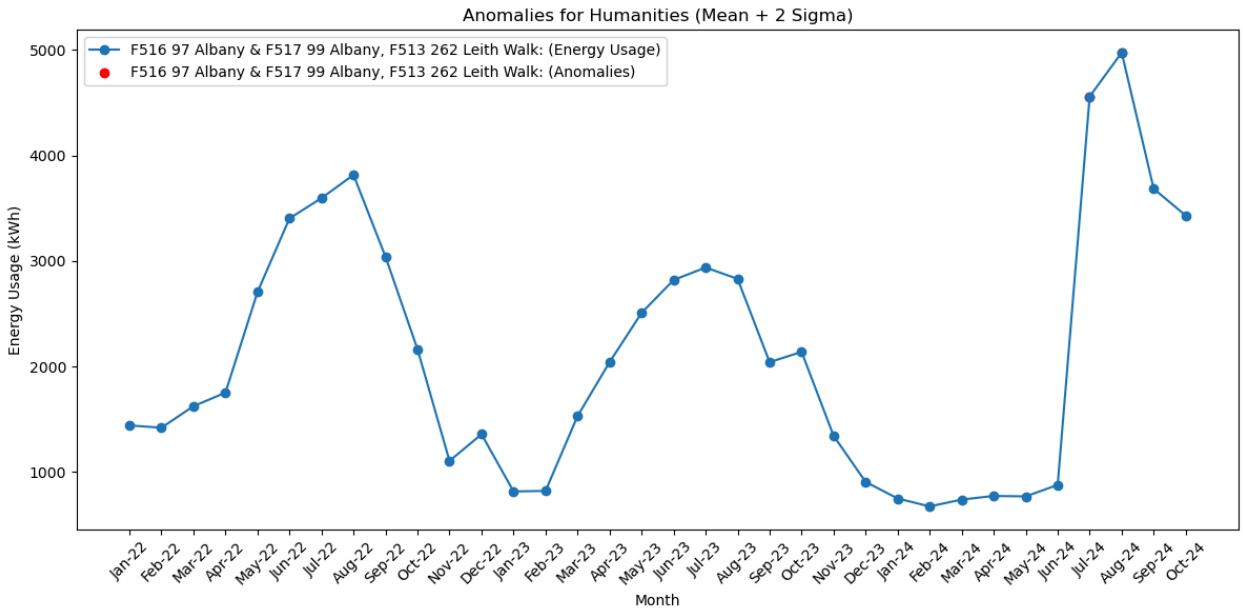
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('stream_data_filtered - copy.csv')

# Define groups and their respective columns
groups = {
    "Libraries": [
        "Libraries Total - kWh",
        "E902 Hocken Library - kWh",
        "F813 UOCOE Robertson Library - kWh",
        "D203 Sayers (at Adams MSB):",
        "F813 Bill Robertson Library 1 MSB:",
        "F419 ISB West Excluding Shops",
        "F505 Richardson Library Block Rising Main:"
    ],
    "Colleges": [
        "Colleges Total - kWh",
        "C405 192 Castle College - kWh",
        "D402 Hayward College - kWh",
        "D40X Cumberland College - kWh",
        "F711 Executive Residence - kWh",
        "F812 UOCOE Owheo Building - kWh",
        "G608 St Margarets College - kWh",
        "H41X Selwyn College - kWh E2",
        "H633 Arana College main - kWh",
        "H71X Studholm College - kWh E2",
        "J126 Carrington College (Kitchen/Dining) - kWh",
        "J14X Aquinas College - kWh",
        "J303 Caroline Freeman College - kWh",
        "K427 Abbey College - kWh"
    ],
    "Science": [
        "Science Total - kWh",
        "D403 Survey & Marine - kWh",
        "E212 Zoology Buildings - kWh",
        "F315 Botany Tin Hut - kWh",
        "F325 Physical Education - kWh",
        "F812 UOCOE Owheo Building - kWh",
        "G401 Mellor Laboratories - kWh",
        "G404 Microbiology - kWh",
        "G413 Science 2 - kWh",
        "J960 Portobello Marine Lab - kWh",
        "G505 Geology north:",
        "G505 Geology south:"
    ],
    "Humanities": [
        "Humanities Total - kWh",
        "F9XX College of Education main (Boiler room) - kWh",
        "F518 Arts 1 Submains MSB:",
        "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
        "F505 1 Richardson Mains:",
```

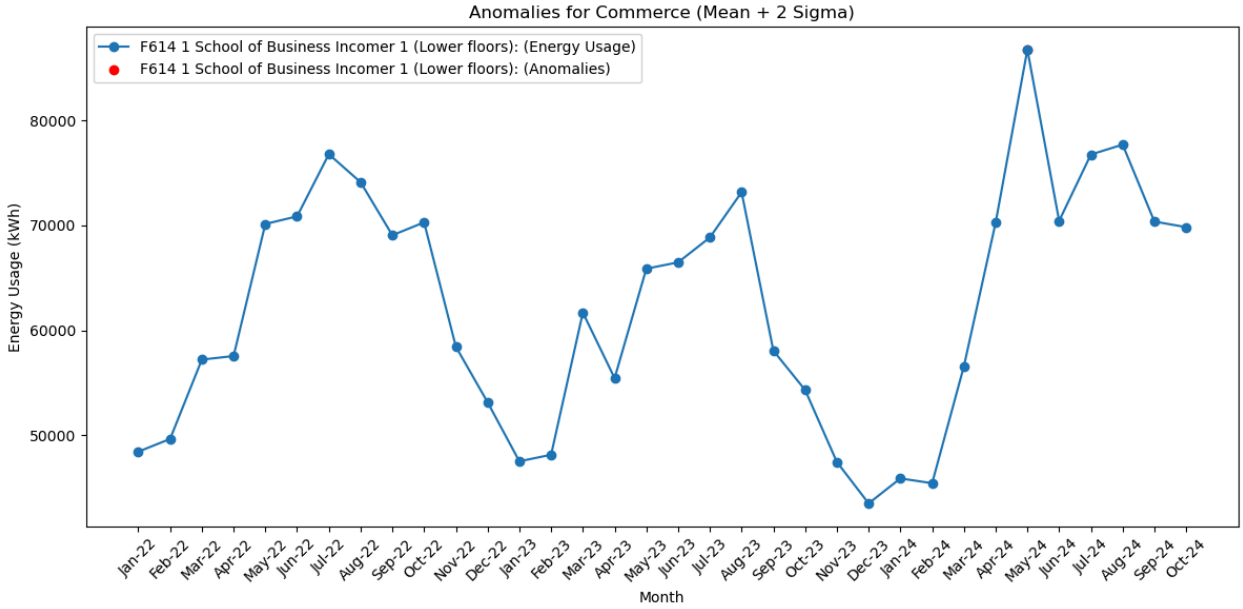
```
        "G506/07 Archway buildings (incl. Allen & Marama Hall):"
    ],
    "Commerce": [
        "Commerce Total - kWh",
        "F614 1 School of Business Incomer 1 (Lower floors):",
        "F614 2 School of Business Incomer 2 (Upper floors):",
        "F618 1  Psychology substation- Goddard:"
    ],
    "ITS Servers": [
        "ITS Servers Total - kWh",
        "F204 444 Great King Street - kWh"
    ]
}

# Initialize a dictionary to store anomalies for each group
anomalies_dict = {}

# Iterate through each group
for group, columns in groups.items():
    group_anomalies = []
    for column in columns:
        if column in df.columns:
            # Calculate mean and standard deviation
            mean = df[column].mean()
            std = df[column].std()

            # Define the threshold for anomalies (Mean + 2 Sigma)
            threshold = mean + 2 * std

            # Identify anomalies
            anomalies = df[df[column] > threshold][["Month", column]]
            group_anomalies.append((column, anomalies))

    anomalies_dict[group] = group_anomalies

# Plot anomalies for each group
for group, anomalies in anomalies_dict.items():
    plt.figure(figsize=(12, 6))
    for column, anomaly_data in anomalies:
        if not anomaly_data.empty:
            plt.plot(df['Month'], df[column], label=f'{column} (Energy Usage)', marker='o')
            plt.scatter(anomaly_data['Month'], anomaly_data[column], label=f'{column} (Anomalies)', color='red')
    plt.title(f"Anomalies for {group} (Mean + 2 Sigma)")
    plt.xlabel("Month")
    plt.ylabel("Energy Usage (kWh)")
    plt.legend()
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

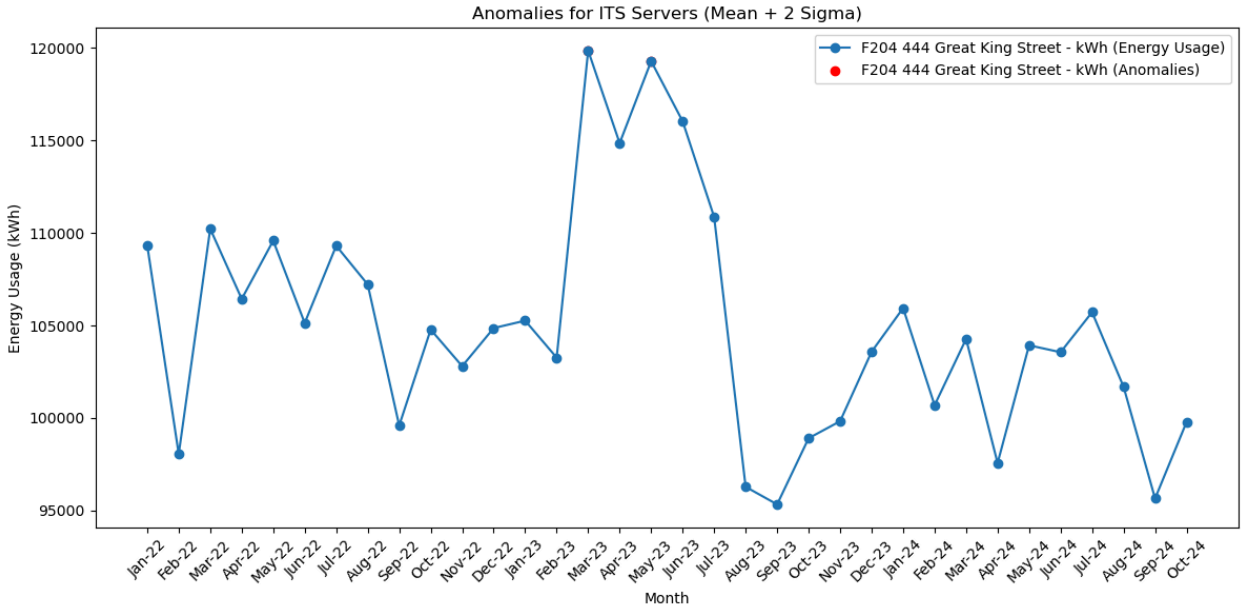



Anomalies for Colleges (Mean + 2 Sigma)
F711 Executive Residence - kWh (Energy Usage)
F711 Executive Residence - kWh (Anomalies)
Energy Usage (kWh)
Month


Anomalies for Science (Mean + 2 Sigma)
G413 Science 2 - kWh (Energy Usage)
G413 Science 2 - kWh (Anomalies)
Energy Usage (kWh)
Month


Anomalies for Humanities (Mean + 2 Sigma)
F516 97 Albany & F517 99 Albany, F513 262 Leith Walk: (Energy Usage)
F516 97 Albany & F517 99 Albany, F513 262 Leith Walk: (Anomalies)
Energy Usage (kWh)
Month

