In [127…

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

Part 1: Data Cleaning and Preprocessing

1.1 Load and Inspect the Dataset

In [2]:

```
data=pd.read_csv("Building_Energy_Benchmarking.csv")
data.shape #Shape of the dataset
```

Out[2]: (494, 31)

In [89]:

```
data.head(3)
```

Out[89]:

	Property Id	Property Name	Address 1	City	Postal Code	Province	Primary Property Type - Self Selected	Number of Buildings	Year Built
0	10176804	Acadia Aquatic & Fitness Centre	9009 Fairmount Dr SE	Calgary	T2H 0Z4	Alberta	Heated Swimming Pool	1	2010
1	6169481	Ad Valorem	2924 11 ST NE	Calgary	t2e7l7	Alberta	Office	1	1981
2	6305956	Alberta Trade Centre	315 10 AV SE	Calgary	T2G 0W2	Alberta	Office	1	1974

3 rows × 31 columns

In [3]:

```
data.columns #column names of the dataset
```

Out[3]:
```
Index(['Property Id', 'Property Name', 'Address 1', 'City', 'Postal Code',
       'Province', 'Primary Property Type - Self Selected',
       'Number of Buildings', 'Year Built',
       'Property GFA - Self-Reported (m²)', 'ENERGY STAR Score',
       'Site Energy Use (GJ)', 'Weather Normalized Site Energy Use (GJ)',
       'Site EUI (GJ/m²)', 'Weather Normalized Site EUI (GJ/m²)',
       'Source Energy Use (GJ)', 'Weather Normalized Source Energy Use (GJ)',
       'Source EUI (GJ/m²)', 'Weather Normalized Source EUI (GJ/m²)',
       'Total GHG Emissions (Metric Tons CO2e)',
       'Total GHG Emissions Intensity (kgCO2e/m²)',
       'Direct GHG Emissions (Metric Tons CO2e)',
       'Direct GHG Emissions Intensity (kgCO2e/m²)',
       'Electricity Use - Grid Purchase (kWh)', 'Natural Gas Use (GJ)',
       'District Hot Water Use (GJ)',
       'Electricity Use – Generated from Onsite Renewable Systems (kWh)',
       'Green Power - Onsite and Offsite (kWh)',
       'Avoided Emissions - Onsite and Offsite Green Power (Metric Tons CO2e)',
       'Year Ending', 'Unique ID'],
      dtype='object')
```

In [4]:
```
data.dtypes #datatypes of each columns
```

```
Out[4]: Property Id                                                          int6
4
Property Name                                                        objec
t
Address 1                                                            objec
t
City                                                                 objec
t
Postal Code                                                          objec
t
Province                                                             objec
t
Primary Property Type - Self Selected                                objec
t
Number of Buildings                                                   int6
4
Year Built                                                            int6
4
Property GFA - Self-Reported (m²)                                    objec
t
ENERGY STAR Score                                                   float6
4
Site Energy Use (GJ)                                                 objec
t
Weather Normalized Site Energy Use (GJ)                              objec
t
Site EUI (GJ/m²)                                                    float6
4
Weather Normalized Site EUI (GJ/m²)                                 float6
4
Source Energy Use (GJ)                                               objec
t
Weather Normalized Source Energy Use (GJ)                            objec
t
Source EUI (GJ/m²)                                                  float6
4
Weather Normalized Source EUI (GJ/m²)                               float6
4
Total GHG Emissions (Metric Tons CO2e)                               objec
t
Total GHG Emissions Intensity (kgCO2e/m²)                           float6
4
Direct GHG Emissions (Metric Tons CO2e)                              objec
t
Direct GHG Emissions Intensity (kgCO2e/m²)                          float6
4
Electricity Use - Grid Purchase (kWh)                                objec
t
Natural Gas Use (GJ)                                                 objec
t
District Hot Water Use (GJ)                                          objec
t
Electricity Use – Generated from Onsite Renewable Systems (kWh)     float6
4
Green Power - Onsite and Offsite (kWh)                              float6
4
Avoided Emissions - Onsite and Offsite Green Power (Metric Tons CO2e)    float6
4
Year Ending                                                           int6
4
```

```
Unique ID                                                                    objec
t
dtype: object
```

In [5]:

```
#Percentage of missing values in each column
missing = (data.isnull().sum() / len(data)) * 100
print(missing)
```

Property Id 0.00000
0
Property Name 0.00000
0
Address 1 0.00000
0
City 0.00000
0
Postal Code 0.00000
0
Province 0.00000
0
Primary Property Type - Self Selected 0.00000
0
Number of Buildings 0.00000
0
Year Built 0.00000
0
Property GFA - Self-Reported (m²) 0.00000
0
ENERGY STAR Score 66.59919
0
Site Energy Use (GJ) 0.00000
0
Weather Normalized Site Energy Use (GJ) 0.00000
0
Site EUI (GJ/m²) 0.00000
0
Weather Normalized Site EUI (GJ/m²) 0.00000
0
Source Energy Use (GJ) 0.00000
0
Weather Normalized Source Energy Use (GJ) 0.40485
8
Source EUI (GJ/m²) 0.00000
0
Weather Normalized Source EUI (GJ/m²) 0.00000
0
Total GHG Emissions (Metric Tons CO2e) 0.00000
0
Total GHG Emissions Intensity (kgCO2e/m²) 0.00000
0
Direct GHG Emissions (Metric Tons CO2e) 0.00000
0
Direct GHG Emissions Intensity (kgCO2e/m²) 0.00000
0
Electricity Use - Grid Purchase (kWh) 0.00000
0
Natural Gas Use (GJ) 2.02429
1
District Hot Water Use (GJ) 96.96356
3
Electricity Use – Generated from Onsite Renewable Systems (kWh) 91.09311
7
Green Power - Onsite and Offsite (kWh) 40.08097
2
Avoided Emissions - Onsite and Offsite Green Power (Metric Tons CO2e) 40.08097
2
Year Ending 0.00000
0

```
Unique ID                                                                  0.00000
0
dtype: float64
```

In [6]:
```
data.isna().sum()
```

Out[6]:
```
Property Id                                                              0
Property Name                                                            0
Address 1                                                                0
City                                                                     0
Postal Code                                                              0
Province                                                                 0
Primary Property Type - Self Selected                                    0
Number of Buildings                                                      0
Year Built                                                               0
Property GFA - Self-Reported (m²)                                        0
ENERGY STAR Score                                                      329
Site Energy Use (GJ)                                                     0
Weather Normalized Site Energy Use (GJ)                                  0
Site EUI (GJ/m²)                                                         0
Weather Normalized Site EUI (GJ/m²)                                      0
Source Energy Use (GJ)                                                   0
Weather Normalized Source Energy Use (GJ)                                2
Source EUI (GJ/m²)                                                       0
Weather Normalized Source EUI (GJ/m²)                                    0
Total GHG Emissions (Metric Tons CO2e)                                   0
Total GHG Emissions Intensity (kgCO2e/m²)                                0
Direct GHG Emissions (Metric Tons CO2e)                                  0
Direct GHG Emissions Intensity (kgCO2e/m²)                               0
Electricity Use - Grid Purchase (kWh)                                    0
Natural Gas Use (GJ)                                                    10
District Hot Water Use (GJ)                                            479
Electricity Use – Generated from Onsite Renewable Systems (kWh)        450
Green Power - Onsite and Offsite (kWh)                                 198
Avoided Emissions - Onsite and Offsite Green Power (Metric Tons CO2e)  198
Year Ending                                                              0
Unique ID                                                                0
dtype: int64
```

In [7]:
```
missing[missing>40].index.tolist() #Columns with more than 40% missing values
```

Out[7]:
```
['ENERGY STAR Score',
 'District Hot Water Use (GJ)',
 'Electricity Use – Generated from Onsite Renewable Systems (kWh)',
 'Green Power - Onsite and Offsite (kWh)',
 'Avoided Emissions - Onsite and Offsite Green Power (Metric Tons CO2e)']
```

In [8]:
```
missing[(missing<40) & (missing > 0.000001)].index.tolist() #Columns with missin
```

Out[8]:
```
['Weather Normalized Source Energy Use (GJ)', 'Natural Gas Use (GJ)']
```

1.2 Handling Missing Data

In [95]:
```
df=data.copy() #made a dataset copy to work further.
```

In [96]:
```
#Dropped columns with more than 40% missing values
df = df.drop(columns=['ENERGY STAR Score','District Hot Water Use (GJ)',
 'Electricity Use – Generated from Onsite Renewable Systems (kWh)',
```

```
 'Green Power - Onsite and Offsite (kWh)',
 'Avoided Emissions - Onsite and Offsite Green Power (Metric Tons CO2e)'])
```

In [97]:
```
#Conerting to the correct datatypes of each columns so that can fill missing val
df['Weather Normalized Source Energy Use (GJ)'] = pd.to_numeric(df['Weather Norm
df['Natural Gas Use (GJ)'] = pd.to_numeric(df['Natural Gas Use (GJ)'].replace(r'
```

In [98]:
```
#Filled missing values with median as the column is numeric type
df['Weather Normalized Source Energy Use (GJ)']=df['Weather Normalized Source En
df['Natural Gas Use (GJ)']=df['Natural Gas Use (GJ)'].fillna(df['Natural Gas Use
df.isna().sum()
```

Out[98]:
```
Property Id                                        0
Property Name                                      0
Address 1                                          0
City                                               0
Postal Code                                        0
Province                                           0
Primary Property Type - Self Selected              0
Number of Buildings                                0
Year Built                                         0
Property GFA - Self-Reported (m²)                  0
Site Energy Use (GJ)                               0
Weather Normalized Site Energy Use (GJ)            0
Site EUI (GJ/m²)                                   0
Weather Normalized Site EUI (GJ/m²)                0
Source Energy Use (GJ)                             0
Weather Normalized Source Energy Use (GJ)          0
Source EUI (GJ/m²)                                 0
Weather Normalized Source EUI (GJ/m²)              0
Total GHG Emissions (Metric Tons CO2e)             0
Total GHG Emissions Intensity (kgCO2e/m²)          0
Direct GHG Emissions (Metric Tons CO2e)            0
Direct GHG Emissions Intensity (kgCO2e/m²)         0
Electricity Use - Grid Purchase (kWh)              0
Natural Gas Use (GJ)                               0
Year Ending                                        0
Unique ID                                          0
dtype: int64
```

1.3 Extracting and Cleaning Data Using Regex

In [99]:
```
#Extract numeric values from text-based numeric columns (e.g., Property GFA,Ener
def extract_number(d):
    if type(d)==str:  #checks the column is object becasue here numeric columns
        n = re.findall(r"\d+\.?\d*", d)#using expression finding the digits, num
        return float(n[0]) if n else None
    return None
numCol = ["Property GFA - Self-Reported (m²)", "Site Energy Use (GJ)","Weather N
          "Total GHG Emissions (Metric Tons CO2e)","Direct GHG Emissions (Metric
for col in numCol:
    df[col] = df[col].apply(extract_number)
df[numCol] = df[numCol].astype(float)
df.dtypes
```

Out[99]:
```
Property Id                                          int64
Property Name                                       object
Address 1                                           object
City                                                object
Postal Code                                         object
Province                                            object
Primary Property Type - Self Selected               object
Number of Buildings                                  int64
Year Built                                           int64
Property GFA - Self-Reported (m²)                  float64
Site Energy Use (GJ)                               float64
Weather Normalized Site Energy Use (GJ)            float64
Site EUI (GJ/m²)                                   float64
Weather Normalized Site EUI (GJ/m²)                float64
Source Energy Use (GJ)                             float64
Weather Normalized Source Energy Use (GJ)          float64
Source EUI (GJ/m²)                                 float64
Weather Normalized Source EUI (GJ/m²)              float64
Total GHG Emissions (Metric Tons CO2e)             float64
Total GHG Emissions Intensity (kgCO2e/m²)          float64
Direct GHG Emissions (Metric Tons CO2e)            float64
Direct GHG Emissions Intensity (kgCO2e/m²)         float64
Electricity Use - Grid Purchase (kWh)              float64
Natural Gas Use (GJ)                               float64
Year Ending                                          int64
Unique ID                                           object
dtype: object
```

In [100…
```
df["Postal Code"].unique() #to see the pattern of cuurent code data
```

Out[100…
```
array(['T2H 0Z4', 't2e7l7', 'T2G 0W2', 'T2G0G2', 'T2G 4M7', 'T2N 2H8',
       'T2R 0G9', 'T2AoK9', 'T3B 0B9', 'T2G0K7', 'T2C 4E1', 'T2W 6G3',
       'T2E 8L9', 'T2B 3E2', 'T2G1W5', 'T2B0M5', 'T2E6R2', 'T2G2N9',
       'T2E 5R1', 'T2K 6K8', 'T2N 3G8', 'T2A0K9', 'T2G 3H2', 'T2G 5E3',
       'T2M 4V8', 'T3E1P1', 'T2A 3E2', 'T2V 5H5', 'T3B 1S4', 'T2K 5J6',
       'T2J 6X3', 'T2R1M4', 'T3E 7B2', 'T3B 5A4', 'T1Y 4Z4', 'T2W 4H7',
       'T2C 2X1', 'T3A 4M8', 'T3H 3R7', 'T2Z 0N2', 'T3H 3E4', 'T3R0N2',
       'T3B 5Y6', 'T3A 5G1', 'T2Y 5G9', 'T2Z 0H3', 'T2E8E1', 'T3P 0A3',
       'T3M 0M2', 'T3L 2Y8', 'T2T3V8', 'T2M 3A3', 'T3C 2C3', 'T2C 3B7',
       'T2N 3Y9', 'T2A0S4', 'T2J 4B5', 'T2L0A2', 'T3E 1P2', 'T2E8A1',
       'T2S 2G4', 'T2G 1J9', 'T2V2W2', 'T3E2J7', 'T3E 3H3', 'T2G 4K8',
       'T2G 4H3', 'T2G4K8', 'T2E 6S2', 'T2S 0A1', 'T2P 2M5', 'T2K 0A2',
       'T3E 0R4', 'T2A 4M6', 'T3S 0A4', 'T3E7H5', 'T2C0B4', 'T3H3P8',
       'T2E 7L7', 'T2L 0A2', 'T2G 0G2', 'T3H 3M4', 'T2G 0K7', 'T2G 1W5',
       'T2K 4Y5', 'T2A 0K9', 'T2E 8A1', 'T2V 2W2', 'T3B 5K9', 'T3E 1P1',
       'T3E 2J7', 'T3B 1C5', 'T2A 0S4', 'T2B 0M5', 'T3H 3P8', 'T3C 1Y3',
       'T3E 7H5', 'T2C 0B4', 'T3C 0E8', 'T2E 4J7', 'T1Y 6C2', 'T2G 2N9',
       'T2R 1M4', 'T2E 8E1', 'T2T 3V8', 'T2N 1N8', 'T3K 0S5', 'T3R 0N2',
       'T3C 3A3', 'T2L 0G6', 'T2V 4S4', 'T1Y 5J1', 'T2G 1T7', 'T2K4Y5',
       'T2A OK9', 'T2E 6R2', 'T3C0E8', 'T3B1C5', 'T3B5K9', 'T2E4J7',
       'T3C1Y3', 't2g4k8', 'T1Y6C2'], dtype=object)
```

In [101…
```
#Standardize Postal Codes to follow the Canadian format (A1A 1A1)
#Convert the string to upper case then removed not alphabet and number objects
df['Postal Code']=df['Postal Code'].str.upper().replace(r"[^A-Z0-9]","",regex=Tr
df['Postal Code']=df['Postal Code'].str.replace(r"(\w{3})(\w{3})",r"\1 \2",regex
```

In [102…
```
df["Postal Code"].unique()#checking the changes have been occured
```

Out[102…
```
array(['T2H 0Z4', 'T2E 7L7', 'T2G 0W2', 'T2G 0G2', 'T2G 4M7', 'T2N 2H8',
       'T2R 0G9', 'T2A OK9', 'T3B 0B9', 'T2G 0K7', 'T2C 4E1', 'T2W 6G3',
       'T2E 8L9', 'T2B 3E2', 'T2G 1W5', 'T2B 0M5', 'T2E 6R2', 'T2G 2N9',
       'T2E 5R1', 'T2K 6K8', 'T2N 3G8', 'T2A 0K9', 'T2G 3H2', 'T2G 5E3',
       'T2M 4V8', 'T3E 1P1', 'T2A 3E2', 'T2V 5H5', 'T3B 1S4', 'T2K 5J6',
       'T2J 6X3', 'T2R 1M4', 'T3E 7B2', 'T3B 5A4', 'T1Y 4Z4', 'T2W 4H7',
       'T2C 2X1', 'T3A 4M8', 'T3H 3R7', 'T2Z 0N2', 'T3H 3E4', 'T3R 0N2',
       'T3B 5Y6', 'T3A 5G1', 'T2Y 5G9', 'T2Z 0H3', 'T2E 8E1', 'T3P 0A3',
       'T3M 0M2', 'T3L 2Y8', 'T2T 3V8', 'T2M 3A3', 'T3C 2C3', 'T2C 3B7',
       'T2N 3Y9', 'T2A 0S4', 'T2J 4B5', 'T2L 0A2', 'T3E 1P2', 'T2E 8A1',
       'T2S 2G4', 'T2G 1J9', 'T2V 2W2', 'T3E 2J7', 'T3E 3H3', 'T2G 4K8',
       'T2G 4H3', 'T2E 6S2', 'T2S 0A1', 'T2P 2M5', 'T2K 0A2', 'T3E 0R4',
       'T2A 4M6', 'T3S 0A4', 'T3E 7H5', 'T2C 0B4', 'T3H 3P8', 'T3H 3M4',
       'T2K 4Y5', 'T3B 5K9', 'T3B 1C5', 'T3C 1Y3', 'T3C 0E8', 'T2E 4J7',
       'T1Y 6C2', 'T2N 1N8', 'T3K 0S5', 'T3C 3A3', 'T2L 0G6', 'T2V 4S4',
       'T1Y 5J1', 'T2G 1T7'], dtype=object)
```

In [103…
```
#extracting meaningful text from column property name and addresses.
#Removing leading and trailing spaces from string and converting first letter of
df["Property Name"]=df["Property Name"].str.strip().str.title()
df["Address 1"]=df["Address 1"].str.strip().str.title()
```

Part 2: Exploratory Data Analysis (EDA) and Aggregations

2.1 Statistical Summary

• Generate summary statistics for numerical features using extracted data. • Identify and explain key observations (e.g., outliers, mean vs. median differences).

In [104…
```
df.describe()
```

Out[104…

	Property Id	Number of Buildings	Year Built	Property GFA - Self-Reported (m^2)	Site Energy Use (GJ)	Weatl Normaliz Site Ener Use ((
count	4.940000e+02	494.000000	494.000000	494.000000	494.000000	494.000(
mean	1.308877e+07	1.060729	1980.091093	1974.198583	3586.745951	3699.120
std	5.659556e+06	0.278281	25.159568	6799.500086	15596.320164	15776.584(
min	6.169481e+06	1.000000	1896.000000	1.000000	1.000000	1.000(
25%	9.563763e+06	1.000000	1970.000000	2.000000	3.000000	3.000(
50%	9.997794e+06	1.000000	1978.000000	216.950000	79.500000	81.000(
75%	2.198860e+07	1.000000	1996.000000	1448.750000	1558.500000	1677.500(
max	2.198863e+07	3.000000	2018.000000	85941.000000	243202.000000	242611.000(

Interpretation: Property Id: Sequential data as it has wide range.
Number of Buildings: mean 1.06, min 1 and max 5. Year Built: Average construction year 1980, oldest is 1896, new built in 2018, mean is higher than median there's likely data skewed.
Property GFA - Self-Reported (m^2): mean 1974,median 216, exterme outliers peresent.
Site Energy Use (GJ): mean 3586, median 79, highly skewed data.
Weather Normalized Site Energy Use (GJ): mean 3699, median 81,highly skewed data,some large properties consugme not proper energy.
Site EUI (GJ/m^2): mean 1, median 1, not skewed.
Weather Normalized Site EUI (GJ/m^2): mean 1, median 1, not skewed.
Source Energy Use (GJ): mean 4556, median 36, highly skewed data.
Weather Normalized Source Energy Use (GJ): mean 10221, median 3144, highly skewed data.
Source EUI (GJ/m^2): mean 2, median 1, normal distribution but also some outliers.
Weather Normalized Source EUI (GJ/m^2): mean 2 ,median 1,normal distribution but also some outliers.
Total GHG Emissions (Metric Tons CO2e): mean 442, median 173, max 13067 which have huge outliers and few properties with very high emissions.
Total GHG Emissions Intensity (kgCO2e/m^2): mean 153, median 117 very few properties have extremely high emissions.
Direct GHG Emissions (Metric Tons CO2e): mean 213, median 74,max 12243,min 0, Highly Skewed Distribution.
Direct GHG Emissions Intensity (kgCO2e/m^2): mean 63, median 43,max 386, min 0, moderate right skew.
Electricity Use - Grid Purchase (kWh): mean 2.539227e+05, median 4.620000e+02, max 9.618602e+06, min 1.000000e+00, Extremely Skewed Distribution.
Natural Gas Use (GJ): mean 5520, median 1569, max 238415,min 3, data is right-skewed.
Year Ending: mean 2020, median 2021 , max 2023,min 2019

2.2 Aggregations

In [105…

```
df.columns
```

Out[105…

```
Index(['Property Id', 'Property Name', 'Address 1', 'City', 'Postal Code',
       'Province', 'Primary Property Type - Self Selected',
       'Number of Buildings', 'Year Built',
       'Property GFA - Self-Reported (m²)', 'Site Energy Use (GJ)',
       'Weather Normalized Site Energy Use (GJ)', 'Site EUI (GJ/m²)',
       'Weather Normalized Site EUI (GJ/m²)', 'Source Energy Use (GJ)',
       'Weather Normalized Source Energy Use (GJ)', 'Source EUI (GJ/m²)',
       'Weather Normalized Source EUI (GJ/m²)',
       'Total GHG Emissions (Metric Tons CO2e)',
       'Total GHG Emissions Intensity (kgCO2e/m²)',
       'Direct GHG Emissions (Metric Tons CO2e)',
       'Direct GHG Emissions Intensity (kgCO2e/m²)',
       'Electricity Use - Grid Purchase (kWh)', 'Natural Gas Use (GJ)',
       'Year Ending', 'Unique ID'],
      dtype='object')
```

• Compute the average Energy Use Intensity (EUI) by Property Type. • Compute the total Greenhouse Gas (GHG) emissions by year. • Identify the top 5 properties with the highest total energy consumption.

In [106…

```
#Found the average energy use intensity by using property type and Site EUI colu
ptype = df.groupby("Primary Property Type - Self Selected")["Site EUI (GJ/m²)"].
print(ptype.sort_values(ascending=False))
```

```
Primary Property Type - Self Selected
Heated Swimming Pool                                  4.805333
Fitness Center/Health Club/Gym                        4.385000
Distribution Center                                   3.286000
Ice/Curling Rink                                      2.182200
Other - Recreation                                    2.165000
Museum                                                1.584000
Social/Meeting Hall                                   1.550000
Other - Public Services                               1.526000
Office                                                1.519636
Performing Arts                                       1.302000
Repair Services (Vehicle, Shoe, Locksmith, etc.)      1.248000
Fire Station                                          1.208827
Self-Storage Facility                                 1.208000
Indoor Arena                                          1.106000
Non-Refrigerated Warehouse                            0.768000
Mixed Use Property                                    0.458000
Other                                                 0.070000
Name: Site EUI (GJ/m²), dtype: float64
```

In [107…

```
#found by the ending of year total emission of greenhouse gas.
gh = df.groupby("Year Ending")["Total GHG Emissions (Metric Tons CO2e)"].sum()
print(gh)
```

```
Year Ending
2019    22799.1
2020    24036.9
2021    24310.5
2022    72301.0
2023    75132.0
Name: Total GHG Emissions (Metric Tons CO2e), dtype: float64
```

In [108…

```
#top 5 property which have highest total energy consumption.
top5 = df.sort_values(by="Site Energy Use (GJ)", ascending=False).head(5)[["Prop
print(top5)
```

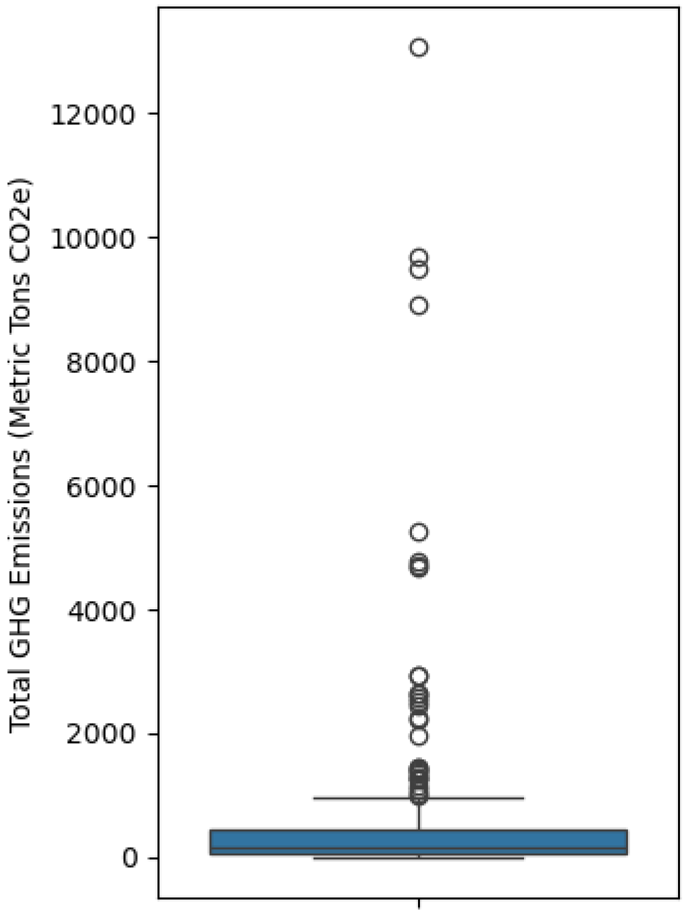
```
                    Property Name  Site Energy Use (GJ)
293        Stoney Transit Facility              243202.0
457        Stoney Transit Facility              160486.0
296  Village Square Leisure Centre               80302.0
307              Municipal Complex               79602.0
73               Municipal Complex               79343.0
```

2.3 Detecting Outliers Using Regex and IQR

o Identify values that do not conform to expected numeric formats. o Remove or correct incorrectly formatted numeric values. • Apply the Interquartile Range (IQR) method to detect outliers in Total GHG Emissions (Metric Tons CO2e). • Replace outliers with the median value for that property type.

In [109…

```
plt.subplot(1, 2, 2)
sns.boxplot(y=df["Total GHG Emissions (Metric Tons CO2e)"])
plt.tight_layout()
plt.show()
```



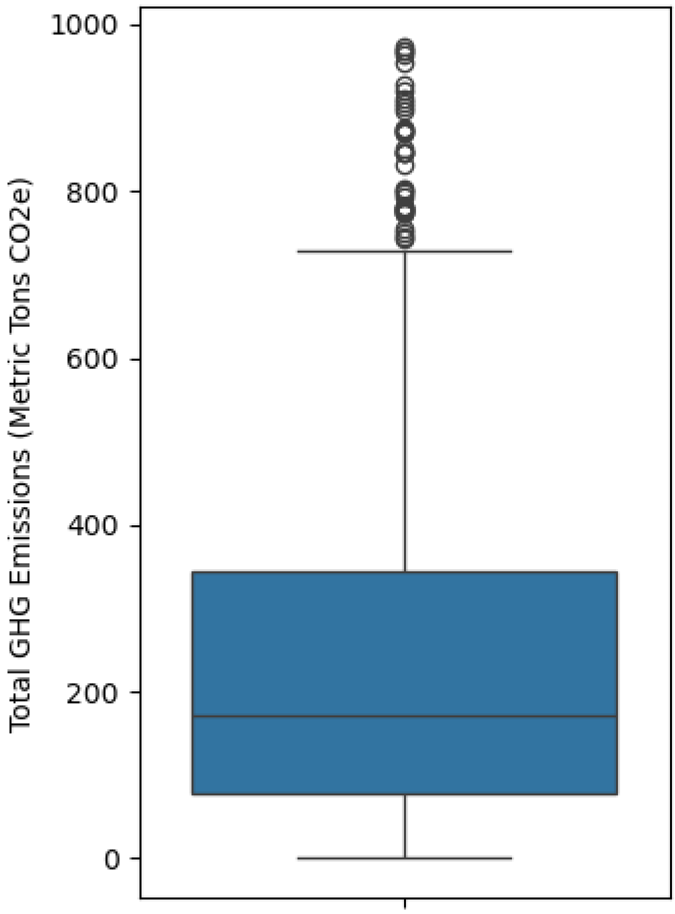
In [110…

```
Q1 = df["Total GHG Emissions (Metric Tons CO2e)"].quantile(0.25)
Q3 = df["Total GHG Emissions (Metric Tons CO2e)"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
median = df["Total GHG Emissions (Metric Tons CO2e)"].median()
df.loc[(df["Total GHG Emissions (Metric Tons CO2e)"] < lower_bound) | (df["Total
```

In [111…

```
plt.subplot(1, 2, 2)
sns.boxplot(y=df["Total GHG Emissions (Metric Tons CO2e)"])
```

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



Interpretation: Before outlier handled the distribution was highly skewed and after handling the extreme values have been removed but still some there even though its controlled.

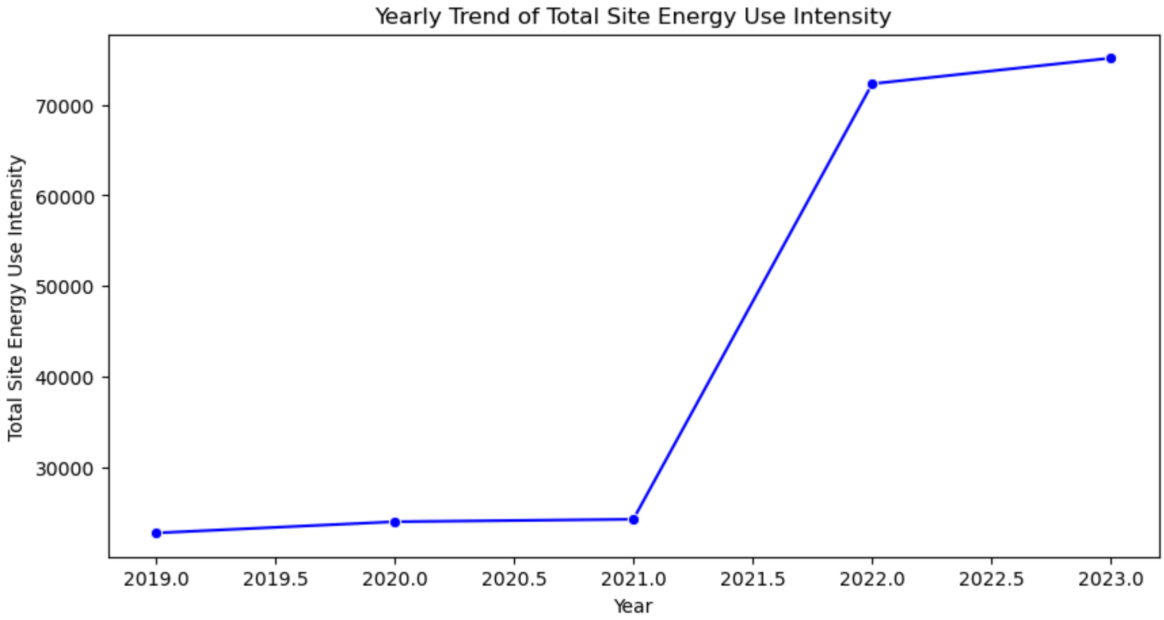
Part 3: Data Visualization

3.1 Time-Series Visualization

Plot the yearly trend of average Site Energy Use Intensity (EUI). • Highlight any significant increases or decreases in energy usage.

In [113...

```
eui = df.groupby("Year Ending")["Site EUI (GJ/m²)"].mean()
plt.figure(figsize=(10, 5))
sns.lineplot(x=gh.index, y=gh.values, marker="o", color="b")
plt.title("Yearly Trend of Total Site Energy Use Intensity")
plt.xlabel("Year")
plt.ylabel("Total Site Energy Use Intensity")
plt.show()
```



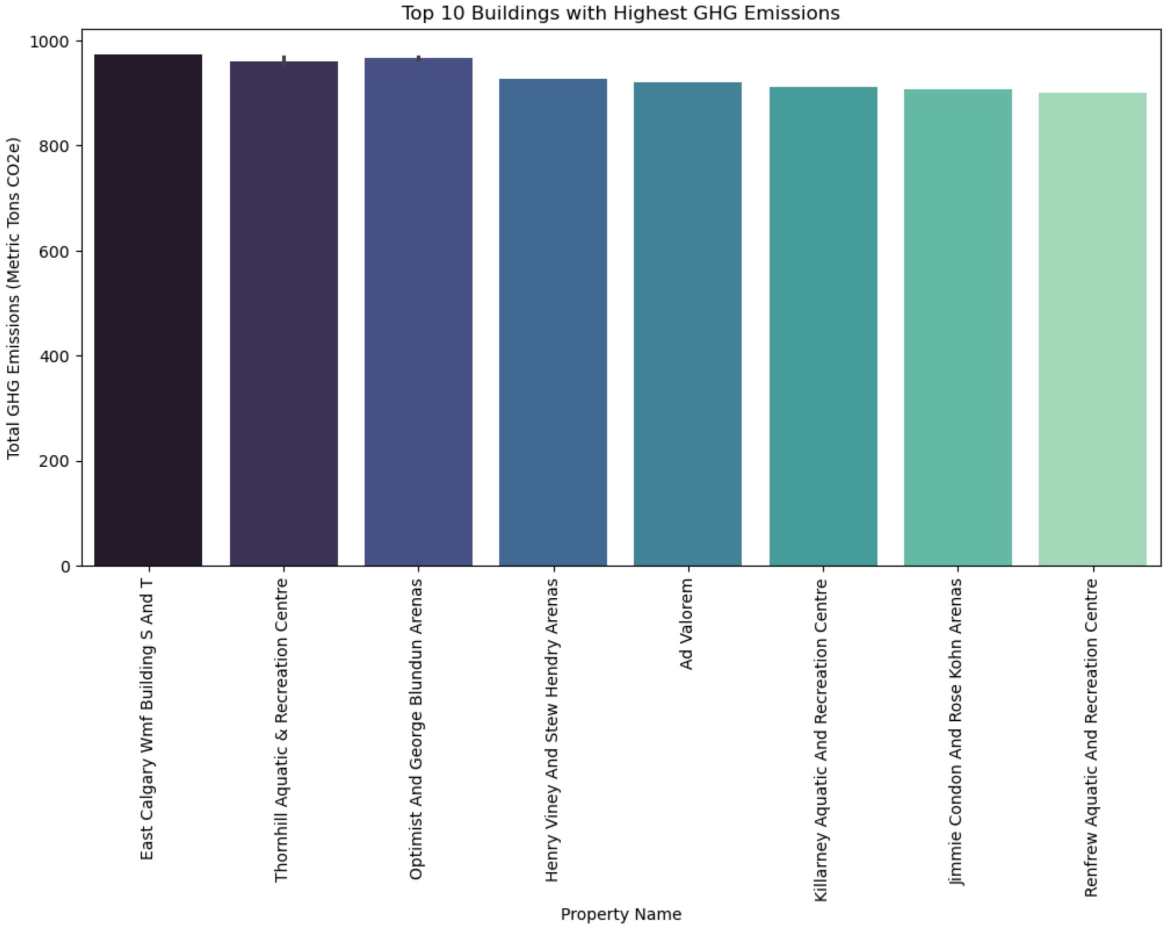
Interpretation: From 2019 to 2021 have a small increase but from 2021 to 2022 sudden spike just from 30000 it went upto 70000 which indicates the usage inrease and at last it remained the same value from 2022 to 2023

3.2 Comparative Bar Charts

• Create a bar chart showing the top 10 buildings with the highest GHG emissions. • Annotate the bar chart with emission values.

In [133…

```
top10_ghg = df.sort_values(by="Total GHG Emissions (Metric Tons CO2e)", ascendin
plt.figure(figsize=(12, 6))
sns.barplot(x=top10_ghg["Property Name"], y=top10_ghg["Total GHG Emissions (Metr
plt.xticks(rotation=90)
plt.title("Top 10 Buildings with Highest GHG Emissions")
plt.xlabel("Property Name")
plt.ylabel("Total GHG Emissions (Metric Tons CO2e)")
plt.show()
```



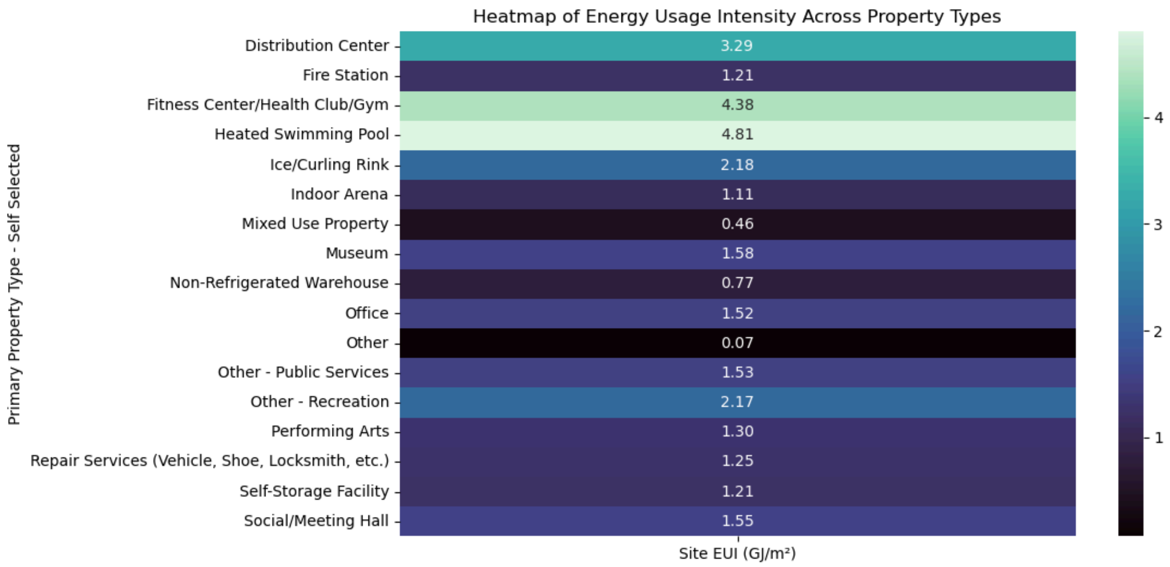
Interpretation: the property which have highest Greenhouse gas emissions are East Calgary Wwft Building S And T,Thornhill Aquatic & Recreation Centre, Optimist And George Blundun Arenas, Henry Viney And Stew Hendry Arenas, Ad Valorem, Killarney Aquatic And Recreation Centre, Jimmie Condon And Rose Kohn Arenas, Renfrew Aquatic And Recreation Centre, the emissions are close to 1000 metric tons of CO2 equivalent

3.3 Heatmap Visualization

Create a heatmap of energy usage intensity (Site EUI (GJ/m²)) across different property types.

In [142…

```
pivot_table = df.pivot_table(values='Site EUI (GJ/m²)', index='Primary Property
plt.figure(figsize=(10, 6))
sns.heatmap(pivot_table, cmap="mako", annot=True, fmt=".2f")
plt.title("Heatmap of Energy Usage Intensity Across Property Types")
plt.show()
```



Interpretation: Heated Swimming Pool: Highest energy usage intensity at 4.81, Fitness Center/Health Club/Gym: Second highest at 4.38.

Part 4: Further Analysis

4.1 Correlation Analysis

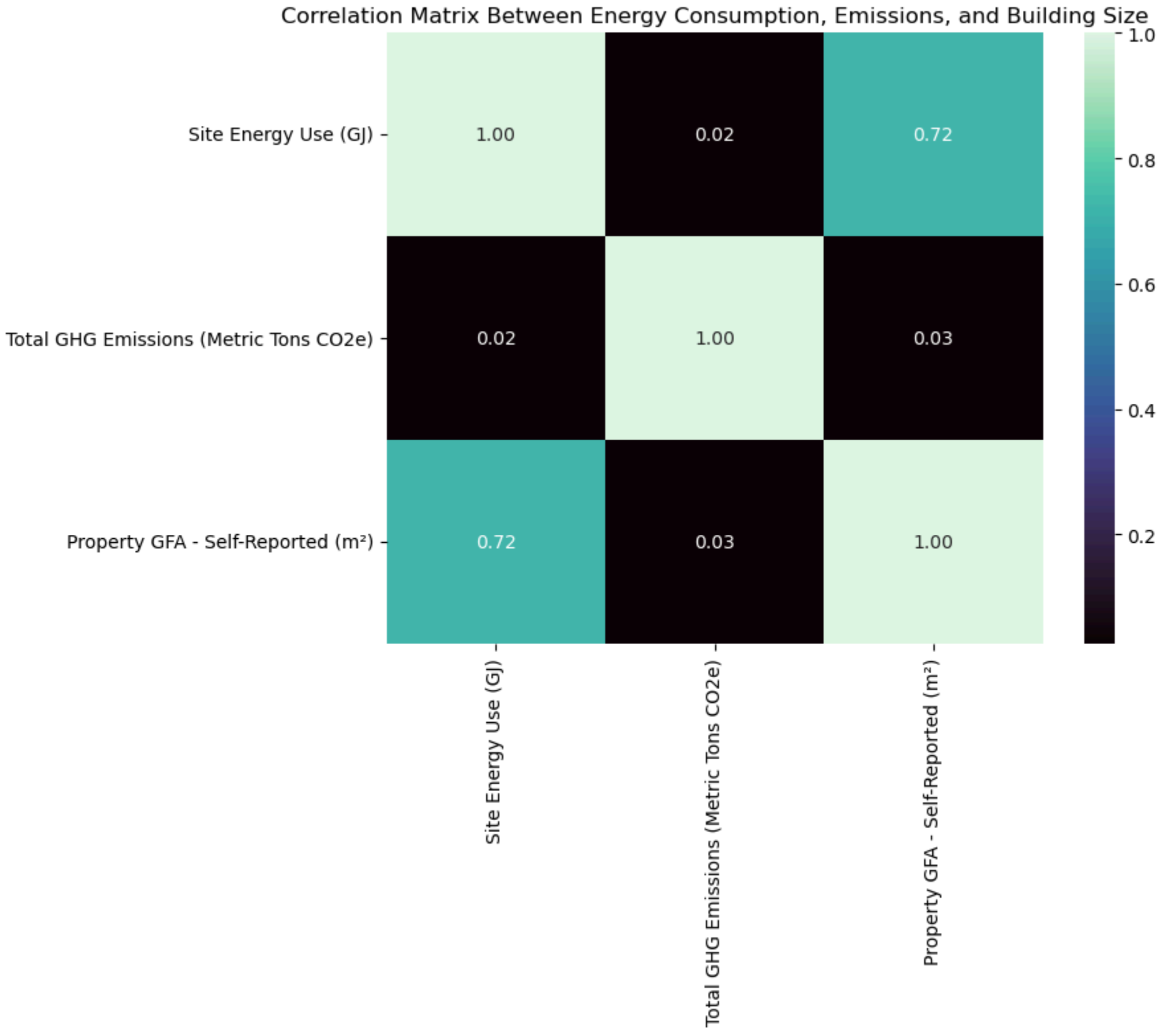
Compute and visualize the correlation matrix between energy consumption, emissions, and building size. • Identify any strong correlations and explain their implications

In [141…

```
corrdf = ['Site Energy Use (GJ)', 'Total GHG Emissions (Metric Tons CO2e)', 'Pro
corrdf=df[corrdf]
```

In [146…

```
corr_matrix = corrdf.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, cmap="mako", annot=True, fmt=".2f")
plt.title("Correlation Matrix Between Energy Consumption, Emissions, and Buildin
plt.show()
```



Interpretation: Strong correlation is there btw Site Energy Use (GJ) and Property GFA - Self-Reported (m^2) which is 0.72 which means when the size of property increases the usage of energy also increases. Weak correlations btw Site Energy Use (GJ) and Total GHG Emissions (Metric Tons CO2e) which is 0.02 that means there is no relationship btw the amount of energy used and total greenhouse gas emission and btw Total GHG Emissions (Metric Tons CO2e) and Property GFA - Self-Reported (m^2) which is 0.03 no relationship btw both of them

4.2 Hypothesis Testing

• Conduct a t-test • Interpret the results and discuss statistical significance.

In [159…

```
office = data[data["Primary Property Type - Self Selected"] == "Office"]["ENERGY
ice= data[data["Primary Property Type - Self Selected"] == "Ice/Curling Rink"]["
print(office.shape,ice.shape)
```

```
(94,) (45,)
```

In [161…

```
from scipy.stats import ttest_ind
stat, p_value = ttest_ind(office, ice, nan_policy='omit')
print("T-Test Results:", stat, p_value)
if p_value < 0.05:
    print("Statistically significant difference in ENERGY STAR Scores between Of
```

```
else:
    print("No significant difference found.")
```

```
T-Test Results: 3.2423927715555587 0.0014886772098221066
Statistically significant difference in ENERGY STAR Scores between Office and Ic
e/Curling Rink
```

Interpretation: p-value is smaller than the significance level of 0.05 we can reject the null hypothesis, the mean ENERGY STAR Scores for Office properties are significantly different from those of Ice/Curling Rink properties.

Part 5: Reporting and Insights

5.1 Summary Report

Energy consumption and efficiency Report

An overview of the dataset indicates a substantial variation in energy use among different property types. Specifically, gyms and heated swimming pools have the highest Site Energy Use Intensity (EUI), at 4.38 and 4.81 GJ/m², respectively. Energy consumption has broadly risen, as evidenced by an acute rise in overall greenhouse gas (GHG) emissions that soared from 24,310 metric tons in 2021 to 75,132 metric tons in 2023. The findings further demonstrate a robust correlation (0.72) between property size and energy consumption, suggesting that larger properties naturally use more energy.

Seasonal and Property Type Differences
The energy consumption is different largely with the type of property. Buildings such as ice, indoor swimming pools, and distribution warehouses exhibit higher per square meter energy use than office buildings and mixed-use complexes. Energy consumption is also seasonal, especially in buildings that need to control their climates, e.g., recreation centers.
A temporal analysis of the energy utilization indicates very consistent use during 2019-2021 and an abrupt spike in 2022. It could be the result of additional working hours or a shift in the policies for energy efficiency.

Regex Use in Data Cleaning

Regular expressions (Regex) were utilized in data cleaning and extraction, mainly in:
--> Standardizing numeric values: Extracting numerical data from textual numeric columns (e.g., "Property GFA - Self-Reported (m²)").
--> Postal Code Formatting: Enforcing consistency by standardizing the Canadian format (A1A 1A1).

--> Removing Unwanted Characters: Cleaning numerical data by removing commas and other non-numerical characters.

Supporting Visualizations

Yearly Energy Use Intensity Trend: Exhibits a peak in 2022, emphasizing monitoring energy spikes.
Top 10 Buildings by GHG Emissions : Selects properties which need urgent intervention.
Heatmap of Energy Use by Property Types : Identifies high-energy properties, facilitating precise energy-saving strategies.
Correlation Matrix Between Energy, Emissions, and Property Size : Shows the correlation between energy consumption and property size.

Github Link:

https://github.com/Megha-R-S/Data-Analysis-of-Building-Energy-Benchmarking-Data