PROJECT REPORT

Supply Chain Sustainability: Tracking $C0_2$ Emissions

IE 6600:
Computation and Visualization for Analytics

Prof. Sivarit Sultornsanee

Group 17:

Gokul Menon	002105689
Ankita Nitin Bandal	002783589
Shreyans Hiteshkumar Trivedi	002766272

Table of Contents

1. Introduction

The prime cause of global warming was identified to be greenhouse gasses such as carbon dioxide, Nitrogen Dioxide, and Sulphur Dioxide. Among these gasses, carbon dioxide is the major contributor. Supply chains are often a company's largest source of carbon emissions and are crucial to the fight against global warming. Investors and consumers have increased their desire for transparency in sustainable development over time. Investors now give a greater importance to a company's sustainability when determining the worth and adaptability of a company. An increasing number of businesses are investing resources, to develop competencies for sustainability reporting and identify the most effective methods for a sustainable supply chain. In this project we are using historical e-commerce data of a Brazilian company Olist. The dataset, which was obtained from Kaggle, has information of 100k orders from 2016 to 2018 made at multiple marketplaces. The CO_2 emissions for each order was calculated using the formula:

$$\boldsymbol{E_{CO2} = W_{goods} \times D \times F_{mode}}$$

$\boldsymbol{E_{CO2}}$ is the emissions in Kilograms of CO_2 equivalent (kgCO2eq)
$\boldsymbol{W_{goods}}$ is the weight of goods (ton)
D is the distance from seller to customer (km)
$\boldsymbol{F_{mode}}$ is the emissions factor for the transportation mode (kgCO2eq/t.km)

Data cleaning, exploratory data analysis, and visualisations were conducted to gain meaningful insights from the data; these functions were conducted on Python using libraries like Numpy, Matplotlib, Pandas, Seaborn, Plotly, and Geopy. Various visualisations like bar plots, scatter plots, and bubble maps were utilized to see which of the products/orders/cities contributed the most to emissions. Visualisation dashboards were created using Tableau.

2. Research Questions

Listed below are a few questions that our project will aim to answer:

- Which cities of Brazil report highest emissions? and by how much?
- Transportation of which product is responsible for the most CO_2 emission?
- How can the emissions be reduced?

3. Summary of Results

The value of the freight and the amount of CO2 released are correlated. As the weight increases, so do the carbon emissions. Because of the direct relationship between travel distance and carbon emissions, emissions increase as distance increases. The top line in the line graph reflects carbon emission for the corresponding month, while the bottom line in the line graph shows the total number of orders placed in that month. It is obvious that the carbon emissions rise in direct proportion to the quantity of orders. The bar graph shows the amount of carbon dioxide released when a product is delivered from one seller state to another to the buyer state. The most orders are placed in So Paulo, and as a result, the emissions are likewise quite high. Amparo, a seller city, generates 12.8 kgCO2eq of the total CO2 emissions.

4. Data Sources

The Olist e-commerce dataset was obtained from Kaggle. Olist is the largest online departmental store in Brazilian marketplaces. Olist effortlessly links small companies from across Brazil to channels, with a single contract. These business owners may use Olist logistics partners to sell their goods through the Olist Store and send them straight to customers.
The dataset contains information about 100,000 orders completed between years 2016 and 2018 at various marketplaces in Brazil. Since this is real commercial data, references to seller companies have been replaced to maintain anonymity.

Table 1 below displays the different datasets and their contents:

Dataset	Columns	Description
Customer's Dataset	customer_id	key to the orders dataset
	customer_unique_id	unique identifier of a customer
	customer_zip_code_prefix	first five digits of customer zip code
	customer_city	customer city name
	customer_state	customer state
Geolocation Dataset	geolocation_zip_code_prefix	first 5 digits of zip code
	geolocation_lat	latitude
	geolocation_lng	longitude
	geolocation_city	city name
	geolocation_state	state
Item Orders	order_id	order unique identifier
	order_item_id	sequential identifying number of items included in the same order
	product_id	product unique identifier
	seller_id	seller unique identifier
	shipping_limit_date	Seller shipping limit date for handing over to the logistic partner
	price	item price
Products	product_id	unique product identifier
	product_name_lenght	number of characters extracted from the product name
	product_description_lenght	number of characters extracted from the product description
	product_photos_qty	number of product published photos
	product_weight_g	product weight measured in grams
	product_length_cm	product length measured in centimeters
	product_height_cm	product height measured in centimeters
	product_width_cm	product width measured in centimeters
Sellers	seller_id	seller unique identifier
	seller_zip_code_prefix	first 5 digits of seller zip code
	seller_city	seller city name
	seller_state	seller state
Orders	order_id	unique identifier of the order
	customer_id	key to the customer dataset
	order_status	Reference to the order status (delivered, shipped, etc)