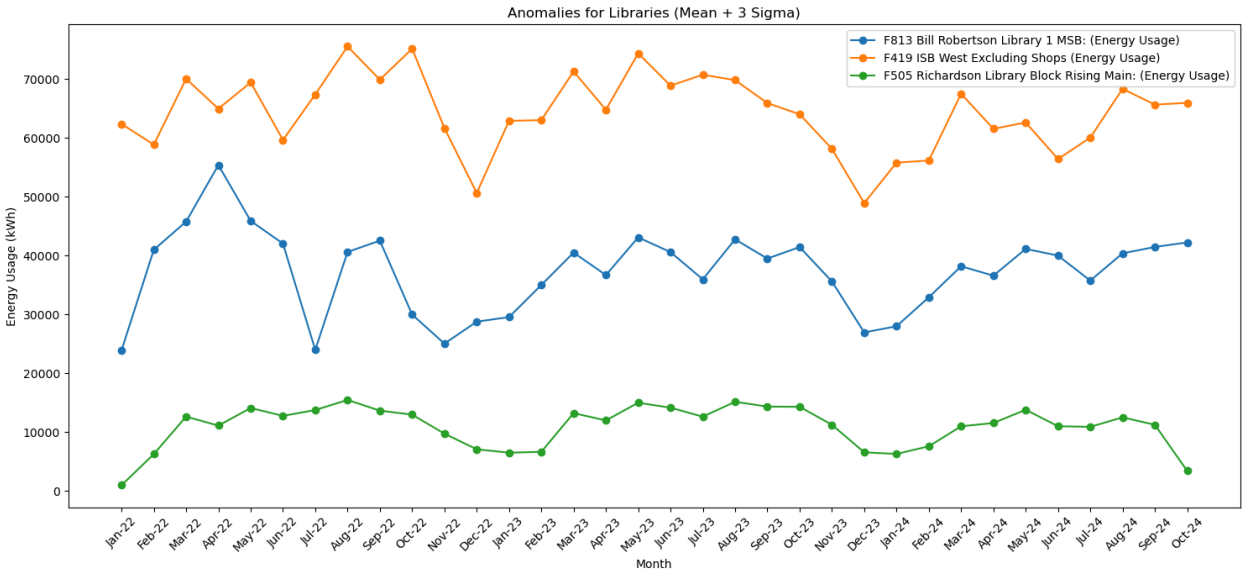


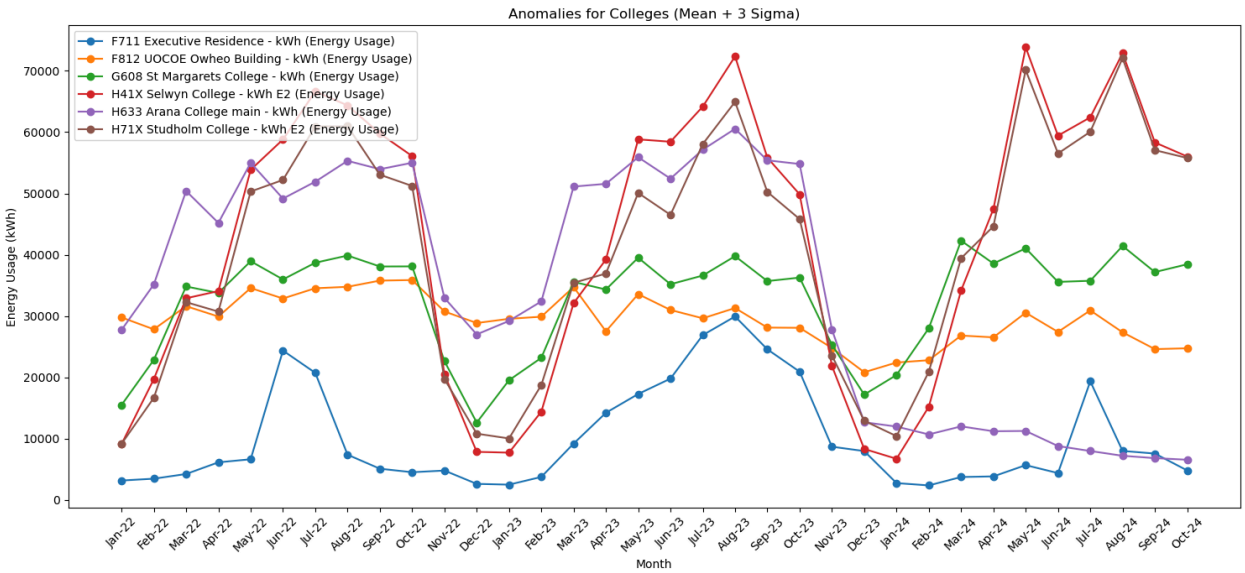


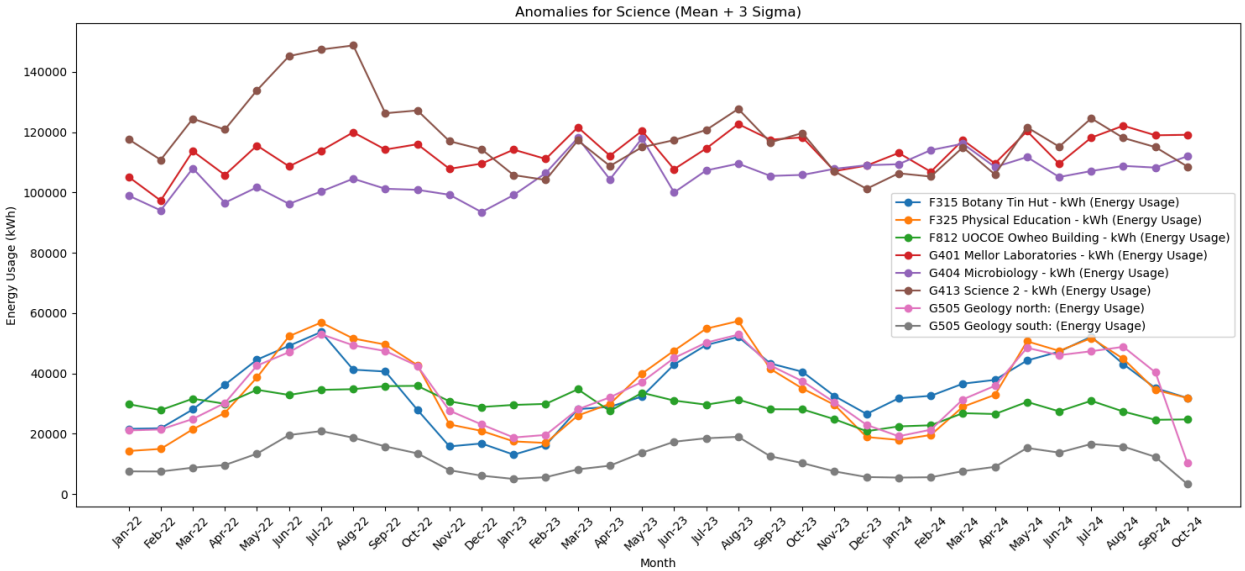
In [290…

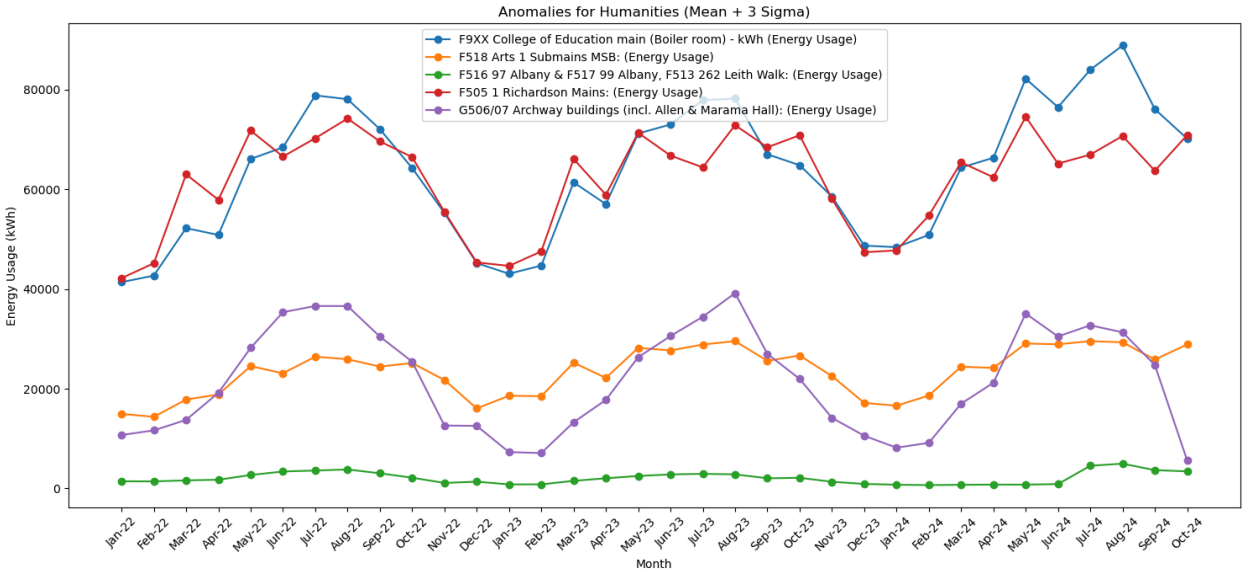
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("stream_data_filtered - Copy.csv")

# Define groups and their respective columns
groups = {
    "Libraries": [
        "F813 Bill Robertson Library 1 MSB:",
        "F419 ISB West Excluding Shops",
        "F505 Richardson Library Block Rising Main:",
    ],
    "Colleges": [
        "F711 Executive Residence - kWh",
        "F812 UOCOE Owheo Building - kWh",
        "G608 St Margarets College - kWh",
        "H41X Selwyn College - kWh E2",
        "H633 Arana College main - kWh",
        "H71X Studholm College - kWh E2",
    ],
    "Science": [
        "F315 Botany Tin Hut - kWh",
        "F325 Physical Education - kWh",
        "F812 UOCOE Owheo Building - kWh",
        "G401 Mellor Laboratories - kWh",
        "G404 Microbiology - kWh",
        "G413 Science 2 - kWh",
        "G505 Geology north:",
        "G505 Geology south:",
    ],
    "Humanities": [
        "F9XX College of Education main (Boiler room) - kWh",
```

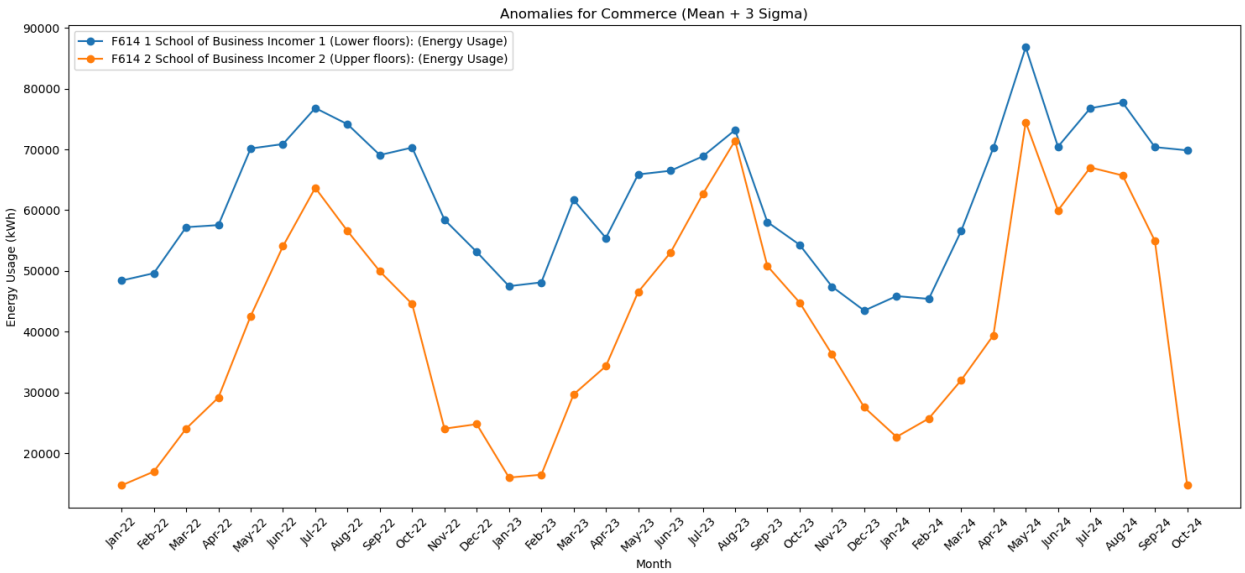
```
        "F518 Arts 1 Submains MSB:",
        "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
        "F505 1 Richardson Mains:",
        "G506/07 Archway buildings (incl. Allen & Marama Hall):",
    ],
    "Commerce": [
        "F614 1 School of Business Incomer 1 (Lower floors):",
        "F614 2 School of Business Incomer 2 (Upper floors):",
        "F618 1  Psychology substation- Goddard:",
    ],
    "ITS Servers": ["F204 444 Great King Street - kWh"],
}


# Iterate through each group
for group, columns in groups.items():
    plt.figure(figsize=(15, 7))  # Create a single figure for each group
    plt.title(f"Anomalies for {group} (Mean + 3 Sigma)")
    plt.xlabel("Month")
    plt.ylabel("Energy Usage (kWh)")
    plt.xticks(rotation=45)  # Rotate x-axis labels

    for column in columns:
        if column in df.columns:
            # Calculate mean and standard deviation
            mean = df[column].mean()
            std = df[column].std()

            # Define the threshold for anomalies (Mean + 3 Sigma)
            threshold = mean + 3 * std

            # Identify anomalies
            anomalies = df[df[column] > threshold][["Month", column]]

            # Plot the energy usage
            plt.plot(
                df["Month"], df[column], label=f"{column} (Energy Usage)", marker="o"
            )

            # Plot anomalies but only add legend entry once
            if not anomalies.empty:
                plt.scatter(
                    anomalies["Month"],
                    anomalies[column],
                    label=f"Anomalies",
                    color="red",
                    s=50,
                )
            # Remove the duplicate legend entries for anomalies by only plotting 'Anomalies' once
            # The check is outside this loop to make sure all column have values plotted already,

    handles, labels = plt.gca().get_legend_handles_labels()
    # Create a new list of handles and labels without duplicates
    new_handles = []
    new_labels = []
    for handle, label in zip(handles, labels):
        if label not in new_labels:
            new_handles.append(handle)
            new_labels.append(label)

    plt.legend(new_handles, new_labels)
    plt.tight_layout()
    plt.show()
```

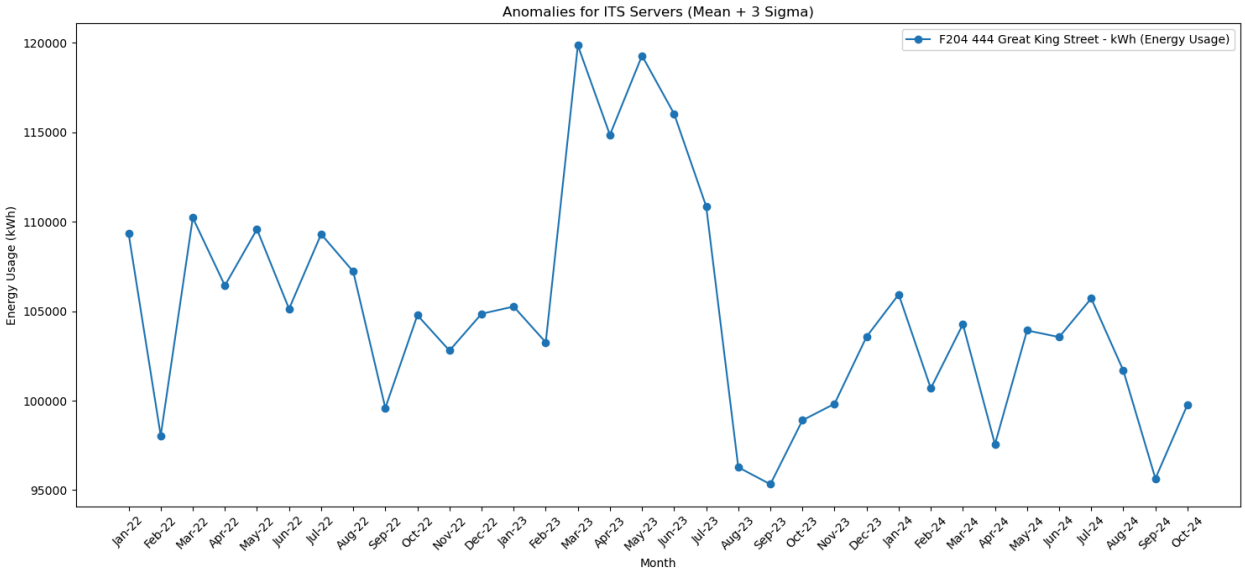



Anomalies for Colleges (Mean + 3 Sigma)
F711 Executive Residence - kWh (Energy Usage)
F812 UOCOE Owheo Building - kWh (Energy Usage)
G608 St Margarets College - kWh (Energy Usage)
H41X Selwyn College - kWh E2 (Energy Usage)
H633 Arana College main - kWh (Energy Usage)
H71X Studholm College - kWh E2 (Energy Usage)
Energy Usage (kWh)
Month


Anomalies for Science (Mean + 3 Sigma)
F315 Botany Tin Hut - kWh (Energy Usage)
F325 Physical Education - kWh (Energy Usage)
F812 UOCOE Owheo Building - kWh (Energy Usage)
G401 Mellor Laboratories - kWh (Energy Usage)
G404 Microbiology - kWh (Energy Usage)
G413 Science 2 - kWh (Energy Usage)
G505 Geology north: (Energy Usage)
G505 Geology south: (Energy Usage)
Energy Usage (kWh)
Month


Anomalies for Humanities (Mean + 3 Sigma)
F9XX College of Education main (Boiler room) - kWh (Energy Usage)
F518 Arts 1 Submains MSB: (Energy Usage)
F516 97 Albany & F517 99 Albany, F513 262 Leith Walk: (Energy Usage)
F505 1 Richardson Mains: (Energy Usage)
G506/07 Archway buildings (incl. Allen & Marama Hall): (Energy Usage)
Energy Usage (kWh)
Month

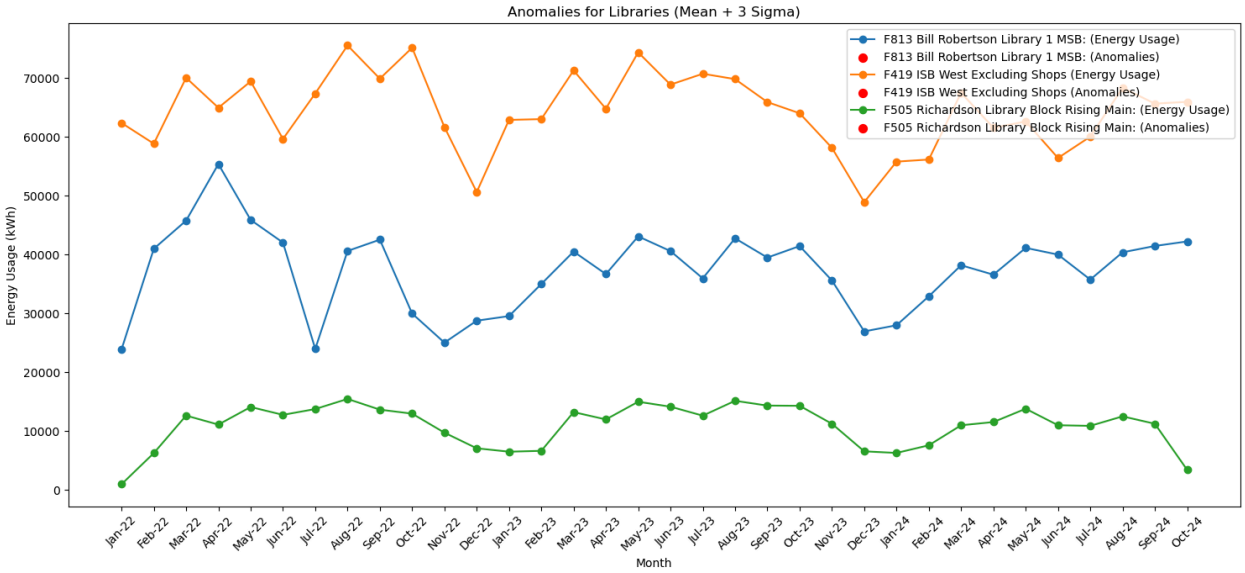


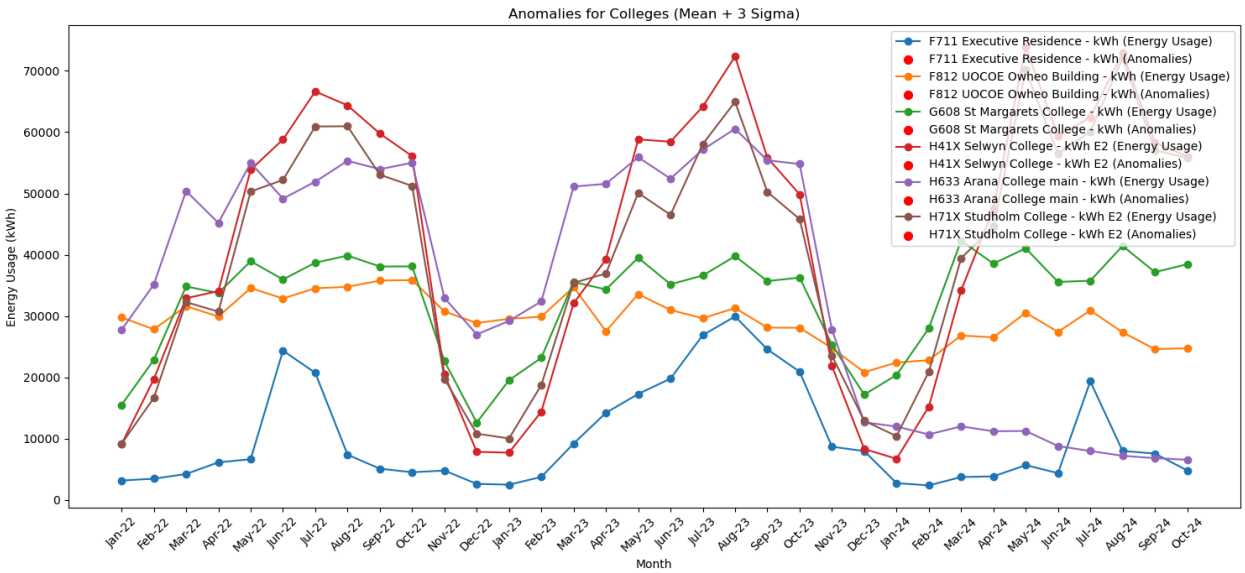


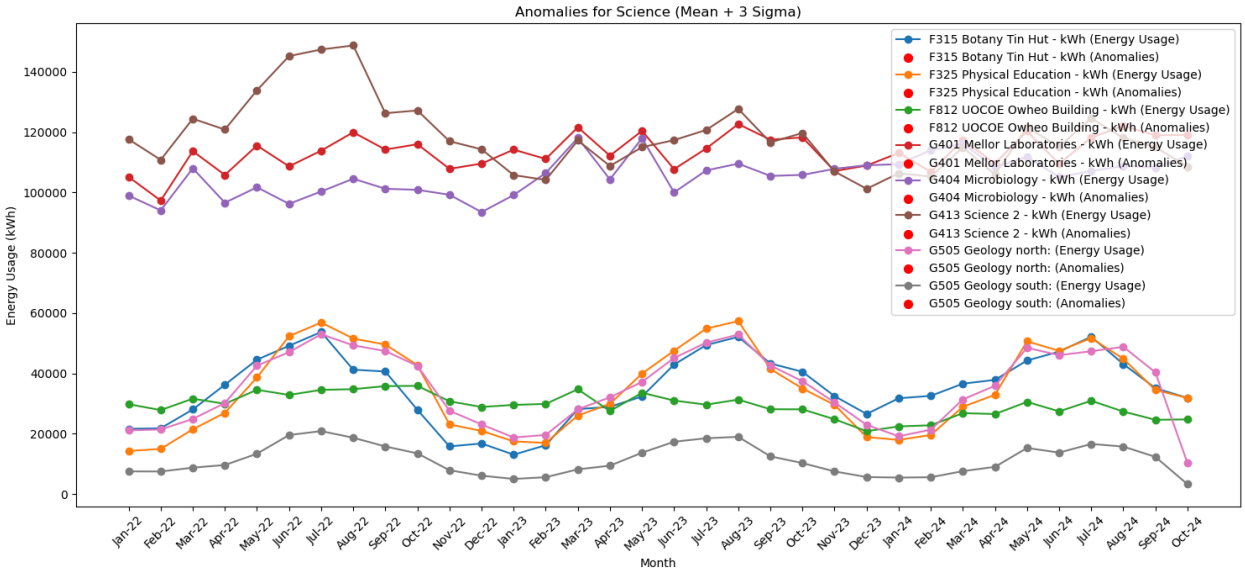
In [288…

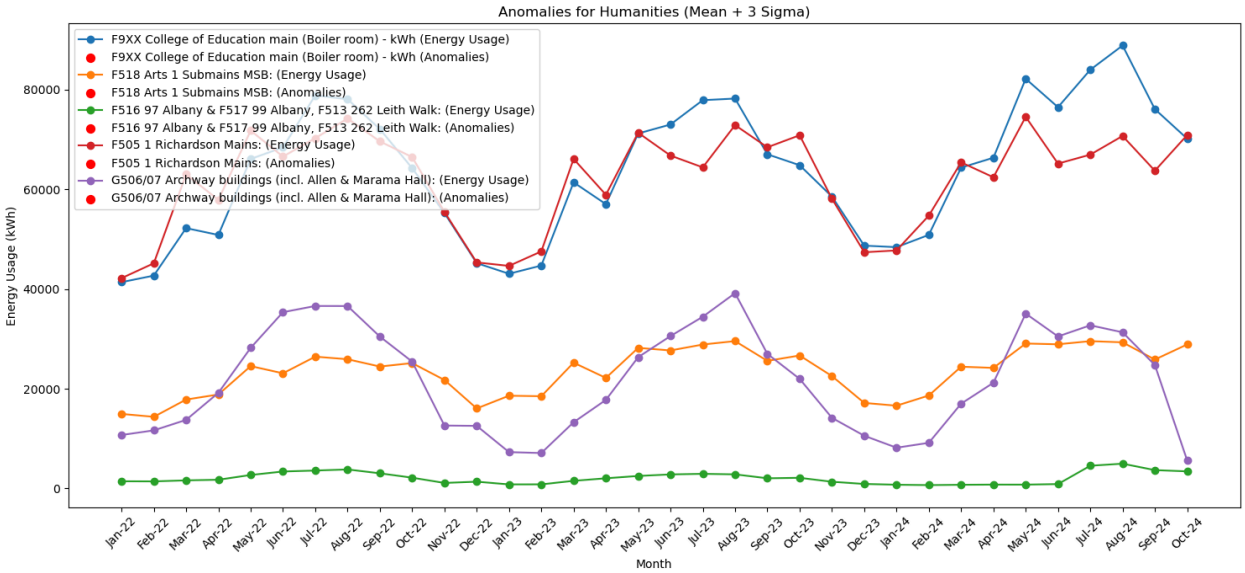
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("stream_data_filtered - Copy.csv")

# Define groups and their respective columns
groups = {
    "Libraries": [
        "F813 Bill Robertson Library 1 MSB:",
        "F419 ISB West Excluding Shops",
        "F505 Richardson Library Block Rising Main:",
    ],
    "Colleges": [
        "F711 Executive Residence - kWh",
        "F812 UOCOE Owheo Building - kWh",
        "G608 St Margarets College - kWh",
        "H41X Selwyn College - kWh E2",
        "H633 Arana College main - kWh",
        "H71X Studholm College - kWh E2",
    ],
    "Science": [
        "F315 Botany Tin Hut - kWh",
        "F325 Physical Education - kWh",
        "F812 UOCOE Owheo Building - kWh",
        "G401 Mellor Laboratories - kWh",
        "G404 Microbiology - kWh",
        "G413 Science 2 - kWh",
        "G505 Geology north:",
        "G505 Geology south:",
    ],
    "Humanities": [
        "F9XX College of Education main (Boiler room) - kWh",
        "F518 Arts 1 Submains MSB:",
        "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
        "F505 1 Richardson Mains:",
        "G506/07 Archway buildings (incl. Allen & Marama Hall):",
```

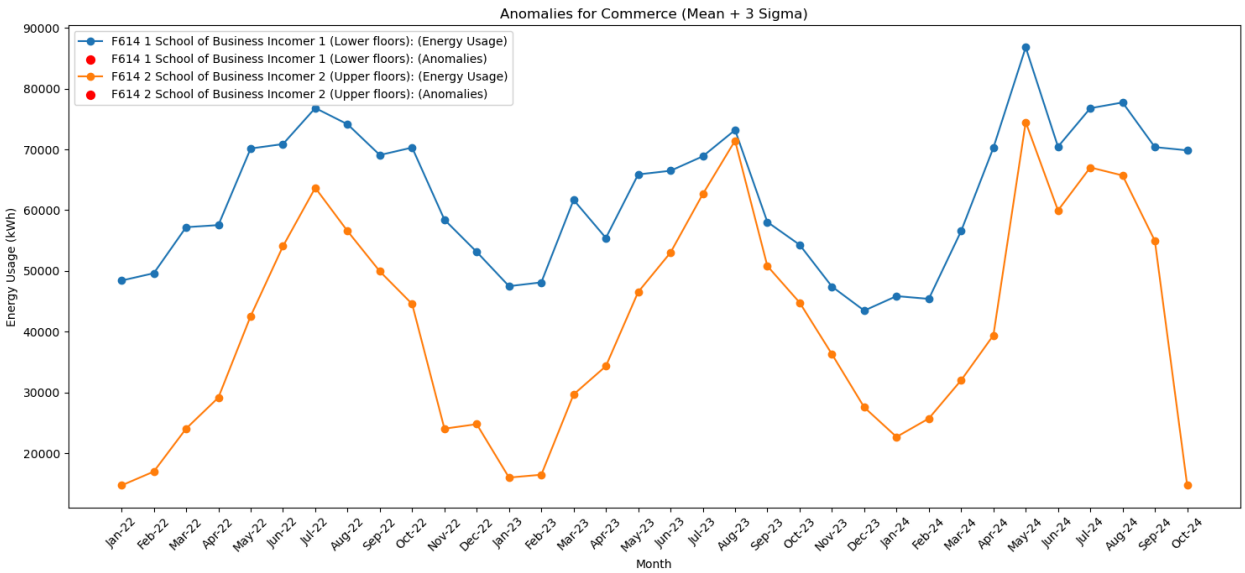
```
    ],
    "Commerce": [
        "F614 1 School of Business Incomer 1 (Lower floors):",
        "F614 2 School of Business Incomer 2 (Upper floors):",
        "F618 1  Psychology substation- Goddard:",
    ],
    "ITS Servers": ["F204 444 Great King Street - kWh"],
}


# Iterate through each group
for group, columns in groups.items():
    plt.figure(figsize=(15, 7))  # Create a single figure for each group
    plt.title(f"Anomalies for {group} (Mean + 3 Sigma)")
    plt.xlabel("Month")
    plt.ylabel("Energy Usage (kWh)")
    plt.xticks(rotation=45)  # Rotate x-axis labels

    for column in columns:
        if column in df.columns:
            # Calculate mean and standard deviation
            mean = df[column].mean()
            std = df[column].std()

            # Define the threshold for anomalies (Mean + 3 Sigma)
            threshold = mean + 3 * std

            # Identify anomalies
            anomalies = df[df[column] > threshold][["Month", column]]

            # Plot the energy usage and highlight anomalies
            plt.plot(
                df["Month"], df[column], label=f"{column} (Energy Usage)", marker="o"
            )
            plt.scatter(
                anomalies["Month"], anomalies[column], label=f"{column} (Anomalies)", color="red", s=50
            )  # Highlight anomalies

    plt.legend()
    plt.tight_layout()
    plt.show()
```

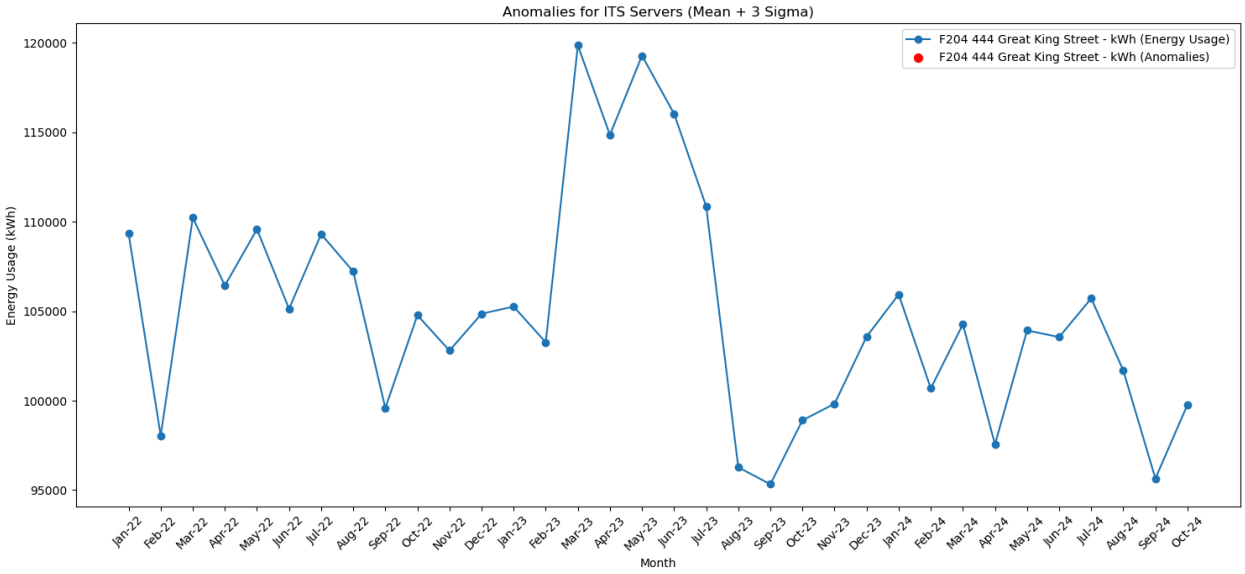



Anomalies for Colleges (Mean + 3 Sigma)
F711 Executive Residence - kWh (Energy Usage)
F711 Executive Residence - kWh (Anomalies)
F812 UOCOE Owheo Building - kWh (Energy Usage)
F812 UOCOE Owheo Building - kWh (Anomalies)
G608 St Margarets College - kWh (Energy Usage)
G608 St Margarets College - kWh (Anomalies)
H41X Selwyn College - kWh E2 (Energy Usage)
H41X Selwyn College - kWh E2 (Anomalies)
H633 Arana College main - kWh (Energy Usage)
H633 Arana College main - kWh (Anomalies)
H71X Studholm College - kWh E2 (Energy Usage)
H71X Studholm College - kWh E2 (Anomalies)
Energy Usage (kWh)
Month


Anomalies for Science (Mean + 3 Sigma)
F315 Botany Tin Hut - kWh (Energy Usage)
F315 Botany Tin Hut - kWh (Anomalies)
F325 Physical Education - kWh (Energy Usage)
F325 Physical Education - kWh (Anomalies)
F812 UOCOE Owheo Building - kWh (Energy Usage)
F812 UOCOE Owheo Building - kWh (Anomalies)
G401 Mellor Laboratories - kWh (Energy Usage)
G401 Mellor Laboratories - kWh (Anomalies)
G404 Microbiology - kWh (Energy Usage)
G404 Microbiology - kWh (Anomalies)
G413 Science 2 - kWh (Energy Usage)
G413 Science 2 - kWh (Anomalies)
G505 Geology north: (Energy Usage)
G505 Geology north: (Anomalies)
G505 Geology south: (Energy Usage)
G505 Geology south: (Anomalies)
Energy Usage (kWh)
Month

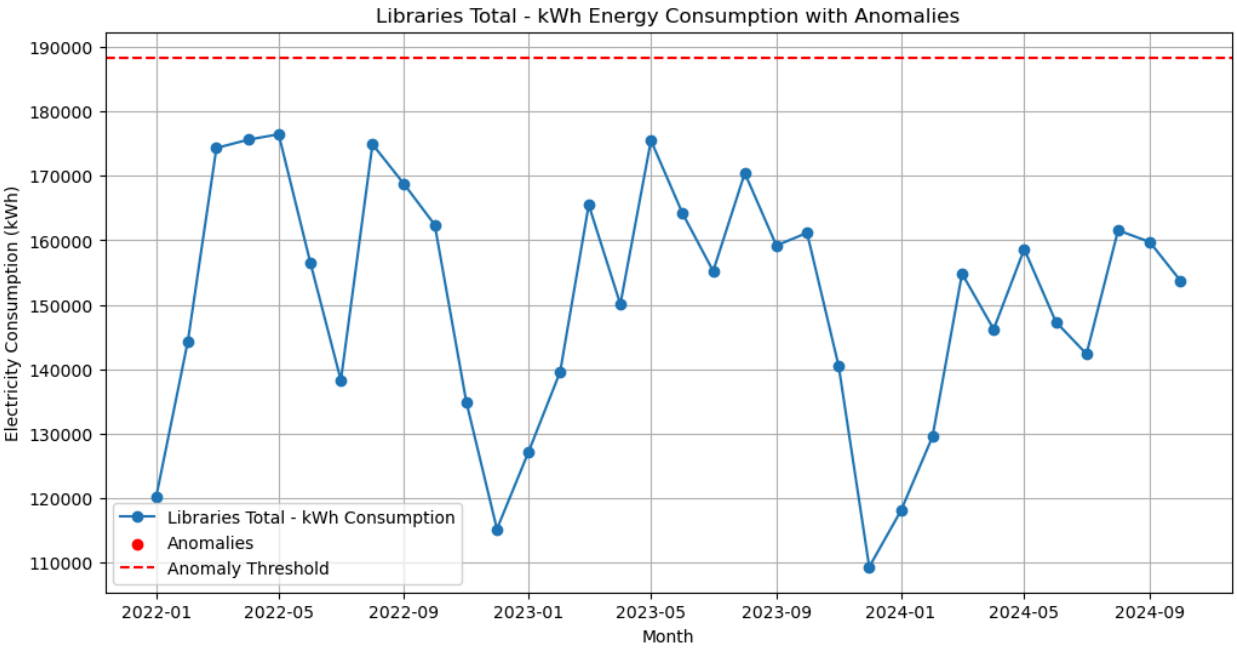

Anomalies for Humanities (Mean + 3 Sigma)
F9XX College of Education main (Boiler room) - kWh (Energy Usage)
F9XX College of Education main (Boiler room) - kWh (Anomalies)
F518 Arts 1 Submains MSB: (Energy Usage)
F518 Arts 1 Submains MSB: (Anomalies)
F516 97 Albany & F517 99 Albany, F513 262 Leith Walk: (Energy Usage)
F516 97 Albany & F517 99 Albany, F513 262 Leith Walk: (Anomalies)
F505 1 Richardson Mains: (Energy Usage)
F505 1 Richardson Mains: (Anomalies)
G506/07 Archway buildings (incl. Allen & Marama Hall): (Energy Usage)
G506/07 Archway buildings (incl. Allen & Marama Hall): (Anomalies)
Energy Usage (kWh)
Month

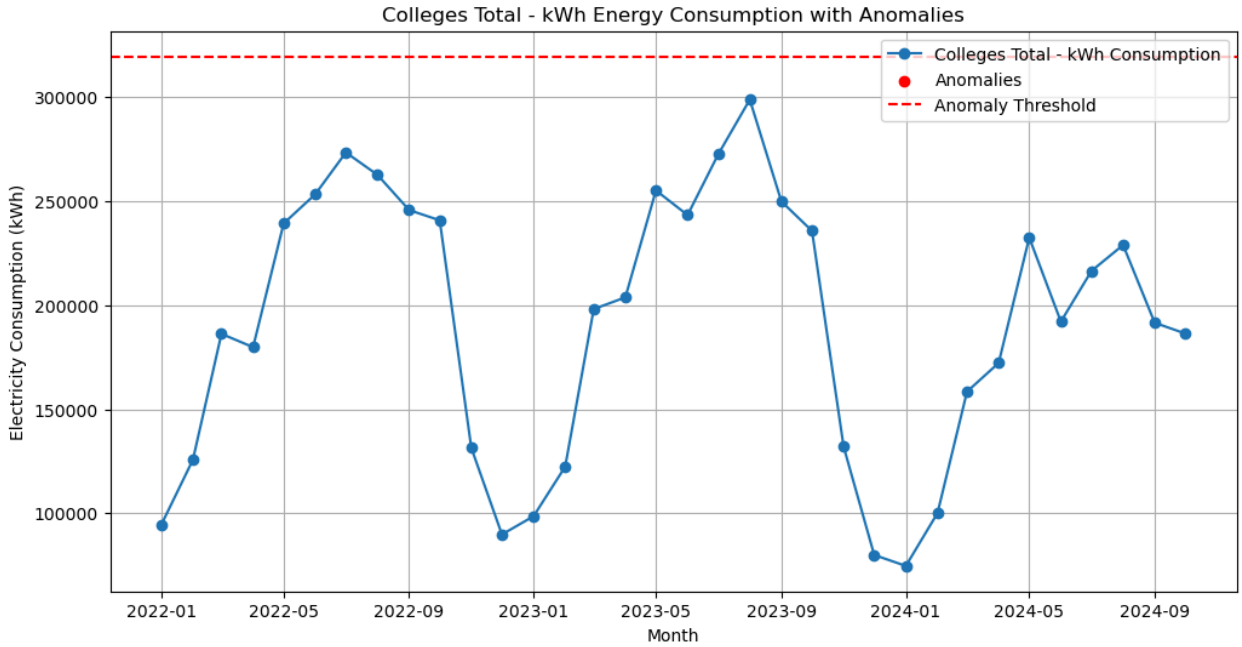


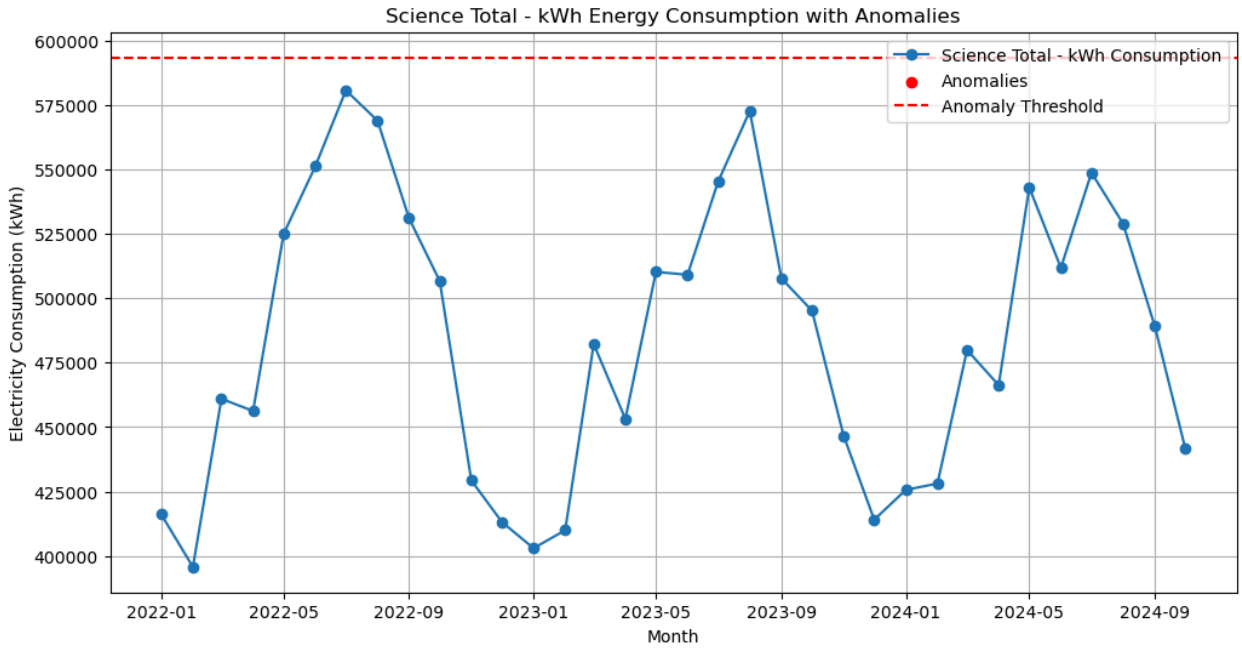


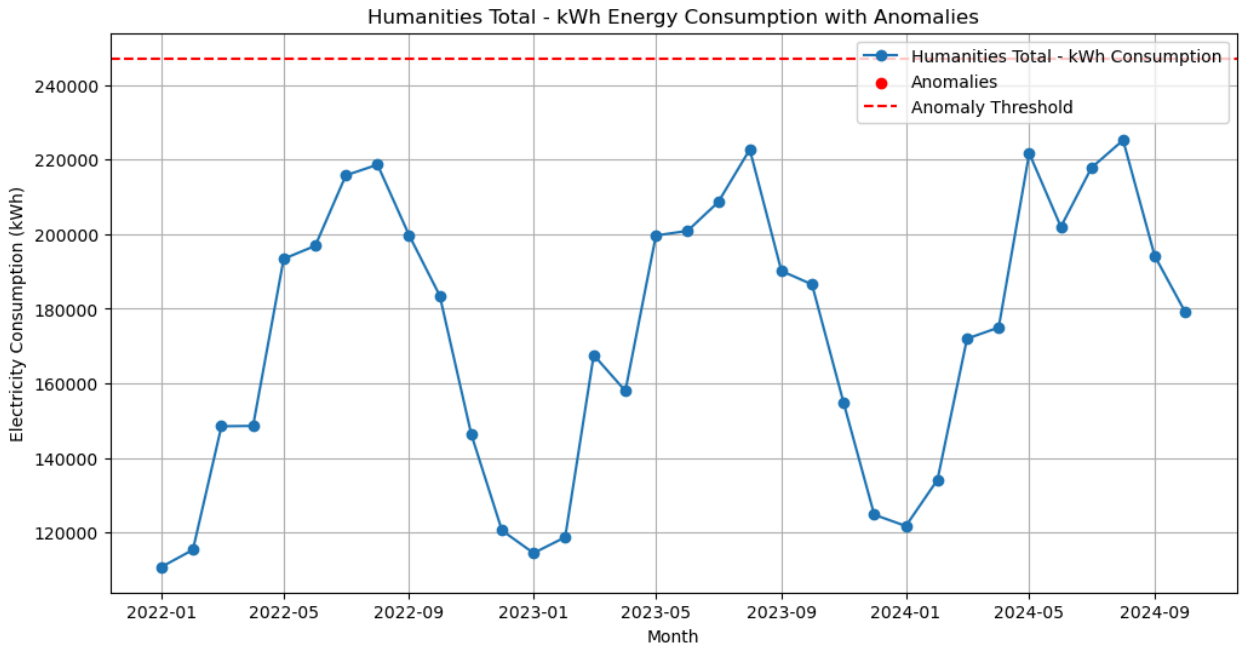
In [84]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
file_path = "stream_data_filtered - Copy.csv"  # Ensure the correct file path
df = pd.read_csv(file_path)

# Define building groups
building_groups = {
    "Libraries Total - kWh": [
        "E902 Hocken Library - kWh", "F813 UOCOE Robertson Library - kWh",
        "D203 Sayers (at Adams MSB):", "F813 Bill Robertson Library 1 MSB:",
        "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Main:"
    ],
    "Colleges Total - kWh": [
        "C405 192 Castle College - kWh", "D402 Hayward College - kWh",
        "D40X Cumberland College - kWh", "F711 Executive Residence - kWh",
        "F812 UOCOE Owheo Building - kWh", "G608 St Margarets College - kWh",
        "H41X Selwyn College - kWh E2", "H633 Arana College main - kWh",
        "H71X Studholm College - kWh E2", "J126 Carrington College (Kitchen/Dining) - kWh",
        "J14X Aquinas College - kWh", "J303 Caroline Freeman College - kWh",
        "K427 Abbey College - kWh"
    ],
    "Science Total - kWh": [
        "D403 Survey & Marine - kWh", "E212 Zoology Buildings - kWh",
        "F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
        "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
        "G404 Microbiology - kWh", "G413 Science 2 - kWh",
        "J960 Portobello Marine Lab - kWh", "G505 Geology north:", "G505 Geology south:"
    ],
    "Humanities Total - kWh": [
        "F9XX College of Education main (Boiler room) - kWh",
        "F518 Arts 1 Submains MSB:", "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:",
        "F505 1 Richardson Mains:", "G506/07 Archway buildings (incl. Allen & Marama Hall):"
    ],
    "Commerce Total - kWh": [
        "F614 1 School of Business Incomer 1 (Lower floors):",
```

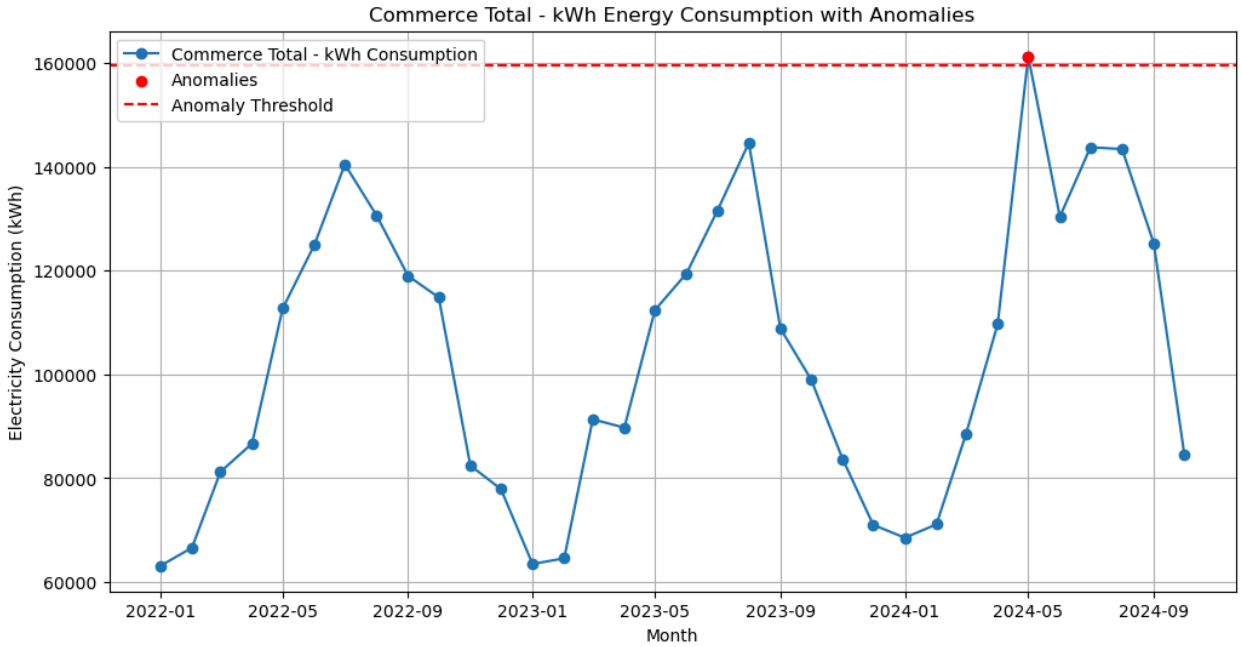
```
        "F614 2 School of Business Incomer 2 (Upper floors):",
        "F618 1 Psychology substation- Goddard:"
    ],
    "ITS Servers Total - kWh": [
        "F204 444 Great King Street - kWh"
    ]
}

# Convert Month column to datetime format for plotting
df["Month"] = pd.to_datetime(df["Month"], format="%b-%y")

# Set Month as index for better visualization
df.set_index("Month", inplace=True)

# Dictionary to store anomalies
anomalies_dict = {}

# Process each group
for group, buildings in building_groups.items():
    valid_buildings = [b for b in buildings if b in df.columns]  # Ensure columns exist
    if not valid_buildings:
        continue  # Skip if no valid columns

    # Calculate total consumption for the group
    df[group] = df[valid_buildings].sum(axis=1)

    # Mean + Sigma Method for anomaly detection
    mean_usage = df[group].mean()
    std_dev_usage = df[group].std()
    threshold = mean_usage + (2 * std_dev_usage)  # 2-Sigma threshold for anomalies

    # Identify anomalies
    anomalies = df[df[group] > threshold][group]
    anomalies_dict[group] = anomalies

    # Plot results
    plt.figure(figsize=(12, 6))
    plt.plot(df.index, df[group], label=f"{group} Consumption", marker='o', linestyle='-')
    plt.scatter(anomalies.index, anomalies, color='red', label="Anomalies", zorder=3)
    plt.axhline(threshold, color='r', linestyle='dashed', label="Anomaly Threshold")
    plt.xlabel("Month")
    plt.ylabel("Electricity Consumption (kWh)")
    plt.title(f"{group} Energy Consumption with Anomalies")
    plt.legend()
    plt.grid(True)
    plt.show()

# Display anomalies for each group
for group, anomalies in anomalies_dict.items():
    print(f"\nAnomalies detected in {group}:\n{anomalies}\n")
```

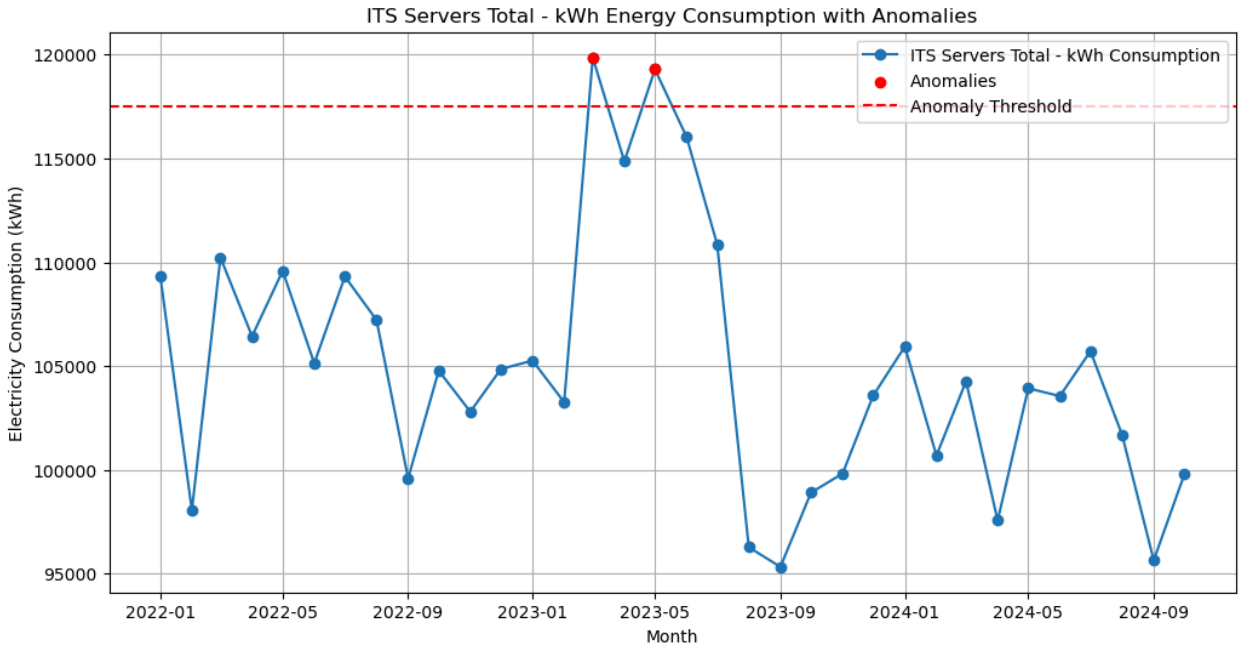



Colleges Total - kWh Energy Consumption with Anomalies
Colleges Total - kWh Consumption
Anomalies
Anomaly Threshold
Electricity Consumption (kWh)
300000
250000
200000
150000
100000
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09
Month


Science Total - kWh Energy Consumption with Anomalies
Science Total - kWh Consumption
Anomalies
Anomaly Threshold
Electricity Consumption (kWh)
600000
575000
550000
525000
500000
475000
450000
425000
400000
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09
Month


Humanities Total - kWh Energy Consumption with Anomalies
Humanities Total - kWh Consumption
Anomalies
Anomaly Threshold
Electricity Consumption (kWh)
240000
220000
200000
180000
160000
140000
120000
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09
Month



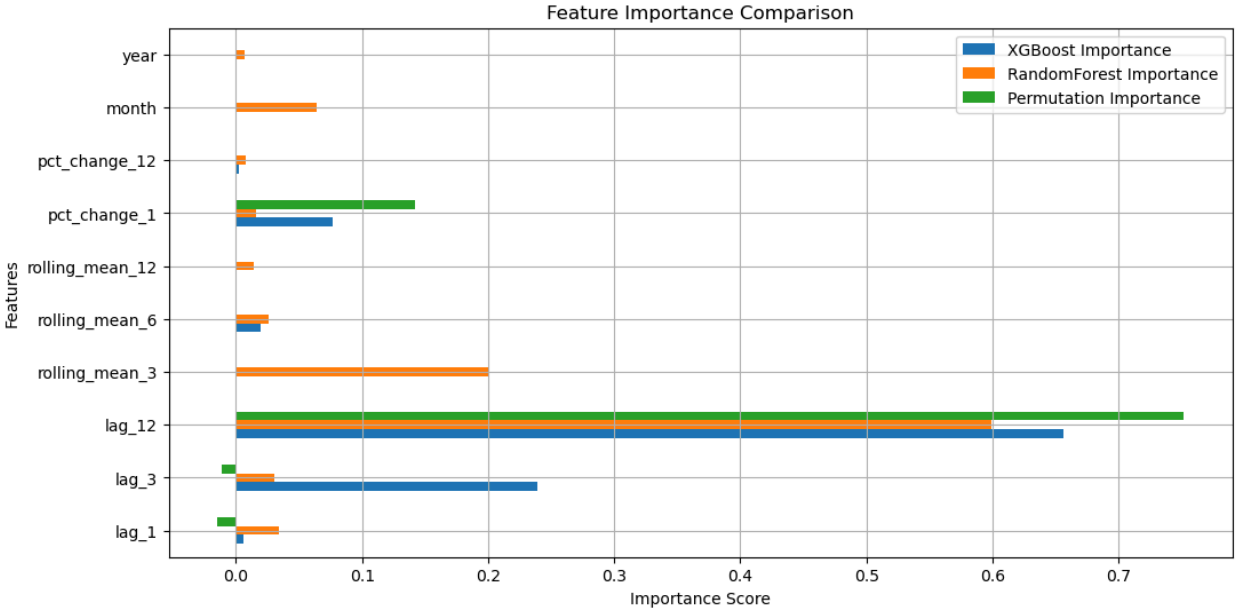



```
Anomalies detected in Libraries Total - kWh:
Series([], Name: Libraries Total - kWh, dtype: float64)


Anomalies detected in Colleges Total - kWh:
Series([], Name: Colleges Total - kWh, dtype: float64)


Anomalies detected in Science Total - kWh:
Series([], Name: Science Total - kWh, dtype: float64)


Anomalies detected in Humanities Total - kWh:
Series([], Name: Humanities Total - kWh, dtype: float64)


Anomalies detected in Commerce Total - kWh:
Month
2024-05-01    161285.13
Name: Commerce Total - kWh, dtype: float64


Anomalies detected in ITS Servers Total - kWh:
Month
2023-03-01    119858.94
2023-05-01    119288.30
Name: ITS Servers Total - kWh, dtype: float64
```

In [96]:
```
import pandas as pd
import numpy as np
```

```python
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.inspection import permutation_importance

# Load dataset
file_path = "stream_data_filtered - Copy.csv"  # Update with correct file path
df = pd.read_csv(file_path)

# Convert Month column to datetime format
df["Month"] = pd.to_datetime(df["Month"], format="%b-%y")
df.set_index("Month", inplace=True)

# Ensure dataset is sorted by time
df = df.sort_index()

# Select target variable: "Total Kwh"
target_variable = "Total kwh"

# Feature Engineering: Lags, Rolling Statistics, Seasonal Effects
df["lag_1"] = df[target_variable].shift(1)   # Previous month
df["lag_3"] = df[target_variable].shift(3)   # 3 months ago
df["lag_12"] = df[target_variable].shift(12) # 12 months ago (yearly seasonality)
df["rolling_mean_3"] = df[target_variable].rolling(window=3).mean()  # Short-term trend
df["rolling_mean_6"] = df[target_variable].rolling(window=6).mean()  # Mid-term trend
df["rolling_mean_12"] = df[target_variable].rolling(window=12).mean()  # Long-term trend
df["pct_change_1"] = df[target_variable].pct_change(1)  # Monthly percentage change
df["pct_change_12"] = df[target_variable].pct_change(12)  # Yearly percentage change

# Extracting Date Features
df["month"] = df.index.month
df["year"] = df.index.year

# Drop rows with NaN values (due to shifting and rolling mean)
df = df.dropna()

# Define features and target
features = ["lag_1", "lag_3", "lag_12", "rolling_mean_3", "rolling_mean_6",
            "rolling_mean_12", "pct_change_1", "pct_change_12", "month", "year"]
X = df[features]
y = df[target_variable]

# Train-Test Split (Time-Series Aware: No Shuffling)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

# Train XGBoost Model
xgb_model = XGBRegressor(objective="reg:squarederror", n_estimators=100, learning_rate=0.1, random_state=42)
xgb_model.fit(X_train, y_train)

# Train RandomForest Model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Extract Feature Importance from XGBoost
xgb_importance = xgb_model.feature_importances_

# Extract Feature Importance from RandomForest
rf_importance = rf_model.feature_importances_

# Compute Permutation Importance
perm_importance = permutation_importance(xgb_model, X_test, y_test, n_repeats=10, random_state=42)

# Convert to DataFrame for Comparison
feature_importance_df = pd.DataFrame({
    "Feature": features,
    "XGBoost Importance": xgb_importance,
    "RandomForest Importance": rf_importance,
    "Permutation Importance": perm_importance.importances_mean
}).set_index("Feature")

# Display Feature Importance Results
print("\nFeature Importance Table:")
print(feature_importance_df)

# Plot Feature Importance (XGBoost vs. RandomForest vs. Permutation)
plt.figure(figsize=(12, 6))
feature_importance_df.plot(kind="barh", figsize=(12, 6), title="Feature Importance Comparison")
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.grid(True)
plt.show()
```

```
Feature Importance Table:
                 XGBoost Importance  RandomForest Importance  \
Feature
lag_1                  5.863366e-03                 0.033921
lag_3                  2.387788e-01                 0.030738
lag_12                 6.558147e-01                 0.599555
rolling_mean_3         7.275987e-07                 0.201188
rolling_mean_6         1.963764e-02                 0.026112
rolling_mean_12        6.217838e-04                 0.013828
pct_change_1           7.711266e-02                 0.015750
pct_change_12          2.170330e-03                 0.008165
month                  0.000000e+00                 0.063622
year                   0.000000e+00                 0.007121

                 Permutation Importance
Feature
lag_1                         -0.014974
lag_3                         -0.011557
lag_12                         0.751844
rolling_mean_3                 0.000000
rolling_mean_6                 0.000000
rolling_mean_12                0.000000
pct_change_1                   0.141710
pct_change_12                  0.000000
month                          0.000000
year                           0.000000
<Figure size 1200x600 with 0 Axes>
```

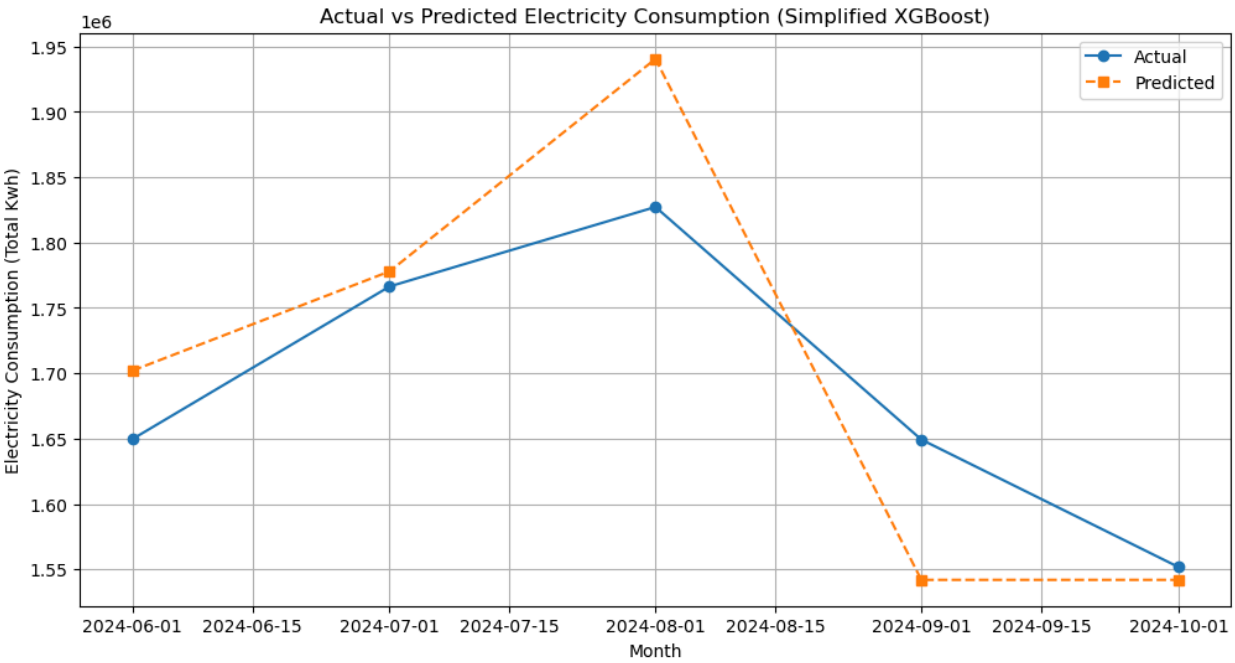


In [132…

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load dataset
file_path = "stream_data_filtered - Copy.csv"  # Update path
df = pd.read_csv(file_path)

# Convert Month column to datetime format
df["Month"] = pd.to_datetime(df["Month"], format="%b-%y")
df.set_index("Month", inplace=True)

# Ensure dataset is sorted by time
df = df.sort_index()

# Select target variable
target_variable = "Total kwh"

# Feature Engineering: Use only the most important features
df["lag_12"] = df[target_variable].shift(12)  # 1-year seasonality
df["lag_3"] = df[target_variable].shift(3)    # 3-month trend

# Drop NaN values (due to shifting)
df = df.dropna()

# Define features & target
features = ["lag_12", "lag_3"]
X = df[features]
y = df[target_variable]

# Train-Test Split (Time-Series Aware: No Shuffling)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

# Train a SIMPLE XGBoost Model
xgb_model = XGBRegressor(
    objective="reg:squarederror",
    n_estimators=500,  # Reduce trees
    learning_rate=0.05,  # Keep a standard learning rate
    max_depth=8,  # Keep tree depth low
    random_state=42
)

xgb_model.fit(X_train, y_train)

# Make Predictions
y_pred = xgb_model.predict(X_test)

# Evaluate Model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"Simple Model Evaluation:\n MAE: {mae}\n MSE: {mse}\n RMSE: {rmse}\n R² Score: {r2}")

# Plot Actual vs. Predicted
plt.figure(figsize=(12, 6))
plt.plot(df.index[-len(y_test):], y_test, label="Actual", marker="o")
plt.plot(df.index[-len(y_test):], y_pred, label="Predicted", marker="s", linestyle="dashed")
plt.xlabel("Month")
plt.ylabel("Electricity Consumption (Total Kwh)")
plt.title("Actual vs Predicted Electricity Consumption (Simplified XGBoost)")
plt.legend()
plt.grid(True)
plt.show()
```

```
Simple Model Evaluation:
 MAE: 58874.3966
 MSE: 5467212028.452863
 RMSE: 73940.5979719725
 R² Score: 0.4197756519306971
```

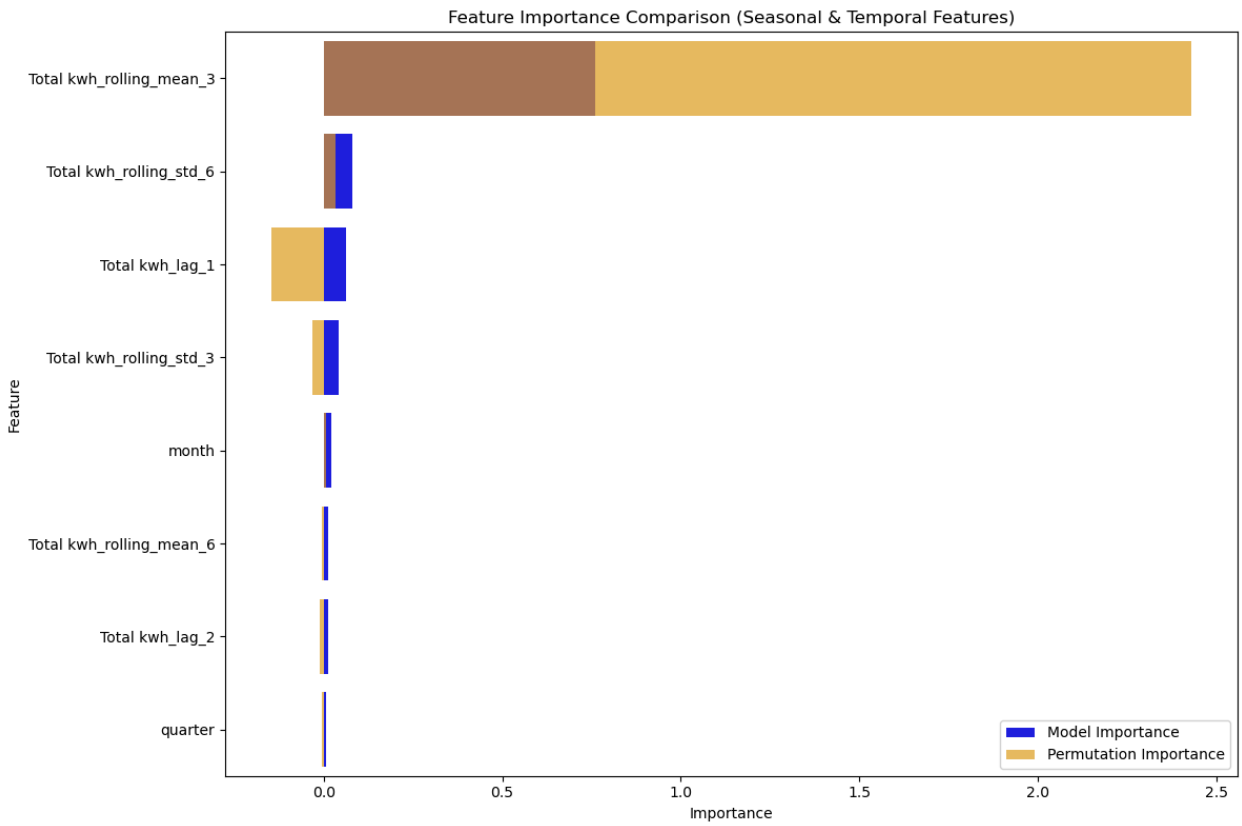


In [116…

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.inspection import permutation_importance
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert 'Month' to datetime and set it as the index for time series operations
df['Month'] = pd.to_datetime(df['Month'], format='%b-%y')
df = df.set_index('Month').sort_index()

# Target variable
target = 'Total kwh'

# Identify building columns (excluding the target)
building_cols = [col for col in df.columns if col != target]
```

```python
# Feature Engineering: Temporal and Seasonal Components

# Add lag features for the target variable
def create_lag_features(data, target_col, n_lags):
    for lag in range(1, n_lags + 1):
        data[f'{target_col}_lag_{lag}'] = data[target_col].shift(lag)
    return data

# Add rolling window features for the target variable
def create_rolling_features(data, target_col, window_sizes):
    for window in window_sizes:
        data[f'{target_col}_rolling_mean_{window}'] = data[target_col].rolling(window=window).mean()
        data[f'{target_col}_rolling_std_{window}'] = data[target_col].rolling(window=window).std()
    return data

# Add seasonal indicators: month and quarter
df['month'] = df.index.month
df['quarter'] = df.index.quarter

# Create lag and rolling features
df = create_lag_features(df, target, n_lags=2)
df = create_rolling_features(df, target, window_sizes=[3, 6])

# Drop building columns
df = df.drop(columns=building_cols)

# Drop rows with NaN values due to lagging/rolling operations
df = df.dropna()

# Prepare the feature matrix (X) and target vector (y)
X = df.drop(columns=[target])  # Exclude the target column from features
y = df[target]

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Regressor
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)

# Calculate feature importance using the model
model_importances = rf.feature_importances_

# Calculate permutation importance for validation
perm_importance = permutation_importance(rf, X_test, y_test, n_repeats=10, random_state=42)

# Combine results into a DataFrame
importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Model Importance': model_importances,
    'Permutation Importance': perm_importance.importances_mean
})

# Sort by model importance for better visualization
importance_df = importance_df.sort_values(by='Model Importance', ascending=False)

# Display the top features based on importance scores
print("Feature Importance:")
print(importance_df)

# Plotting feature importance comparison
plt.figure(figsize=(12, 8))

sns.barplot(x='Model Importance', y='Feature', data=importance_df, color='blue', label='Model Importance')
sns.barplot(x='Permutation Importance', y='Feature', data=importance_df, color='orange', label='Permutation Importance', alpha=0.

plt.title('Feature Importance Comparison (Seasonal & Temporal Features)')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.legend()
plt.tight_layout()
plt.show()
```

```
Feature Importance:
                    Feature  Model Importance  Permutation Importance
4  Total kwh_rolling_mean_3          0.760119                2.430119
7   Total kwh_rolling_std_6          0.079791                0.032989
2           Total kwh_lag_1          0.063012               -0.147019
5   Total kwh_rolling_std_3          0.041478               -0.031665
0                     month          0.021728                0.005323
6  Total kwh_rolling_mean_6          0.013398               -0.004256
3           Total kwh_lag_2          0.012972               -0.009911
1                   quarter          0.007503               -0.004359
```

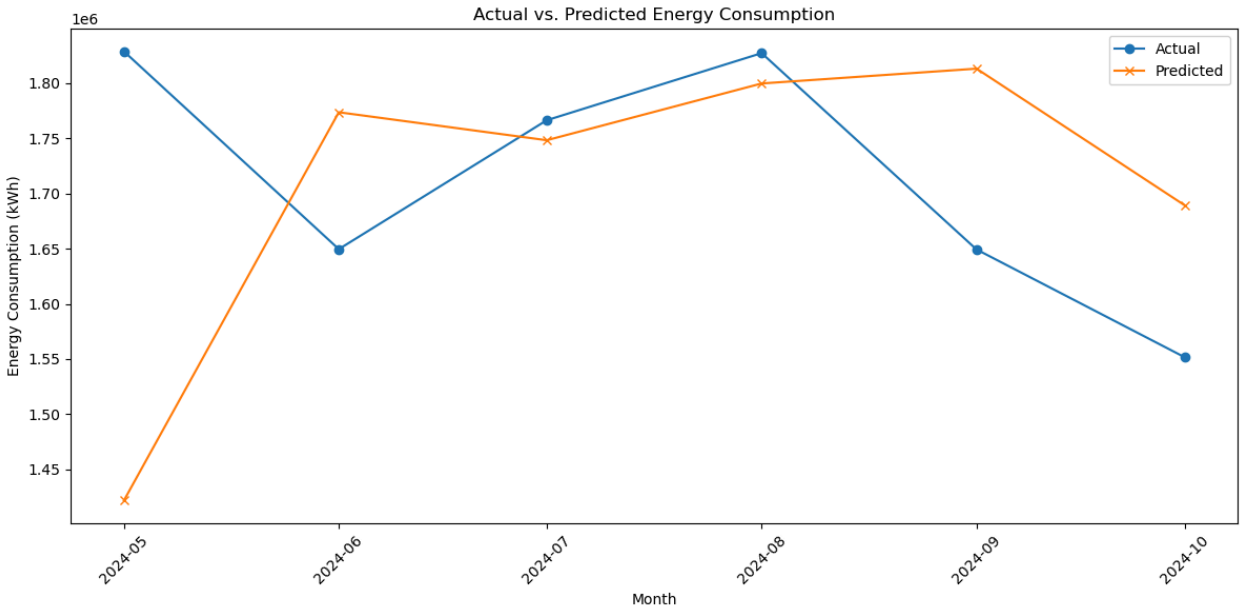


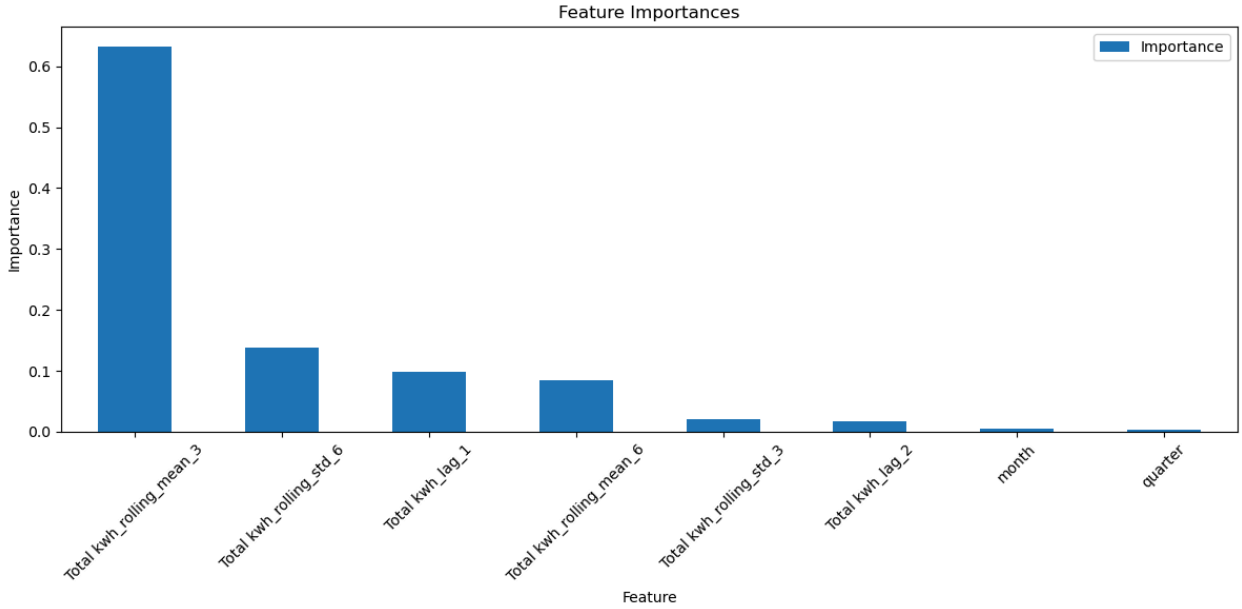
In [128…

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler

# Load the dataset
df = pd.read_csv('stream_data_filtered - Copy.csv')

# Convert 'Month' to datetime and set it as the index
df['Month'] = pd.to_datetime(df['Month'], format='%b-%y')
df = df.set_index('Month').sort_index()

# Target variable
target = 'Total kwh'

# Identify building columns (excluding the target)
building_cols = [col for col in df.columns if col != target]

# Feature Engineering: Temporal and Seasonal Components

# Add lag features for the target variable
def create_lag_features(data, target_col, n_lags):
    for lag in range(1, n_lags + 1):
        data[f'{target_col}_lag_{lag}'] = data[target_col].shift(lag)
    return data

# Add rolling window features for the target variable
def create_rolling_features(data, target_col, window_sizes):
    for window in window_sizes:
        data[f'{target_col}_rolling_mean_{window}'] = data[target_col].rolling(window=window).mean()
        data[f'{target_col}_rolling_std_{window}'] = data[target_col].rolling(window=window).std()
    return data

# Add seasonal indicators: month and quarter
df['month'] = df.index.month
df['quarter'] = df.index.quarter

# Create lag and rolling features
df = create_lag_features(df, target, n_lags=2)
df = create_rolling_features(df, target, window_sizes=[3, 6])

# Drop building columns
df = df.drop(columns=building_cols)

# Drop rows with NaN values due to lagging/rolling operations
df = df.dropna()

# Prepare the feature matrix (X) and target vector (y)
X = df.drop(columns=[target])  # Exclude the target column from features
y = df[target]
```

```python
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=False)

# Scale the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a Random Forest Regressor
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = rf.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Plotting the predictions
plt.figure(figsize=(12, 6))
plt.plot(df.index[-len(y_test):], y_test, label='Actual', marker='o')
plt.plot(df.index[-len(y_pred):], y_pred, label='Predicted', marker='x')
plt.title('Actual vs. Predicted Energy Consumption')
plt.xlabel('Month')
plt.ylabel('Energy Consumption (kWh)')
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Feature Importance from the trained Random Forest
feature_importances = pd.DataFrame({'Feature': X.columns, 'Importance': rf.feature_importances_})
feature_importances = feature_importances.sort_values('Importance', ascending=False)
feature_importances.plot(x='Feature', y='Importance', kind='bar', figsize=(12, 6))
plt.title('Feature Importances')
plt.xlabel('Feature')
plt.ylabel('Importance')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Mean Squared Error: 38023061020.98598
R-squared: -2.595592490646989
```



