# Project Breakdown: Carbon Emissions Analysis

## Introduction

In this project, I aim to analyze the carbon emissions trends of major oil and gas companies over time. By leveraging Python for data processing, categorization, normalization, and visualization, I can extract meaningful insights about how different-sized companies contribute to emissions. The goal is to determine whether emission patterns have changed over time and whether companies with different emission levels exhibit distinct trends.

The specific question I sought to answer was:  
"How have total emissions from major oil & gas companies changed over time?"

To address this, I:

* Collected and cleaned the data, ensuring accuracy.
* Categorized companies into emission groups (*Low, Medium, High*) based on quantiles.
* Applied normalization to fairly compare emissions across companies of different sizes.
* Created visualizations to observe emission trends over time for each category.

This approach allows for a clearer understanding of how carbon emissions have evolved and whether certain company groups have changed their behaviors more than others.

## Coding Details

### 1️ Overview of the Project

The purpose of this project was to analyze **carbon emissions trends** from major oil and gas companies over time.  
This involved:

* **Cleaning and transforming the data**
* **Applying binning and normalization**
* **Visualizing the results using Matplotlib**

### 2️ Data Loading & Preprocessing

#### 🔹 Loading the Data

*import pandas as pd*

*data = pd.read\_csv("C:/Users/marti/Documents/Documets +/UpSkill/Coding/2025/Carbon Projects/emissions\_high\_granularity.csv")*

✅ **Purpose:**

* Loads the dataset into a **Pandas DataFrame**.
* *pd.read\_csv()* reads CSV files and stores them in a structured table format.

🔄 **Alternative Options:**

* *pd.read\_excel("file.xlsx")* → If the data was in an Excel file.
* *pd.read\_sql(query, connection)* → If the data was in a database.

#### 🔹 Selecting Relevant Columns

*dataQ1 = data[["year", "total\_emissions\_MtCO2e", "parent\_entity"]]*

✅ **Purpose:**

* Extracts only the **necessary columns** from the dataset.
* **Avoids memory overload** by removing unneeded columns.

🔄 **Alternative Options:**

* *data.drop(columns=["col1", "col2"])* → Drops specific columns instead of selecting.
* *data.iloc[:, [0, 2, 5]]* → Selects columns by their index position.

#### 🔹 Handling Duplicates & Missing Values

*print("No. Missing Column Values:\n", dataQ1.isnull().sum())*  
*print(dataQ1.duplicated().sum())*

*dataQ1 = dataQ1.drop\_duplicates()*

✅ **Purpose:**

* Identifies and removes duplicate records to **prevent data distortion**.

🔄 **Alternative Options:**

* *dataQ1.dropna()* → Removes rows with missing values.
* *dataQ1.fillna(value)* → Replaces missing values with a default.

#### 🔹 Converting Year Column to Datetime Format

*dataQ1["year"] = pd.to\_datetime(dataQ1["year"], format="%Y")*

✅ **Purpose:**

* Ensures that *year* is recognized as a **time-based** variable.
* Allows **time-series operations** like resampling.

🔄 **Alternative Options:**

* *dataQ1["year"] = dataQ1["year"].astype(int)* → If *year* was numeric.
* *dataQ1["year"] = pd.to\_datetime(dataQ1["year"], errors="coerce")* → Avoids crashes if invalid values exist.

### 3️ Categorizing Data: Binning

**🔹 Creating Bins for Company Emissions**

*quantiles = dataQ1["total\_emissions\_MtCO2e"].quantile([0, 0.33, 0.66, 1.0]).values*  
*bin\_names = ["Low", "Medium", "High"]*

*dataQ1 = dataQ1.copy()*  
*dataQ1.loc[:, "emission\_category"] = pd.cut(dataQ1["total\_emissions\_MtCO2e"], bins=quantiles, labels=bin\_names, include\_lowest=True)*

✅ **Purpose:**

* **Categorizes** companies into *"Low"*, *"Medium"*, and *"High"* emission groups.
* ***.loc[] is used*** to modify the DataFrame safely (prevents warnings).

🔄 **Alternative Binning Methods:**

* **Equal-width bins**:  
  *bins = np.linspace(min(dataQ1["total\_emissions\_MtCO2e"]), max(dataQ1["total\_emissions\_MtCO2e"]), 4)*  
  → Creates bins with equal ranges rather than **equal-sized groups**.
* **Manual Bins**:  
  *bins = [0, 1000, 5000, np.inf]*  
  *bin\_names = ["Small", "Medium", "Large"]*  
  → Custom bin ranges.

### 4️ Aggregating Data

*emissions\_by\_category = dataQ1.groupby(["year", "emission\_category"], observed=True)["total\_emissions\_MtCO2e"].sum().unstack()*

✅ **Purpose:**

* Groups emissions by **year** and **category**, then sums them up.

🔄 **Alternative Grouping Methods:**

* **Monthly Aggregation:**  
  *dataQ1.resample("M", on="year")["total\_emissions\_MtCO2e"].sum()*  
  → Useful for more granular time analysis.
* **Mean Instead of Sum:**  
  *dataQ1.groupby("year")["total\_emissions\_MtCO2e"].mean()*  
  → Compares **average** emissions rather than total.

### 5️ Normalization

*from sklearn.preprocessing import MinMaxScaler*

*scaler = MinMaxScaler(feature\_range=(0, 1))*  
*normalized\_emissions = pd.DataFrame(scaler.fit\_transform(emissions\_by\_category), index=emissions\_by\_category.index, columns=emissions\_by\_category.columns)*

✅ **Purpose:**

* Ensures **all values** fall between **0 and 1** for fair comparison.

🔄 **Alternative Scaling Methods:**

* **Standardization (Z-score):**  
  *from sklearn.preprocessing import StandardScaler*  
  *scaler = StandardScaler()*  
  → Centers data around **mean = 0, std = 1**.

**6️ Plotting the Line Chart**

*plt.figure(figsize=(18, 9))*

*for category in normalized\_emissions.columns:*  
*plt.plot(normalized\_emissions.index, normalized\_emissions[category], marker="o", linestyle="-", label=category)*

✅ **Purpose:**

* Plots **each emission category’s trend over time**.

🔄 **Alternative *marker* Styles:**

* *"o"* → Circles (default)
* *"s"* → Squares
* *"D"* → Diamonds

🔄 **Alternative *linestyle* Options:**

* *"-"* → Solid line
* *"--"* → Dashed line
* *":"* → Dotted line

**🔹 X-Axis Formatting**

*import matplotlib.dates as mdates*

*plt.xticks(rotation=30, ha="right")*  
*plt.gca().xaxis.set\_major\_locator(mdates.YearLocator(10))*  
*plt.gca().xaxis.set\_major\_formatter(mdates.DateFormatter('%Y'))*

✅ **Purpose:**

* Prevents **overlapping years** and ensures **readability**.

**🔹 Final Touches**

*plt.xlabel("Year", fontsize=14)*  
*plt.ylabel("Normalized Emissions (0-1 Scale)", fontsize=14)*  
*plt.title("Normalized Emissions Trends by Company Size (Low, Medium, High)", fontsize=16)*

*plt.grid(True, linestyle="--", alpha=0.7)*  
*plt.legend(title="Emission Category", bbox\_to\_anchor=(1.05, 1), loc="upper left")*

*plt.savefig('Matplot\_Linechart.png', dpi=300, bbox\_inches="tight")*  
*plt.show()*

✅ **Purpose:**

* Adds **labels, title, legend, and styling** for a **professional** look.

**🔹 Summary of What You've Achieved**

✔ **Loaded & cleaned the dataset**  
✔ **Created emission categories** using binning  
✔ **Grouped & normalized emissions for fair comparisons**  
✔ **Plotted clear, labeled visualizations**

🔥 **Well done! This is professional-level data analysis!** 🚀

Would you like to move forward with **GitHub upload or next question analysis?**

## Findings & Conclusion

From the analysis, the following insights were drawn:

1️⃣ **General Increase in Emissions Over Time**

* Emissions have generally **increased over time**, particularly from the **mid-20th century onward**.
* This aligns with the **global rise in industrial activity and fossil fuel reliance**.

2️⃣ **Different Growth Patterns Among Company Sizes**

* **High-emission companies** show a **gradual increase**, accelerating in the latter half of the 20th century.
* **Medium-emission companies** exhibit **more fluctuations**, likely due to market shifts or regulatory impacts.
* **Low-emission companies** had smaller changes, but their trend still reflects an upward shift.

3️⃣ **Recent Declines in Some Categories**

* The **late-2000s to 2020s** show **some declines**, possibly due to:
  + **Stronger climate policies**
  + **Economic slowdowns**
  + **Advancements in cleaner energy sources**

## Further Areas of Exploration

While this analysis provides **useful trends**, several additional questions could be explored with further data:

1️⃣ **What are the key factors driving emission changes?**

* Does this align with **policy changes** (e.g., carbon taxes, emissions trading schemes)?
* Can we correlate emission reductions with **company investments in renewables**?

2️⃣ **Which companies are leading the transition to clean energy?**

* Identify **which major oil & gas firms** are reducing emissions **faster than others**.

3️⃣ **How do emissions compare across different geographical regions?**

* The current dataset focuses on **individual companies**, but analyzing **regional emissions trends** could provide more **policy-relevant insights**.

4️⃣ **Impact of carbon pricing & regulations on emissions**

* Does the presence of **carbon taxes** or **cap-and-trade programs** correspond to emission reductions?

## Additional Data Needed for a Deeper Analysis

To explore these further, I would need additional datasets such as:

✅ **Government climate policy data** (carbon taxes, subsidies for renewables).  
✅ **Financial reports of oil & gas companies** (investments in clean energy).  
✅ **Geographical emissions data** (regional breakdowns of CO₂ contributions).  
✅ **Macroeconomic data** (GDP, energy consumption trends, industrial output).

By integrating these additional datasets, I could **identify specific drivers of emissions changes**, distinguish between **policy-driven reductions vs. market-driven shifts**, and **pinpoint companies that are leading in sustainability efforts**.

## Final Thoughts

This project has demonstrated how **Python enables efficient data analysis** of large-scale emissions data, offering insights that would be difficult to extract in traditional tools like Excel. The findings suggest that **emissions have risen significantly over time**, but **recent trends indicate potential shifts towards reductions in some categories**.

Future research could **dive deeper into company-level policies**, **regional differences**, and **external economic influences** to better understand the mechanisms behind these emission trends.

🔥 **Next Steps?** Would you like to:

* **Push this project to GitHub?**
* **Start the next analysis question?**
* **Improve the visualization further?**

Let me know how you’d like to proceed! 🚀