Table 1

5. Methods

a. Data Cleaning

All six datasets (products, order items, sellers, customers, orders, geolocation) contained a lot of Null values. Since these incomplete rows will significantly affect the visualizations, these rows were dropped using the dropna() function. Next step was dropping duplicate rows using drop_duplicates() function.

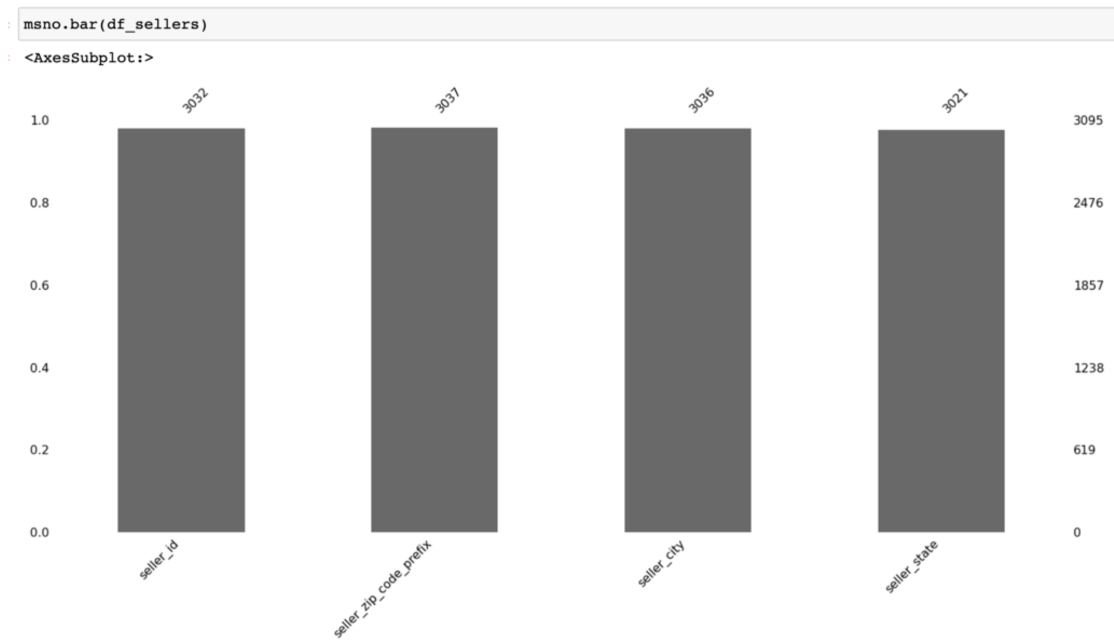

Figure 1: Bar plot of missing values in columns of Sellers Dataset

Using drop() function, all unnecessary columns that were irrelevant to our project were dropped.

The Seller Zip code datatype was modified from string to int. Zip codes are supposed to have 5 digits, however it was discovered that some of the Zip codes were missing a digit. These Zip Codes were corrected by adding a “0” before it.

```
df2['seller_zip_code_prefix'] = df2['seller_zip_code_prefix'].astype(int)
df2['seller_zip_code_prefix'] = df2['seller_zip_code_prefix'].apply(lambda x : str(x).zfill(5))
```

Figure 2: Code for datatype modification and correcting zip code

b. Merging Datasets

Figure 3 below displays the datasets and how they are interconnected using different primary keys. The primary keys are product_id, seller_id, zip_code_prefix, customer_id, and order_id.

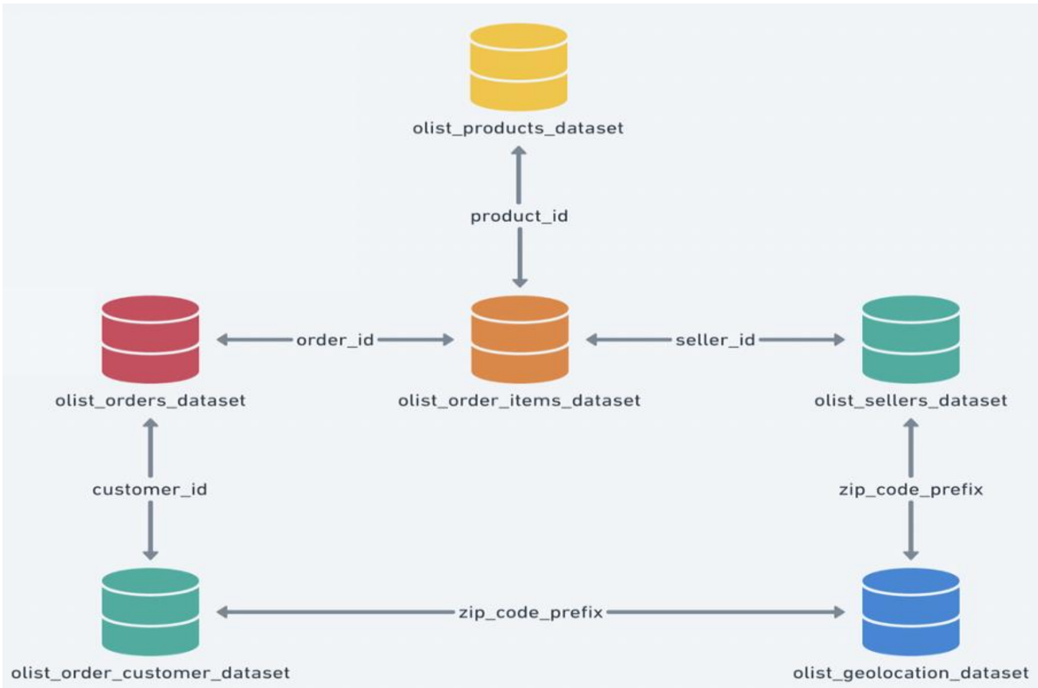

Figure 3: Data Schema

c. Exploratory Data Analysis

Initial investigation was conducted on the merged dataset to form a first impression of the data by try to find patterns and anomalies.

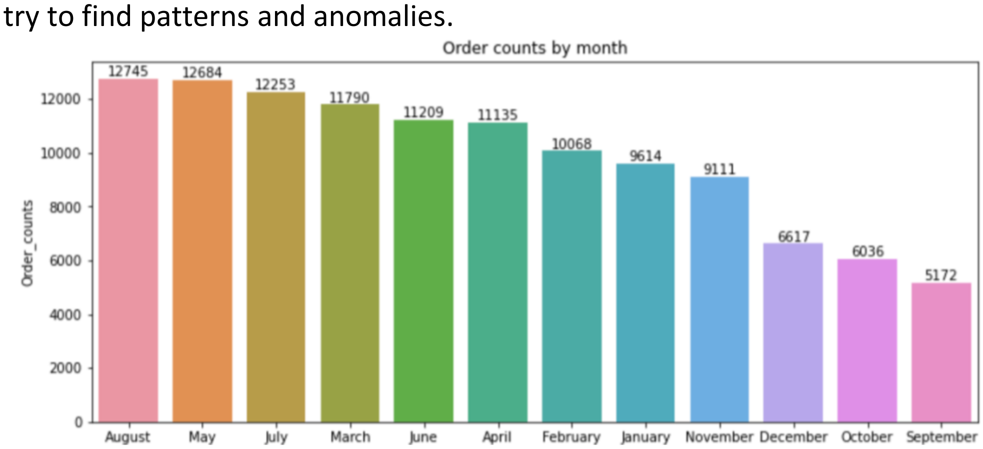

Figure 4: Bar plot showing order counts by month

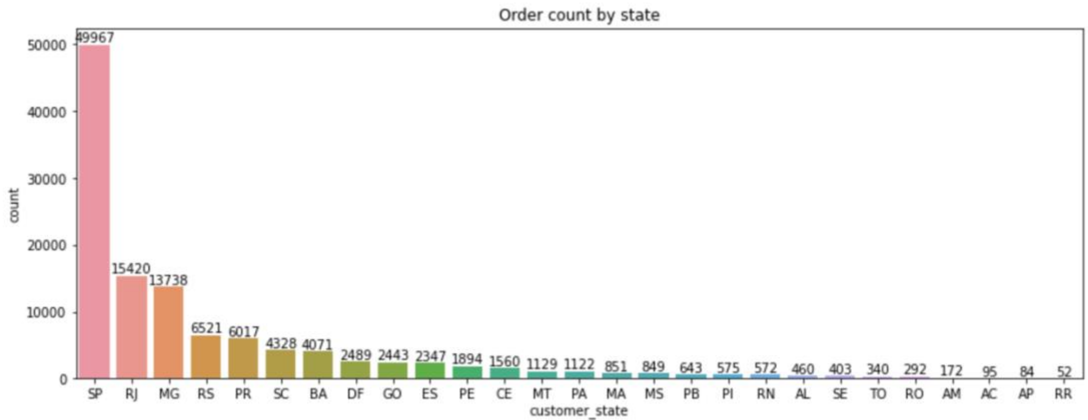

Figure 5: Bar plot showing order count by states in Brazil

We can observe from Figure 4 that month of August has highest order count, and Figure 5 shows that São Paulo is the state with highest number of orders. We can infer that orders to São Paulo and the month of August will be the major contributors to CO_2 emissions.

d. Distance Calculation

The function distance() from the Geopy library was used to calculate the distance from Seller to Customer coordinates (Figure 6). Inputs for this function was seller and customer latitudes and longitudes.

```
for i in final_df.index:
    final_df.loc[i,'Distance in KM'] = distance.distance((final_df.loc[i,"cust_lat"], final_df.loc[i, "cust_lng"]),
                                                         ((final_df.loc[i, "sell_lat"], final_df.loc[i, "sell_lng"])))
```

Figure 6: Python code for distance calculation using coordinates

e. CO_2 Emission Calculation

After the data was converted to the required units, CO2 emission was calculated using distance, weight of goods, and emission factor of transportation mode. For this project, we consider the transportation mode to be trucks.

```
for i in final_df.index:
    final_df.loc[i,'CO2_Emission'] = ( final_df.loc[i,'Distance in KM']
                                      * final_df.loc[i,'product_weight_ton']
                                      * 0.042)
```

Figure 7: Python code for carbon emission calculation

6. Results

a. Python Visualizations

A few visualizations were put together in Python using Seaborn, Plotly, and Matplotlib.

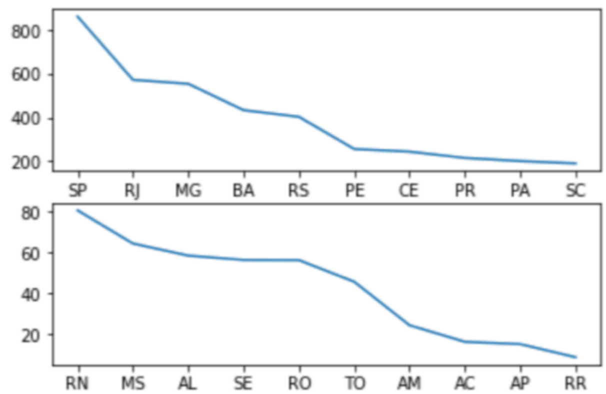

Figure 8: 10 states with Highest and Lowest emissions

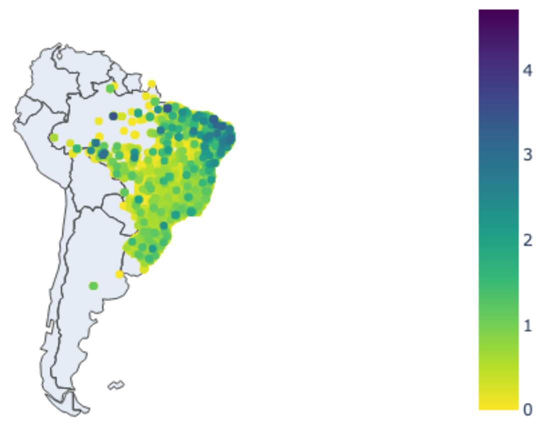

Figure 9: Geo-Scatter plot for emissions across Brazil

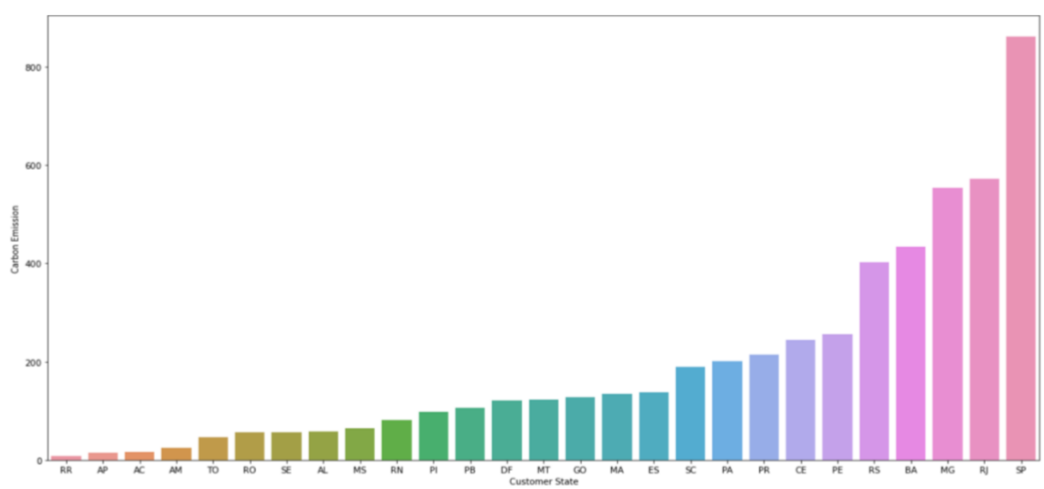

Figure 10: Bar plot showing total emissions per state in ascending order

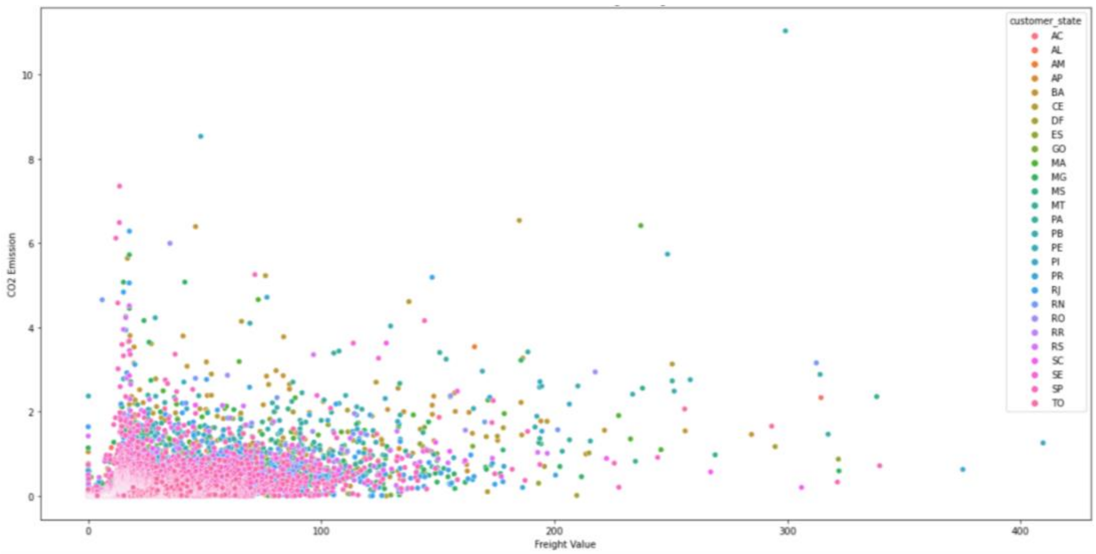

Figure 11: Bar plot showing carbon emissions according to Freight Value

b. Tableau Visualizations

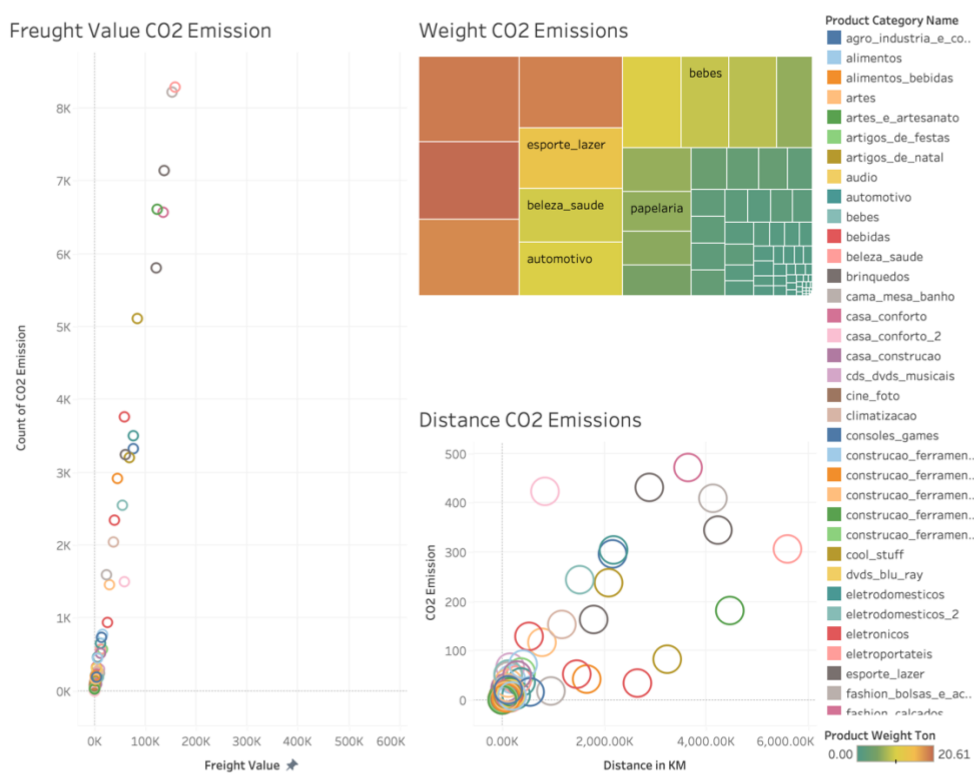

Figure 12: Tableau Dashboard 1

As we can see from Figure 12, there is a direct correlation between freight value and CO_2 emissions. As the weight increases, so does the carbon emission. Carbon emission is directly proportional to transportation distance, i.e., emissions increase as the distance increases.

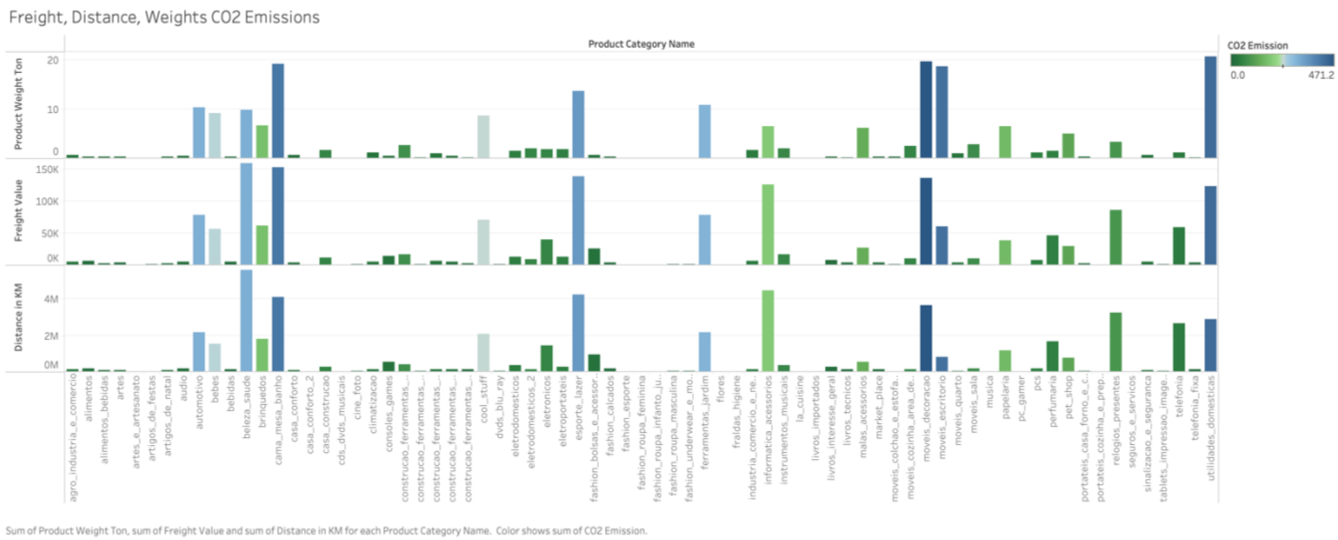

Figure 13: Comparison of Carbon Emission with 3 factors

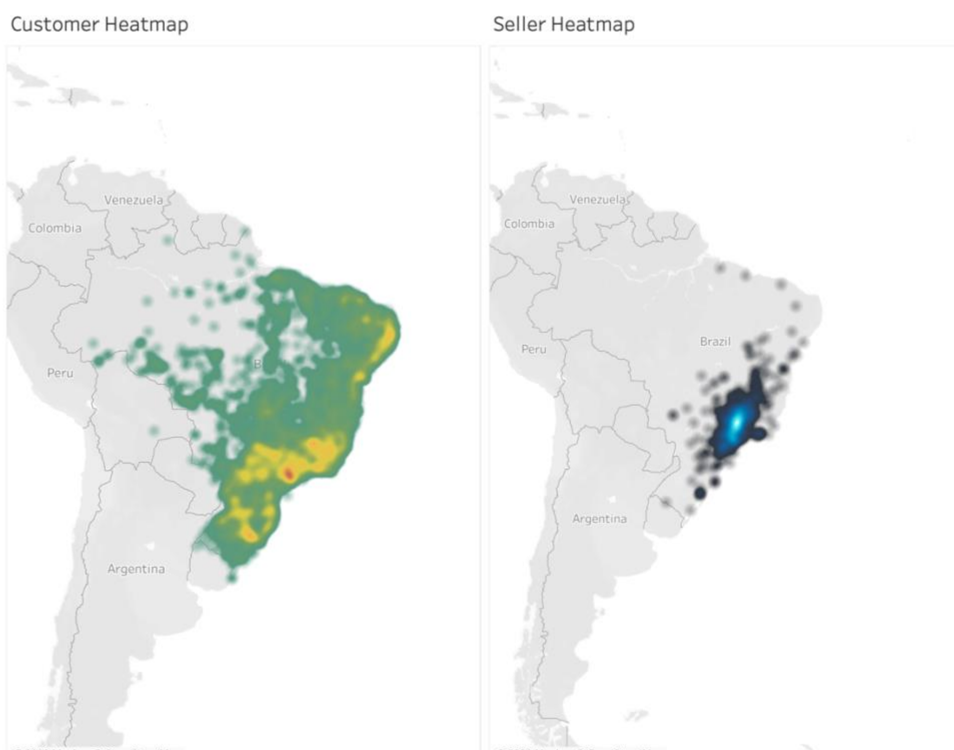

Figure 14: Carbon Emission Heatmap for Customer and Seller States in Brazil

From Figure 14 we can infer that, customer city Barueri has the highest carbon emission of 9.4 kgCO2eq. Seller city Amparo contributes the most towards CO_2 emissions with 12.8 kgCO2eq.

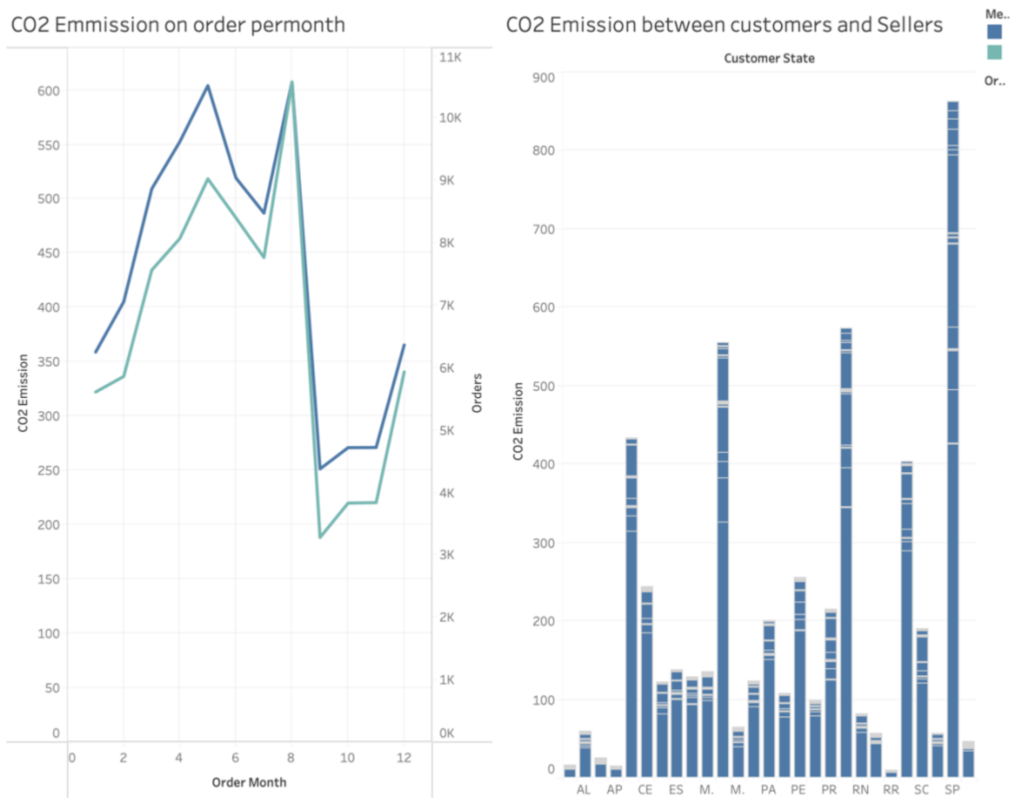

Figure 15: Emissions of orders per month, and emission between customer and seller

The line graph in Figure 15, lower line represents total number of orders placed in a month, and the upper line represents carbon emission for the corresponding month. It is clear that as the number of orders increases, so does the carbon emissions. The bar chart represents carbon emission when a product is shipped from a particular seller state to customer state. Number of orders within São Paulo are highest, hence the emissions are also very high.

7. Limitations and Future Work

a. Limitations

The exact transportation mode used for shipping the product from seller to customer was not recorded. Since the shipments are within Brazil, the mode of transportation was assumed to be Trucks. The exact routes used for shipping are not known, so the shortest distance from seller to customer was calculated.

b. Future Work

The future scope of this project is to gather more detailed shipping data that describes the shipping routes, and mode of transportation used. With this data we can optimize the routes, and suggest better transportation modes and routes to reduce carbon emissions in the supply chain. Demand forecasting can be conducted using historical data, which can optimize inventory levels, which can help reduce emissions further.

8. References

Source Dataset: "Brazilian E-Commerce Public Dataset by Olist." *Brazilian E-Commerce Public Dataset by Olist | Kaggle*, /datasets/olistbr/brazilian-ecommerce.

Saci, Samir. "Supply Chain Sustainability Reporting With Python." *Medium*, 10 Dec. 2022, towardsdatascience.com/supply-chain-sustainability-reporting-with-python-161c1f63f267.

Molin, Stefanie. "Hands-On Data Analysis With Pandas." *A Python Data Science Handbook for Data Collection, Wrangling, Analysis, and Visualization, 2nd Edition*, 2021.

Belorkar, Abha, et al. "Interactive Data Visualization With Python." *Present Your Data as an Effective and Compelling Story, 2nd Edition*, 2020.

Vaingast, Shai. "Beginning Python Visualization." *Crafting Visual Transformation Scripts*, 2014.

"Plotly." *Plotly Python Graphing Library*, plotly.com/python.

"Get Started Mapping With Tableau." *Get Started Mapping With Tableau - Tableau*, help.tableau.com/current/pro/desktop/en-us/buildexamples_maps.htm.

Dekanovsky, Vaclav. "Driving Distance Between Two or More Places in Python." *Medium*, 15 Dec. 2020, towardsdatascience.com/driving-distance-between-two-or-more-places-in-python-89779d691def.