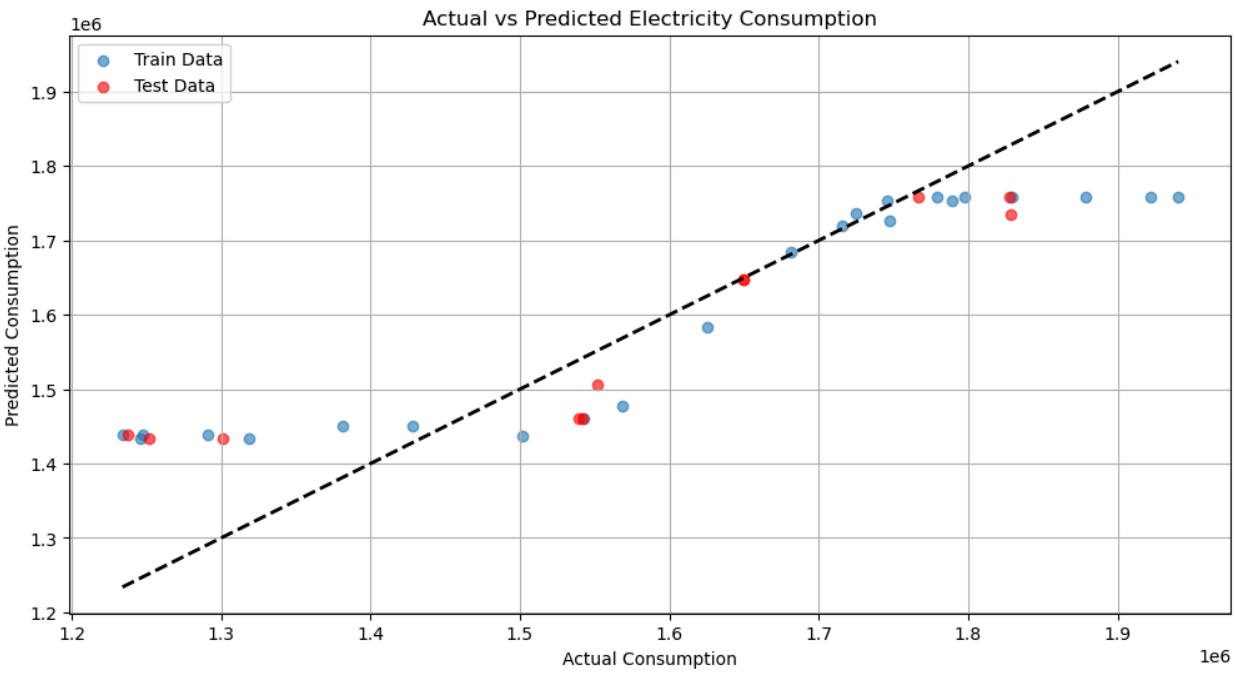


In [226…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load dataset
file_path = "stream_data_filtered - Copy.csv"  # Update path
df = pd.read_csv(file_path)

# Convert Month column to datetime format
df["Month"] = pd.to_datetime(df["Month"], format="%b-%y")
df.set_index("Month", inplace=True)

# Ensure dataset is sorted by time
df = df.sort_index()

# Select target variable
target_variable = "Total kwh"

# Feature Engineering: Lags, Rolling Statistics, Seasonal Effects
#df["lag_1"] = df[target_variable].shift(1)   # Previous month
#df["lag_3"] = df[target_variable].shift(3)   # 3 months ago
#df["lag_12"] = df[target_variable].shift(12) # 12 months ago (yearly seasonality)
df["rolling_mean_1"] = df[target_variable].rolling(window=1).mean()  # Short-term trend
df["rolling_mean_3"] = df[target_variable].rolling(window=3).mean()  # Short-term trend
df["rolling_std_3"] = df[target_variable].rolling(window=3).std()  # Short-term trend
#df["rolling_mean_6"] = df[target_variable].rolling(window=6).mean()  # Mid-term trend
#df["rolling_mean_12"] = df[target_variable].rolling(window=12).mean()  # Long-term trend
#df["pct_change_1"] = df[target_variable].pct_change(1)  # Monthly percentage change
#df["pct_change_12"] = df[target_variable].pct_change(12)  # Yearly percentage change

# Extracting Date Features
df["month"] = df.index.month
df["year"] = df.index.year

# Drop rows with NaN values (due to shifting and rolling mean)
#df = df.dropna()

# Define features and target
features = [ "rolling_mean_1","rolling_mean_3", "rolling_std_3",
             "month", "year"]
X = df[features]
y = df[target_variable]

# Train-Test Split (Time-Series Aware: No Shuffling)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, shuffle=False)

# Updated Hyperparameters to Avoid Overfitting
rf_model = RandomForestRegressor(
    n_estimators=200,       # Lower tree count
    max_depth=3,            # Reduce complexity
    min_samples_split=13,   # Increase min samples split
    min_samples_leaf=7,     # Higher leaf nodes for generalization
    random_state=42
)

# Train the Model
rf_model.fit(X_train, y_train)

# Make Predictions
```

```python
y_train_pred = rf_model.predict(X_train)
y_test_pred = rf_model.predict(X_test)

# Calculate Errors
mae_train = mean_absolute_error(y_train, y_train_pred)
mse_train = mean_squared_error(y_train, y_train_pred)
rmse_train = np.sqrt(mse_train)
r2_train = r2_score(y_train, y_train_pred)

mae_test = mean_absolute_error(y_test, y_test_pred)
mse_test = mean_squared_error(y_test, y_test_pred)
rmse_test = np.sqrt(mse_test)
r2_test = r2_score(y_test, y_test_pred)

# Print Evaluation Metrics
print(f"\nTrain Set Evaluation:\n MAE: {mae_train}\n MSE: {mse_train}\n RMSE: {rmse_train}\n R² Score: {r2_train}\n")
print(f"Test Set Evaluation:\n MAE: {mae_test}\n MSE: {mse_test}\n RMSE: {rmse_test}\n R² Score: {r2_test}\n")

# Scatter Plot: Actual vs Predicted
plt.figure(figsize=(12, 6))
plt.scatter(y_train, y_train_pred, label="Train Data", alpha=0.6)
plt.scatter(y_test, y_test_pred, label="Test Data", alpha=0.6, color='r')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)  # Perfect prediction line
plt.xlabel("Actual Consumption")
plt.ylabel("Predicted Consumption")
plt.title("Actual vs Predicted Electricity Consumption")
plt.legend()
plt.grid(True)
plt.show()

# Error Distribution Plot
errors = y_test - y_test_pred
plt.figure(figsize=(10, 5))
plt.hist(errors, bins=20, edgecolor='black', alpha=0.7)
plt.axvline(errors.mean(), color='red', linestyle='dashed', linewidth=2)
plt.xlabel("Prediction Error")
plt.ylabel("Frequency")
plt.title("Distribution of Prediction Errors")
plt.grid(True)
plt.show()
```

```
Train Set Evaluation:
 MAE: 82388.25138431278
 MSE: 11184972605.34914
 RMSE: 105759.03084535684
 R² Score: 0.7828716469890324

Test Set Evaluation:
 MAE: 81317.05954961744
 MSE: 10820858285.458088
 RMSE: 104023.35451934862
 R² Score: 0.7460664387576379
```



In [194…
```
y
```

Out[194…
```
Month
2022-01-01    1247984.205
2022-02-01    1290956.561
2022-03-01    1568301.282
2022-04-01    1501357.756
2022-05-01    1747537.419
2022-06-01    1797779.574
2022-07-01    1878862.983
2022-08-01    1921710.689
2022-09-01    1779010.618
2022-10-01    1724854.740
2022-11-01    1381030.457
2022-12-01    1233968.724
2023-01-01    1246137.036
2023-02-01    1318343.900
2023-03-01    1625166.633
2023-04-01    1542413.669
2023-05-01    1788816.288
2023-06-01    1745500.769
2023-07-01    1829589.552
2023-08-01    1940578.363
2023-09-01    1715935.128
2023-10-01    1681190.850
2023-11-01    1427750.680
2023-12-01    1237653.497
2024-01-01    1251995.802
2024-02-01    1301487.391
2024-03-01    1541910.889
2024-04-01    1539002.644
2024-05-01    1828852.601
2024-06-01    1649565.532
2024-07-01    1766434.299
2024-08-01    1827277.248
2024-09-01    1649254.967
2024-10-01    1551662.345
Name: Total kwh, dtype: float64
```

In [196…
```
X
```

Out[196…

Month	lag_12	rolling_mean_1	rolling_mean_3	rolling_mean_6	rolling_std_3	rolling_mean_12	pct_change_1	pct_change_12	month
2022-01-01	NaN	1247984.205	NaN	NaN	NaN	NaN	NaN	NaN	1
2022-02-01	NaN	1290956.561	NaN	NaN	NaN	NaN	0.034433	NaN	2
2022-03-01	NaN	1568301.282	1.369081e+06	NaN	173862.851823	NaN	0.214837	NaN	3
2022-04-01	NaN	1501357.756	1.453539e+06	NaN	144723.986032	NaN	-0.042685	NaN	4
2022-05-01	NaN	1747537.419	1.605732e+06	NaN	127286.720278	NaN	0.163971	NaN	5
2022-06-01	NaN	1797779.574	1.682225e+06	1.525653e+06	158637.214423	NaN	0.028750	NaN	6
2022-07-01	NaN	1878862.983	1.808060e+06	1.630799e+06	66263.611324	NaN	0.045102	NaN	7
2022-08-01	NaN	1921710.689	1.866118e+06	1.735925e+06	62940.933143	NaN	0.022805	NaN	8
2022-09-01	NaN	1779010.618	1.859861e+06	1.771043e+06	73223.096275	NaN	-0.074257	NaN	9
2022-10-01	NaN	1724854.740	1.808525e+06	1.808293e+06	101692.702165	NaN	-0.030442	NaN	10
2022-11-01	NaN	1381030.457	1.628299e+06	1.747208e+06	215845.703089	NaN	-0.199335	NaN	11
2022-12-01	NaN	1233968.724	1.446618e+06	1.653240e+06	251929.676563	1.589446e+06	-0.106487	NaN	12
2023-01-01	1247984.205	1246137.036	1.287045e+06	1.547785e+06	81620.520458	1.589292e+06	0.009861	-0.001480	1
2023-02-01	1290956.561	1318343.900	1.266150e+06	1.447224e+06	45608.970884	1.591575e+06	0.057945	0.021215	2
2023-03-01	1568301.282	1625166.633	1.396549e+06	1.421584e+06	201253.346182	1.596313e+06	0.232733	0.036259	3
2023-04-01	1501357.756	1542413.669	1.495308e+06	1.391177e+06	158742.718424	1.599735e+06	-0.050920	0.027346	4
2023-05-01	1747537.419	1788816.288	1.652132e+06	1.459141e+06	125395.051260	1.603175e+06	0.159751	0.023621	5
2023-06-01	1797779.574	1745500.769	1.692244e+06	1.544396e+06	131551.545679	1.598818e+06	-0.024215	-0.029080	6
2023-07-01	1878862.983	1829589.552	1.787969e+06	1.641638e+06	42050.796009	1.594712e+06	0.048175	-0.026225	7
2023-08-01	1921710.689	1940578.363	1.838556e+06	1.745344e+06	97847.421423	1.596284e+06	0.060663	0.009818	8
2023-09-01	1779010.618	1715935.128	1.828701e+06	1.760472e+06	112324.253312	1.591028e+06	-0.115761	-0.035455	9
2023-10-01	1724854.740	1681190.850	1.779235e+06	1.783602e+06	140803.426678	1.587389e+06	-0.020248	-0.025315	10
2023-11-01	1381030.457	1427750.680	1.608292e+06	1.723424e+06	157315.691468	1.591283e+06	-0.150750	0.033830	11
2023-12-01	1233968.724	1237653.497	1.448865e+06	1.638783e+06	222521.250768	1.591590e+06	-0.133145	0.002986	12
2024-01-01	1246137.036	1251995.802	1.305800e+06	1.542517e+06	105855.576013	1.592078e+06	0.011588	0.004702	1
2024-02-01	1318343.900	1301487.391	1.263712e+06	1.436002e+06	33491.006548	1.590673e+06	0.039530	-0.012786	2
2024-03-01	1625166.633	1541910.889	1.365131e+06	1.406998e+06	155082.576629	1.583735e+06	0.184730	-0.051229	3
2024-04-01	1542413.669	1539002.644	1.460800e+06	1.383300e+06	137976.695911	1.583451e+06	-0.001886	-0.002211	4
2024-05-01	1788816.288	1828852.601	1.636589e+06	1.450150e+06	166511.762121	1.586787e+06	0.188336	0.022381	5
2024-06-01	1745500.769	1649565.532	1.672474e+06	1.518802e+06	146276.566183	1.578793e+06	-0.098033	-0.054961	6
2024-07-01	1829589.552	1766434.299	1.748284e+06	1.604542e+06	91011.177185	1.573530e+06	0.070848	-0.034519	7

Month	lag_12	rolling_mean_1	rolling_mean_3	rolling_mean_6	rolling_std_3	rolling_mean_12	pct_change_1	pct_change_12	month
2024-08-01	1940578.363	1827277.248	1.747759e+06	1.692174e+06	90315.767493	1.564088e+06	0.034444	-0.058385	8
2024-09-01	1715935.128	1649254.967	1.747656e+06	1.710065e+06	90484.614552	1.558531e+06	-0.097425	-0.038859	9
2024-10-01	1681190.850	1551662.345	1.676065e+06	1.712174e+06	139749.673319	1.547737e+06	-0.059174	-0.077046	10

In [180…

```
df
```

Out[180…

Month	F204 444 Great King Street - kWh	F813 Bill Robertson Library 1 MSB:	F419 ISB West Excluding Shops	F505 Richardson Library Block Rising Main:	F518 Arts 1 Submains MSB:	F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:	F614 1 School of Business Incomer 1 (Lower floors):	F614 2 School of Business Incomer 2 (Upper floors):	F618 1 Psychology substation-Goddard:	F505 1 Richardson Mains:	...	G60X UNICOL - kWh	H4 Selw Colleg kWh
2022-12-01	104859.89	28787.15	50649.84	7102.20	16056.58	1358.19	53123.07	24773.63	64457.73	45352.19	...	32150.064	7859.0
2023-01-01	105261.77	29568.10	62910.13	6525.37	18610.10	816.88	47486.46	15986.18	67116.03	44661.50	...	41933.872	7731.8
2023-02-01	103261.91	35017.51	63041.74	6671.54	18506.53	821.36	48112.13	16455.17	85586.95	47564.80	...	53350.896	14412.8
2023-03-01	119858.94	40530.64	71311.41	13250.50	25238.78	1530.34	61686.79	29668.86	84045.82	66122.49	...	74208.144	32099.4
2023-04-01	114857.85	36706.53	64798.02	12007.07	22181.70	2040.94	55425.02	34327.04	79712.26	58931.71	...	62701.632	39227.0
2023-05-01	119288.30	43112.02	74385.01	15011.97	28210.91	2507.66	65879.55	46530.56	98728.96	71376.13	...	77504.880	58825.3
2023-06-01	116037.35	40615.52	68902.80	14172.85	27693.47	2819.33	66482.18	53007.36	102009.86	66795.77	...	70256.208	58440.4
2023-07-01	110863.88	35981.25	70751.08	12645.97	28881.02	2937.60	68857.34	62631.43	108556.29	64429.05	...	74350.400	64171.2
2023-08-01	96278.33	42779.20	69839.57	15174.93	29567.90	2830.11	73185.28	71435.27	120033.28	72911.62	...	78171.200	72355.2
2023-09-01	95328.88	39511.27	65938.74	14360.27	25602.53	2041.44	58049.54	50806.27	101186.56	68458.75	...	73683.664	55839.5
2023-10-01	98905.80	41467.78	64035.31	14325.63	26680.10	2138.46	54295.04	44785.15	104367.10	70875.13	...	75035.264	49855.4
2023-11-01	99825.87	35617.89	58159.92	11266.37	22550.14	1342.98	47420.41	36296.70	98383.88	58141.18	...	46442.784	21868.0
2023-12-01	103582.65	26966.18	48941.78	6586.27	17165.63	906.98	43467.27	27548.67	95669.25	47395.59	...	31819.072	8332.5
2024-01-01	105940.95	28010.66	55816.44	6313.95	16609.09	749.68	45851.14	22679.04	95974.40	47764.22	...	32038.880	6726.4
2024-02-01	100690.24	32932.86	56173.07	7613.73	18652.74	674.80	45396.99	25709.06	83159.04	54753.02	...	47696.416	15219.0
2024-03-01	104272.47	38208.32	67525.13	11014.46	24431.68	739.36	56550.40	32000.00	90723.33	65436.41	...	69449.040	34178.9
2024-04-01	97562.71	36607.43	61563.70	11562.37	24194.37	774.00	70272.52	39407.11	94680.06	62444.29	...	67209.376	47464.5
2024-05-01	103927.94	41156.86	62638.77	13822.85	29080.70	769.60	86793.22	74491.91	120805.38	74567.68	...	82622.288	73830.8
2024-06-01	103556.12	40019.52	56435.31	11023.78	28928.58	878.05	70441.98	59953.15	102550.53	65186.56	...	67335.632	59431.3
2024-07-01	105718.17	35773.12	60062.58	10917.12	29561.66	4558.11	76777.98	67025.92	116712.45	66982.40	...	66946.544	62414.4
2024-08-01	101686.08	40405.50	68360.89	12528.96	29324.35	4973.25	77723.13	65711.62	114256.90	70731.01	...	80172.224	72886.4
2024-09-01	95648.13	41482.43	65673.35	11271.33	25865.92	3683.20	70376.96	54946.82	98271.23	63757.57	...	71153.728	58321.3
2024-10-01	99781.97	42255.17	65975.72	3435.04	28894.98	3430.02	69843.97	14749.70	90706.95	70908.93	...	77810.704	56030.5

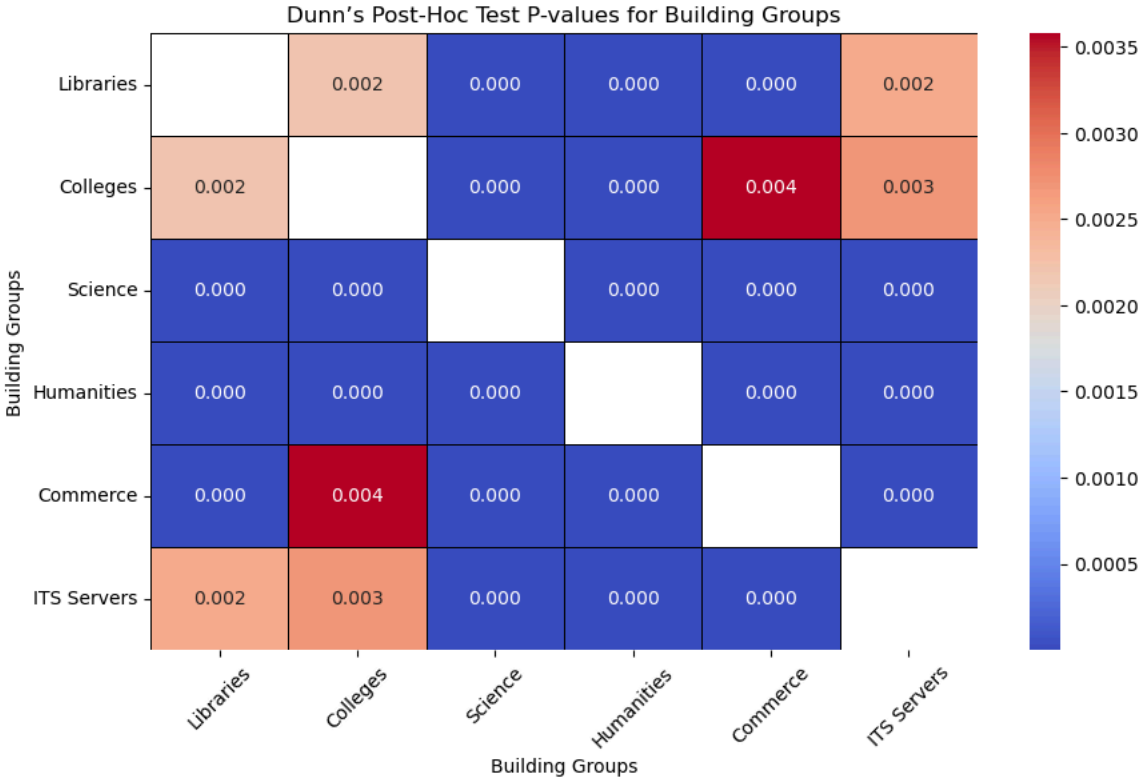
23 rows × 35 columns

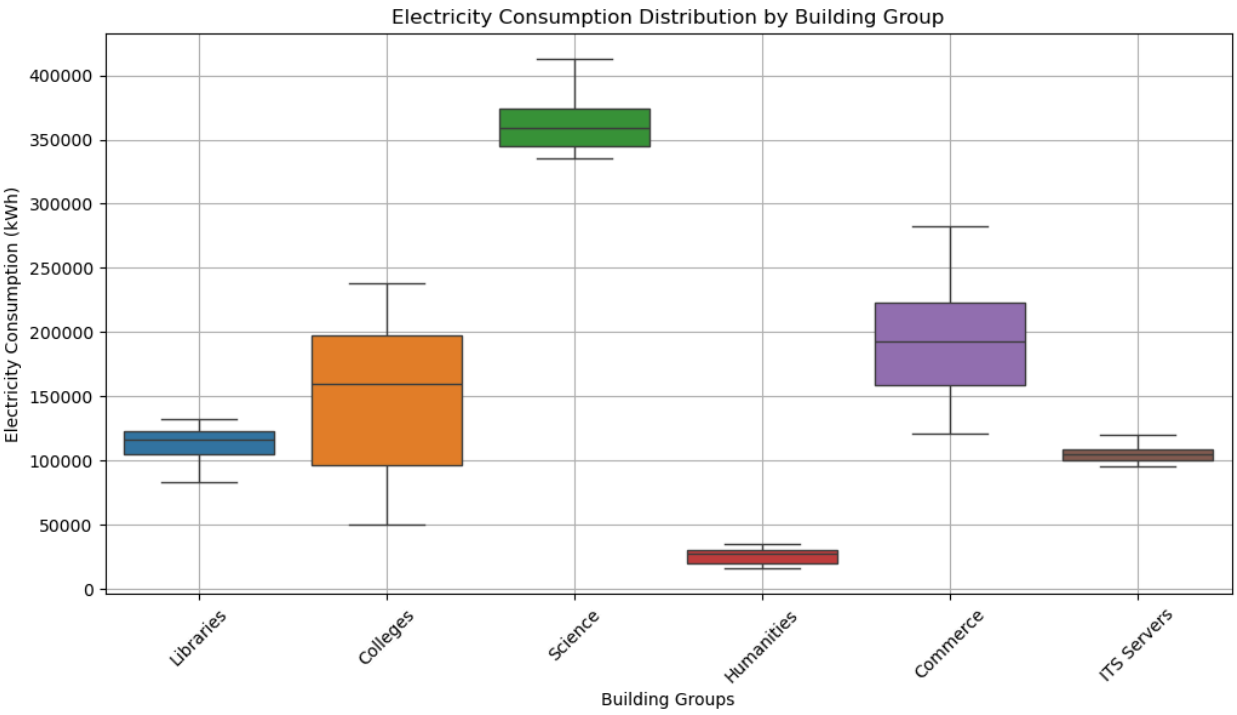
In [260…

```
# Import necessary libraries
import pandas as pd
import numpy as np
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
from itertools import combinations
from scipy.stats import mannwhitneyu
```

```
# Load dataset
file_path = "stream_data_filtered - Copy.csv"  # Update the correct path if necessary
df = pd.read_csv(file_path)

# Define building group categories (adjust if needed based on dataset structure)
building_groups = {
    "Libraries": ["F813 Bill Robertson Library 1 MSB:", "F419 ISB West Excluding Shops", "F505 Richardson Library Block Rising Ma
    "Colleges": ["G608 St Margarets College - kWh", "H41X Selwyn College - kWh E2", "H633 Arana College main - kWh", "H71X Studhc
    "Science": ["G404 Microbiology - kWh", "G413 Science 2 - kWh", "G403 Biochemistry - kWh"],
    "Humanities": ["F518 Arts 1 Submains MSB:", "F516 97 Albany & F517 99 Albany, F513 262 Leith Walk:"],
    "Commerce": ["F614 1 School of Business Incomer 1 (Lower floors):", "F614 2 School of Business Incomer 2 (Upper floors):",
                 "F618 1 Psychology substation- Goddard"],
    "ITS Servers": ["F204 444 Great King Street - kWh"]
}

# Aggregate electricity consumption per building group
grouped_consumption = {}
for group, buildings in building_groups.items():
    grouped_consumption[group] = df[buildings].sum(axis=1)

# Convert to DataFrame
grouped_df = pd.DataFrame(grouped_consumption)
grouped_df["Month"] = df.index  # Retain Month column

# Perform Kruskal-Wallis Test
kruskal_stat, kruskal_p = stats.kruskal(*[grouped_df[group] for group in grouped_df.columns if group != "Month"])

# Print Kruskal-Wallis test results
print(f"Kruskal-Wallis Test Statistic: {kruskal_stat}")
print(f"P-value: {kruskal_p}")

# Perform Dunn’s post-hoc test manually if Kruskal-Wallis test is significant
if kruskal_p < 0.05:
    print("\nSignificant differences detected. Performing Dunn’s post-hoc test...\n")

    # Compute pairwise comparisons using Mann-Whitney U test (alternative to Dunn's test)
    group_pairs = list(combinations([group for group in grouped_df.columns if group != "Month"], 2))
    dunn_results = {}

    for g1, g2 in group_pairs:
        stat, p_value = mannwhitneyu(grouped_df[g1], grouped_df[g2], alternative='two-sided')
        dunn_results[f"{g1} vs {g2}"] = p_value

    # Convert results to DataFrame
    dunn_df = pd.DataFrame.from_dict(dunn_results, orient="index", columns=["P-value"])
    dunn_df["Significant"] = dunn_df["P-value"] < 0.05  # Mark significant differences

    print(dunn_df)

    # Convert Dunn's test results into a pivot format suitable for a heatmap
    building_groups_list = list(building_groups.keys())
    dunn_matrix = np.zeros((len(building_groups_list), len(building_groups_list)))

    for i, g1 in enumerate(building_groups_list):
        for j, g2 in enumerate(building_groups_list):
            if g1 == g2:
                dunn_matrix[i, j] = np.nan  # No comparison within the same group
            else:
                key = f"{g1} vs {g2}"
                reverse_key = f"{g2} vs {g1}"
                if key in dunn_results:
                    dunn_matrix[i, j] = dunn_results[key]
                elif reverse_key in dunn_results:
                    dunn_matrix[i, j] = dunn_results[reverse_key]

    # Convert to DataFrame
    dunn_heatmap_df = pd.DataFrame(dunn_matrix, index=building_groups_list, columns=building_groups_list)

    # Plot Heatmap for Dunn’s Test P-values
    plt.figure(figsize=(10, 6))
    sns.heatmap(dunn_heatmap_df, annot=True, fmt=".3f", cmap="coolwarm", linewidths=0.5, linecolor="black")
    plt.title("Dunn’s Post-Hoc Test P-values for Building Groups")
    plt.xlabel("Building Groups")
    plt.ylabel("Building Groups")
    plt.xticks(rotation=45)
    plt.yticks(rotation=0)
    plt.show()

# Boxplot to show distribution of electricity consumption across building groups
plt.figure(figsize=(12, 6))
sns.boxplot(data=grouped_df.drop(columns=["Month"]))
plt.xlabel("Building Groups")
plt.ylabel("Electricity Consumption (kWh)")
plt.title("Electricity Consumption Distribution by Building Group")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

```
Kruskal-Wallis Test Statistic: 169.94806312769003
P-value: 7.484859883892976e-35

Significant differences detected. Performing Dunn's post-hoc test...

                              P-value  Significant
Libraries vs Colleges      2.211481e-03         True
Libraries vs Science       1.407033e-12         True
Libraries vs Humanities    1.407033e-12         True
Libraries vs Commerce      5.223219e-12         True
Libraries vs ITS Servers   2.499029e-03         True
Colleges vs Science        1.407033e-12         True
Colleges vs Humanities     1.407033e-12         True
Colleges vs Commerce       3.578955e-03         True
Colleges vs ITS Servers    2.709314e-03         True
Science vs Humanities      1.407033e-12         True
Science vs Commerce        1.407033e-12         True
Science vs ITS Servers     1.407033e-12         True
Humanities vs Commerce     1.407033e-12         True
Humanities vs ITS Servers  1.407033e-12         True
Commerce vs ITS Servers    1.407033e-12         True
```





In [ ]: