In [1]:
```
import pandas as pd
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib.dates as mdates
```

In [2]:
```
stream_file_path = "Stream data.xlsx"
stream_data = pd.read_excel(stream_file_path, engine='openpyxl')
print(stream_data.iloc[:10, 5])  # First 10 rows, first 10 columns
```

```
0    440424.382
1    444172.076
2    519503.259
3    487604.940
4    606794.981
5    705035.506
6    757101.765
7    456433.794
8    676192.258
9    646846.514
Name: 0000 Ring Main #3  - kWh, dtype: float64
```

castle_file_path = "192 Castle street.xlsx" castle_data = pd.read_excel(castle_file_path, engine='openpyxl') print(castle_data .iloc[:10, 5]) # First 10 rows, first 10 columns

janitza_file_path = "Janitza Reading.xlsx" janitza_data = pd.read_excel(janitza_file_path, engine='openpyxl') print(janitza_data .iloc[:10, 5]) # First 10 rows, first 10 columns

gas_file_path = "Gas Data.xlsx" gas_data = pd.read_excel(gas_file_path, engine='openpyxl') print(gas_data.iloc[:10, 0]) # 9 to 19 (Python uses zero-based indexing)

cfi_file_path = "CFI.xlsx" cfi_data = pd.read_excel(cfi_file_path , engine='openpyxl') print(cfi_data.iloc[:10, 9:20]) # 9 to 19 (Python uses zero-based indexing)

In [7]:
```
# ------- 🌏 Remove All "PF" Columns (Power Factor) ---------
pf_columns = [col for col in stream_data.columns if "PF" in col]
stream_data_filtered = stream_data.drop(columns=pf_columns)
```

In [8]:
```
print("\n✅ Removed Power Factor (PF) Columns. Remaining Columns:")
print(stream_data_filtered.columns)
```

```
✅ Removed Power Factor (PF) Columns. Remaining Columns:
Index(['Date', '0000 Ring Main #1 MP4889 - kWh', '0000 Ring Main #2  - kWh',
       '0000 Ring Main #3  - kWh', 'A161 Taieri Farm - kWh',
       'C405 192 Castle College - kWh', 'D20X Med School Sub Main - kWh',
       'D402 Hayward College - kWh', 'D403 Survey & Marine - kWh',
       'D40X Cumberland College - kWh', 'E201 School of Dentistry - kWh',
       'E212 Zoology Buildings - kWh', 'E214 Otago Dental School - kWh',
       'E301 Hunter Centre - kWh', 'E305 Physiotherapy - kWh',
       'E308 Student Health - kWh', 'E325 Research Support Facility - kWh',
       'E902 Hocken Library - kWh', 'F204 444 Great King Street - kWh',
       'F315 Botany Tin Hut - kWh', 'F325 Physical Education - kWh',
       'F711 Executive Residence - kWh', 'F812 UOCOE Owheo Building - kWh',
       'F813 UOCOE Robertson Library - kWh', 'F940 Plaza Building - kWh',
       'F9XX College of Education main (Boiler room) - kWh',
       'G401 Mellor Laboratories - kWh', 'G403 Biochemistry - kWh',
       'G404 Microbiology - kWh', 'G413 Science 2 - kWh',
       'G608 St Margarets College - kWh', 'G60X UNICOL - 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',
       'J960 Portobello Marine Lab - kWh', 'K427 Abbey College - kWh',
       'XC01 UoO School of Medicine ChCh - kWh', 'Ring Mains Total - kWh',
       'Libraries Total - kWh', 'Colleges Total - kWh', 'Science Total - kWh',
       'Health Science Total -kWh', 'Humanities Total - kWh',
       'OBS & Psychology Total - kWh', 'Total Stream DN Electricity - kWh',
       'ITS Servers Total - kWh', 'Unnamed: 90'],
      dtype='object')
```

In [9]:
```python
# Basic Dataset Information
print("\n Basic Information:")
print(stream_data_filtered.info())
```

```
 Basic Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75 entries, 0 to 74
Data columns (total 51 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   Date                                                  36 non-null     object
 1   0000 Ring Main #1 MP4889 - kWh                        35 non-null     float64
 2   0000 Ring Main #2  - kWh                              35 non-null     float64
 3   0000 Ring Main #3  - kWh                              35 non-null     float64
 4   A161 Taieri Farm - kWh                                35 non-null     float64
 5   C405 192 Castle College - kWh                         35 non-null     float64
 6   D20X Med School Sub Main - kWh                        35 non-null     float64
 7   D402 Hayward College - kWh                            35 non-null     float64
 8   D403 Survey & Marine - kWh                            35 non-null     float64
 9   D40X Cumberland College - kWh                         35 non-null     float64
 10  E201 School of Dentistry - kWh                        0 non-null      float64
 11  E212 Zoology Buildings - kWh                          35 non-null     float64
 12  E214 Otago Dental School - kWh                        35 non-null     float64
 13  E301 Hunter Centre - kWh                              35 non-null     float64
 14  E305 Physiotherapy - kWh                              35 non-null     float64
 15  E308 Student Health - kWh                             35 non-null     float64
 16  E325 Research Support Facility - kWh                  35 non-null     float64
 17  E902 Hocken Library - kWh                             35 non-null     float64
 18  F204 444 Great King Street - kWh                      35 non-null     float64
 19  F315 Botany Tin Hut - kWh                             35 non-null     float64
 20  F325 Physical Education - kWh                         35 non-null     float64
 21  F711 Executive Residence - kWh                        35 non-null     float64
 22  F812 UOCOE Owheo Building - kWh                       35 non-null     float64
 23  F813 UOCOE Robertson Library - kWh                    35 non-null     float64
 24  F940 Plaza Building - kWh                             35 non-null     float64
 25  F9XX College of Education main (Boiler room) - kWh    35 non-null     float64
 26  G401 Mellor Laboratories - kWh                        35 non-null     float64
 27  G403 Biochemistry - kWh                               35 non-null     float64
 28  G404 Microbiology - kWh                               35 non-null     float64
 29  G413 Science 2 - kWh                                  35 non-null     float64
 30  G608 St Margarets College - kWh                       35 non-null     float64
 31  G60X UNICOL - kWh                                     35 non-null     float64
 32  H41X Selwyn College - kWh E2                          35 non-null     float64
 33  H633 Arana College main - kWh                         35 non-null     float64
 34  H71X Studholm College - kWh E2                        35 non-null     float64
 35  J126 Carrington College (Kitchen/Dining) - kWh        35 non-null     float64
 36  J14X Aquinas College - kWh                            35 non-null     float64
 37  J303 Caroline Freeman College - kWh                   35 non-null     float64
 38  J960 Portobello Marine Lab - kWh                      35 non-null     float64
 39  K427 Abbey College - kWh                              35 non-null     float64
 40  XC01 UoO School of Medicine ChCh - kWh                35 non-null     float64
 41  Ring Mains Total - kWh                                60 non-null     object
 42  Libraries Total - kWh                                 61 non-null     object
 43  Colleges Total - kWh                                  73 non-null     object
 44  Science Total - kWh                                   65 non-null     object
 45  Health Science Total -kWh                             60 non-null     object
 46  Humanities Total - kWh                                59 non-null     object
 47  OBS & Psychology Total - kWh                          36 non-null     object
 48  Total Stream DN Electricity - kWh                     54 non-null     object
 49  ITS Servers Total - kWh                               56 non-null     object
 50  Unnamed: 90                                           7 non-null      object
dtypes: float64(40), object(11)
memory usage: 30.0+ KB
None
```

In [10]:
```
print("\n First Few Rows:")
print(stream_data_filtered.head())
```

First Few Rows:

	Date	0000 Ring Main #1 MP4889 - kWh	0000 Ring Main #2 - kWh
0	jan 22	251350.978	0.0
1	feb 22	258506.224	0.0
2	mar 22	305807.132	0.0
3	apr 22	301477.281	0.0
4	may 22	368266.870	0.0

	0000 Ring Main #3 - kWh	A161 Taieri Farm - kWh
0	440424.382	5108.36
1	444172.076	4832.68
2	519503.259	5474.11
3	487604.940	6012.11
4	606794.981	3417.19

	C405 192 Castle College - kWh	D20X Med School Sub Main - kWh
0	11006.160	327840.03
1	14045.024	334840.35
2	23767.872	374717.28
3	20786.656	363227.76
4	24610.720	409979.49

	D402 Hayward College - kWh	D403 Survey & Marine - kWh
0	9460.99	9963.728
1	15397.23	10737.416
2	25299.55	13596.736
3	22042.35	13572.792
4	26364.83	17680.424

	D40X Cumberland College - kWh	...	Ring Mains Total - kWh
0	18225.27	...	691775.36
1	25812.61	...	702678.3
2	43444.23	...	825310.391
3	38529.47	...	789082.221
4	48061.39	...	975061.851

	Libraries Total - kWh	Colleges Total - kWh	Science Total - kWh
0	207424.52	187025.792	519811.102
1	229916.898	274032.066	493387.592
2	270178.388	436309.404	574577.334
3	263960.132	446513.608	565962.862
4	272543.018	610869.994	648713.356

	Health Science Total -kWh	Humanities Total - kWh
0	757910.414	110716.888
1	731577.462	115342.064
2	831066.47	148488.08
3	777692.678	148583.6
4	853381.752	193396.944

	OBS & Psychology Total - kWh	Total Stream DN Electricity - kWh
0	120681	2666022.785
1	132638	2702224.627
2	156541	3254557.831
3	131581	3125000.171
4	166315	3683255.374

	ITS Servers Total - kWh	Unnamed: 90
0	237083.43	NaN
1	215193.92	NaN

```
2                 245081.2          NaN
3                235901.71          NaN
4                246812.64          NaN

[5 rows x 51 columns]
```

In [11]:
```
# Check for missing values
print("\n Missing Values in Each Column:")
print(stream_data_filtered.isnull().sum())
```

```
 Missing Values in Each Column:
Date                                                  39
0000 Ring Main #1 MP4889 - kWh                        40
0000 Ring Main #2  - kWh                              40
0000 Ring Main #3  - kWh                              40
A161 Taieri Farm - kWh                                40
C405 192 Castle College - kWh                         40
D20X Med School Sub Main - kWh                        40
D402 Hayward College - kWh                            40
D403 Survey & Marine - kWh                            40
D40X Cumberland College - kWh                         40
E201 School of Dentistry - kWh                        75
E212 Zoology Buildings - kWh                          40
E214 Otago Dental School - kWh                        40
E301 Hunter Centre - kWh                              40
E305 Physiotherapy - kWh                              40
E308 Student Health - kWh                             40
E325 Research Support Facility - kWh                  40
E902 Hocken Library - kWh                             40
F204 444 Great King Street - kWh                      40
F315 Botany Tin Hut - kWh                             40
F325 Physical Education - kWh                         40
F711 Executive Residence - kWh                        40
F812 UOCOE Owheo Building - kWh                       40
F813 UOCOE Robertson Library - kWh                    40
F940 Plaza Building - kWh                             40
F9XX College of Education main (Boiler room) - kWh    40
G401 Mellor Laboratories - kWh                        40
G403 Biochemistry - kWh                               40
G404 Microbiology - kWh                               40
G413 Science 2 - kWh                                  40
G608 St Margarets College - kWh                       40
G60X UNICOL - kWh                                     40
H41X Selwyn College - kWh E2                          40
H633 Arana College main - kWh                         40
H71X Studholm College - kWh E2                        40
J126 Carrington College (Kitchen/Dining) - kWh        40
J14X Aquinas College - kWh                            40
J303 Caroline Freeman College - kWh                   40
J960 Portobello Marine Lab - kWh                      40
K427 Abbey College - kWh                              40
XC01 UoO School of Medicine ChCh - kWh                40
Ring Mains Total - kWh                                15
Libraries Total - kWh                                 14
Colleges Total - kWh                                   2
Science Total - kWh                                   10
Health Science Total -kWh                             15
Humanities Total - kWh                                16
OBS & Psychology Total - kWh                          39
Total Stream DN Electricity - kWh                     21
ITS Servers Total - kWh                               19
Unnamed: 90                                           68
dtype: int64
```

In [12]:
```
# Summary statistics after PF removal
print("\n📊 Summary Statistics:")
print(stream_data_filtered.describe())
```

📊 Summary Statistics:

	0000 Ring Main #1 MP4889 - kWh	0000 Ring Main #2 - kWh	\
count	35.000000	35.000000	
mean	218377.356371	8475.225771	
std	57390.667993	48834.970550	
min	128801.481000	0.000000	
25%	177447.207000	0.000000	
50%	202655.105000	0.000000	
75%	254928.601000	0.000000	
max	368266.870000	289035.212000	

	0000 Ring Main #3 - kWh	A161 Taieri Farm - kWh	\
count	35.000000	35.000000	
mean	594647.633657	3473.196000	
std	121479.610180	797.364857	
min	401875.122000	2566.770000	
25%	472980.518500	3060.350000	
50%	606794.981000	3271.790000	
75%	688662.135500	3774.055000	
max	806203.431000	6012.110000	

	C405 192 Castle College - kWh	D20X Med School Sub Main - kWh	\
count	35.000000	35.000000	
mean	19678.156800	361508.416286	
std	4777.368021	31601.196587	
min	9163.376000	305257.860000	
25%	15092.984000	335163.075000	
50%	22019.872000	366398.580000	
75%	23052.488000	385562.190000	
max	24610.720000	410549.730000	

	D402 Hayward College - kWh	D403 Survey & Marine - kWh	\
count	35.000000	35.000000	
mean	25064.488571	16561.315314	
std	7735.407016	3388.254264	
min	9460.990000	9963.728000	
25%	21090.265000	14131.864000	
50%	26218.290000	16537.568000	
75%	29000.800000	18955.960000	
max	41268.900000	23438.272000	

	D40X Cumberland College - kWh	E201 School of Dentistry - kWh	...	\
count	35.000000	0.0	...	
mean	40543.660571	NaN	...	
std	13837.657920	NaN	...	
min	13771.850000	NaN	...	
25%	33129.210000	NaN	...	
50%	44898.770000	NaN	...	
75%	47140.825000	NaN	...	
max	81384.580000	NaN	...	

	G60X UNICOL - kWh	H41X Selwyn College - kWh E2	\
count	35.000000	35.000000	
mean	65160.296229	43125.367714	
std	16738.978711	22064.724747	
min	31056.272000	6726.402000	
25%	51106.360000	21177.171000	
50%	71153.728000	53866.554000	
75%	77810.704000	59128.350000	
max	84484.704000	73830.822000	

```
       H633 Arana College main - kWh  H71X Studholm College - kWh E2  \
count                      35.000000                       35.000000
mean                    33744.619429                    41265.910286
std                     20041.663112                    19214.266014
min                      6576.912000                     9162.460000
25%                     11625.408000                    22240.815000
50%                     32963.872000                    46512.880000
75%                     53187.400000                    56157.540000
max                     60545.264000                    72074.530000

       J126 Carrington College (Kitchen/Dining) - kWh  \
count                                       35.000000
mean                                     30312.675714
std                                      12213.900395
min                                       7481.590000
25%                                      18268.165000
50%                                      34132.430000
75%                                      39770.150000
max                                      46195.990000

       J14X Aquinas College - kWh  J303 Caroline Freeman College - kWh  \
count                   35.000000                            35.000000
mean                 44613.789943                        102962.457286
std                  22250.278881                         49585.352240
min                   8067.390000                         20119.320000
25%                  23422.336000                         50511.990000
50%                  50143.510000                        119910.330000
75%                  60835.184000                        142673.580000
max                  76083.536000                        173416.530000

       J960 Portobello Marine Lab - kWh  K427 Abbey College - kWh  \
count                         35.000000                35.000000
mean                       34107.890857             30253.224000
std                         7295.912453             15239.101875
min                        22589.980000              6909.050000
25%                        28373.165000             14283.235000
50%                        33108.150000             35495.620000
75%                        39555.690000             42891.630000
max                        49317.510000             53417.450000

       XC01 UoO School of Medicine ChCh - kWh
count                              35.000000
mean                           186316.221943
std                              9085.192698
min                            172476.864000
25%                            179128.032000
50%                            184149.864000
75%                            195110.316000
max                            204770.040000

[8 rows x 40 columns]
```

In [13]:
```
# Export the filtered stream data to a CSV file
stream_data_filtered.to_csv('stream_data_filtered.csv', index=False)
```

In [80]:
```
import pandas as pd
import numpy as np

# Read the CSV file
```

```
df = pd.read_csv('stream_data_filtered.csv')

# Drop the last column
df = df.iloc[:, :-1]

# Drop rows from index 36 onwards
df = df.iloc[:36]

# Fill empty cells with 0
df = df.fillna(0)

# Convert all columns to numeric (except Date) and round to 2 decimal points
numeric_columns = df.columns.drop('Date')
df[numeric_columns] = df[numeric_columns].apply(pd.to_numeric, errors='coerce').

# Create a list of month-year labels for 36 rows
month_labels = [f"{month}-{year}" for year in range(22, 25) for month in ['Jan',

# Replace the existing Date column with the new labels
df['Date'] = month_labels

# Save the processed DataFrame to a new CSV file
df.to_csv('processed_stream_data.csv', index=False)

print("CSV file has been processed and saved as 'processed_stream_data.csv'")
```

```
CSV file has been processed and saved as 'processed_stream_data.csv'
```

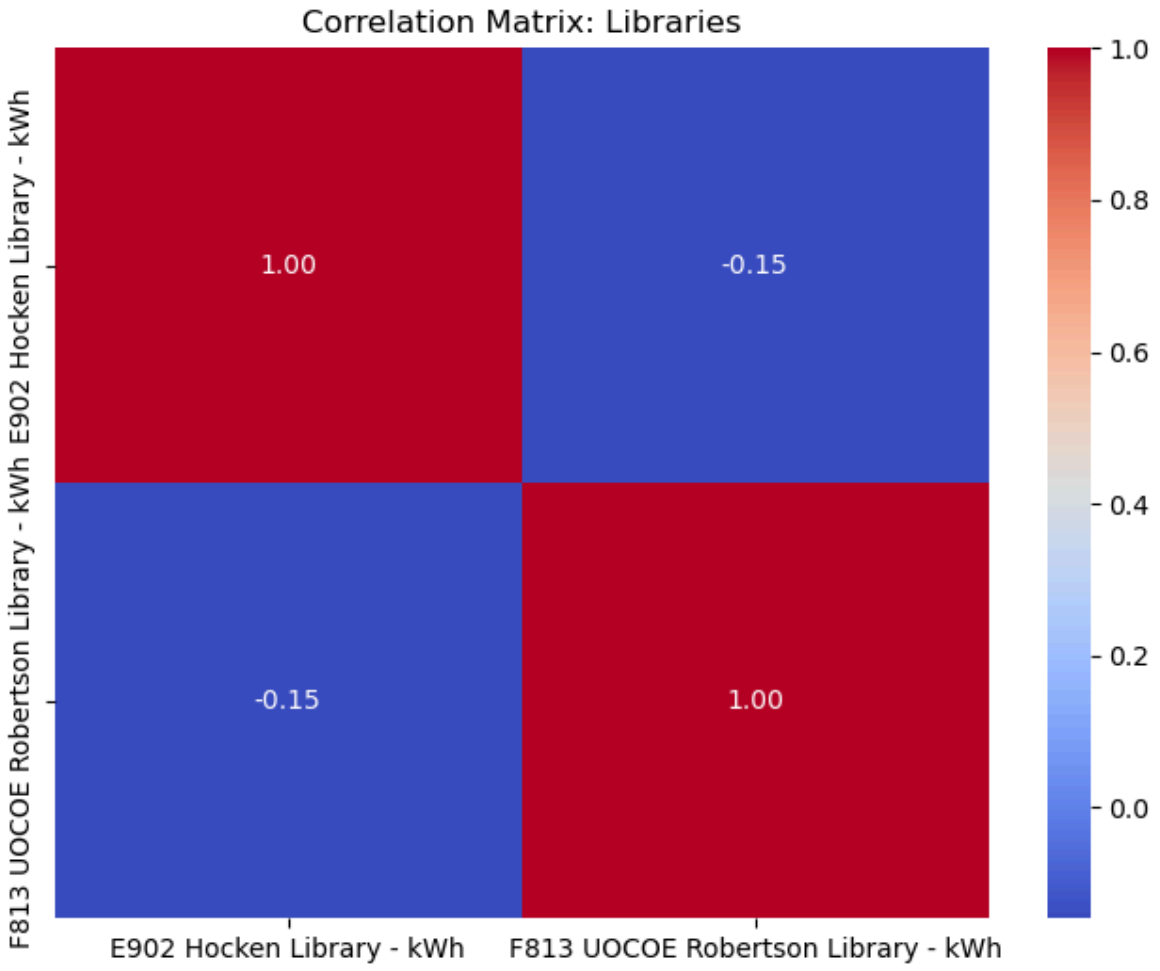
In [14]:
```
import pandas as pd

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

# Drop all rows from row 37 onwards
filtered_data = data.iloc[:36]

# Drop the last two columns
modified_data = filtered_data.iloc[:, :-2]

# Convert all values to numeric, fill empty cells with 0, and round to 2 decimal
modified_data = modified_data.apply(pd.to_numeric, errors='coerce').fillna(0).ro

# Save the modified dataset as a CSV file
modified_data.to_csv('Stream filtered new data.csv', index=False)
```

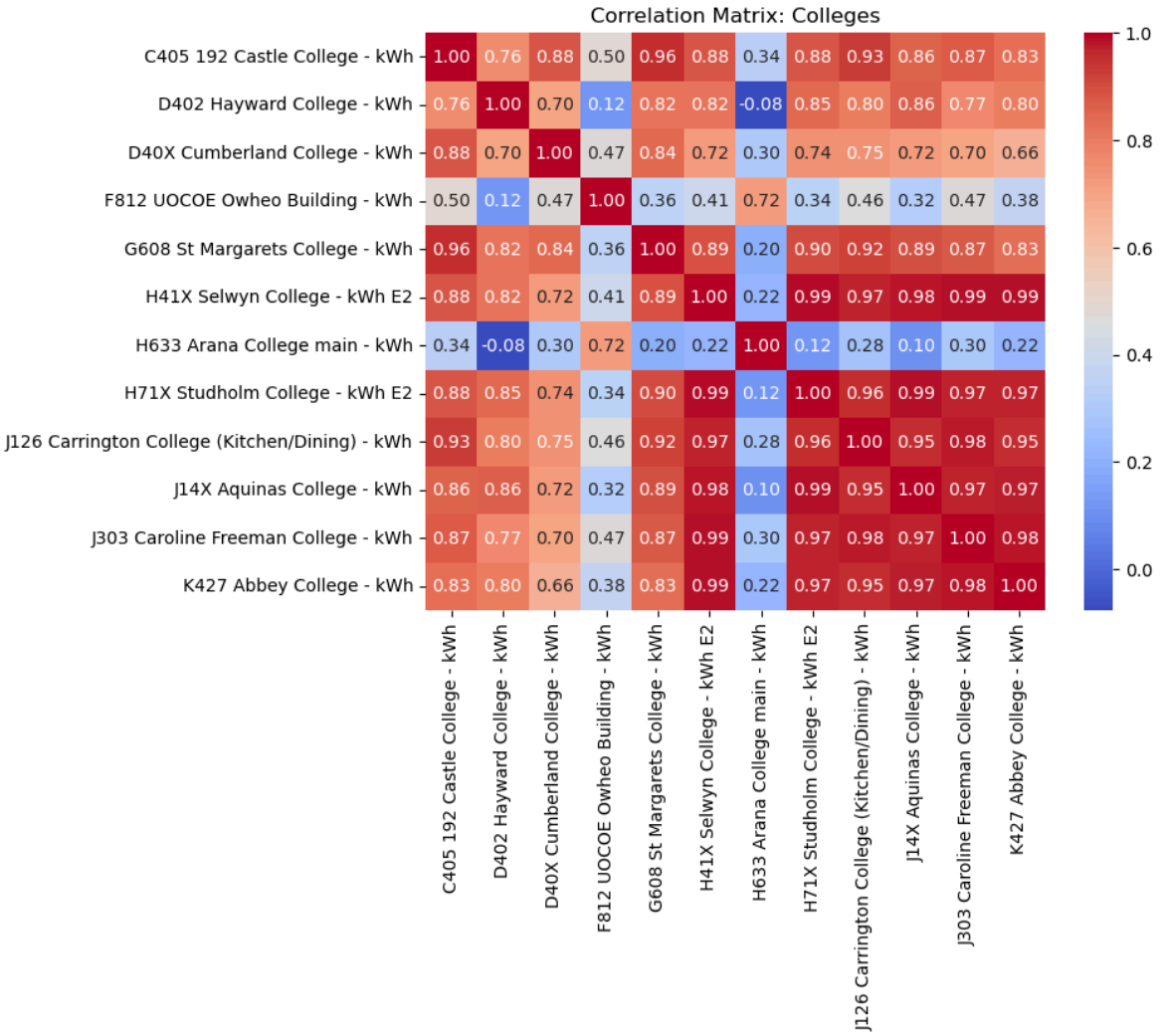
In [15]:
```
# Define column groups based on the screenshot
groups = {
    "Libraries": [
        "E902 Hocken Library - kWh", "F813 UOCOE Robertson Library - kWh",
        "D203 Sayers (at Adams) - kWh", "F813 Bill Robertson Library - kWh",
        "F419 ISB West Excludir - kWh", "F505 Richardson Library - kWh"
    ],
    "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"
```

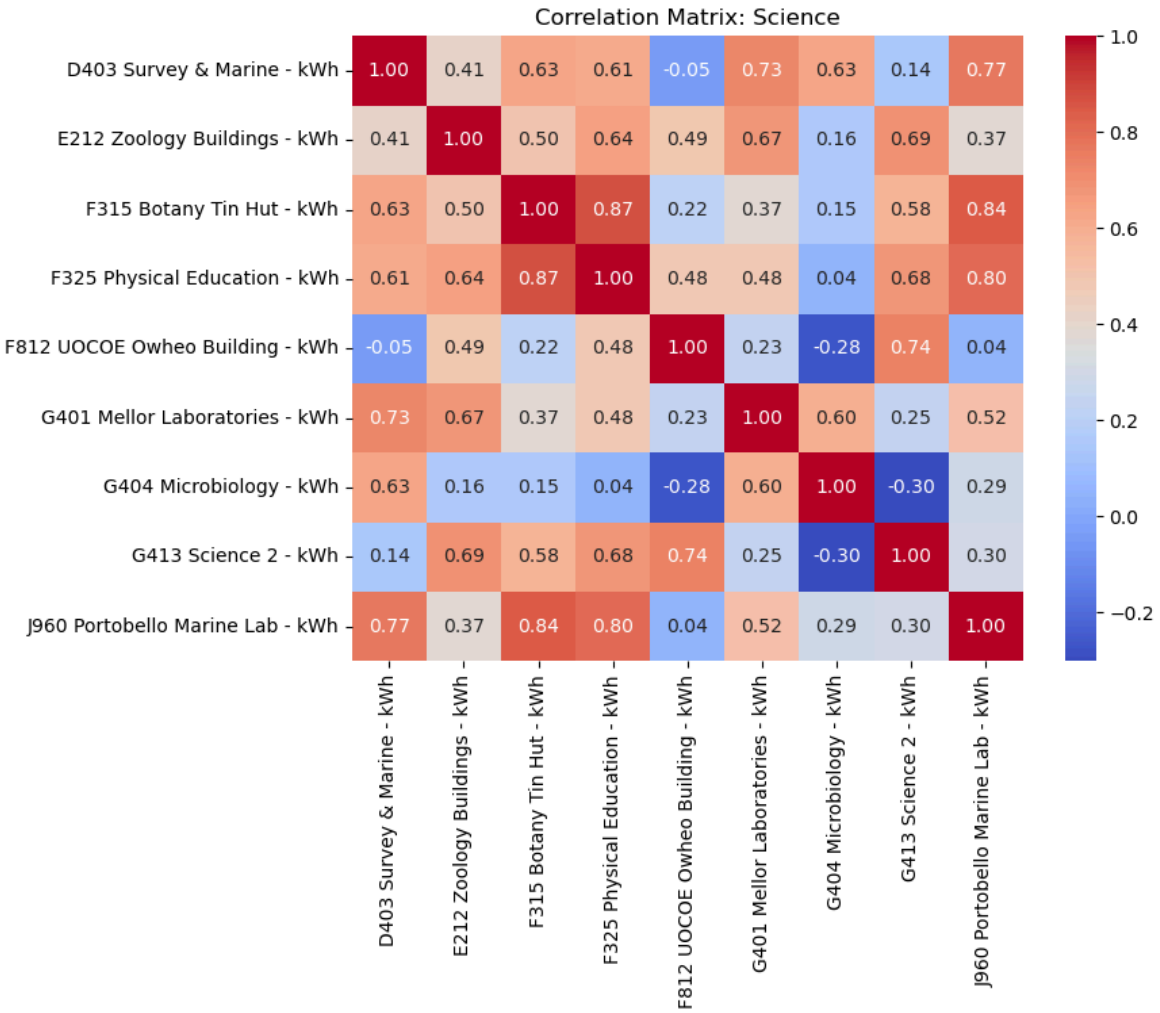
```
    ],
    "Science": [
        "D403 Survey & Marine - kWh", "E212 Zoology Buildings - kWh",
        "F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
        "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
        "G404 Microbiology - kWh", "G413 Science 2 - kWh",
        "J960 Portobello Marine Lab - kWh", "G505 Geology north:",
        "G505 Geology south:"
    ],
    "Health Science": [
        "A161 Taieri Farm - kWh", "D20X Med School Sub Main - kWh",
        "E214 Otago Dental School - kWh", "E301 Hunter Centre - kWh",
        "E305 Physiotherapy - kWh", "E325 Research Support Facility - kWh"
    ],
    "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
    ],
    "Commerce": [
        "F614 1 School of Business Incomer 1 (Lower floors) - kWh",
        "F614 2 School of Business Incomer 2 (Upper floors) - kWh",
        "F618 1 Psychology substation - Goddard - kWh"
    ],
    "Total Electricity": [
        "Total Stream DN Electricity - kWh"
    ],
    "ITS Servers": [
        "F204 444 Great King Street - kWh", "325 Gt King Server (325-PHYS) - kWh
    ]
}
```

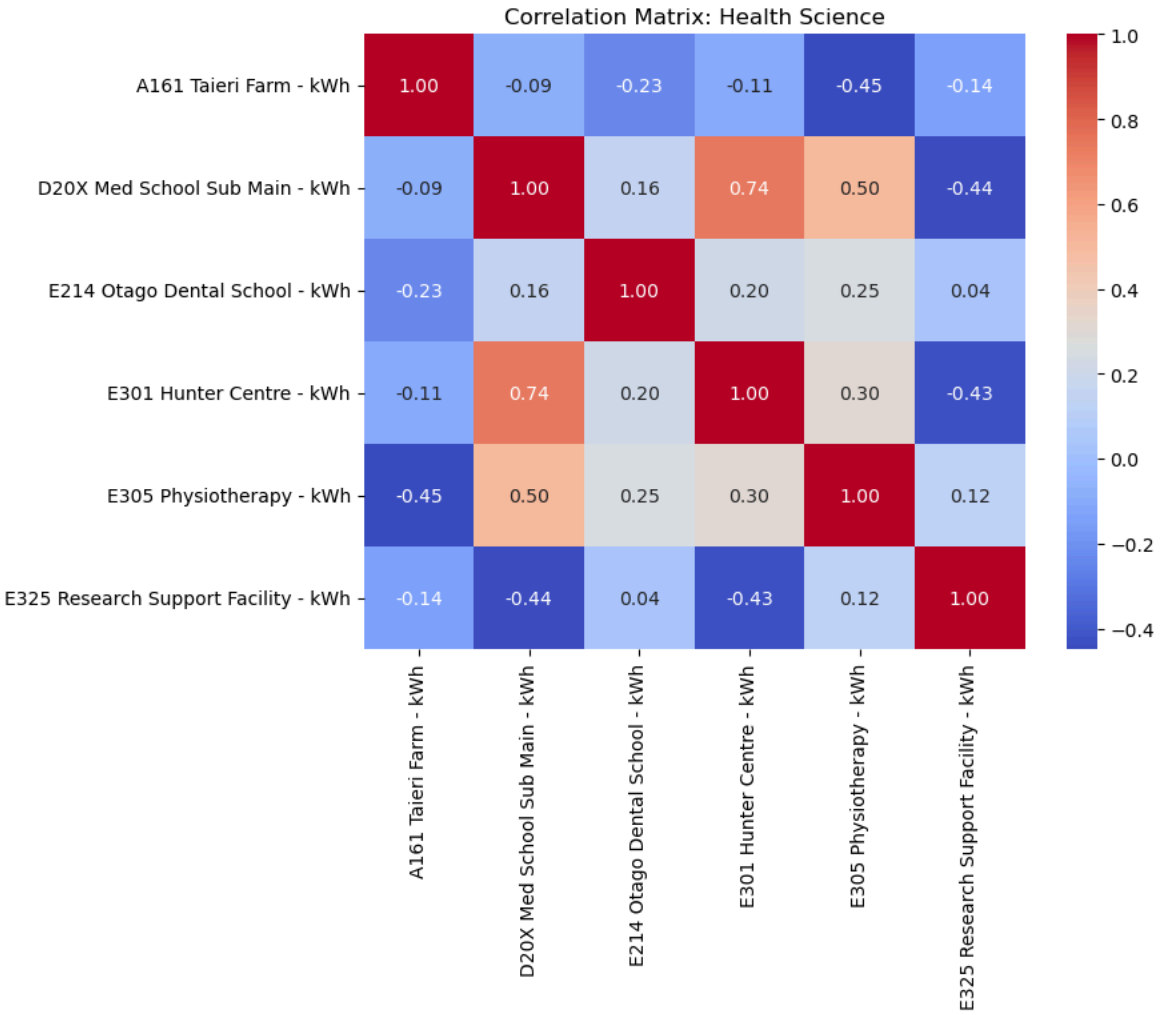
In [16]:

```
# Perform correlation analysis for each group and plot heatmaps
for group_name, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if len(available_columns) > 1:  # Only process groups with more than one val
        plt.figure(figsize=(8,6))
        corr_matrix = stream_data_filtered[available_columns].corr(numeric_only=
        sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
        plt.title(f"Correlation Matrix: {group_name}")
        plt.show()
    else:
        print(f"\nSkipping {group_name} - Not enough valid columns for correlati
```


Correlation Matrix: Libraries
F813 UOCOE Robertson Library - kWh E902 Hocken Library - kWh
1.00
-0.15
-0.15
1.00
E902 Hocken Library - kWh
F813 UOCOE Robertson Library - kWh
1.0
0.8
0.6
0.4
0.2
0.0


Correlation Matrix: Colleges
C405 192 Castle College - kWh 1.00 0.76 0.88 0.50 0.96 0.88 0.34 0.88 0.93 0.86 0.87 0.83
D402 Hayward College - kWh 0.76 1.00 0.70 0.12 0.82 0.82 -0.08 0.85 0.80 0.86 0.77 0.80
D40X Cumberland College - kWh 0.88 0.70 1.00 0.47 0.84 0.72 0.30 0.74 0.75 0.72 0.70 0.66
F812 UOCOE Owheo Building - kWh 0.50 0.12 0.47 1.00 0.36 0.41 0.72 0.34 0.46 0.32 0.47 0.38
G608 St Margarets College - kWh 0.96 0.82 0.84 0.36 1.00 0.89 0.20 0.90 0.92 0.89 0.87 0.83
H41X Selwyn College - kWh E2 0.88 0.82 0.72 0.41 0.89 1.00 0.22 0.99 0.97 0.98 0.99 0.99
H633 Arana College main - kWh 0.34 -0.08 0.30 0.72 0.20 0.22 1.00 0.12 0.28 0.10 0.30 0.22
H71X Studholm College - kWh E2 0.88 0.85 0.74 0.34 0.90 0.99 0.12 1.00 0.96 0.99 0.97 0.97
J126 Carrington College (Kitchen/Dining) - kWh 0.93 0.80 0.75 0.46 0.92 0.97 0.28 0.96 1.00 0.95 0.98 0.95
J14X Aquinas College - kWh 0.86 0.86 0.72 0.32 0.89 0.98 0.10 0.99 0.95 1.00 0.97 0.97
J303 Caroline Freeman College - kWh 0.87 0.77 0.70 0.47 0.87 0.99 0.30 0.97 0.98 0.97 1.00 0.98
K427 Abbey College - kWh 0.83 0.80 0.66 0.38 0.83 0.99 0.22 0.97 0.95 0.97 0.98 1.00
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
1.0
0.8
0.6
0.4
0.2
0.0


Correlation Matrix: 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
1.00 0.41 0.63 0.61 -0.05 0.73 0.63 0.14 0.77
0.41 1.00 0.50 0.64 0.49 0.67 0.16 0.69 0.37
0.63 0.50 1.00 0.87 0.22 0.37 0.15 0.58 0.84
0.61 0.64 0.87 1.00 0.48 0.48 0.04 0.68 0.80
-0.05 0.49 0.22 0.48 1.00 0.23 -0.28 0.74 0.04
0.73 0.67 0.37 0.48 0.23 1.00 0.60 0.25 0.52
0.63 0.16 0.15 0.04 -0.28 0.60 1.00 -0.30 0.29
0.14 0.69 0.58 0.68 0.74 0.25 -0.30 1.00 0.30
0.77 0.37 0.84 0.80 0.04 0.52 0.29 0.30 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

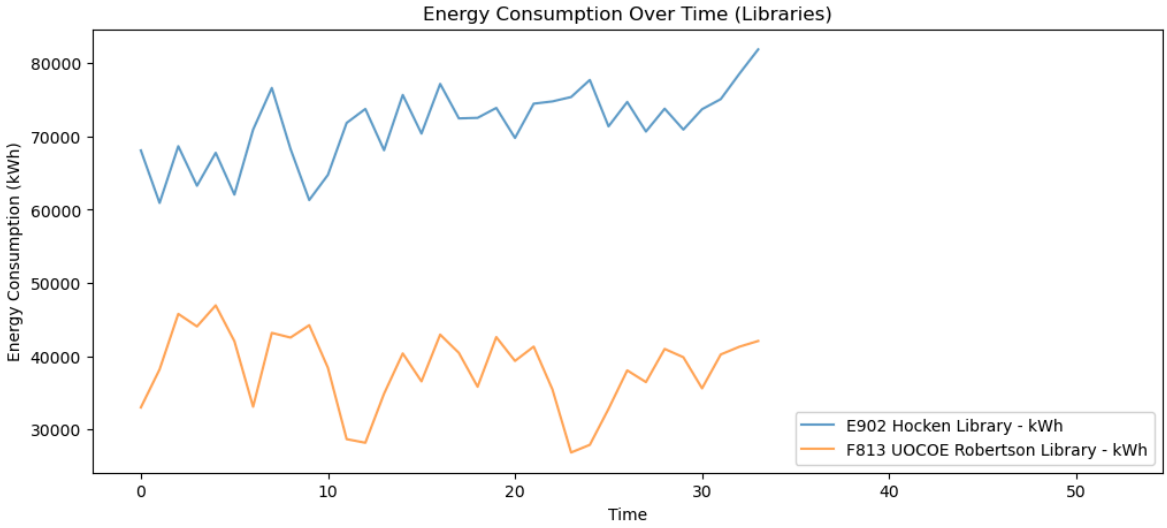


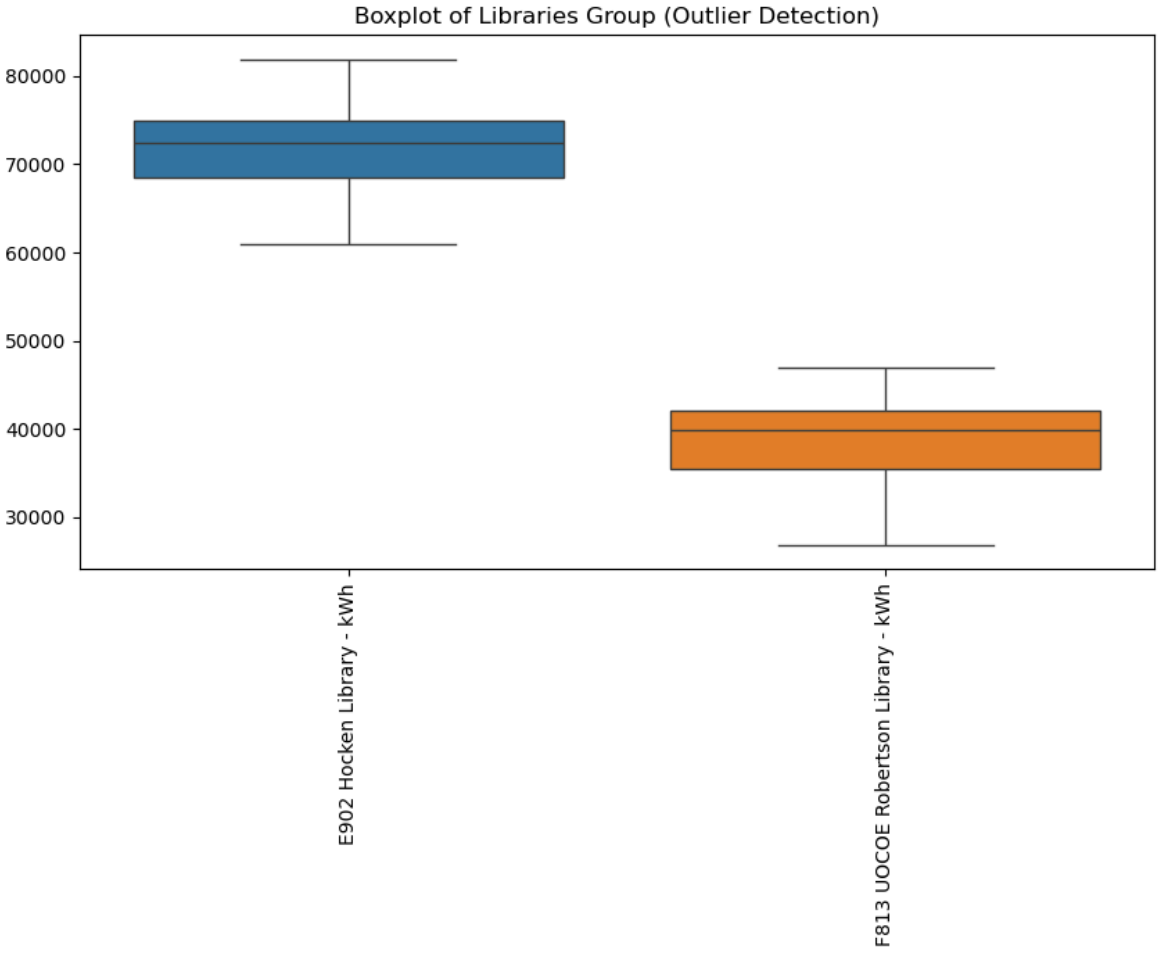

```
Skipping Humanities - Not enough valid columns for correlation analysis.

Skipping Commerce - Not enough valid columns for correlation analysis.

Skipping Total Electricity - Not enough valid columns for correlation analysis.

Skipping ITS Servers - Not enough valid columns for correlation analysis.
```

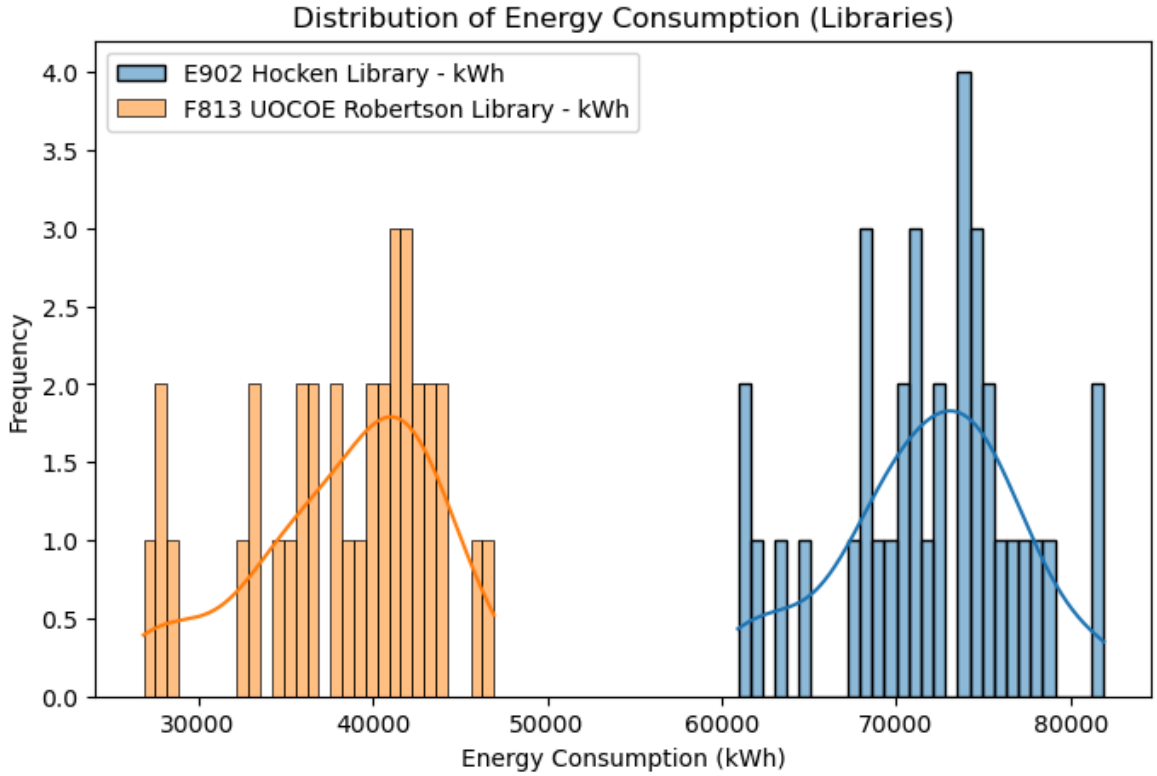
In [17]:
```
# Iterate over each group and perform analyses
for group_name, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if len(available_columns) > 1:  # Only process groups with more than one val
        print(f"\n 🔹 Analyzing {group_name} Group")

        # -------  Time Series Trend Analysis ---------
        plt.figure(figsize=(12,5))
        for col in available_columns:
            plt.plot(stream_data_filtered.index, stream_data_filtered[col], labe
        plt.xlabel("Time")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Over Time ({group_name})")
        plt.legend()
        plt.show()

        # ------- 🚨 Boxplot for Outlier Detection ---------
        plt.figure(figsize=(10,5))
        sns.boxplot(data=stream_data_filtered[available_columns])
        plt.xticks(rotation=90)
        plt.title(f"Boxplot of {group_name} Group (Outlier Detection)")
```

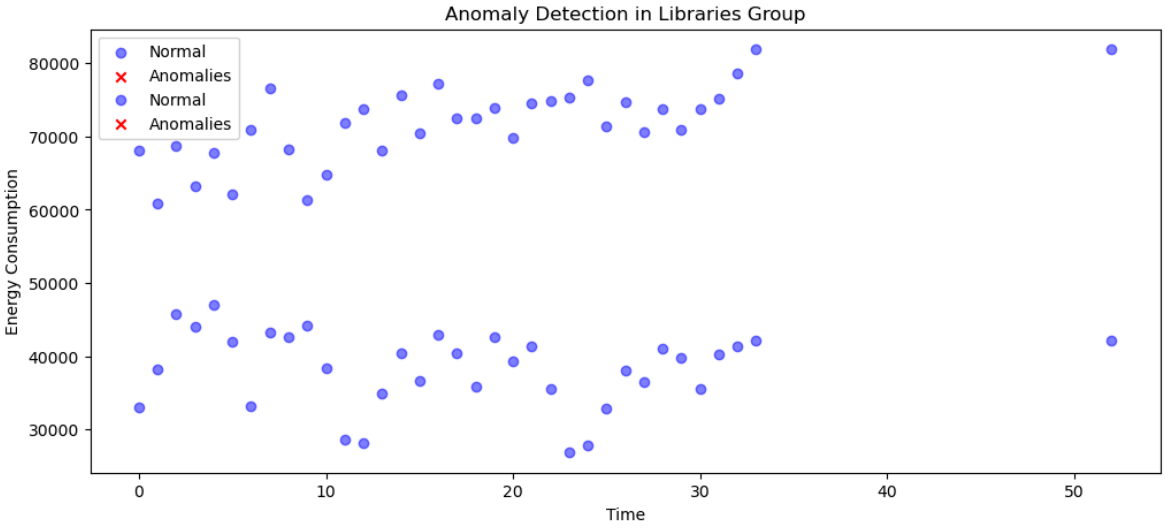
```
        plt.show()

        # ------- 📉 Histogram for Distribution ---------
        plt.figure(figsize=(8,5))
        for col in available_columns:
            sns.histplot(stream_data_filtered[col], bins=30, kde=True, label=col
        plt.xlabel("Energy Consumption (kWh)")
        plt.ylabel("Frequency")
        plt.title(f"Distribution of Energy Consumption ({group_name})")
        plt.legend()
        plt.show()

        # ------- 🎈 Anomaly Detection using Z-score ---------
        z_scores = stream_data_filtered[available_columns].apply(lambda x: (x -
        anomalies = z_scores[(z_scores.abs() > 3).any(axis=1)]

        # Plot anomalies
        plt.figure(figsize=(12,5))
        for col in available_columns:
            plt.scatter(stream_data_filtered.index, stream_data_filtered[col], l
            plt.scatter(anomalies.index, anomalies[col], color='red', label="Anc
        plt.xlabel("Time")
        plt.ylabel("Energy Consumption")
        plt.title(f"Anomaly Detection in {group_name} Group")
        plt.legend()
        plt.show()

    else:
        print(f"\nSkipping {group_name} - Not enough valid columns for analysis.
```

- Analyzing Libraries Group

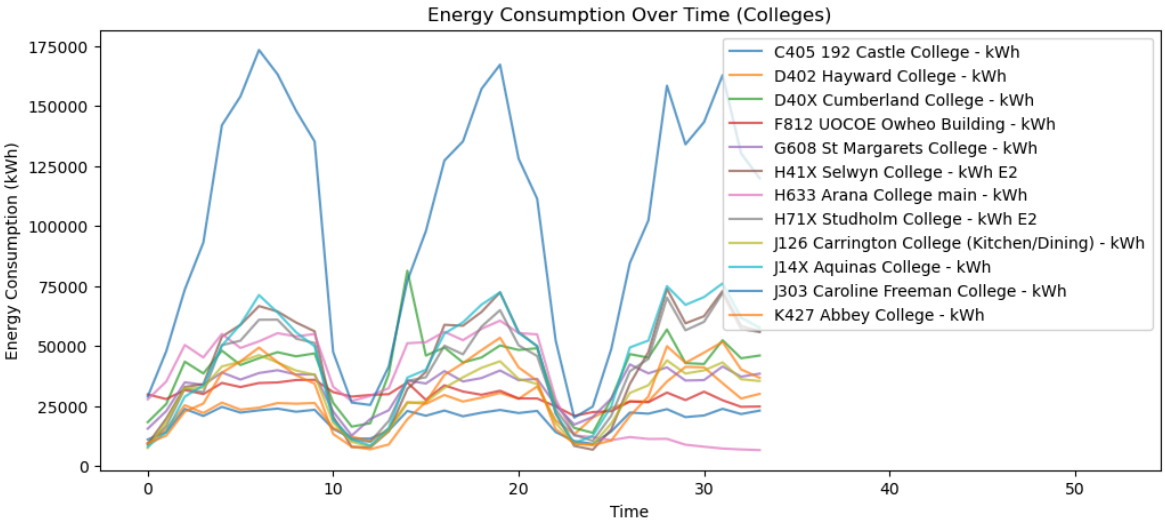


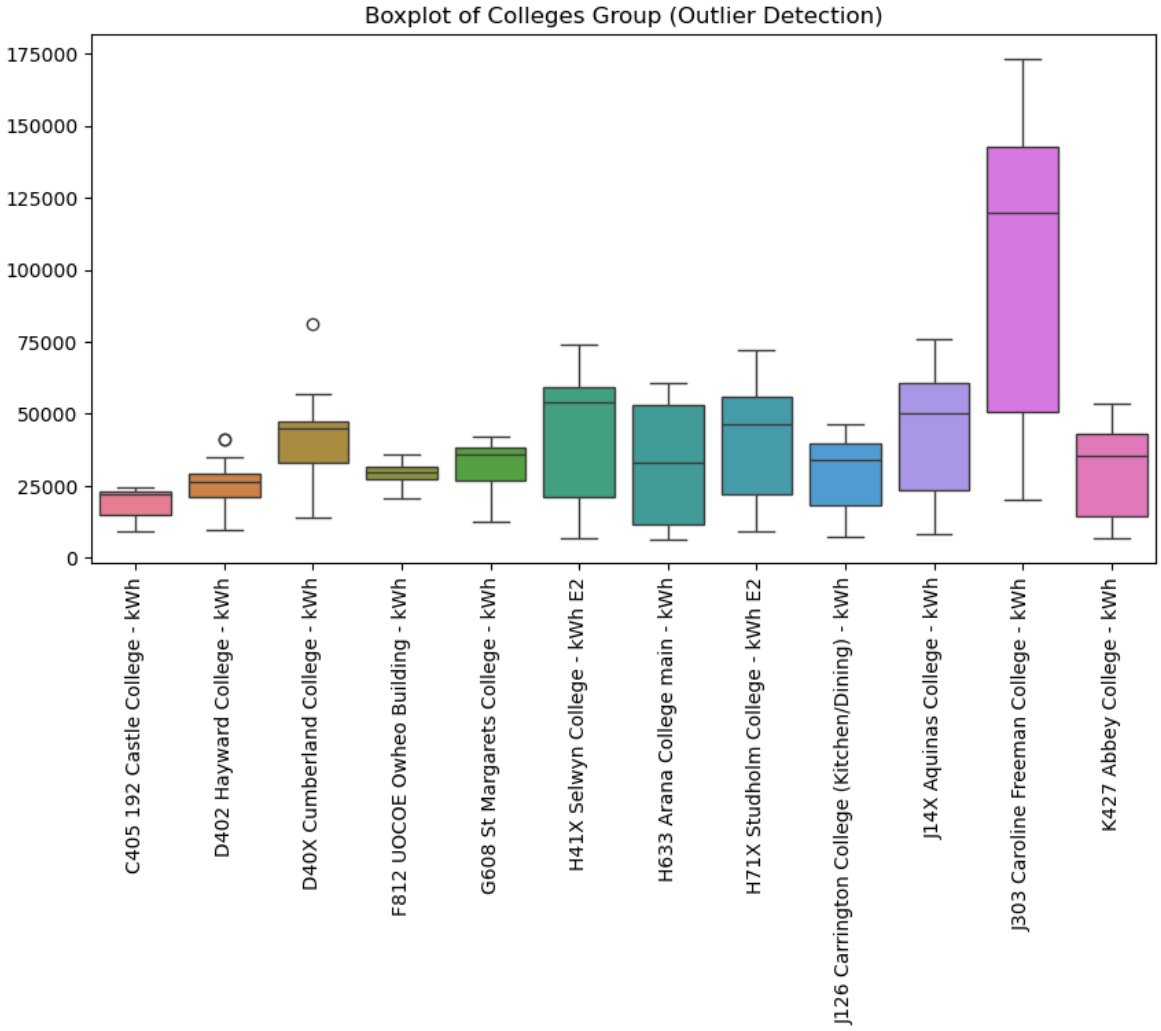

Boxplot of Libraries Group (Outlier Detection)
80000
70000
60000
50000
40000
30000
E902 Hocken Library - kWh
F813 UOCOE Robertson Library - kWh

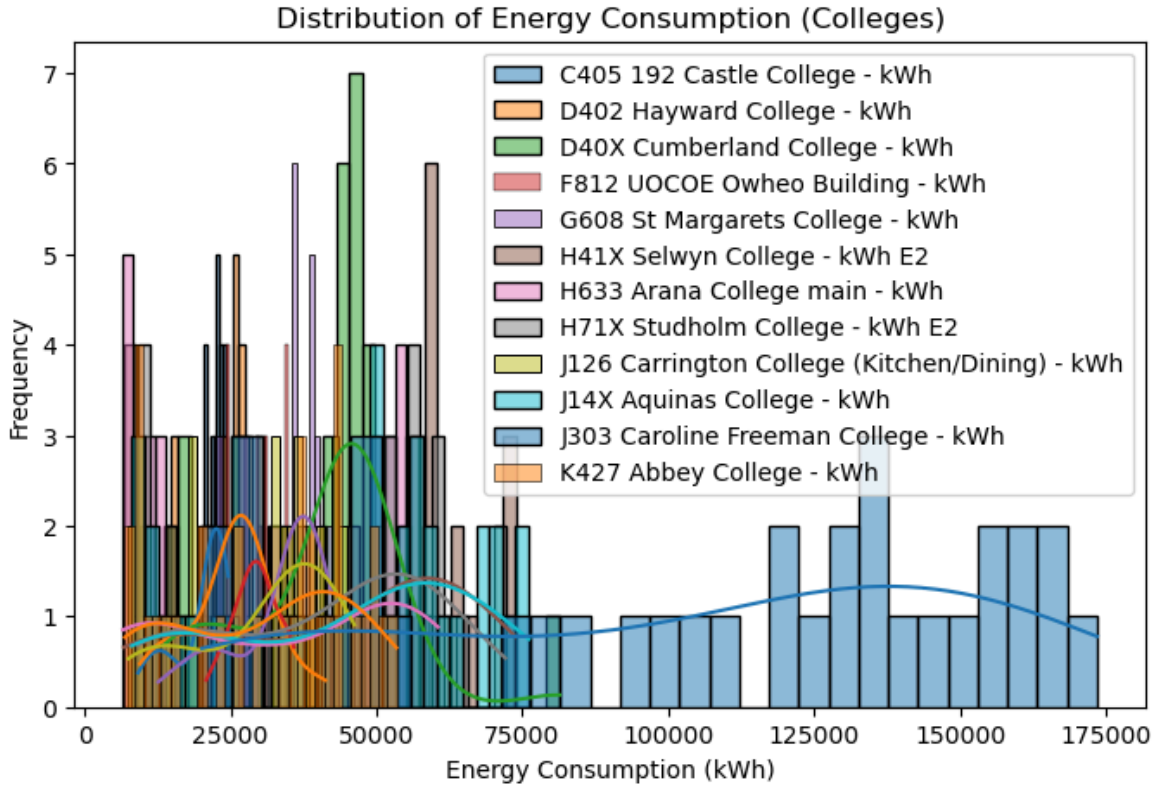

Distribution of Energy Consumption (Libraries)
E902 Hocken Library - kWh
F813 UOCOE Robertson Library - kWh
Frequency
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
30000
40000
50000
60000
70000
80000
Energy Consumption (kWh)

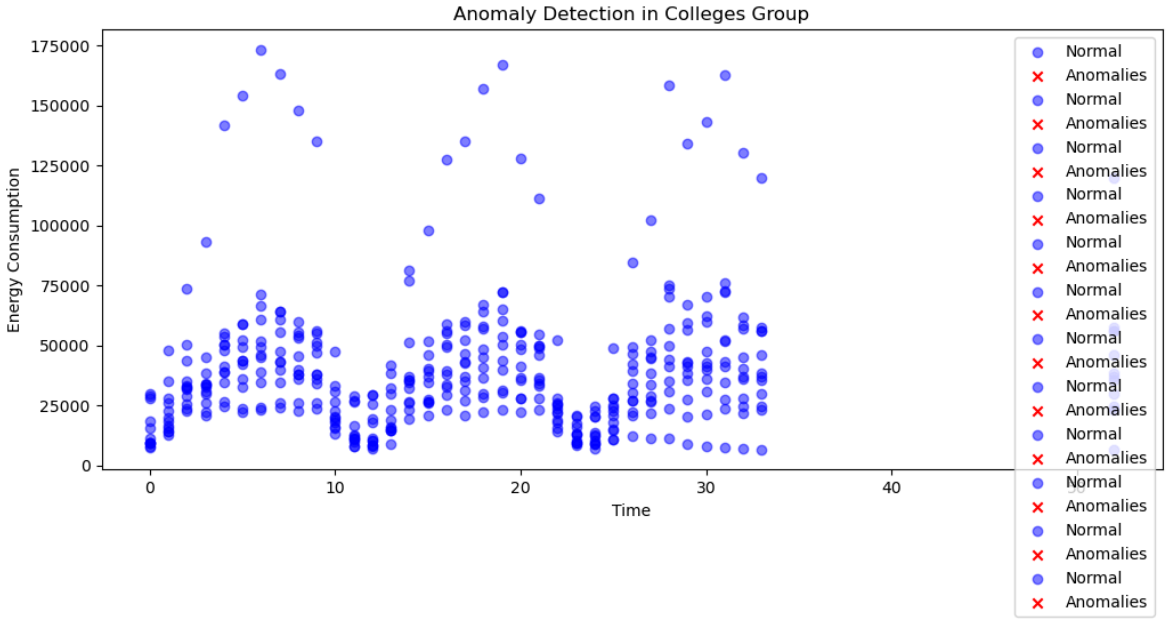


- Analyzing Colleges Group

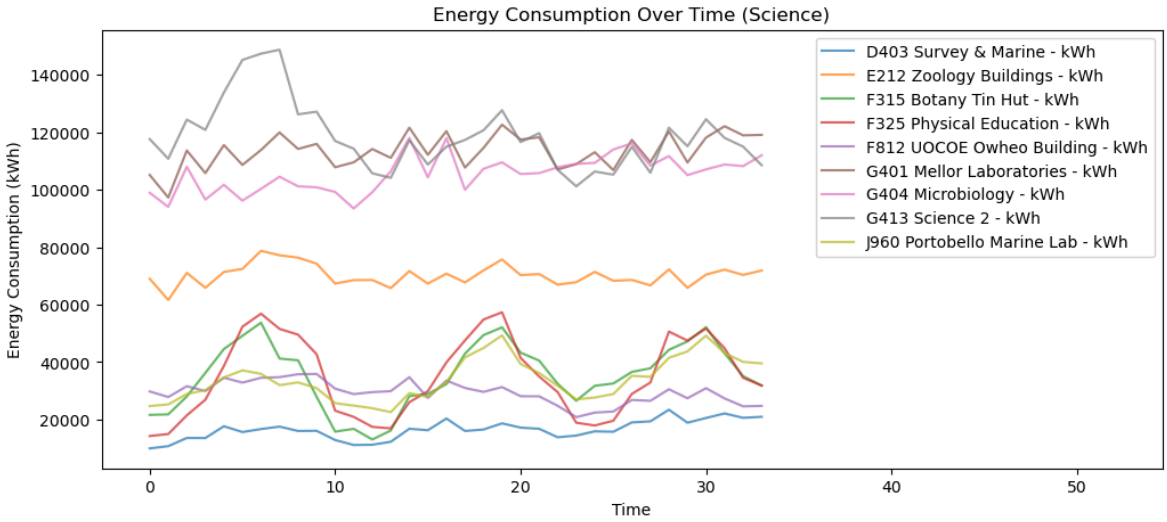


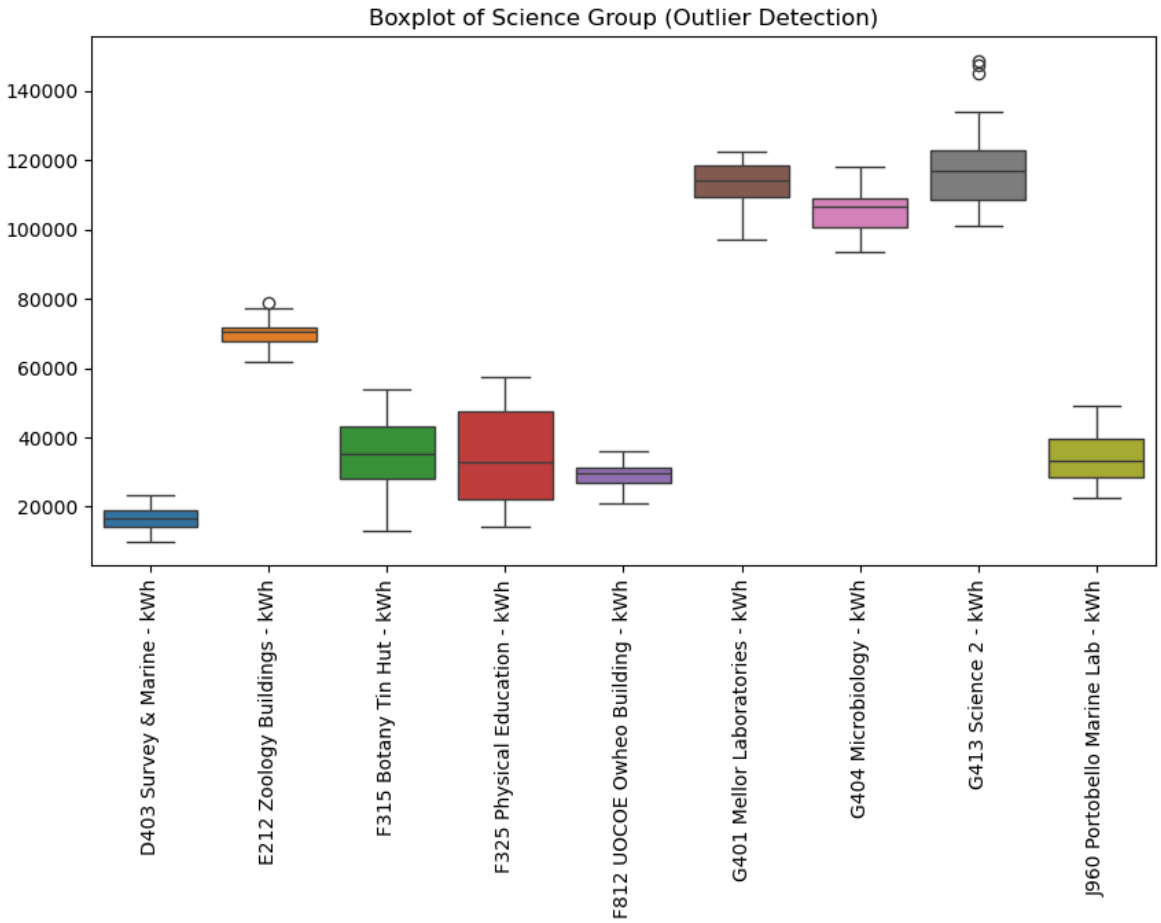

Boxplot of Colleges Group (Outlier Detection)
175000
150000
125000
100000
75000
50000
25000
0
C405 192 Castle College - kWh
D402 Hayward College - kWh
D40X Cumberland College - kWh
F812 UOCOE Owheo Building - kWh
G608 St Margarets College - kWh
H41X Selwyn College - kWh E2
H633 Arana College main - kWh
H71X Studholm College - kWh E2
J126 Carrington College (Kitchen/Dining) - kWh
J14X Aquinas College - kWh
J303 Caroline Freeman College - kWh
K427 Abbey College - kWh

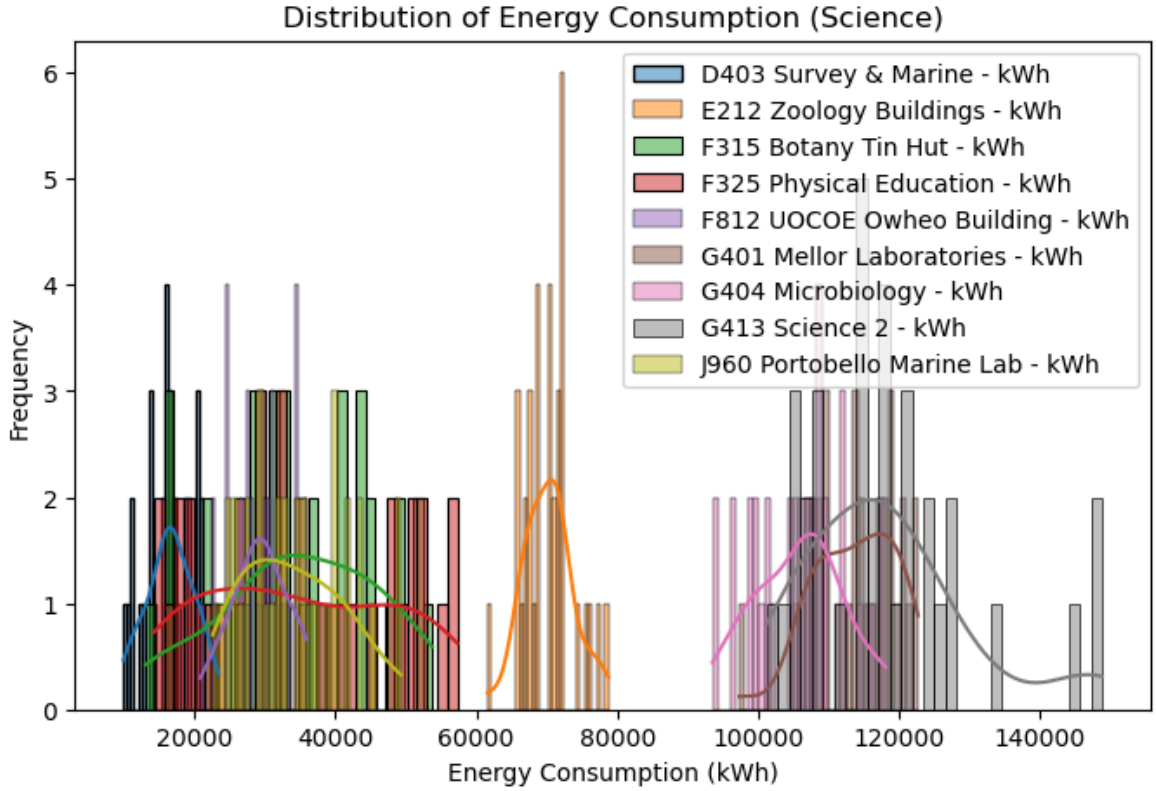

Distribution of Energy Consumption (Colleges)
Frequency
Energy Consumption (kWh)
0
25000
50000
75000
100000
125000
150000
175000
C405 192 Castle College - kWh
D402 Hayward College - kWh
D40X Cumberland College - kWh
F812 UOCOE Owheo Building - kWh
G608 St Margarets College - kWh
H41X Selwyn College - 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

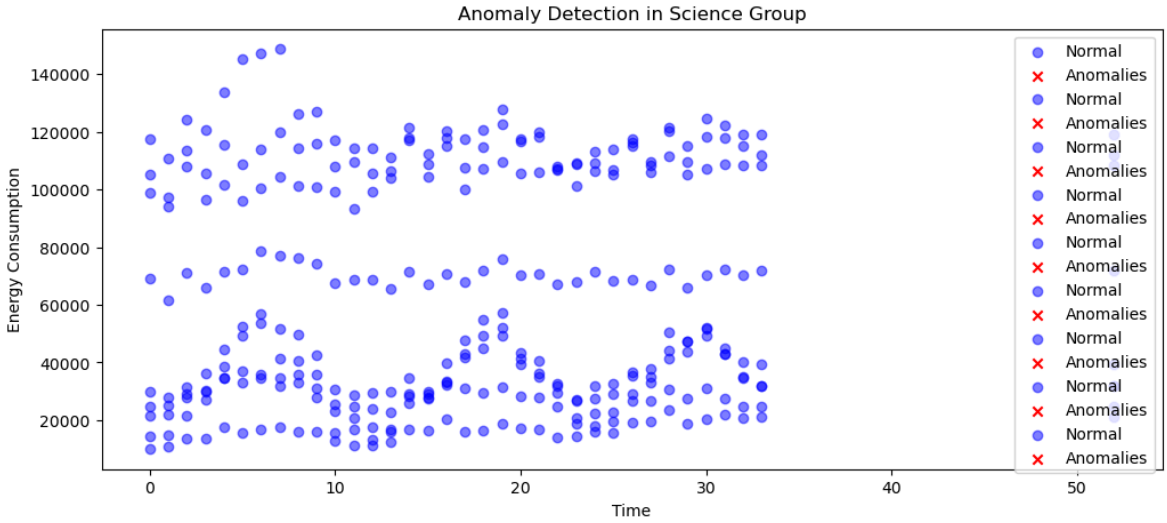


- Analyzing Science Group

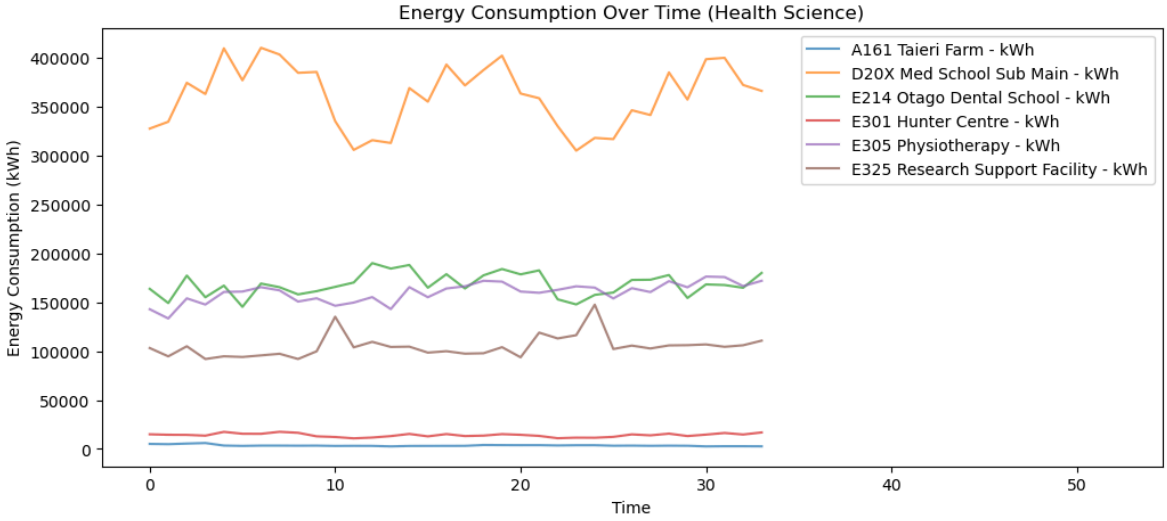


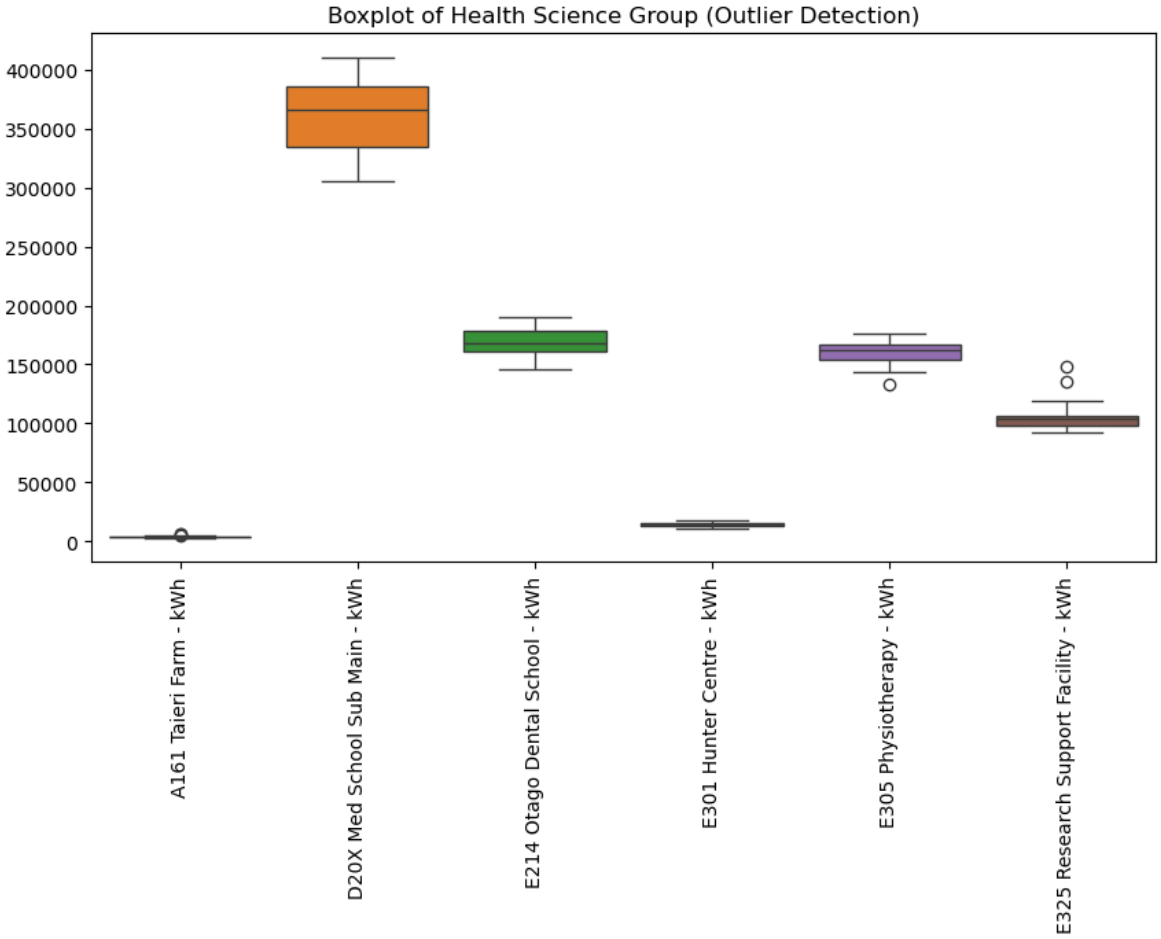

Boxplot of Science Group (Outlier Detection)
140000
120000
100000
80000
60000
40000
20000
D403 Survey & Marine - kWh
E212 Zoology Buildings - kWh
F315 Botany Tin Hut - kWh
F325 Physical Education - kWh
F812 UOCOE Owheo Building - kWh
G401 Mellor Laboratories - kWh
G404 Microbiology - kWh
G413 Science 2 - kWh
J960 Portobello Marine Lab - kWh

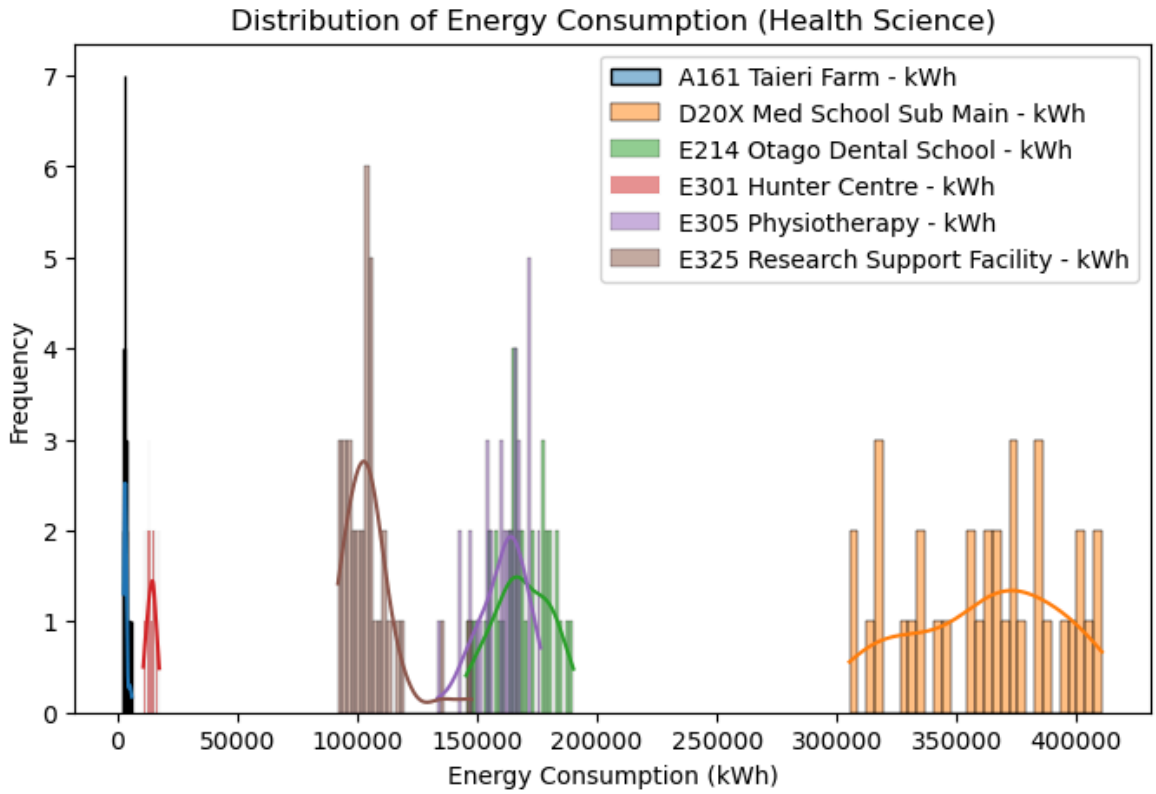

Distribution of Energy Consumption (Science)
Frequency
Energy Consumption (kWh)
0
1
2
3
4
5
6
20000
40000
60000
80000
100000
120000
140000
D403 Survey & Marine - kWh
E212 Zoology Buildings - kWh
F315 Botany Tin Hut - kWh
F325 Physical Education - kWh
F812 UOCOE Owheo Building - kWh
G401 Mellor Laboratories - kWh
G404 Microbiology - kWh
G413 Science 2 - kWh
J960 Portobello Marine Lab - kWh

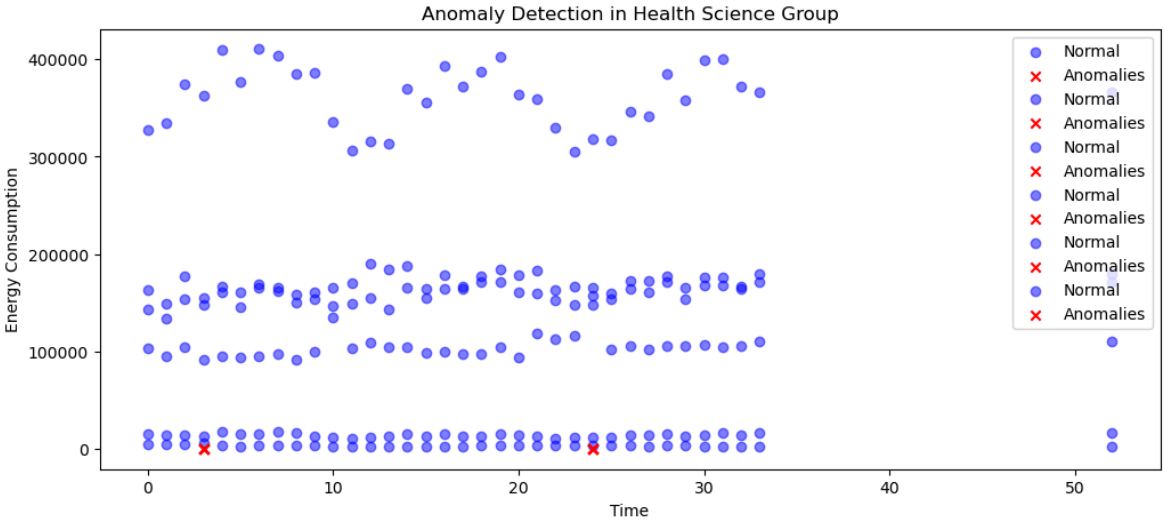


- Analyzing Health Science Group



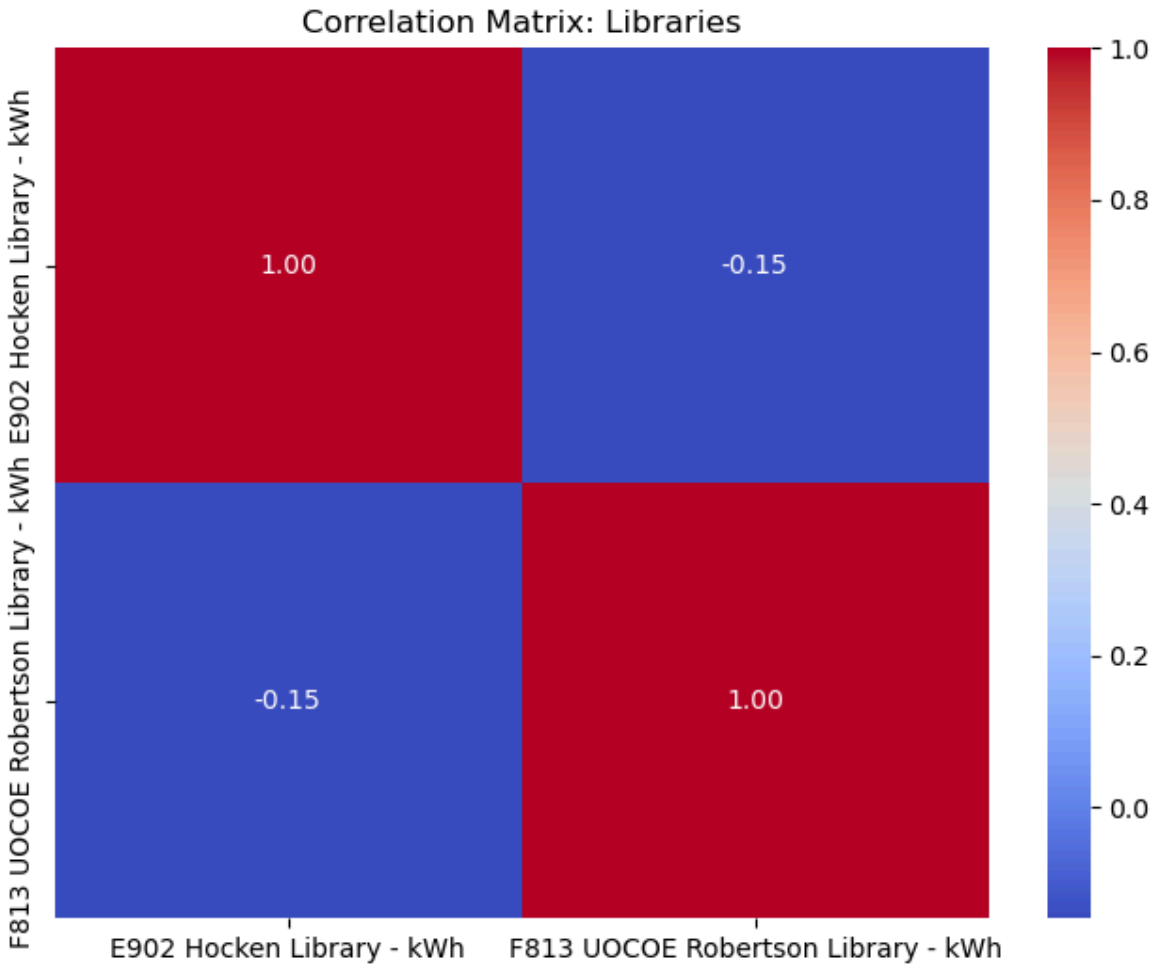

Boxplot of Health Science Group (Outlier Detection)
400000
350000
300000
250000
200000
150000
100000
50000
0
A161 Taieri Farm - kWh
D20X Med School Sub Main - kWh
E214 Otago Dental School - kWh
E301 Hunter Centre - kWh
E305 Physiotherapy - kWh
E325 Research Support Facility - kWh


Distribution of Energy Consumption (Health Science)
A161 Taieri Farm - kWh
D20X Med School Sub Main - kWh
E214 Otago Dental School - kWh
E301 Hunter Centre - kWh
E305 Physiotherapy - kWh
E325 Research Support Facility - kWh
Frequency
0
1
2
3
4
5
6
7
0
50000
100000
150000
200000
250000
300000
350000
400000
Energy Consumption (kWh)




```
Skipping Humanities - Not enough valid columns for analysis.

Skipping Commerce - Not enough valid columns for analysis.

Skipping Total Electricity - Not enough valid columns for analysis.

Skipping ITS Servers - Not enough valid columns for analysis.
```

In [18]:
```
# Libraries Group
libraries_cols = [
    "E902 Hocken Library - kWh", "F813 UOCOE Robertson Library - kWh",
    "D203 Sayers (at Adams) - kWh", "F813 Bill Robertson Library - kWh",
    "F419 ISB West Excludir - kWh", "F505 Richardson Library - kWh"
]

available_libraries = [col for col in libraries_cols if col in stream_data_filte

if len(available_libraries) > 1:
    plt.figure(figsize=(8,6))
    corr_matrix = stream_data_filtered[available_libraries].corr(numeric_only=Tr
    sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
    plt.title("Correlation Matrix: Libraries")
    plt.show()
else:
    print("\nSkipping Libraries Group - Not enough valid columns for correlation
```

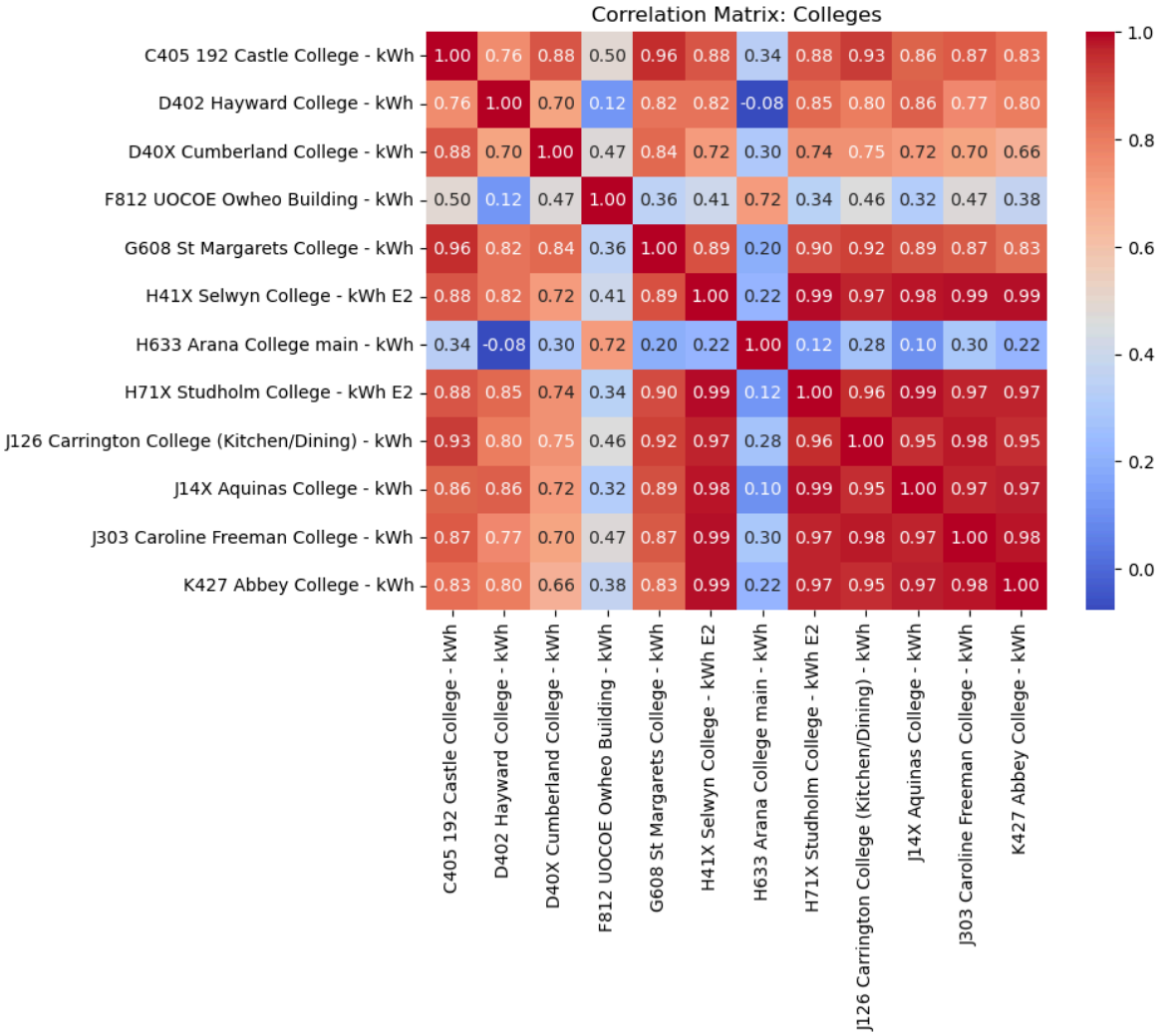


In [19]:
```
""" The correlation matrix reveals a weak negative correlation (-0.15) between e
```

Out[19]:
```
' The correlation matrix reveals a weak negative correlation (-0.15) between en
ergy consumption in Hocken Library and Robertson Library, indicating that their
energy usage patterns are largely independent. This suggests that factors such
as operational schedules, HVAC settings, and occupancy trends may differ betwee
n the two libraries, leading to minimal influence on each other's consumption.A
dditionally, external factors like weather conditions, infrastructure efficienc
y,or varying visitor counts might play a more significant role in determining e
nergy usage. To gain deeper insights, further analysis such as time-series tren
d examination, seasonal variation studies, or clustering similar buildings coul
d help identify key drivers of energy consumption across different facilities.'
```

In [20]:
```
# Colleges Group
colleges_cols = [
    "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"
]

available_colleges = [col for col in colleges_cols if col in stream_data_filtere

if len(available_colleges) > 1:
    plt.figure(figsize=(8,6))
    corr_matrix = stream_data_filtered[available_colleges].corr(numeric_only=Tru
    sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
```

```
    plt.title("Correlation Matrix: Colleges")
    plt.show()
else:
    print("\nSkipping Colleges Group - Not enough valid columns for correlation
```

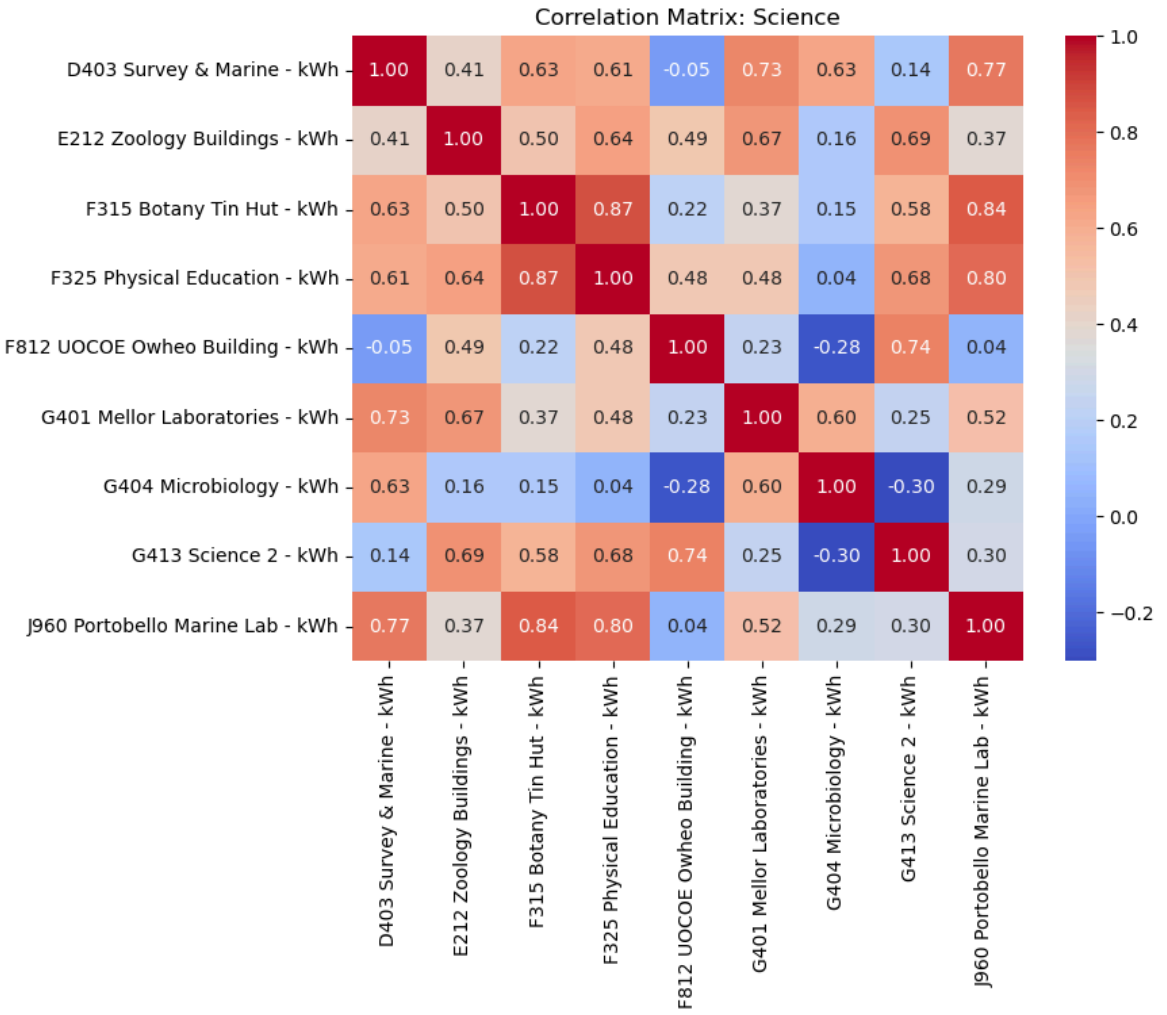


In [21]:
```
"""The correlation matrix for colleges reveals strong positive correlations (abo
```

Out[21]:
```
'The correlation matrix for colleges reveals strong positive correlations (abov
e 0.8) among most buildings, indicating similar energy consumption patterns, li
kely due to shared occupancy levels, HVAC systems, or operational schedules. No
tably, Selwyn, Studholm, Carrington, Caroline Freeman, and Aquinas Colleges (~
0.97-0.99) exhibit highly synchronized energy usage, suggesting similar demand
cycles. In contrast, Arana College (-0.08 to 0.34) and UOCOE Owheo Building (~
0.12 to 0.50) show weaker correlations, implying distinct energy consumption be
haviors, possibly due to different infrastructure or operational strategies. Fu
rther analysis through clustering, external factor integration, and forecasting
can help optimize energy management and efficiency across these buildings. '
```

In [22]:
```
# Science Group
science_cols = [
    "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:"
]
available_science = [col for col in science_cols if col in stream_data_filtered.
```

```
if len(available_science) > 1:
    plt.figure(figsize=(8,6))
    corr_matrix = stream_data_filtered[available_science].corr(numeric_only=True
    sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
    plt.title("Correlation Matrix: Science")
    plt.show()
else:
    print("\nSkipping Science Group - Not enough valid columns for correlation a
```

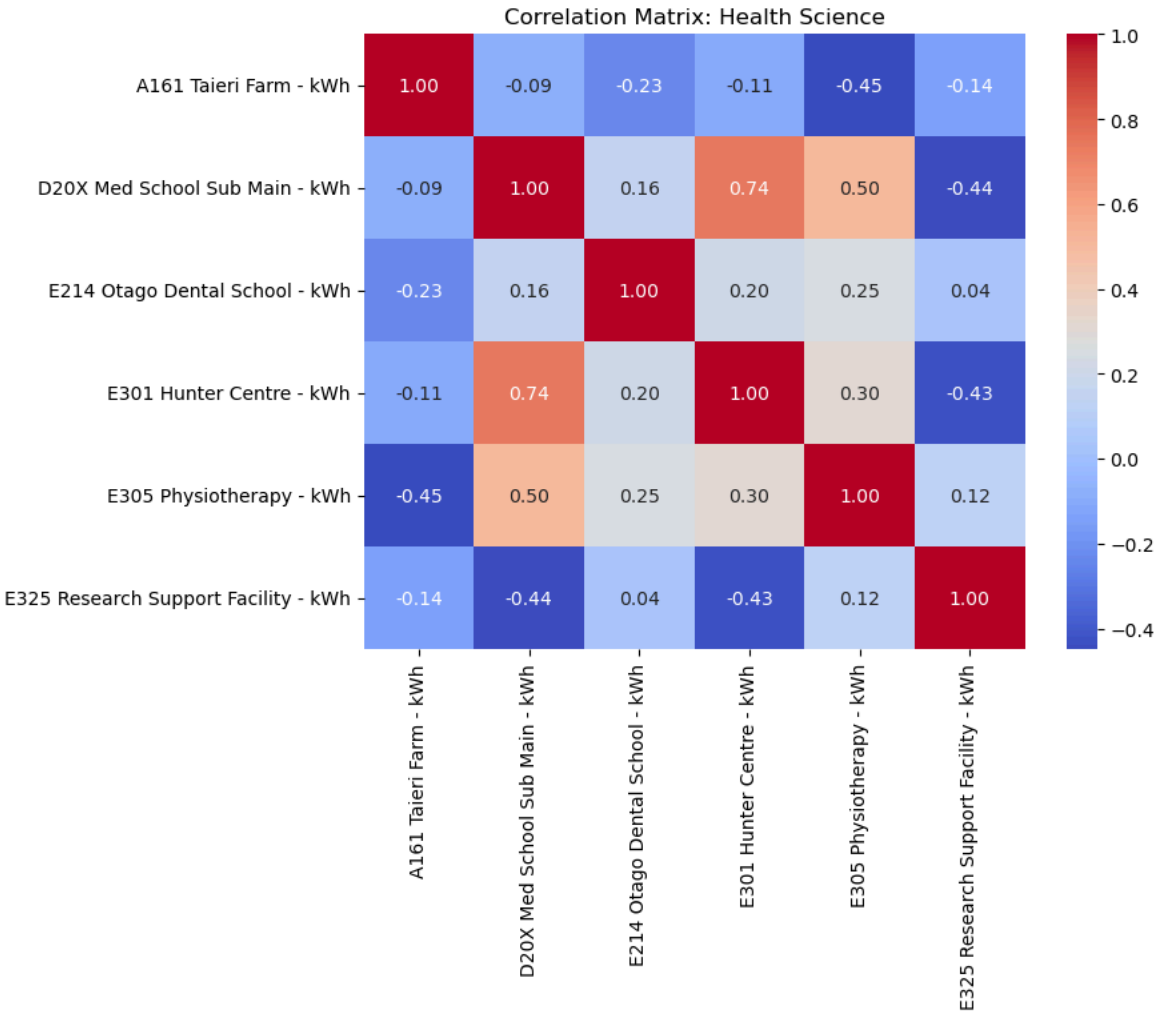


In [23]:
```
"""The correlation matrix for science buildings reveals strong positive correlat
```

Out[23]:
```
'The correlation matrix for science buildings reveals strong positive correlati
ons between Botany Tin Hut and Physical Education (~0.87) and Survey & Marine w
ith Portobello Marine Lab (~0.77), suggesting similar energy consumption patter
ns, likely due to comparable research equipment usage and operational schedule
s. Additionally, Mellor Laboratories and Survey & Marine (~0.73) indicate align
ed energy demands, potentially from high-intensity research activities. In cont
rast, UOCOE Owheo Building exhibits weak or negative correlations (-0.05 to 0.2
2) with most buildings, implying distinct operational patterns or different ene
rgy infrastructures. Similarly, Microbiology shows mixed correlations (-0.30 to
0.63), indicating fluctuating energy demands possibly due to irregular lab expe
riments. To optimize energy efficiency, further analysis should explore operati
onal factors, clustering of similar buildings, and seasonal variations in consu
mption trends. '
```

In [24]:
```
# Health Science Group
health_science_cols = [
    "A161 Taieri Farm - kWh", "D20X Med School Sub Main - kWh",
    "E214 Otago Dental School - kWh", "E301 Hunter Centre - kWh",
```

```
    "E305 Physiotherapy - kWh", "E325 Research Support Facility - kWh"
]

available_health_science = [col for col in health_science_cols if col in stream_

if len(available_health_science) > 1:
    plt.figure(figsize=(8,6))
    corr_matrix = stream_data_filtered[available_health_science].corr(numeric_on
    sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
    plt.title("Correlation Matrix: Health Science")
    plt.show()
else:
    print("\nSkipping Health Science Group - Not enough valid columns for correl
```



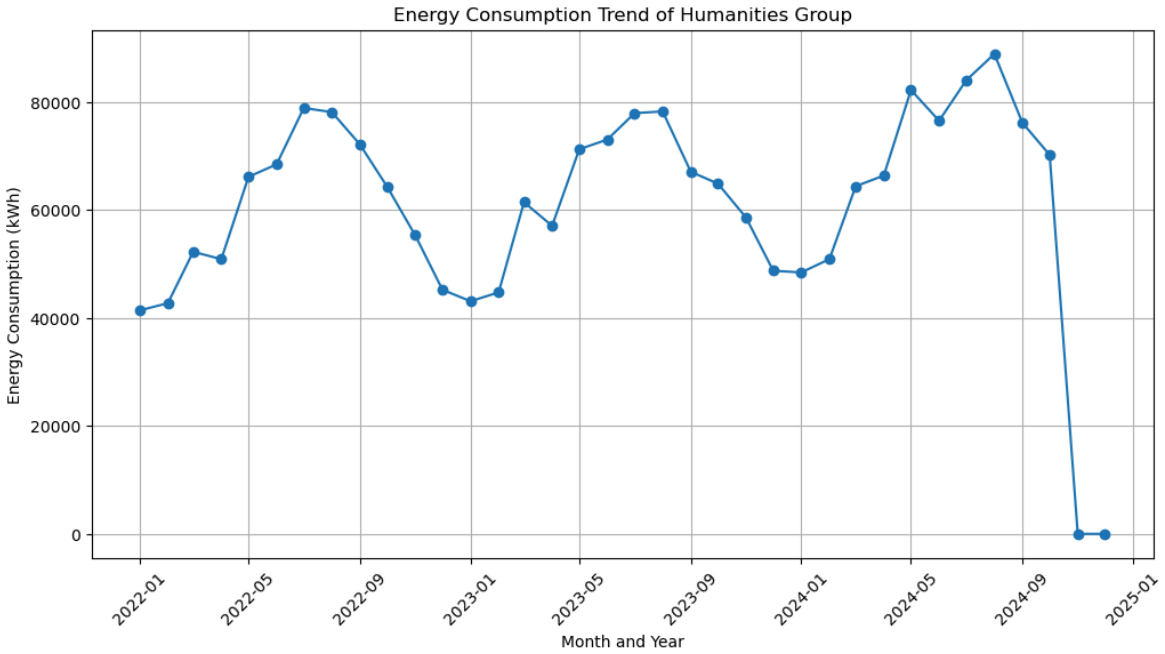
In [25]:

```
# Humanities Group
humanities_cols = [
    "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 & M
]

available_humanities = [col for col in humanities_cols if col in stream_data_fil

if len(available_humanities) > 1:
    plt.figure(figsize=(8,6))
    corr_matrix = stream_data_filtered[available_humanities].corr(numeric_only=T
    sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
    plt.title("Correlation Matrix: Humanities")
    plt.show()
```

```
else:
    print("\nSkipping Humanities Group - Not enough valid columns for correlatio
```

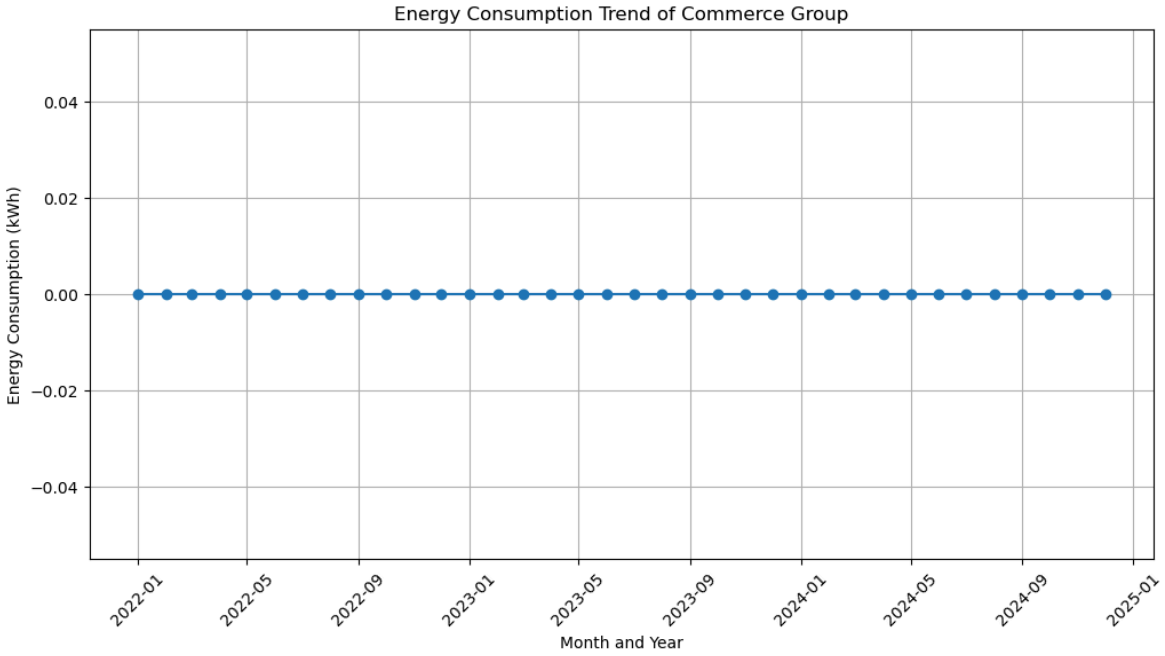
```
Skipping Humanities Group - Not enough valid columns for correlation analysis.
```

In [26]:
```
stream_data_filtered['Date'] = pd.to_datetime(stream_data_filtered['Date'], form

# Drop rows where 'Date' conversion failed
stream_data_filtered = stream_data_filtered.dropna(subset=['Date'])
available_humanities_cols = [col for col in humanities_cols if col in stream_dat

# Sum the energy consumption across the humanities group and ensure numeric valu
stream_data_filtered['Humanities Total - kWh'] = pd.to_numeric(
    stream_data_filtered[available_humanities_cols].sum(axis=1), errors='coerce'
)

# Sort the dataframe by date
stream_data_filtered = stream_data_filtered.sort_values(by='Date')

# Plot the energy consumption trend
plt.figure(figsize=(12, 6))
plt.plot(stream_data_filtered['Date'], stream_data_filtered['Humanities Total -

# Formatting the plot
plt.xlabel("Month and Year")
plt.ylabel("Energy Consumption (kWh)")
plt.title("Energy Consumption Trend of Humanities Group")
plt.xticks(rotation=45)
plt.grid(True)

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



In [27]:
```
commerce_cols = [
    "F506 Commerce (Main) - kWh", "F511 Commerce (Annex) - kWh",
    "F512 Commerce (Castle) - kWh", "F513 Commerce (Wall St) - kWh"
]

available_commerce_cols = [col for col in commerce_cols if col in stream_data_fi
```

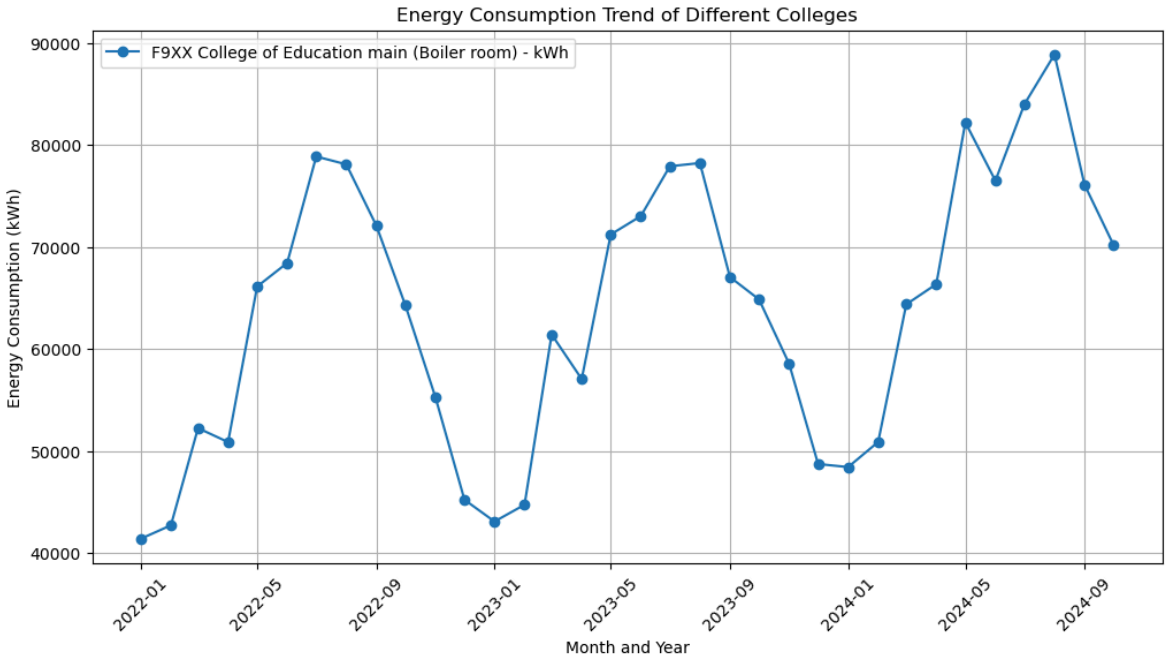
```
# Sum the energy consumption across the commerce group and ensure numeric values
stream_data_filtered['Commerce Total - kWh'] = pd.to_numeric(
    stream_data_filtered[available_commerce_cols].sum(axis=1), errors='coerce'
)

# Sort the dataframe by date
stream_data_filtered = stream_data_filtered.sort_values(by='Date')

# Plot the energy consumption trend for Commerce Group
plt.figure(figsize=(12, 6))
plt.plot(stream_data_filtered['Date'], stream_data_filtered['Commerce Total - kW

# Formatting the plot
plt.xlabel("Month and Year")
plt.ylabel("Energy Consumption (kWh)")
plt.title("Energy Consumption Trend of Commerce Group")
plt.xticks(rotation=45)
plt.grid(True)

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



In [28]:

```
college_cols = [
    "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",
    "F506 Commerce (Main) - kWh",
    "F511 Commerce (Annex) - kWh",
    "F512 Commerce (Castle) - kWh",
    "F513 Commerce (Wall St) - kWh"
]

# Check if the columns exist in the dataset
available_college_cols = [col for col in college_cols if col in stream_data_filt

# Sort the dataframe by date
stream_data_filtered = stream_data_filtered.sort_values(by='Date')
```

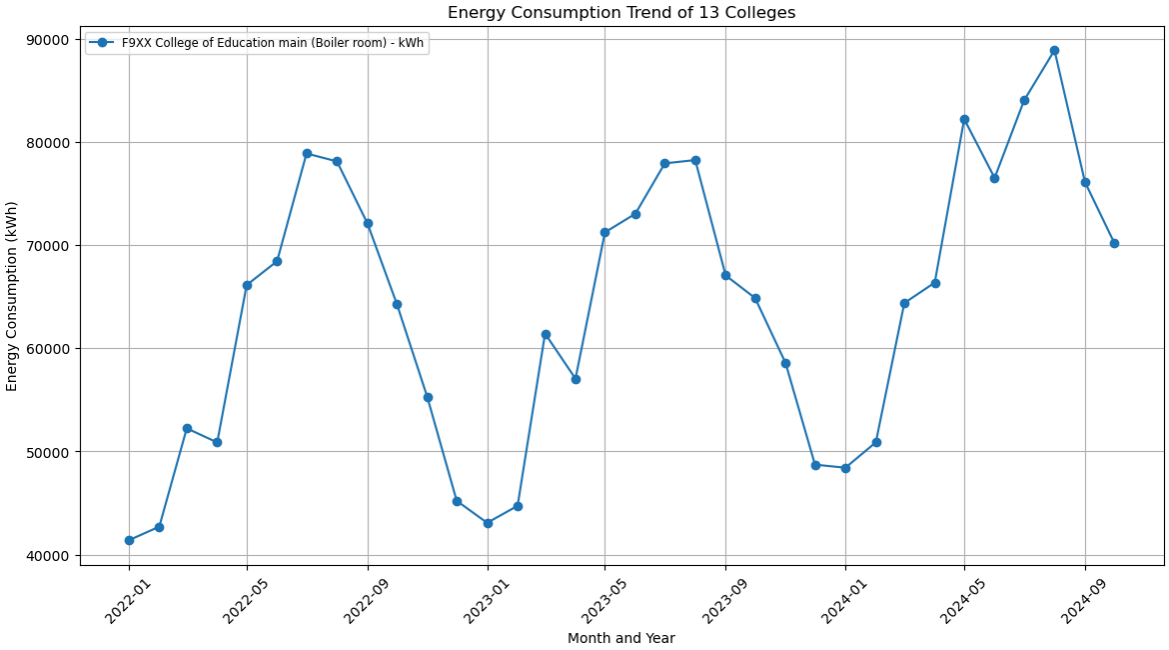
```
# Plot energy consumption trends for each college in different colors
plt.figure(figsize=(12, 6))

for col in available_college_cols:
    plt.plot(stream_data_filtered['Date'], stream_data_filtered[col], marker='o'

# Formatting the plot
plt.xlabel("Month and Year")
plt.ylabel("Energy Consumption (kWh)")
plt.title("Energy Consumption Trend of Different Colleges")
plt.xticks(rotation=45)
plt.grid(True)
plt.legend(loc='best')

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



In [29]:

```
college_cols = [
    "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",
    "F506 Commerce (Main) - kWh",
    "F511 Commerce (Annex) - kWh",
    "F512 Commerce (Castle) - kWh",
    "F513 Commerce (Wall St) - kWh",
    "F601 Science (Main) - kWh",
    "F602 Science (Annex) - kWh",
    "F603 Science (Research) - kWh",
    "F604 Science (Teaching) - kWh"
]

# Check if the columns exist in the dataset
available_college_cols = [col for col in college_cols if col in stream_data_filt

# Sort the dataframe by date
stream_data_filtered = stream_data_filtered.sort_values(by='Date')
```

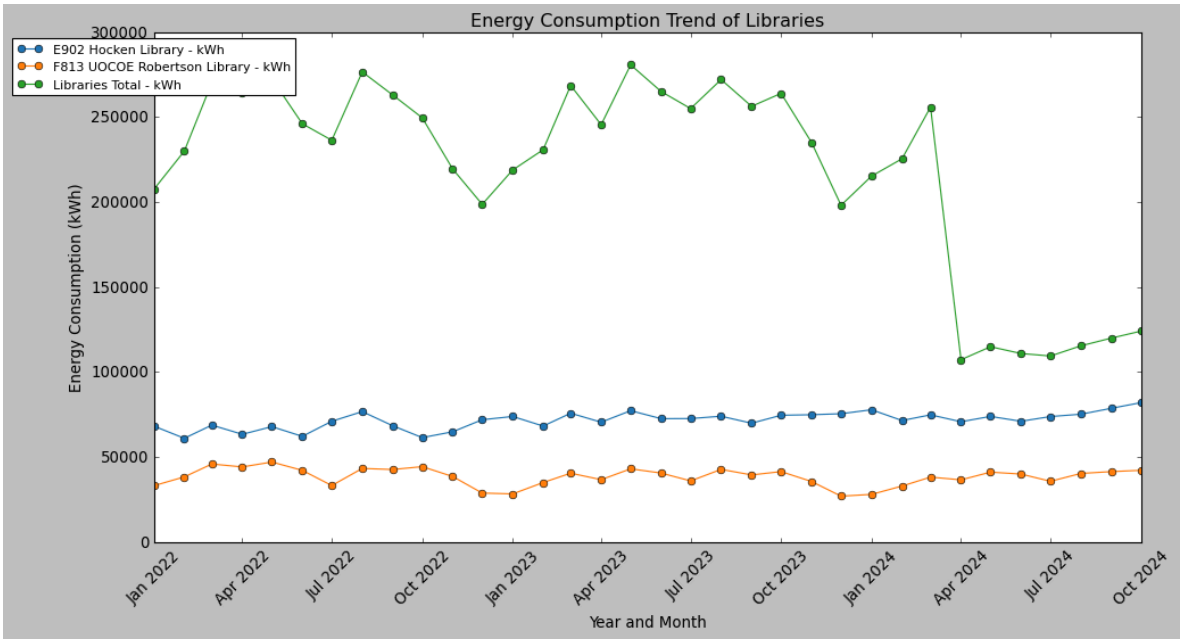
```
# Plot energy consumption trends for each of the 13 colleges in different colors
plt.figure(figsize=(14, 7))

for col in available_college_cols:
    plt.plot(stream_data_filtered['Date'], stream_data_filtered[col], marker='o'

# Formatting the plot
plt.xlabel("Month and Year")
plt.ylabel("Energy Consumption (kWh)")
plt.title("Energy Consumption Trend of 13 Colleges")
plt.xticks(rotation=45)
plt.grid(True)
plt.legend(loc='best', fontsize='small')

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



In [30]:

```
library_cols = [
    "E902 Hocken Library - kWh",
    "F813 UOCOE Robertson Library - kWh",
    "Libraries Total - kWh"
]

# Filter only the columns that exist in the dataset
available_library_cols = [col for col in library_cols if col in stream_data_filt

# Convert library energy consumption columns to numeric
for col in available_library_cols:
    stream_data_filtered[col] = pd.to_numeric(stream_data_filtered[col], errors=

# Set a plain white background
plt.style.use('classic')

# Define a dark color palette for better contrast
dark_palette = sns.color_palette("tab10", len(available_library_cols))  # Using

# Set the figure size
plt.figure(figsize=(14, 7))
```

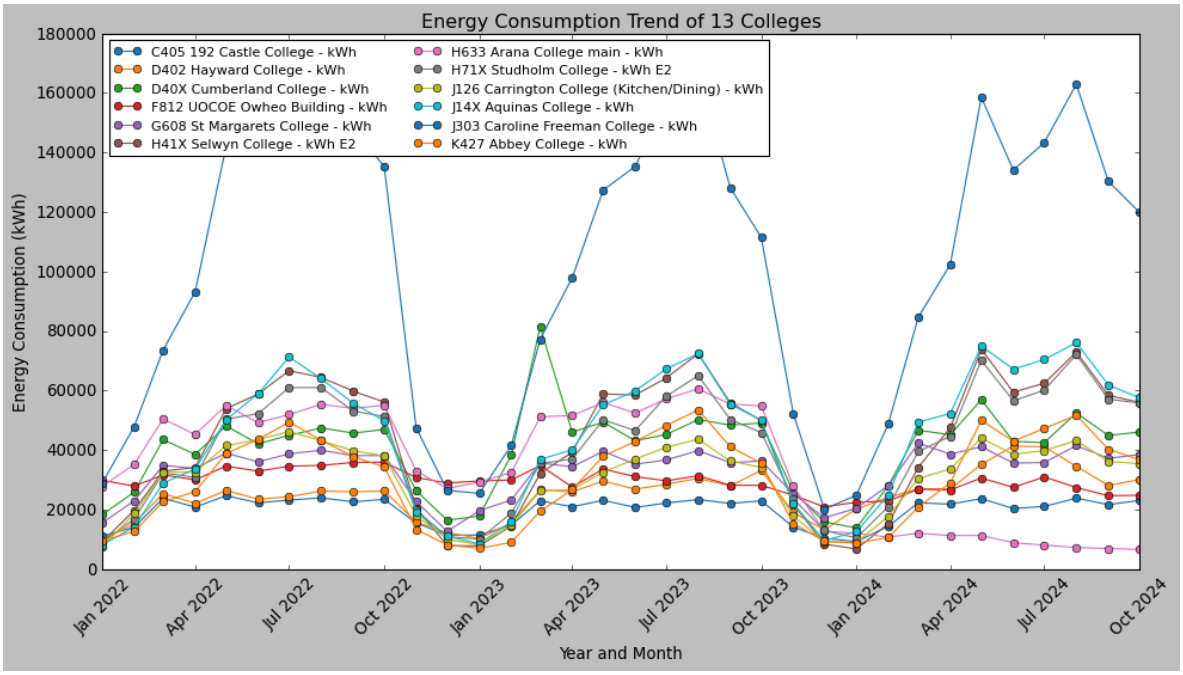
```
# Plot energy consumption trends for each of the libraries in different colors
for idx, col in enumerate(available_library_cols):
    plt.plot(stream_data_filtered['Date'], stream_data_filtered[col],
             marker='o', linestyle='-', color=dark_palette[idx % 10], label=col)

# Formatting the plot
plt.xlabel("Year and Month")
plt.ylabel("Energy Consumption (kWh)")
plt.title("Energy Consumption Trend of Libraries")
plt.xticks(rotation=45)
plt.legend(loc='upper left', fontsize='small', ncol=1, bbox_to_anchor=(-0.15, 1)

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



In [31]:

```
import seaborn as sns
import matplotlib.pyplot as plt
college_cols = [
    "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 only the columns that exist in the dataset
available_college_cols = [col for col in college_cols if col in stream_data_filt
# Set a plain white background
plt.style.use('classic')

# Define a dark color palette for better contrast
dark_palette = sns.color_palette("tab10", len(available_college_cols))  # Using
```

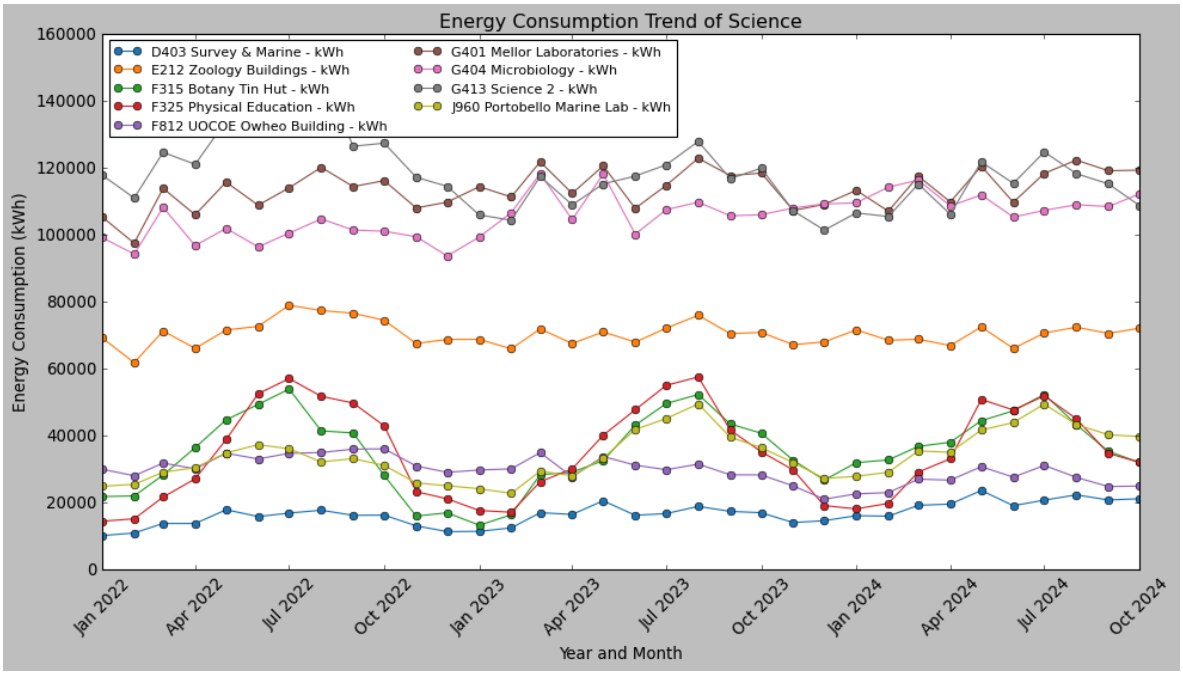
```
# Set the figure size
plt.figure(figsize=(14, 7))

# Plot energy consumption trends for each of the 13 colleges in different colors
for idx, col in enumerate(available_college_cols):
    plt.plot(stream_data_filtered['Date'], stream_data_filtered[col],
             marker='o', linestyle='-', color=dark_palette[idx % 10], label=col)

# Formatting the plot
plt.xlabel("Year and Month")
plt.ylabel("Energy Consumption (kWh)")
plt.title("Energy Consumption Trend of 13 Colleges")
plt.xticks(rotation=45)
plt.legend(loc='upper left', fontsize='small', ncol=2)  # Adjust legend to fit a

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



In [32]:

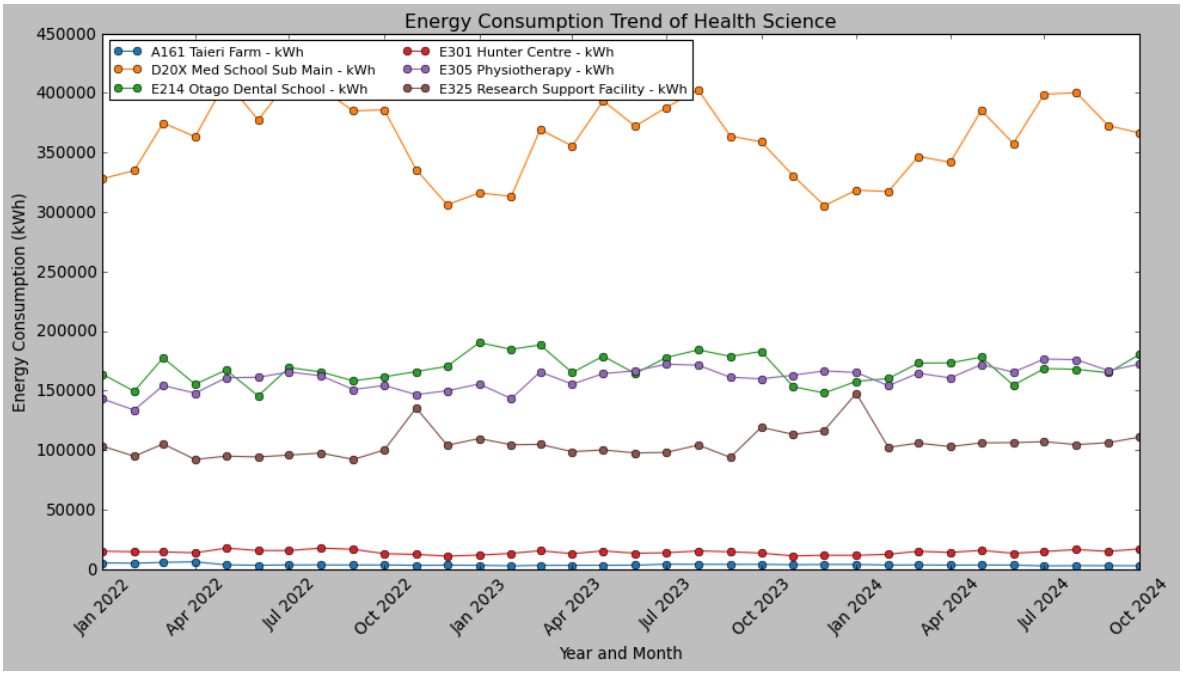
```
categories = {
    "Science": [
        "D403 Survey & Marine - kWh", "E212 Zoology Buildings - kWh",
        "F315 Botany Tin Hut - kWh", "F325 Physical Education - kWh",
        "F812 UOCOE Owheo Building - kWh", "G401 Mellor Laboratories - kWh",
        "G404 Microbiology - kWh", "G413 Science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

}

# Set a plain white background
plt.style.use('classic')

# Iterate over each category to plot individual figures
for category, college_cols in categories.items():
    available_college_cols = [col for col in college_cols if col in stream_data_

    if available_college_cols:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
```

```
        dark_palette = sns.color_palette("tab10", len(available_college_cols))

        for idx, col in enumerate(available_college_cols):
            plt.plot(stream_data_filtered['Date'], stream_data_filtered[col],
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Trend of {category}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=2)

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



In [33]:
```
categories = {

    "Health Science": [
        "A161 Taieri Farm - kWh", "D20X Med School Sub Main - kWh",
        "E214 Otago Dental School - kWh", "E301 Hunter Centre - kWh",
        "E305 Physiotherapy - kWh", "E325 Research Support Facility - kWh"
    ],

}

# Set a plain white background
plt.style.use('classic')

# Iterate over each category to plot individual figures
for category, college_cols in categories.items():
    available_college_cols = [col for col in college_cols if col in stream_data_

    if available_college_cols:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_college_cols))

        for idx, col in enumerate(available_college_cols):
            plt.plot(stream_data_filtered['Date'], stream_data_filtered[col],
```

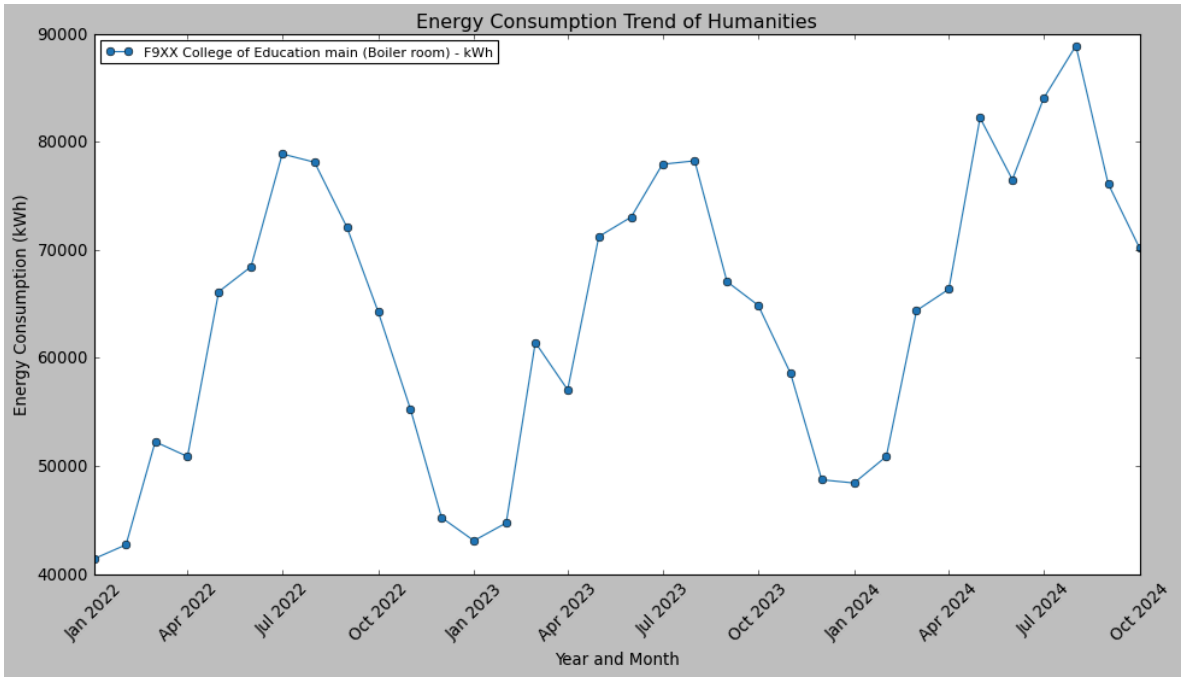
```
                    marker='o', linestyle='-', color=dark_palette[idx % 10], la

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Trend of {category}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=2)

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



In [34]:

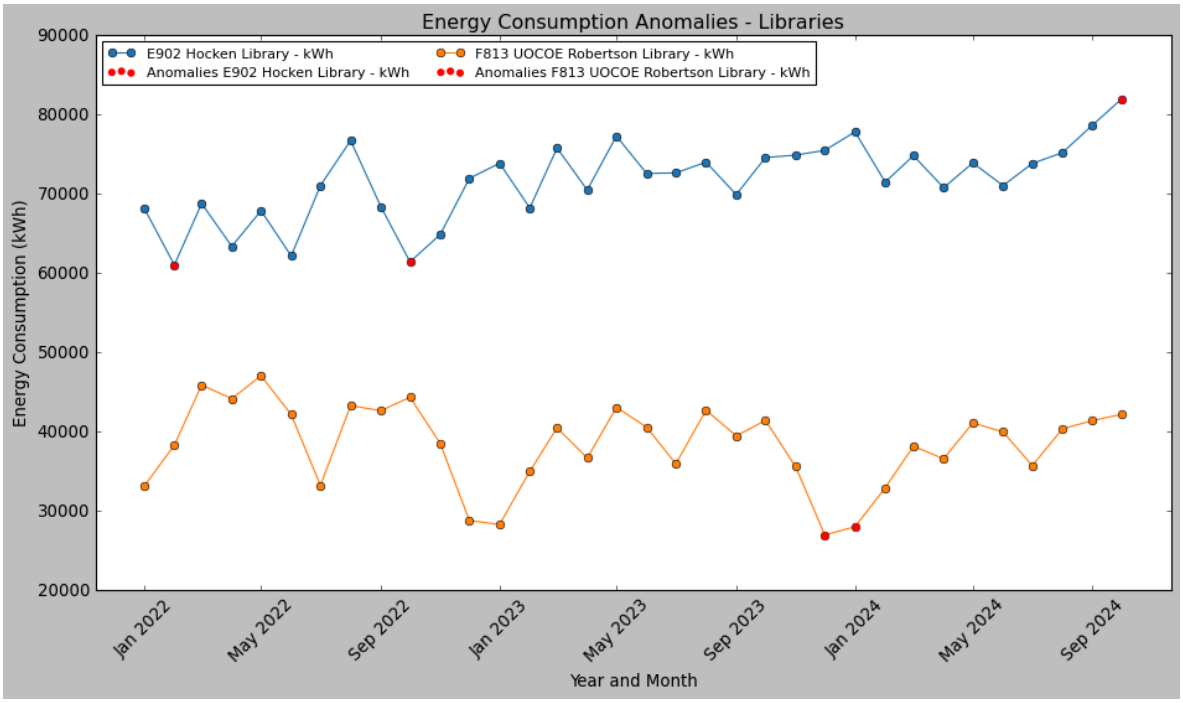
```
categories = {

    "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
    ]
}

# Set a plain white background
plt.style.use('classic')

# Iterate over each category to plot individual figures
for category, college_cols in categories.items():
    available_college_cols = [col for col in college_cols if col in stream_data_

    if available_college_cols:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_college_cols))

        for idx, col in enumerate(available_college_cols):
            plt.plot(stream_data_filtered['Date'], stream_data_filtered[col],
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
```

```
        plt.title(f"Energy Consumption Trend of {category}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=2)

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



In [35]:
```
# Define groups and their corresponding column names
groups = {
    "Libraries": [
        "E902 Hocken Library - kWh", "F813 UOCOE Robertson Library - kWh",
        "D203 Sayers (at Adams) - kWh", "F813 Bill Robertson Library - kWh",
        "F419 ISB West Excludir - kWh", "F505 Richardson Library - kWh"
    ],

}

# Set a plain white background
plt.style.use('classic')

# Iterate over each group to plot anomalies separately
for group, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if available_columns:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_columns))

        for idx, col in enumerate(available_columns):
            data_series = stream_data_filtered[col]
            mean_val, std_dev = np.mean(data_series), np.std(data_series)
            anomalies = data_series[(data_series > mean_val + 2 * std_dev) | (da

            plt.plot(stream_data_filtered['Date'], data_series,
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

            # Highlight anomalies
            plt.scatter(stream_data_filtered['Date'][anomalies.index], anomalies
```

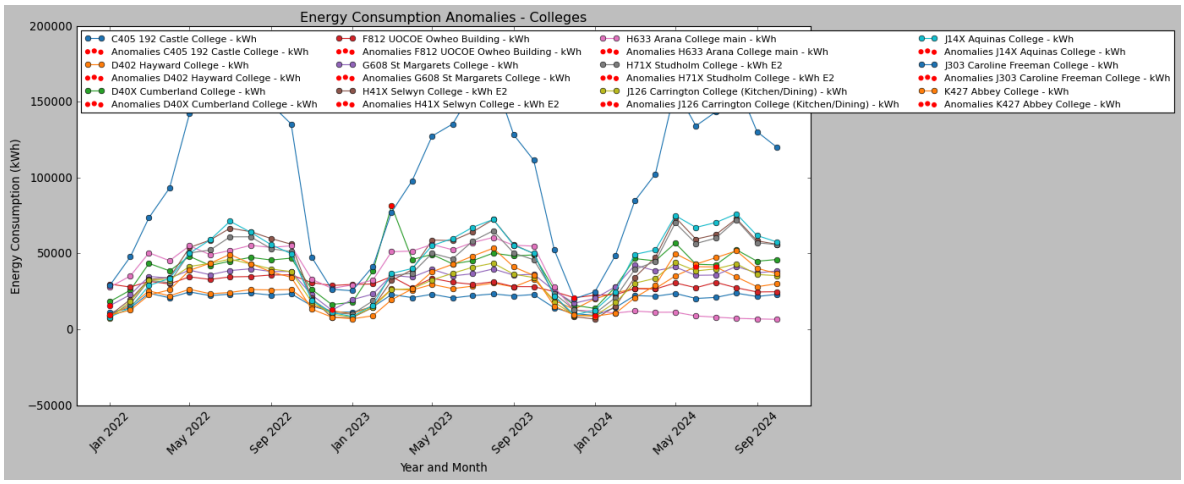
```
        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Anomalies - {group}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=2)

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



In [36]:

```
# Define groups and their corresponding column names
groups = {

    "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"
    ],

}

# Set a plain white background
plt.style.use('classic')

# Iterate over each group to plot anomalies separately
for group, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if available_columns:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_columns))

        for idx, col in enumerate(available_columns):
            data_series = stream_data_filtered[col]
            mean_val, std_dev = np.mean(data_series), np.std(data_series)
```

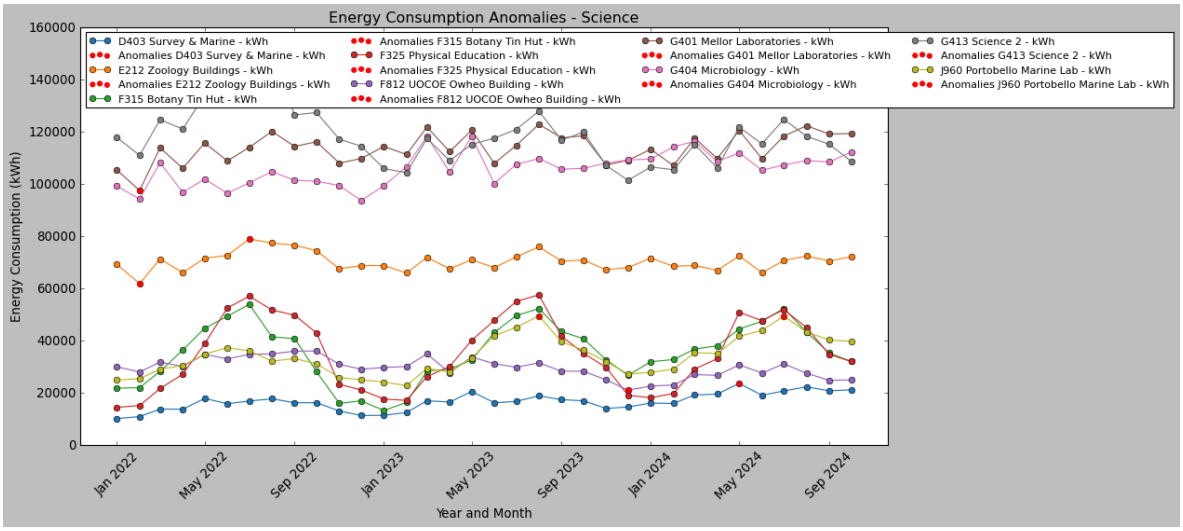
```
            anomalies = data_series[(data_series > mean_val + 2 * std_dev) | (da

            plt.plot(stream_data_filtered['Date'], data_series,
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

            # Highlight anomalies
            plt.scatter(stream_data_filtered['Date'][anomalies.index], anomalies

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Anomalies - {group}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=4)

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



In [37]:

```
# Define groups and their corresponding column names
groups = {

    "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:"
    ],

}

# Set a plain white background
plt.style.use('classic')

# Iterate over each group to plot anomalies separately
for group, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if available_columns:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_columns))

        for idx, col in enumerate(available_columns):
            data_series = stream_data_filtered[col]
```

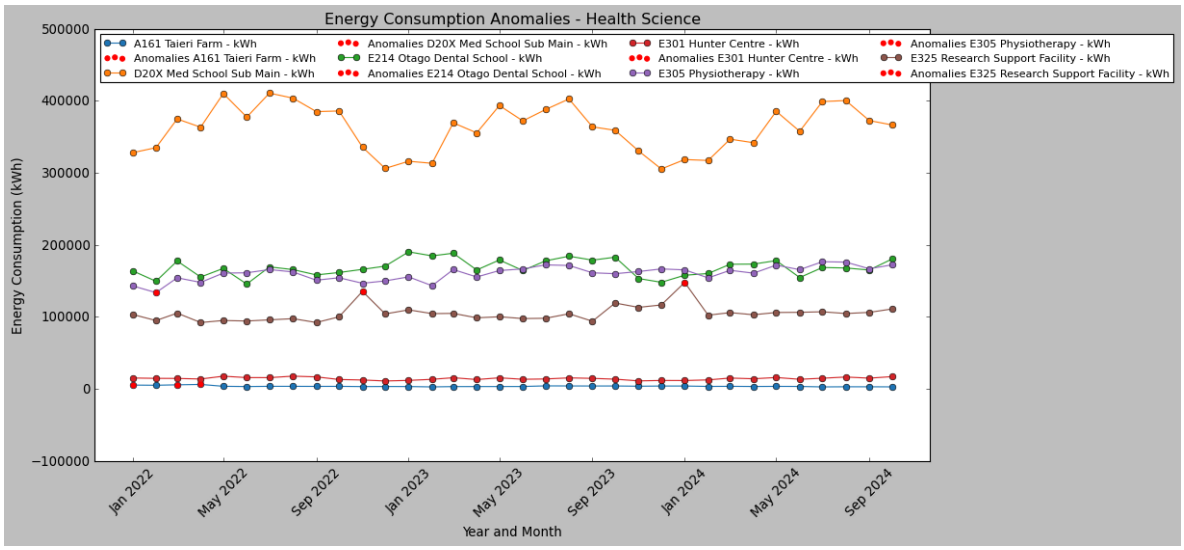
```
            mean_val, std_dev = np.mean(data_series), np.std(data_series)
            anomalies = data_series[(data_series > mean_val + 2 * std_dev) | (da

            plt.plot(stream_data_filtered['Date'], data_series,
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

            # Highlight anomalies
            plt.scatter(stream_data_filtered['Date'][anomalies.index], anomalies

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Anomalies - {group}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=4)

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



In [38]:

```
# Define groups and their corresponding column names
groups = {

    "Health Science": [
        "A161 Taieri Farm - kWh", "D20X Med School Sub Main - kWh",
        "E214 Otago Dental School - kWh", "E301 Hunter Centre - kWh",
        "E305 Physiotherapy - kWh", "E325 Research Support Facility - kWh"
    ],

}

# Set a plain white background
plt.style.use('classic')

# Iterate over each group to plot anomalies separately
for group, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if available_columns:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_columns))

        for idx, col in enumerate(available_columns):
            data_series = stream_data_filtered[col]
            mean_val, std_dev = np.mean(data_series), np.std(data_series)
```

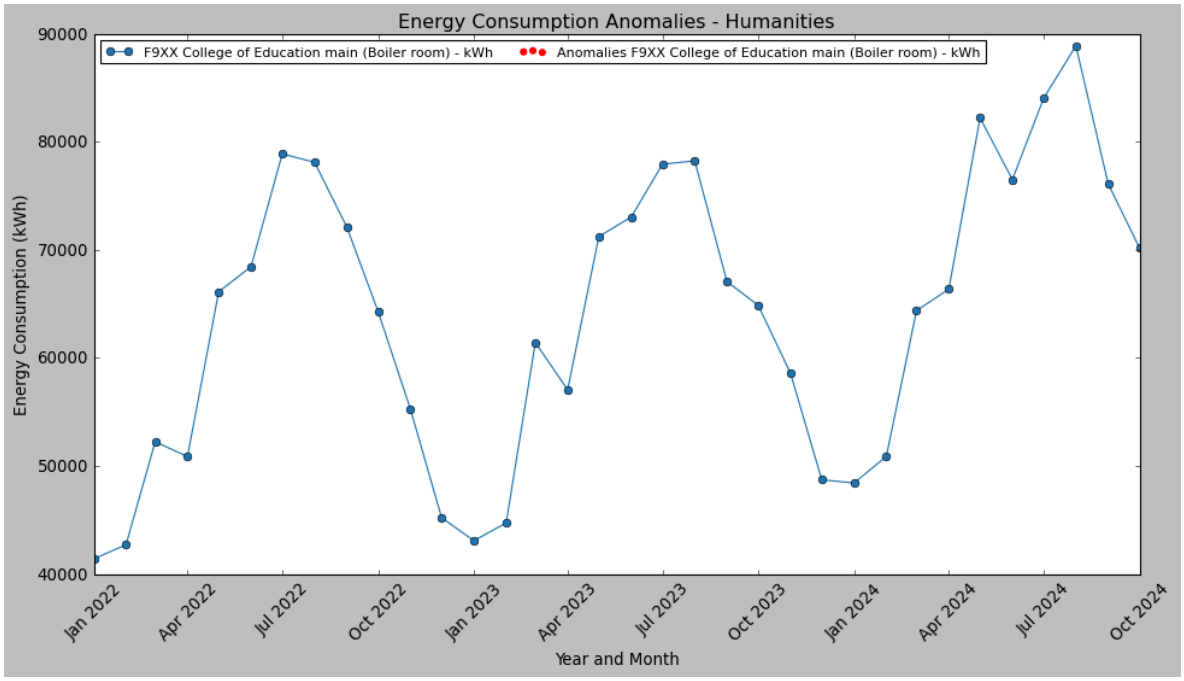
```
            anomalies = data_series[(data_series > mean_val + 2 * std_dev) | (da

            plt.plot(stream_data_filtered['Date'], data_series,
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

            # Highlight anomalies
            plt.scatter(stream_data_filtered['Date'][anomalies.index], anomalies

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Anomalies - {group}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=4)

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



In [39]:

```
# Define groups and their corresponding column names
groups = {

    "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
    ]
}

# Set a plain white background
plt.style.use('classic')

# Iterate over each group to plot anomalies separately
for group, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c

    if available_columns:  # Ensure there are valid columns to plot
        plt.figure(figsize=(14, 7))
        dark_palette = sns.color_palette("tab10", len(available_columns))

        for idx, col in enumerate(available_columns):
            data_series = stream_data_filtered[col]
            mean_val, std_dev = np.mean(data_series), np.std(data_series)
            anomalies = data_series[(data_series > mean_val + 2 * std_dev) | (da
```

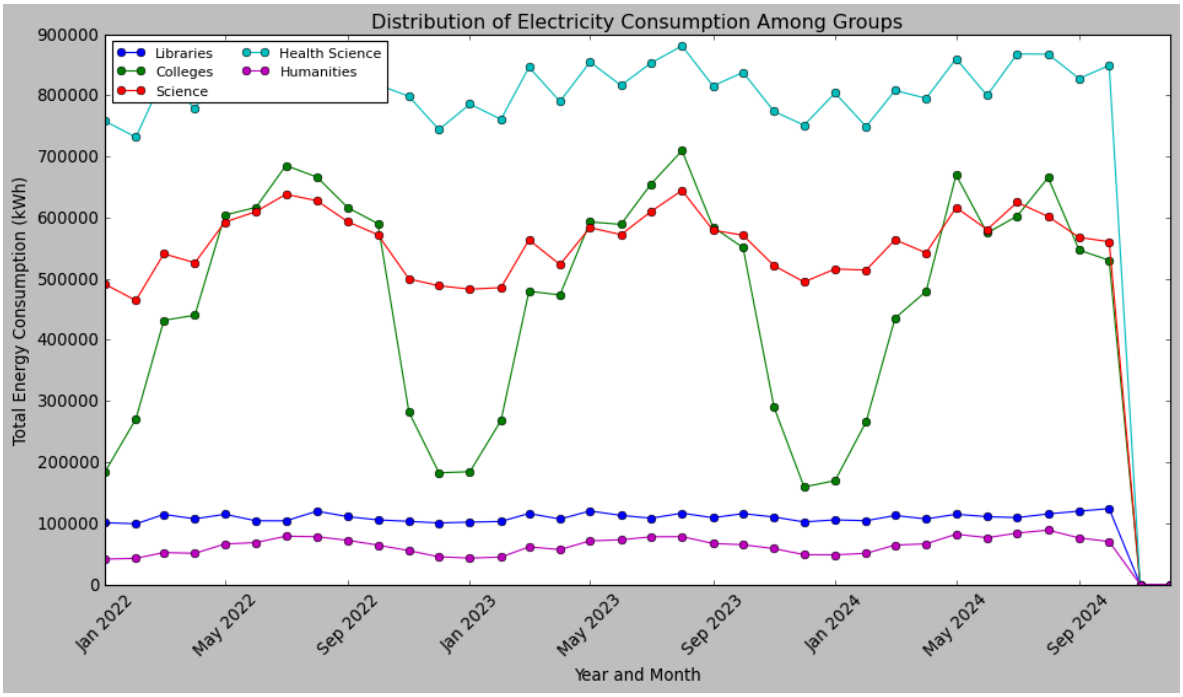
```
            plt.plot(stream_data_filtered['Date'], data_series,
                     marker='o', linestyle='-', color=dark_palette[idx % 10], la

            # Highlight anomalies
            plt.scatter(stream_data_filtered['Date'][anomalies.index], anomalies

        # Formatting the plot
        plt.xlabel("Year and Month")
        plt.ylabel("Energy Consumption (kWh)")
        plt.title(f"Energy Consumption Anomalies - {group}")
        plt.xticks(rotation=45)
        plt.legend(loc='upper left', fontsize='small', ncol=2)

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



In [40]:

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

# Define groups and their corresponding column names
groups = {
    "Libraries": [
        "E902 Hocken Library - kWh", "F813 UOCOE Robertson Library - kWh",
        "D203 Sayers (at Adams) - kWh", "F813 Bill Robertson Library - kWh",
        "F419 ISB West Excludir - kWh", "F505 Richardson Library - kWh"
    ],
    "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"
    ],
    "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:"
    ],
    "Health Science": [
        "A161 Taieri Farm - kWh", "D20X Med School Sub Main - kWh",
        "E214 Otago Dental School - kWh", "E301 Hunter Centre - kWh",
        "E305 Physiotherapy - kWh", "E325 Research Support Facility - kWh"
    ],
    "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
    ]
}

# Set a plain white background
plt.style.use('classic')

# Compute and plot distribution of electricity consumption per group
group_totals = {}

for group, columns in groups.items():
    available_columns = [col for col in columns if col in stream_data_filtered.c
    if available_columns:
        group_totals[group] = stream_data_filtered[available_columns].sum(axis=1

# Convert to DataFrame
distribution_df = pd.DataFrame(group_totals)

# Plot distribution
plt.figure(figsize=(14, 7))
for group in distribution_df.columns:
    plt.plot(stream_data_filtered['Date'], distribution_df[group], marker='o', l

# Formatting the plot
plt.xlabel("Year and Month")
plt.ylabel("Total Energy Consumption (kWh)")
plt.title("Distribution of Electricity Consumption Among Groups")
plt.xticks(rotation=45)
plt.legend(loc='upper left', fontsize='small', ncol=2)

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



In [41]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Read the CSV file
df = pd.read_csv('stream_data_filtered.csv')

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

# Select columns for buildings F, G, and H
f_columns = [col for col in df.columns if col.startswith('F')]
g_columns = [col for col in df.columns if col.startswith('G')]
h_columns = [col for col in df.columns if col.startswith('H')]

# Combine the selected columns
selected_columns = ['Date'] + f_columns + g_columns + h_columns
buildings_df = df[selected_columns]

# Convert numeric columns, replacing non-numeric values with NaN
for col in buildings_df.columns:
    if col != 'Date':
        buildings_df[col] = pd.to_numeric(buildings_df[col], errors='coerce')

# Calculate total energy usage for each building group
buildings_df['F_Total'] = buildings_df[f_columns].sum(axis=1)
buildings_df['G_Total'] = buildings_df[g_columns].sum(axis=1)
buildings_df['H_Total'] = buildings_df[h_columns].sum(axis=1)

# Set the style for the plots
plt.style.use('ggplot')

# Create a figure with subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 10))

# 1. Line Plot of Total Energy Usage
ax1.plot(buildings_df['Date'], buildings_df['F_Total'], label='F Buildings', mar
ax1.plot(buildings_df['Date'], buildings_df['G_Total'], label='G Buildings', mar
ax1.plot(buildings_df['Date'], buildings_df['H_Total'], label='H Buildings', mar
```

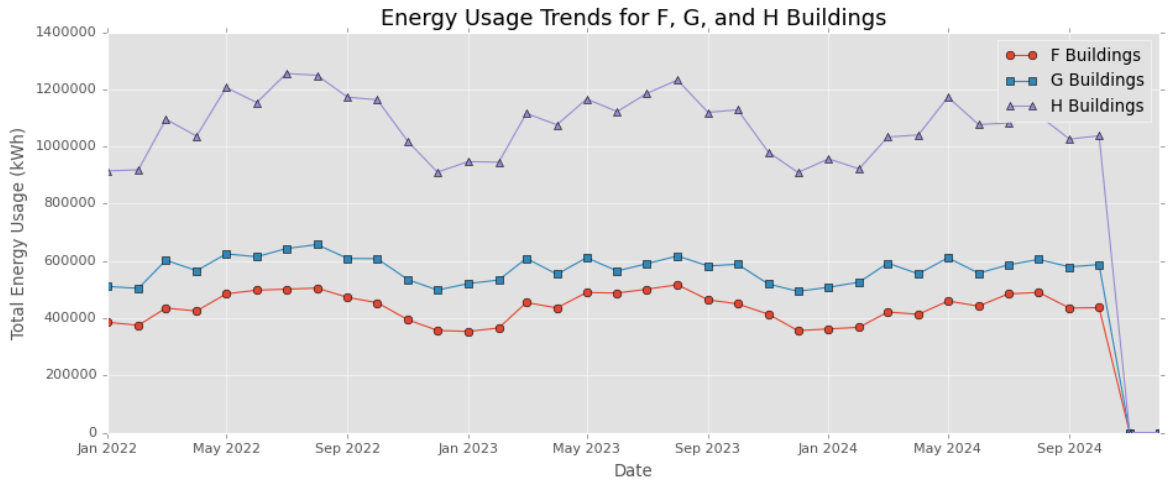
```
ax1.set_title('Energy Usage Trends for F, G, and H Buildings', fontsize=16)
ax1.set_xlabel('Date', fontsize=12)
ax1.set_ylabel('Total Energy Usage (kWh)', fontsize=12)
ax1.legend()
ax1.grid(True)

# 2. Stacked Bar Plot
building_groups = ['F_Total', 'G_Total', 'H_Total']
stacked_data = buildings_df[building_groups].fillna(0)  # Fill NaN with 0
ax2.bar(range(len(stacked_data)), stacked_data['F_Total'], label='F Buildings')
ax2.bar(range(len(stacked_data)), stacked_data['G_Total'], bottom=stacked_data['
ax2.bar(range(len(stacked_data)), stacked_data['H_Total'],
        bottom=stacked_data['F_Total'] + stacked_data['G_Total'], label='H Build
ax2.set_title('Cumulative Energy Usage by Building Group', fontsize=16)
ax2.set_xlabel('Time Periods', fontsize=12)
ax2.set_ylabel('Cumulative Energy Usage (kWh)', fontsize=12)
ax2.legend()

plt.tight_layout()
plt.savefig('building_energy_trends.png')

# Calculate insights with error handling
insights = {
    'F_Buildings': {
        'Total_Usage': buildings_df['F_Total'].sum(),
        'Average_Monthly_Usage': buildings_df['F_Total'].mean(),
        'Monthly_Variation': buildings_df['F_Total'].std()
    },
    'G_Buildings': {
        'Total_Usage': buildings_df['G_Total'].sum(),
        'Average_Monthly_Usage': buildings_df['G_Total'].mean(),
        'Monthly_Variation': buildings_df['G_Total'].std()
    },
    'H_Buildings': {
        'Total_Usage': buildings_df['H_Total'].sum(),
        'Average_Monthly_Usage': buildings_df['H_Total'].mean(),
        'Monthly_Variation': buildings_df['H_Total'].std()
    }
}

# Print insights
print("\nEnergy Usage Insights:")
for building, data in insights.items():
    print(f"\n{building}:")
    for metric, value in data.items():
        print(f"{metric}: {value:.2f}")

# Correlation analysis
correlation_matrix = buildings_df[['F_Total', 'G_Total', 'H_Total']].corr()
print("\nCorrelation Matrix:")
print(correlation_matrix)

# Calculate percentage contribution
total_energy = buildings_df['F_Total'] + buildings_df['G_Total'] + buildings_df[
buildings_df['F_Percentage'] = (buildings_df['F_Total'] / total_energy * 100).fi
buildings_df['G_Percentage'] = (buildings_df['G_Total'] / total_energy * 100).fi
buildings_df['H_Percentage'] = (buildings_df['H_Total'] / total_energy * 100).fi

print("\nAverage Percentage Contribution:")
print(f"F Buildings: {buildings_df['F_Percentage'].mean():.2f}%")
```

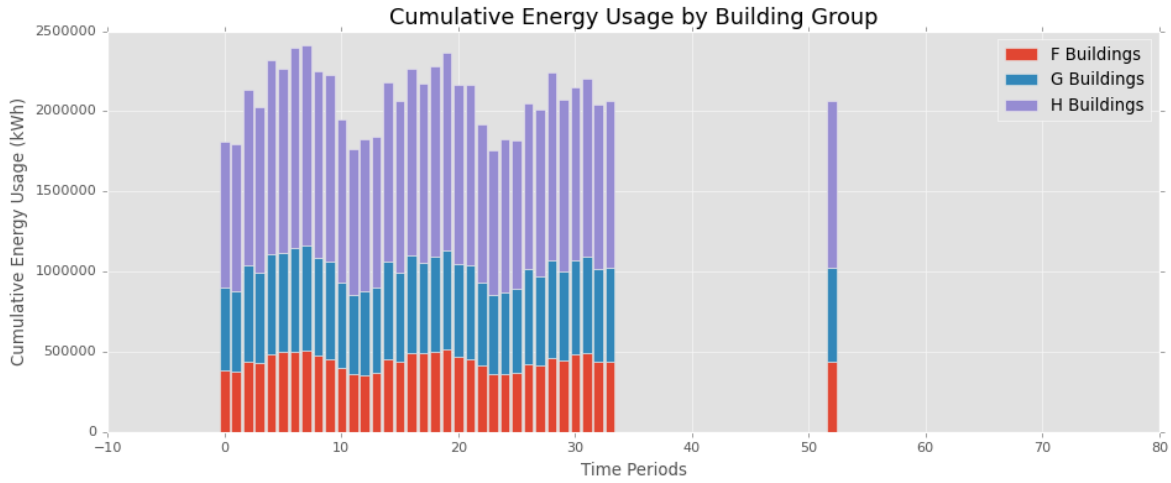
```
print(f"G Buildings: {buildings_df['G_Percentage'].mean():.2f}%")
print(f"H Buildings: {buildings_df['H_Percentage'].mean():.2f}%")
```

```
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:23: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df[col] = pd.to_numeric(buildings_df[col], errors='coerce')
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:26: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df['F_Total'] = buildings_df[f_columns].sum(axis=1)
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:27: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df['G_Total'] = buildings_df[g_columns].sum(axis=1)
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:28: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df['H_Total'] = buildings_df[h_columns].sum(axis=1)
```

```
Energy Usage Insights:

F_Buildings:
Total_Usage: 15325086.74
Average_Monthly_Usage: 204334.49
Monthly_Variation: 222481.27

G_Buildings:
Total_Usage: 20049307.25
Average_Monthly_Usage: 267324.10
Monthly_Variation: 289236.24

H_Buildings:
Total_Usage: 37511244.70
Average_Monthly_Usage: 500149.93
Monthly_Variation: 542848.88

Correlation Matrix:
          F_Total   G_Total   H_Total
F_Total  1.000000  0.997049  0.998473
G_Total  0.997049  1.000000  0.998650
H_Total  0.998473  0.998650  1.000000

Average Percentage Contribution:
F Buildings: 9.80%
G Buildings: 12.86%
H Buildings: 24.01%
```

```
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:94: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df['F_Percentage'] = (buildings_df['F_Total'] / total_energy * 100).f
illna(0)
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:95: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df['G_Percentage'] = (buildings_df['G_Total'] / total_energy * 100).f
illna(0)
C:\Users\sugan\AppData\Local\Temp\ipykernel_6004\436797195.py:96: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  buildings_df['H_Percentage'] = (buildings_df['H_Total'] / total_energy * 100).f
illna(0)
```


Energy Usage Trends for F, G, and H Buildings
Total Energy Usage (kWh)
1400000
1200000
1000000
800000
600000
400000
200000
0
Jan 2022
May 2022
Sep 2022
Jan 2023
May 2023
Sep 2023
Jan 2024
May 2024
Sep 2024
Date
F Buildings
G Buildings
H Buildings


Cumulative Energy Usage by Building Group
Cumulative Energy Usage (kWh)
2500000
2000000
1500000
1000000
500000
0
−10
0
10
20
30
40
50
60
70
80
Time Periods
F Buildings
G Buildings
H Buildings

In [ ]: