# SUMMARY REPORT

## Key Trends in Energy Consumption and Efficiency

The Dataset reveals a strong correlation between building size and energy use, indicating that larger buildings tend to consume significantly more energy. The Stoney Transit Facility has the highest GHG Emission of more than 10,000 Tons, which is significantly the highest over the rest of the buildings in a great percentage. The Site Energy Use Intensity (EUI) Time series visualization shows noticeable peaks and significant drop. Correlation between energy consumption and GHG emissions confirms that high energy use directly contributes to environmental impact.

## Seasonal and Property Type Variations

An analysis of property types suggests that commercial buildings (offices, warehouses and retail spaces) generally have higher energy consumption than residential properties. This could be because of longer operating hours, heating and cooling requirements. Seasonal Trends like increased energy consumption during winter months due to higher heating demands.

## Recommendations For Improving Energy Efficiency And Reducing Emissions

**1.** Energy Efficiency Programs for Large Buildings: Since larger buildings account for the majority of energy consumption, incentivizing energy-efficient retrofits (e.g., insulation, LED lighting, efficient HVAC systems) can yield significant reductions in energy use and emissions.

**2.** Targeted Interventions for High GHG Emitters: Buildings with extremely high GHG emissions should be prioritized for carbon reduction strategies such as renewable energy integration and smart energy management systems.

**3.** Peak Load Management Strategies: Seasonal analysis highlights peak winter energy demand, which could be mitigated through demand response programs, improved insulation, and energy storage solutions.

**4.** Implementing Smart Building Technology and Real Time Energy Monitoring can help building owners optimize energy use and identify inefficiencies.

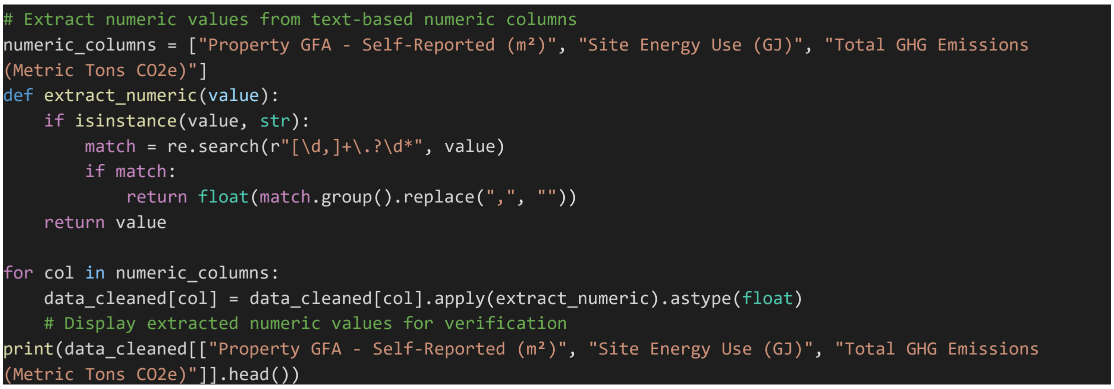
**5.** Investing in Renewable Energy Sources such as Installation of Solar panels, wind turbines and green roofs can support long-term sustainability goals.

**6.** Public Awareness and Education Programs: Providing training and awareness campaigns on energy conservation practices can empower building operators and occupants to adopt more sustainable practices and behaviour.

## HOW REGEX WAS USED IN THIS ASSIGNMENT

### 1. Extracting Numeric Values from Text

Some records contained **text-based numeric values**, requiring extraction before converting them into numbers. To extract only valid numeric portions from such entries, this **Regex extraction function** was applied:



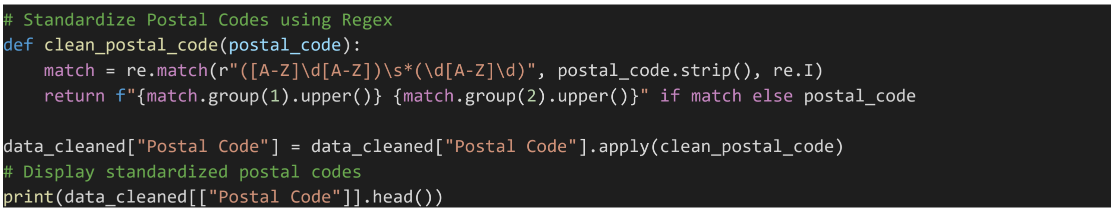
```
# Extract numeric values from text-based numeric columns
numeric_columns = ["Property GFA - Self-Reported (m²)", "Site Energy Use (GJ)", "Total GHG Emissions
(Metric Tons CO2e)"]
def extract_numeric(value):
    if isinstance(value, str):
        match = re.search(r"[\d,]+\.?\d*", value)
        if match:
            return float(match.group().replace(",", ""))
    return value

for col in numeric_columns:
    data_cleaned[col] = data_cleaned[col].apply(extract_numeric).astype(float)
    # Display extracted numeric values for verification
print(data_cleaned[["Property GFA - Self-Reported (m²)", "Site Energy Use (GJ)", "Total GHG Emissions
(Metric Tons CO2e)"]].head())
```

Match = re.search(r”[\d, ]+\.?\d*”, value) - This pattern captures whole numbers, decimals, comma-seperated values.

return float(match.group().replace(",", "")) - This removes commas before converting the values to float.

### 2. Standardizing Postal Codes

Canadian postal codes follow the **A1A 1A1** format (e.g., "T2M 2M5"). However, there were inconsistencies such as, Missing spaces (e.g., "T2P2M5") and lowercase letters (e.g., "t2p 2m5")



```
# Standardize Postal Codes using Regex
def clean_postal_code(postal_code):
    match = re.match(r"([A-Z]\d[A-Z])\s*(\d[A-Z]\d)", postal_code.strip(), re.I)
    return f"{match.group(1).upper()} {match.group(2).upper()}" if match else postal_code

data_cleaned["Postal Code"] = data_cleaned["Postal Code"].apply(clean_postal_code)
# Display standardized postal codes
print(data_cleaned[["Postal Code"]].head())
```

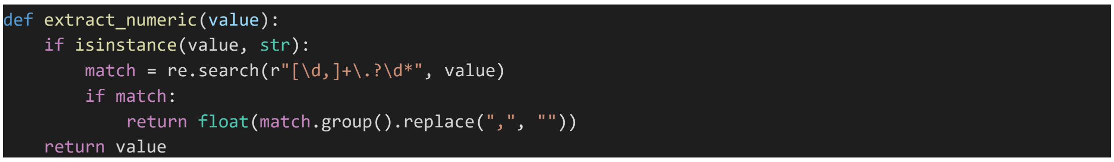
([A-Z]\d[A-Z])\s*(\d[A-Z]\d) captures **two postal code segments**.

\s* allows for **optional spaces**.

.upper() ensures all letters are capitalized.

## 3. Identifying Invalid Numeric Values

Some numeric columns (e.g., **Property GFA**, **Site Energy Use**, **Total GHG Emissions**) contained non-standard values like: Numbers with commas (e.g., "1,200" instead of 1200), Text mixed with numbers (e.g., "~500", "N/A"), Scientific notation (e.g., "4.5E+3").



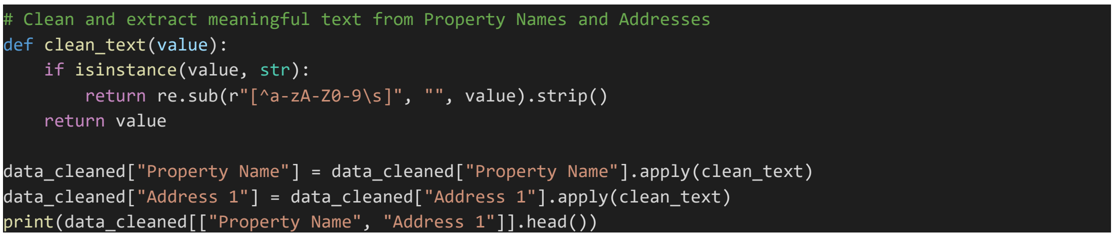
```
def extract_numeric(value):
    if isinstance(value, str):
        match = re.search(r"[\d,]+\.?\d*", value)
        if match:
            return float(match.group().replace(",", ""))
    return value
```

The pattern \d+(\.\d+)? ensures the value is a **valid number** (e.g., 123, 45.67).

· .replace(",", "") removes commas to accommodate numbers formatted as "1,000".

. If a value **does not** match this pattern, it is flagged as **invalid**.

## 4. Clean and Extract meaningful text from property Names and Addresses



```
# Clean and extract meaningful text from Property Names and Addresses
def clean_text(value):
    if isinstance(value, str):
        return re.sub(r"[^a-zA-Z0-9\s]", "", value).strip()
    return value

data_cleaned["Property Name"] = data_cleaned["Property Name"].apply(clean_text)
data_cleaned["Address 1"] = data_cleaned["Address 1"].apply(clean_text)
print(data_cleaned[["Property Name", "Address 1"]].head())
```

[^a-zA-Z0-9\s]

- ^ → **Negation**, meaning "not the following characters."
- a-zA-Z → Allows **uppercase and lowercase letters**.
- 0-9 → Allows **numbers**.
- \s → Allows **whitespace (spaces, tabs, new lines).**

· This pattern **removes anything that is NOT** a **letter, number, or space.**

**GITBUB REPOSITORY LINK:** github.com/Oloma-Prince-Eworitsemoghan/Data-601-HW4