ghg-week-1

June 20, 2025

1 Importing Libraries

[ ]:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
```

2 Load Dataset & Data Preprocessing

[74]:

```
import pandas as pd

# Path to your Excel file
file_path = (r"C:\Users\manas\Downloads\edunet␣
 ↪internship\SupplyChainEmissionFactorsforUSIndustriesCommodities.xlsx")
years = range(2010, 2017)

# Function to load and clean one year's data
def load_year_data(year):
    sheets = [("Commodity", f"{year}_Detail_Commodity"), ("Industry",␣
 ↪f"{year}_Detail_Industry")]
    data_frames = []

    for source, sheet in sheets:
        df = pd.read_excel(file_path, sheet_name=sheet)
        df.columns = df.columns.str.strip()
        df["Source"] = source
        df["Year"] = year
        df.rename(columns={
            f"{source} Code": "Code",
            f"{source} Name": "Name"
        }, inplace=True)
        data_frames.append(df)

    return pd.concat(data_frames, ignore_index=True)
```

```
# Load and stack all years
df = pd.concat([load_year_data(y) for y in years], ignore_index=True)

# Drop irrelevant column if present
df.drop(columns=["Unnamed: 7"], errors="ignore", inplace=True)

# Create a truly unique identifier
df["Code"] = df["Code"].astype(str)
df["Unique_Code"] = df["Code"] + "_" + df["Source"] + "_" + df["Year"].
 ↪astype(str)

# Done! Take a peek
print(f" Final shape: {df.shape}")
df.head()
```

```
 Final shape: (22092, 15)
```

[74]:

```
     Code                                               Name      Substance   \
0  1111A0  Fresh soybeans, canola, flaxseeds, and other o…  carbon dioxide
1  1111A0  Fresh soybeans, canola, flaxseeds, and other o…         methane
2  1111A0  Fresh soybeans, canola, flaxseeds, and other o…   nitrous oxide
3  1111A0  Fresh soybeans, canola, flaxseeds, and other o…      other GHGs
4  1111B0          Fresh wheat, corn, rice, and other grains  carbon dioxide

                                Unit  \
0       kg/2018 USD, purchaser price
1       kg/2018 USD, purchaser price
2       kg/2018 USD, purchaser price
3  kg CO2e/2018 USD, purchaser price
4       kg/2018 USD, purchaser price

   Supply Chain Emission Factors without Margins  \
0                                          0.398
1                                          0.001
2                                          0.002
3                                          0.002
4                                          0.659

   Margins of Supply Chain Emission Factors  \
0                                     0.073
1                                     0.001
2                                     0.000
3                                     0.000
4                                     0.081

   Supply Chain Emission Factors with Margins  \
0                                       0.470
```

```
1                                       0.002
2                                       0.002
3                                       0.002
4                                       0.740

   DQ ReliabilityScore of Factors without Margins  \
0                                               4
1                                               4
2                                               4
3                                               3
4                                               4

   DQ TemporalCorrelation of Factors without Margins  \
0                                                  3
1                                                  3
2                                                  3
3                                                  3
4                                                  3

   DQ GeographicalCorrelation of Factors without Margins  \
0                                                  1
1                                                  1
2                                                  1
3                                                  1
4                                                  1

   DQ TechnologicalCorrelation of Factors without Margins  \
0                                                  4
1                                                  1
2                                                  4
3                                                  3
4                                                  4

   DQ DataCollection of Factors without Margins     Source  Year  \
0                                             1  Commodity  2010
1                                             1  Commodity  2010
2                                             1  Commodity  2010
3                                             1  Commodity  2010
4                                             1  Commodity  2010

             Unique_Code
0  1111A0_Commodity_2010
1  1111A0_Commodity_2010
2  1111A0_Commodity_2010
3  1111A0_Commodity_2010
4  1111B0_Commodity_2010
```

[67]:

```
all_data = []

for year in years:
    try:
        df_com = pd.read_excel(excel_file,␣
 ↪sheet_name=f'{year}_Detail_Commodity')
        df_ind = pd.read_excel(excel_file, sheet_name=f'{year}_Detail_Industry')

        df_com['Source'] = 'Commodity'
        df_ind['Source'] = 'Industry'
        df_com['Year'] = df_ind['Year'] = year

        df_com.columns = df_com.columns.str.strip()
        df_ind.columns = df_ind.columns.str.strip()

        df_com.rename(columns={
            'Commodity Code': 'Code',
            'Commodity Name': 'Name'
        }, inplace=True)

        df_ind.rename(columns={
            'Industry Code': 'Code',
            'Industry Name': 'Name'
        }, inplace=True)

        all_data.append(pd.concat([df_com, df_ind], ignore_index=True))

    except Exception as e:
        print(f"Error processing year {year}: {e}")
```

[48]:

```
len(all_data)
```

[48]:

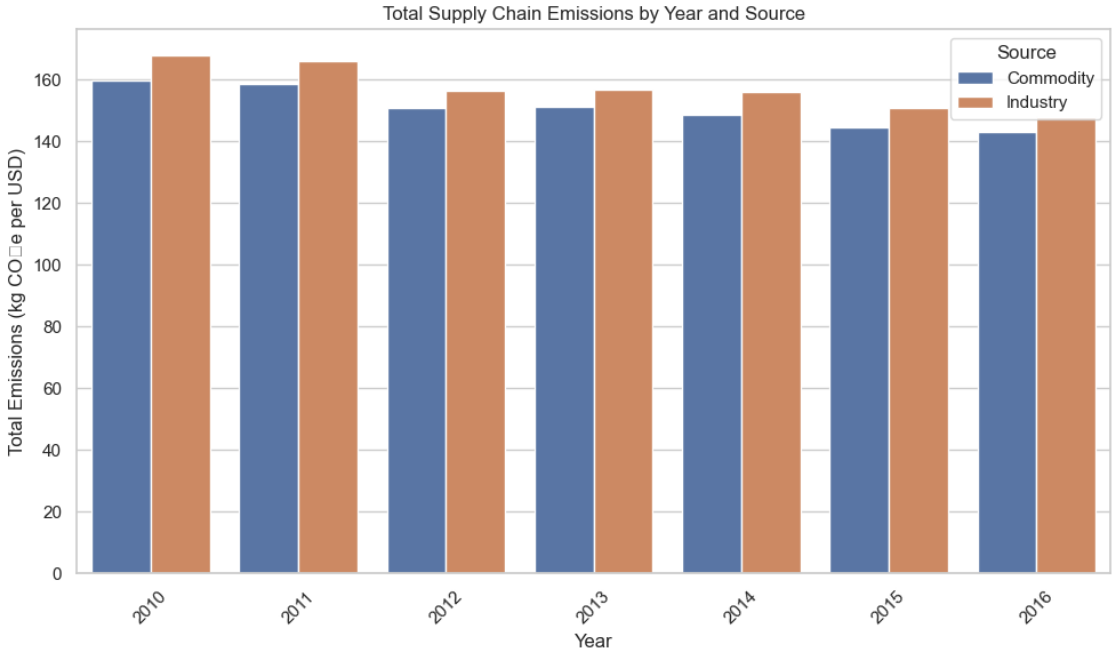
```
7
```

[59]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Set style
sns.set_theme(style="whitegrid")

# Aggregate emissions with margins
df_grouped = df.groupby(['Year', 'Source'])['Supply Chain Emission Factors with␣
 ↪Margins'].sum().reset_index()

plt.figure(figsize=(10, 6))
sns.barplot(data=df_grouped, x='Year', y='Supply Chain Emission Factors with␣
 ↪Margins', hue='Source')
```

```
plt.title('Total Supply Chain Emissions by Year and Source')
plt.ylabel('Total Emissions (kg CO e per USD)')
plt.xlabel('Year')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

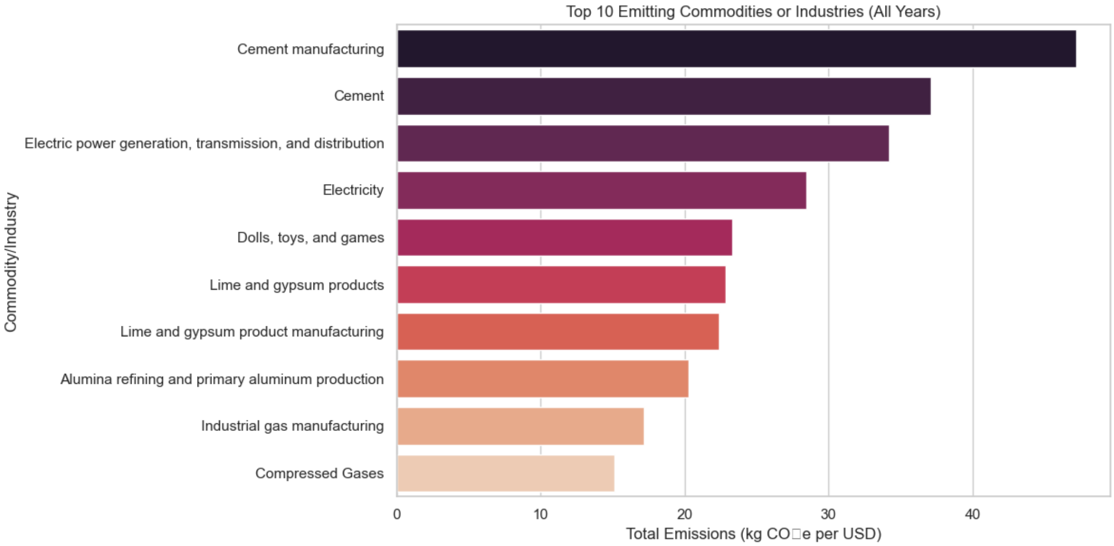
```
C:\Users\manas\AppData\Local\Temp\ipykernel_24836\3608682049.py:16: UserWarning:
Glyph 8322 (\N{SUBSCRIPT TWO}) missing from font(s) Arial.
  plt.tight_layout()
C:\Users\manas\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170:
UserWarning: Glyph 8322 (\N{SUBSCRIPT TWO}) missing from font(s) Arial.
  fig.canvas.print_figure(bytes_io, **kw)
```



[61]:

```
top_emitters = (
    df.groupby('Name')['Supply Chain Emission Factors with Margins']
    .sum()
    .sort_values(ascending=False)
    .head(10)
)

plt.figure(figsize=(12, 6))
sns.barplot(x=top_emitters.values, y=top_emitters.index, palette='rocket')
plt.title('Top 10 Emitting Commodities or Industries (All Years)')
plt.xlabel('Total Emissions (kg CO e per USD)')
plt.ylabel('Commodity/Industry')
plt.tight_layout()
plt.show()
```

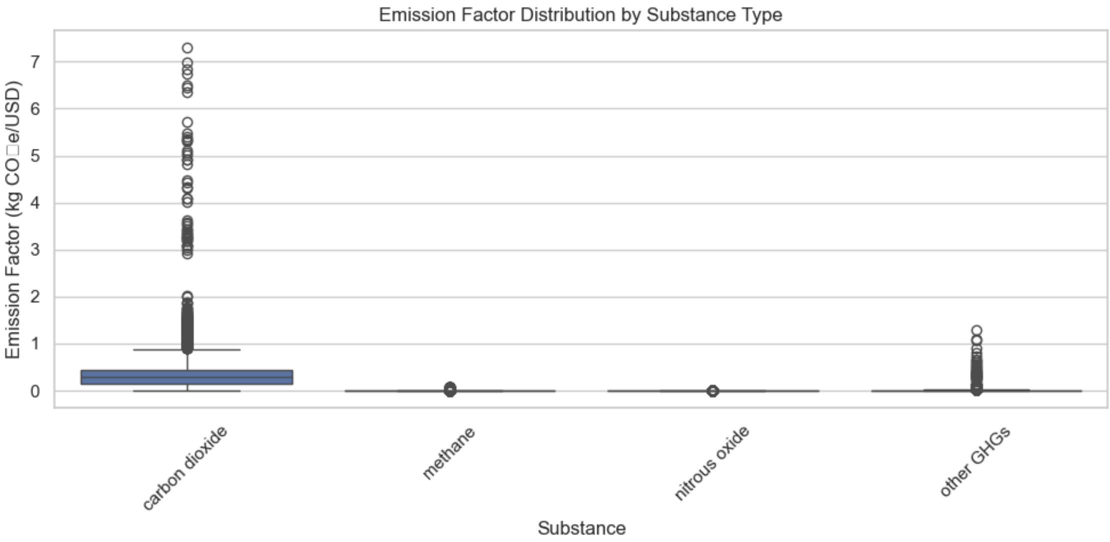
```
C:\Users\manas\AppData\Local\Temp\ipykernel_24836\270884559.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=top_emitters.values, y=top_emitters.index, palette='rocket')
C:\Users\manas\AppData\Local\Temp\ipykernel_24836\270884559.py:13: UserWarning:
Glyph 8322 (\N{SUBSCRIPT TWO}) missing from font(s) Arial.
  plt.tight_layout()
C:\Users\manas\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170:
UserWarning: Glyph 8322 (\N{SUBSCRIPT TWO}) missing from font(s) Arial.
  fig.canvas.print_figure(bytes_io, **kw)
```



[63]:
```
plt.figure(figsize=(10, 5))
sns.boxplot(data=df, x='Substance', y='Supply Chain Emission Factors with␣
  ↪Margins')
plt.title('Emission Factor Distribution by Substance Type')
plt.ylabel('Emission Factor (kg CO e/USD)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\manas\AppData\Local\Temp\ipykernel_24836\1790072078.py:6: UserWarning:
Glyph 8322 (\N{SUBSCRIPT TWO}) missing from font(s) Arial.
  plt.tight_layout()
```

```
C:\Users\manas\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170:
UserWarning: Glyph 8322 (\N{SUBSCRIPT TWO}) missing from font(s) Arial.
  fig.canvas.print_figure(bytes_io, **kw)
```



[ ]: