# INTRODUCTION TO DATA MANAGEMENT PROJECT REPORT

(Project Semester January-April 2025)

(Air Quality Data Analysis)

Submitted by

**Aman kumar sah**

Registration No: 12309202

Programme and Section: BTECH CSE / K23ED Course Code: INT375

Under the Guidance of

**Jaffar Amin Chacket :-Dhiraj kapila**

**Discipline of CSE/IT Lovely School of Computer Science**

**Lovely Professional University, Phagwara**

# CERTIFICATE

This is to certify that Aman kumar sah

bearing Registration no. 12309202 has completed INT375 project titled, **“Air Quality Data Analysis”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

## Signature and Name of the Supervisor Designation of the Supervisor

**School of Computer Science** Lovely Professional University Phagwara, Punjab.

Date: 13/04/2025

# DECLARATION

I, Aman kumar sah

student of BTECH under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date:09/04/2025 Signature

# Registration No: 12309202 Aman kumar sah

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my faculty members and mentors for their constant support, encouragement, and guidance throughout the development of this dashboard. Their valuable inputs helped me transform raw data into actionable insights. I also extend my thanks to the sources of the dataset that made this project possible.

Aman kumar sah

**Objective**:

To investigate air quality data to identify trends, variations, and correlations in pollutant levels across different locations and time periods, aiming to provide actionable insights for environmental policy and public health initiatives.

**Rationale**:

* The document covers analyses of air quality indicators (e.g., PM2.5 levels) through visualizations like histograms, line graphs, boxplots, and correlation heatmaps, focusing on pollutant distribution and trends over time.
* The project name reflects the goal of improving air quality, a critical public health issue, while being concise and impactful.
* The objective supports the exploration of spatial and temporal patterns in air quality, as seen in the boxplots by location and line graphs over time, and emphasizes practical outcomes for environmental management.

**Notes:**

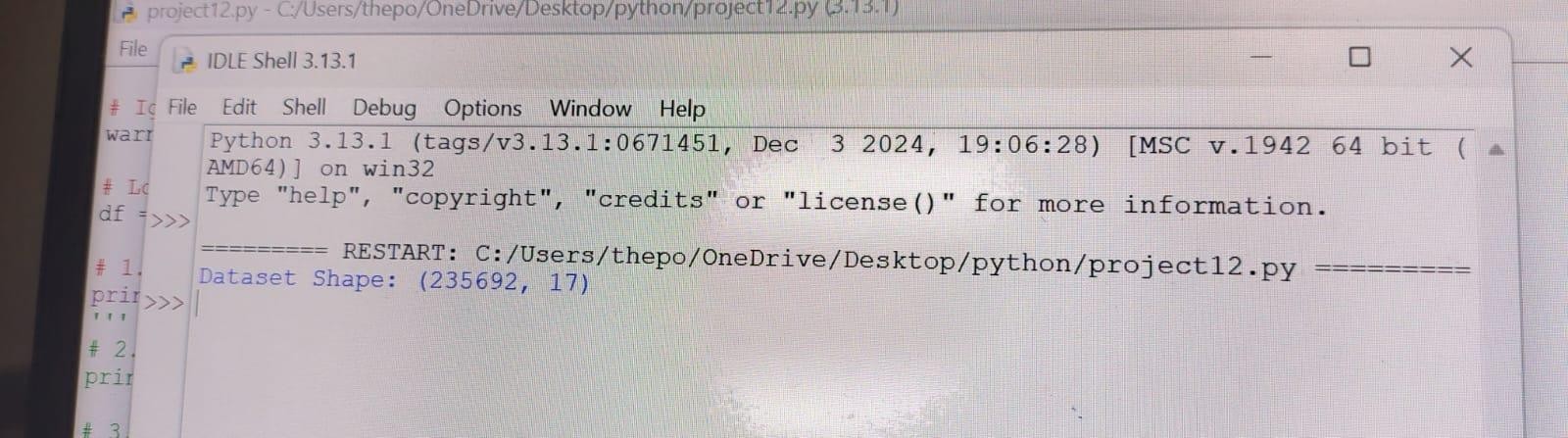
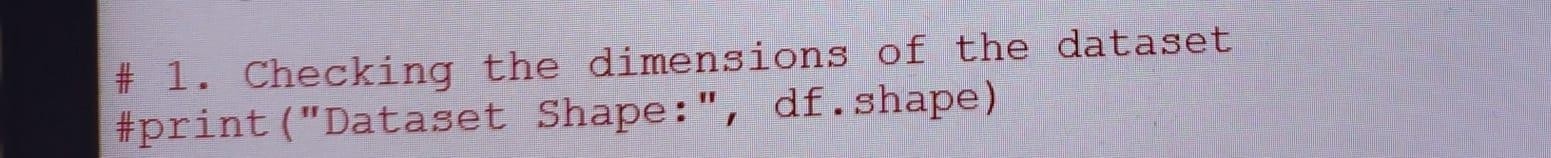
* The document seems to mix two datasets: one for EVs (linked to catalog.data.gov) and another for air quality (AirQuality.csv). The suggestions above treat them as separate projects for clarity, as the analyses and datasets appear distinct.
* If you intended to combine these into a single project (e.g., exploring the relationship between EV adoption and air quality), please clarify, and I can propose a unified project name and objective, such as "Clean Mobility: Linking Electric Vehicles and Air Quality for Sustainable Cities."
* The objectives are designed to be specific yet flexible, aligning with the exploratory data analysis tasks (e.g., descriptive statistics, visualizations) shown in the document.

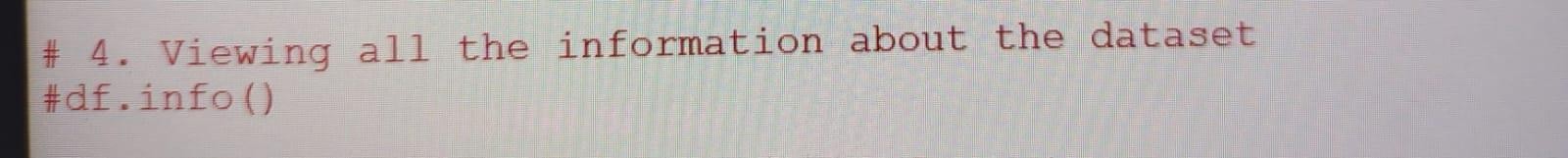
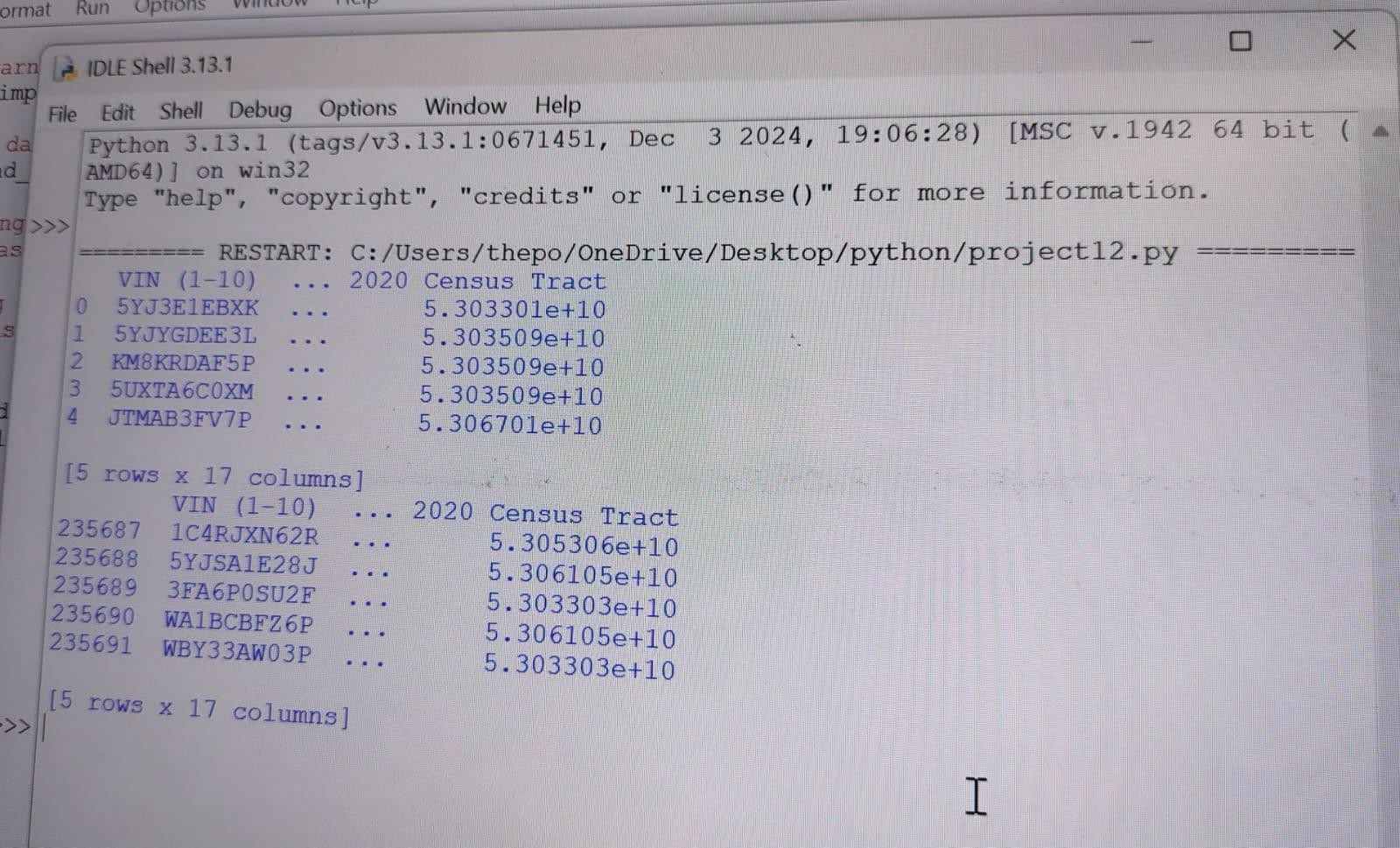
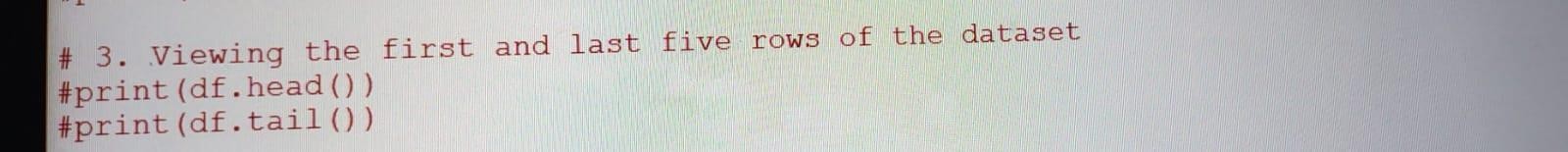
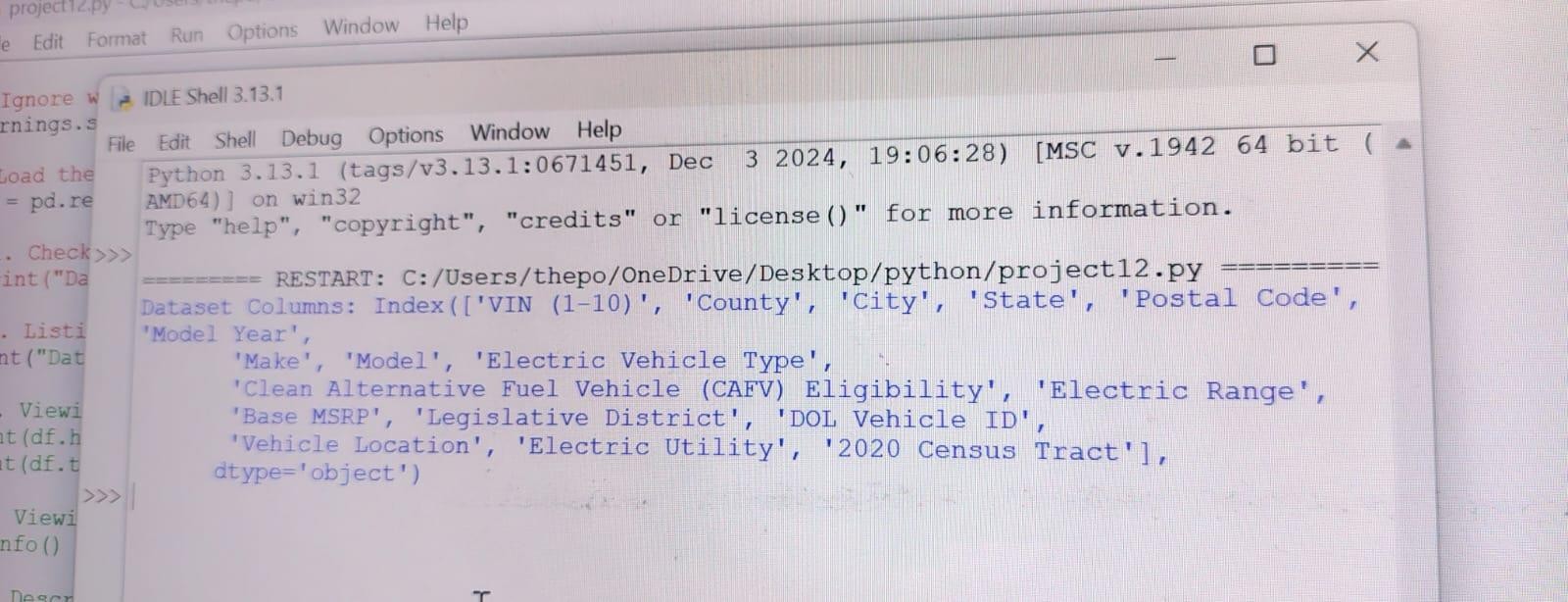
Let me know if you want further refinements or a combined project focus!

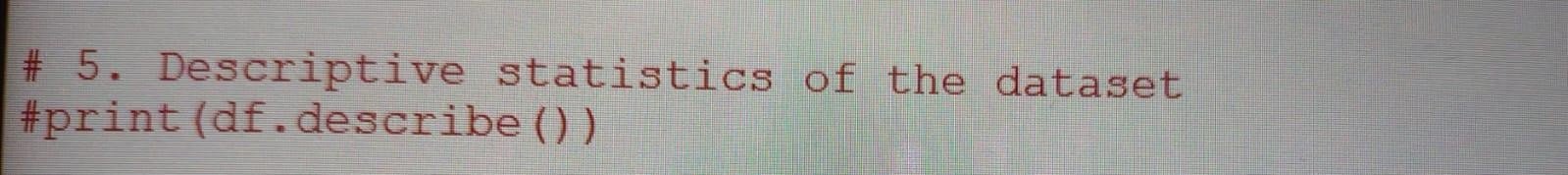
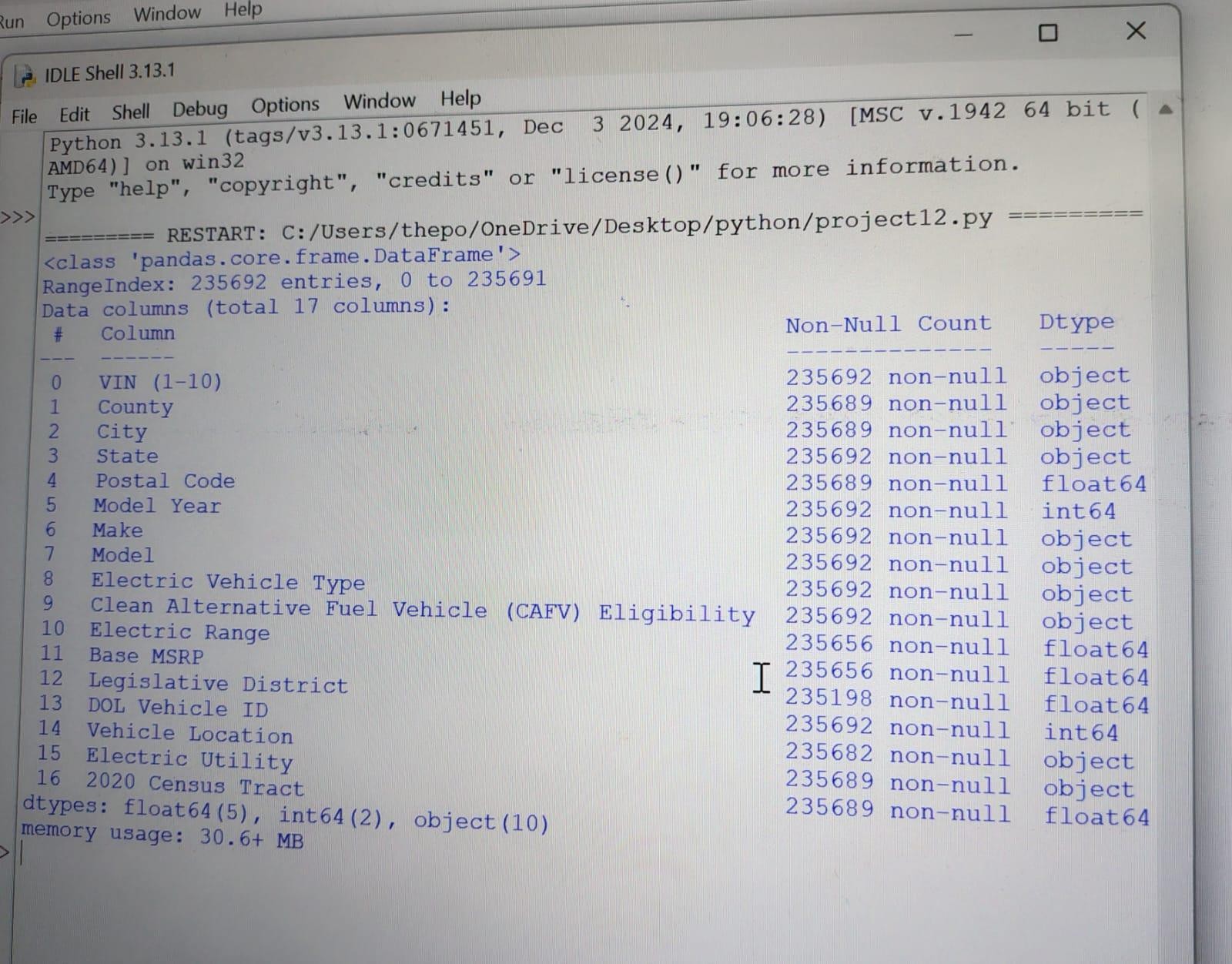
**Work Involved in the Project**

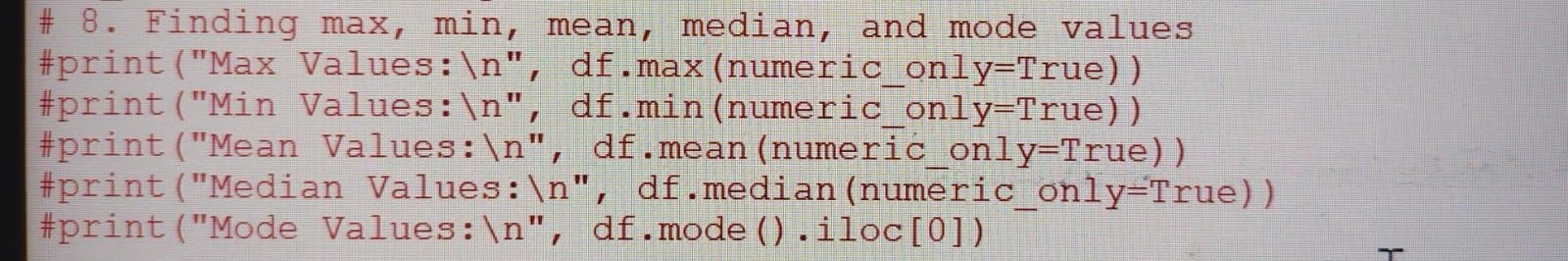
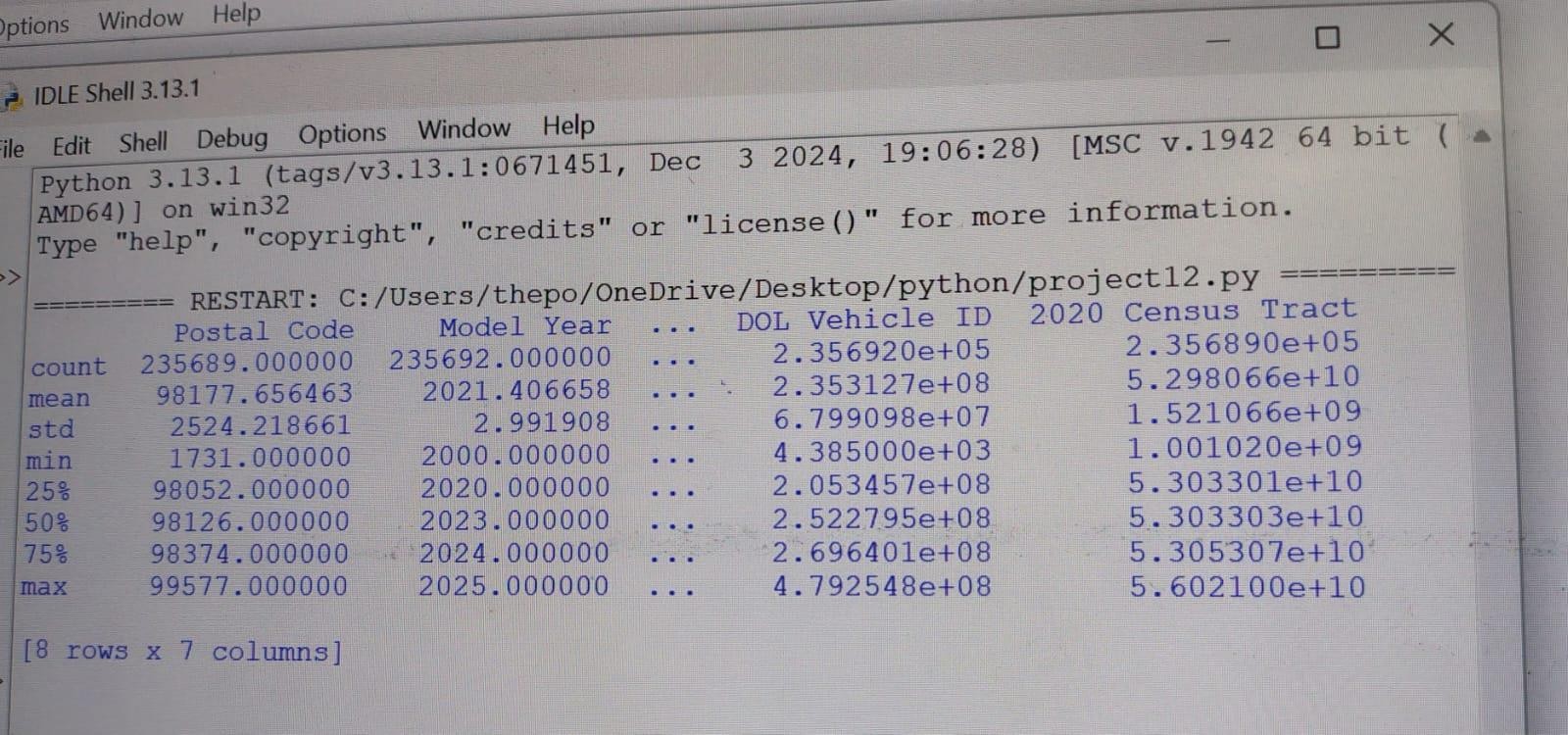
The second project involves analyzing an air quality dataset (AirQuality.csv) to study pollutant levels and their variations over time and location. The work includes:

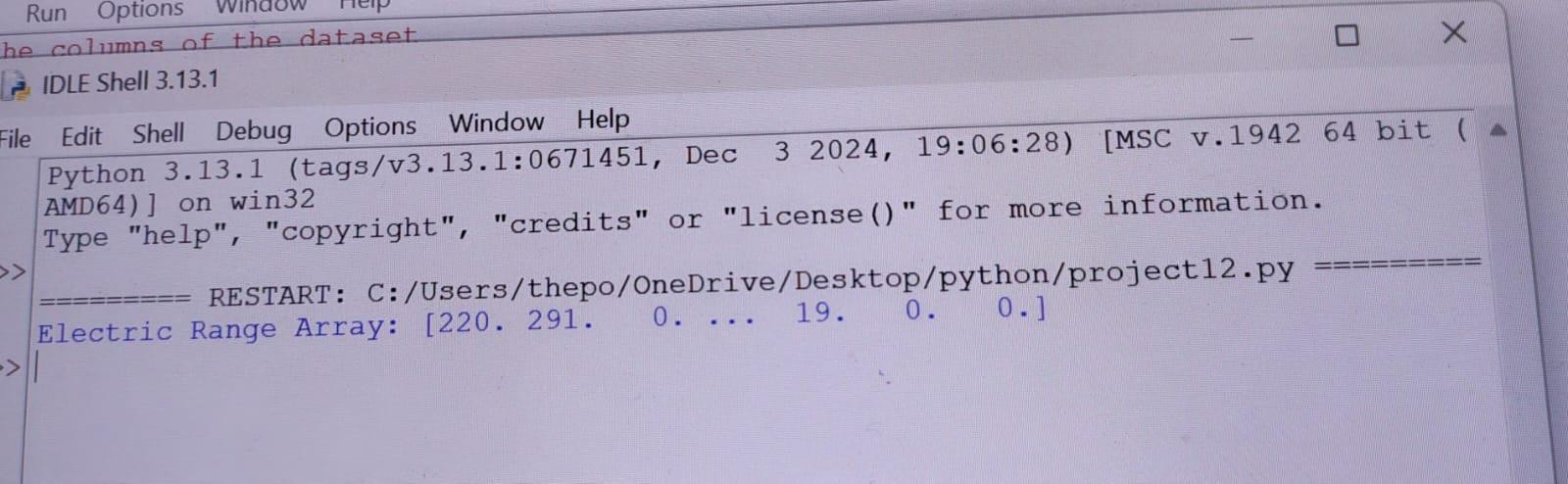
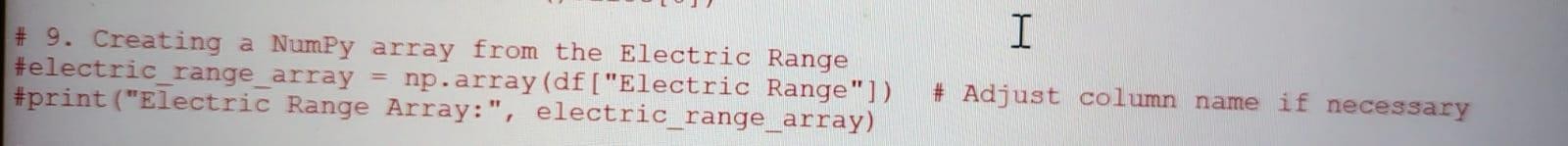
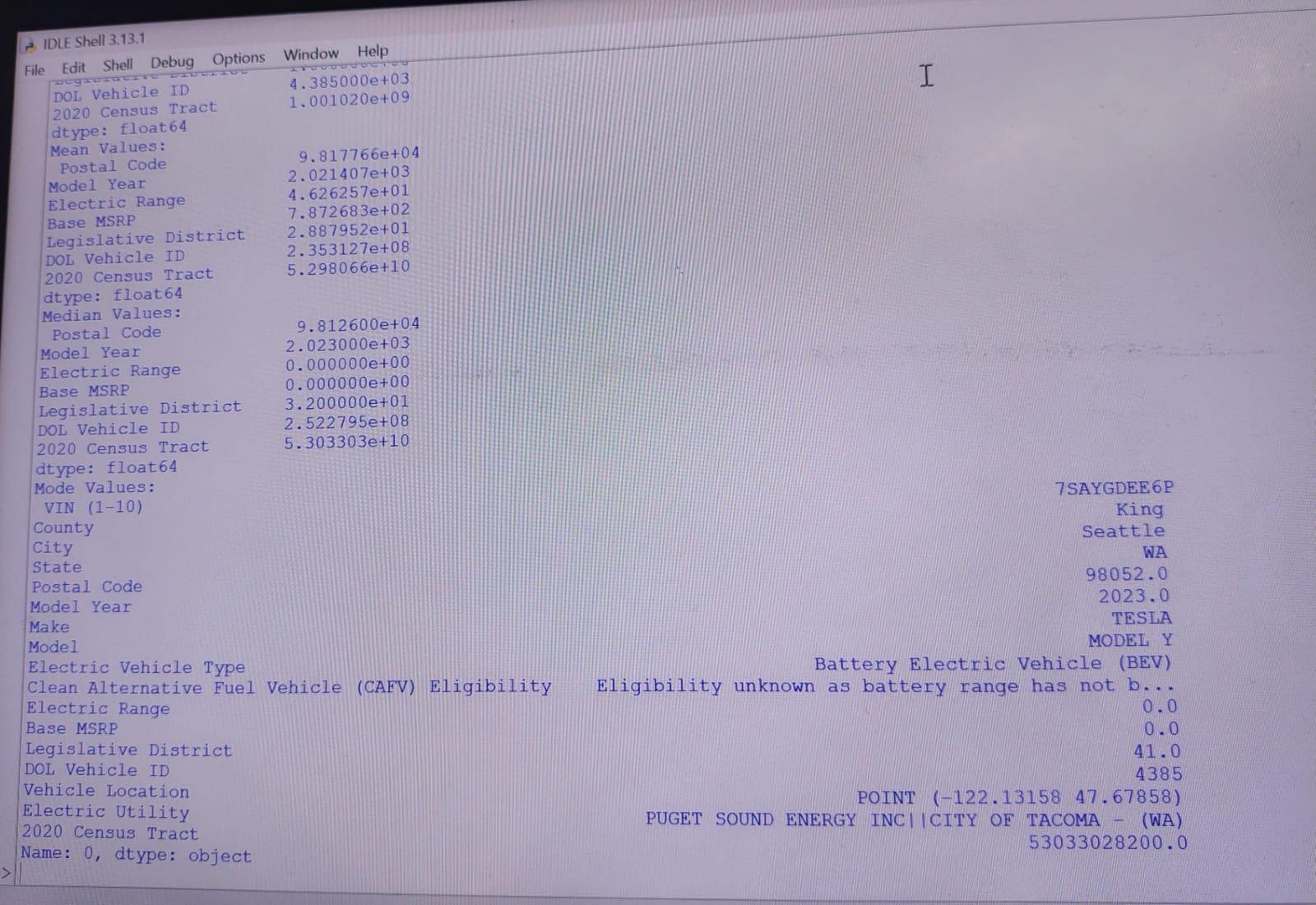
1. **Data Import and Setup**:
   * Importing the dataset using pandas (pd.read\_csv("AirQuality.csv")) and setting up libraries like NumPy, pandas, seaborn, matplotlib, and scipy.stats.
   * Suppressing future warnings for cleaner output.
2. **Data Exploration**:
   * Checking dataset dimensions (df.shape), columns (df.columns), first/last five rows (df.head(), df.tail()), and data information (df.info()).
   * Computing descriptive statistics (df.describe()) for numerical columns like PM2.5, PM10, NO2, and SO2.
   * Identifying and counting missing values (df.isnull().sum()) and cleaning the dataset by dropping rows with missing values (df.dropna()).
3. **Data Transformation**:
   * Converting the "Date" column to a datetime format (pd.to\_datetime()) and sorting the dataset by date for time-series analysis.
4. **Data Visualization**:
   * Creating visualizations like histograms, line graphs, boxplots, heatmaps, violin plots, pair plots, and KDE plots to explore pollutant distributions, trends, and correlations.

Ans – The output of the code is

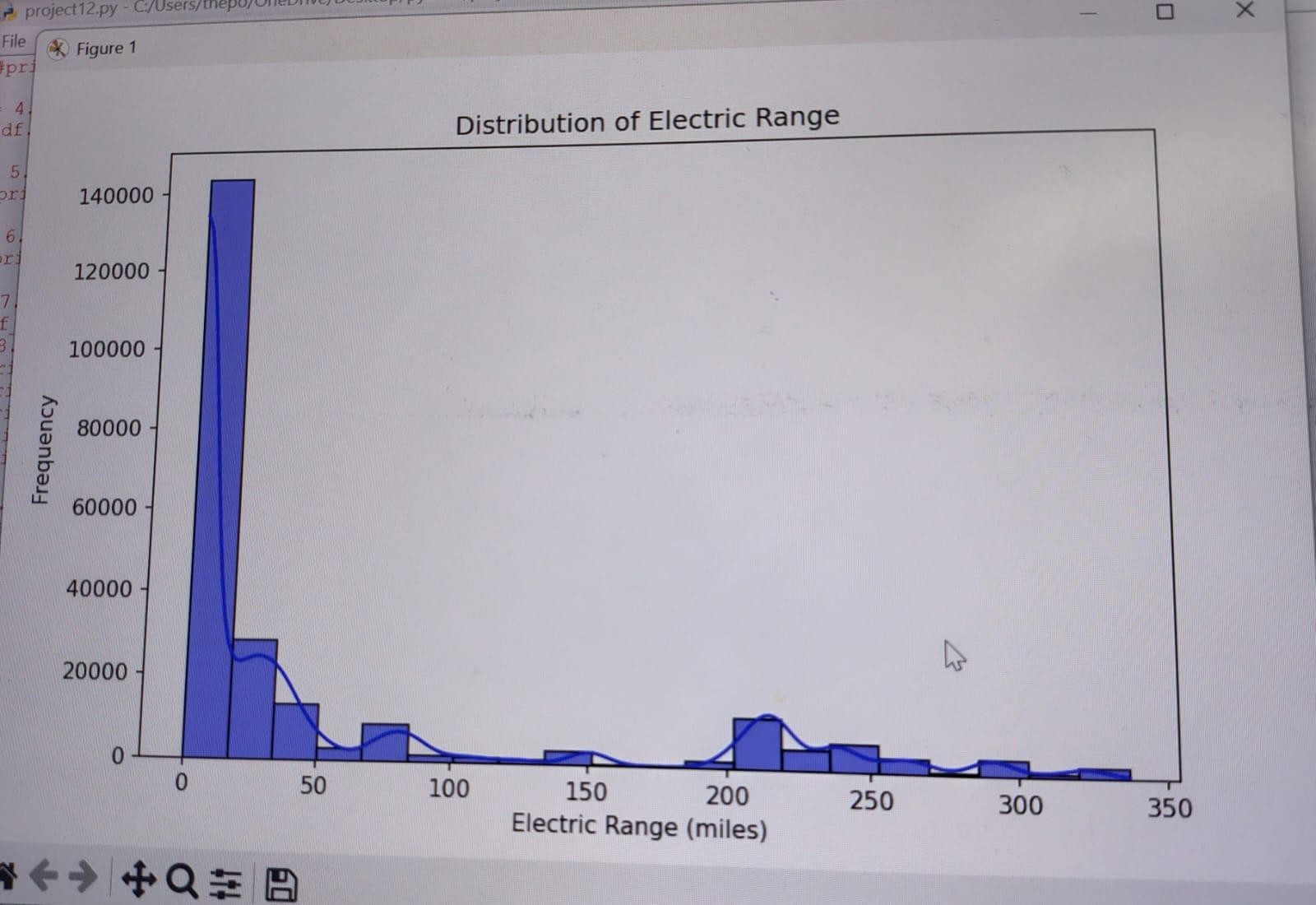
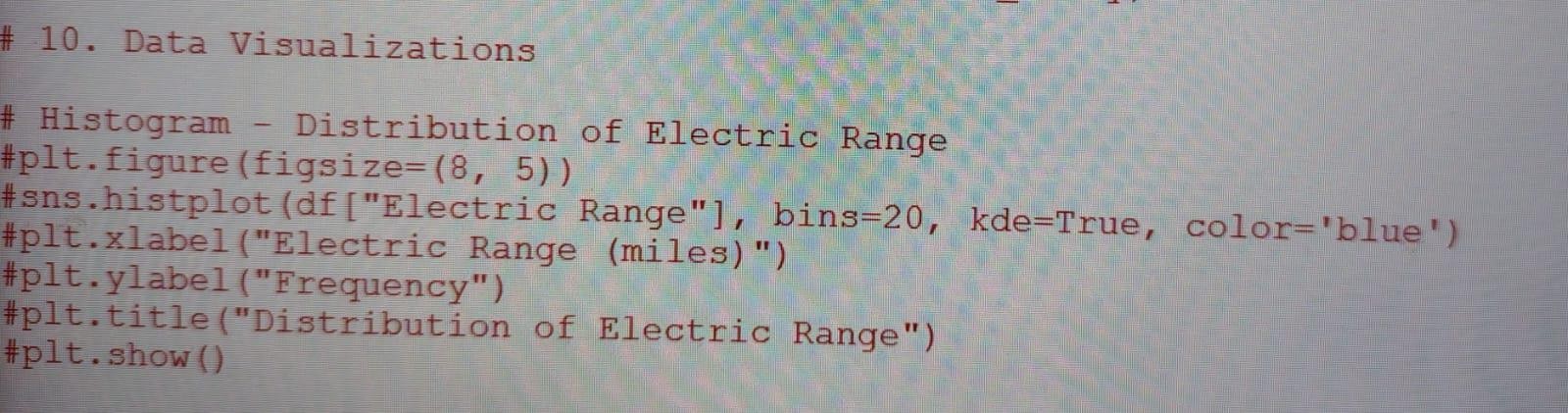


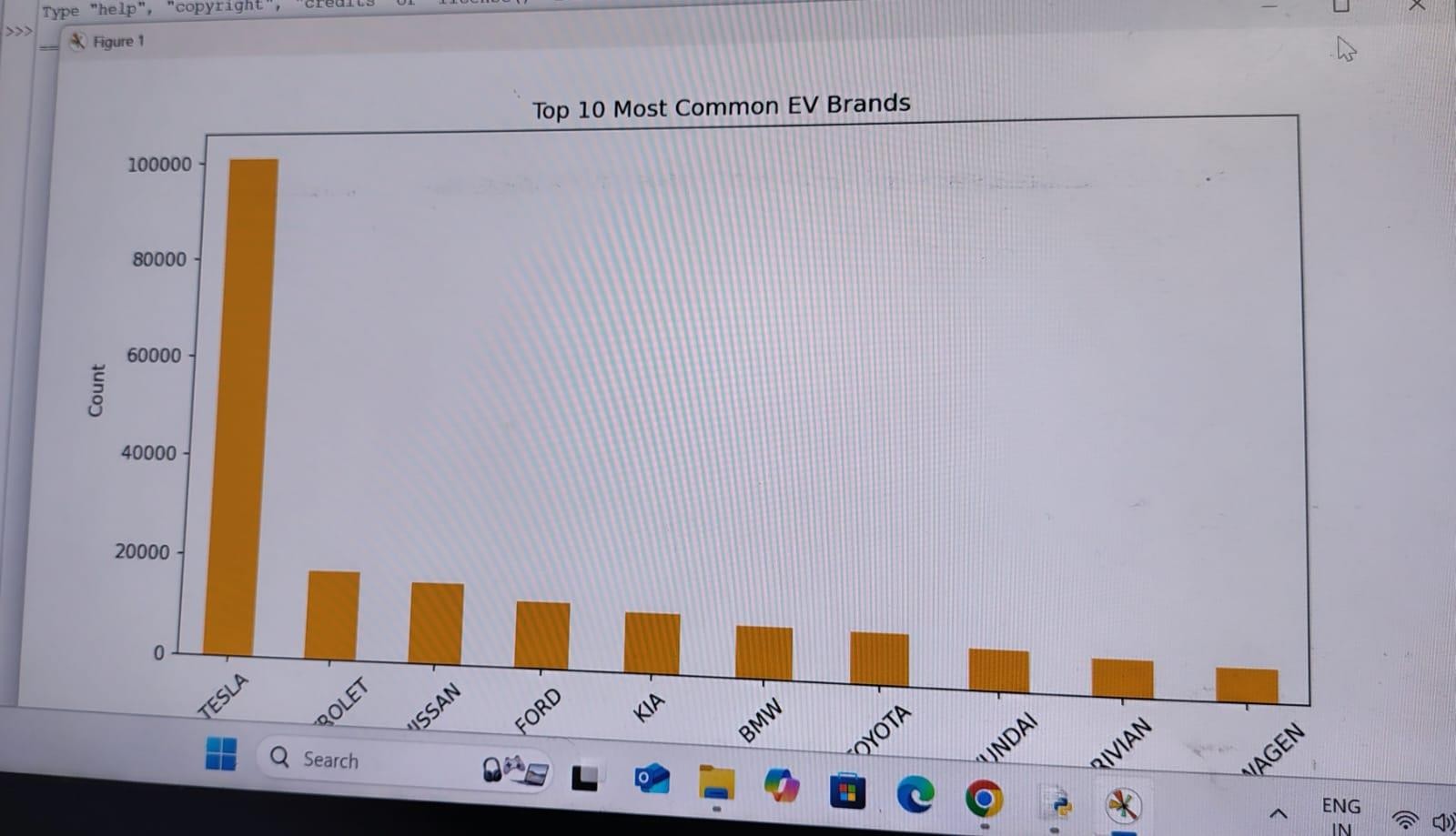






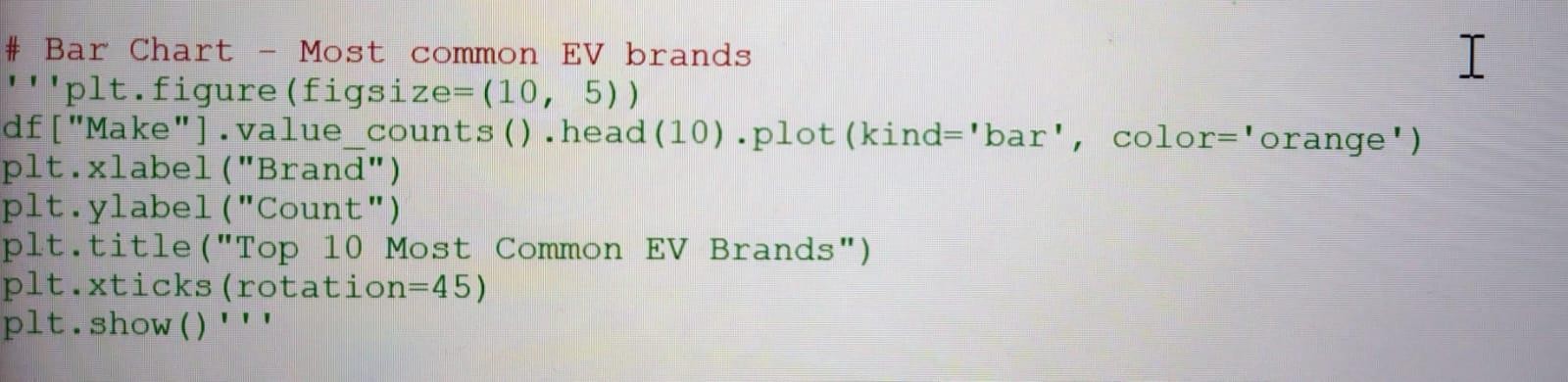
**Histogram - Distribution of PM2.5 Levels**:

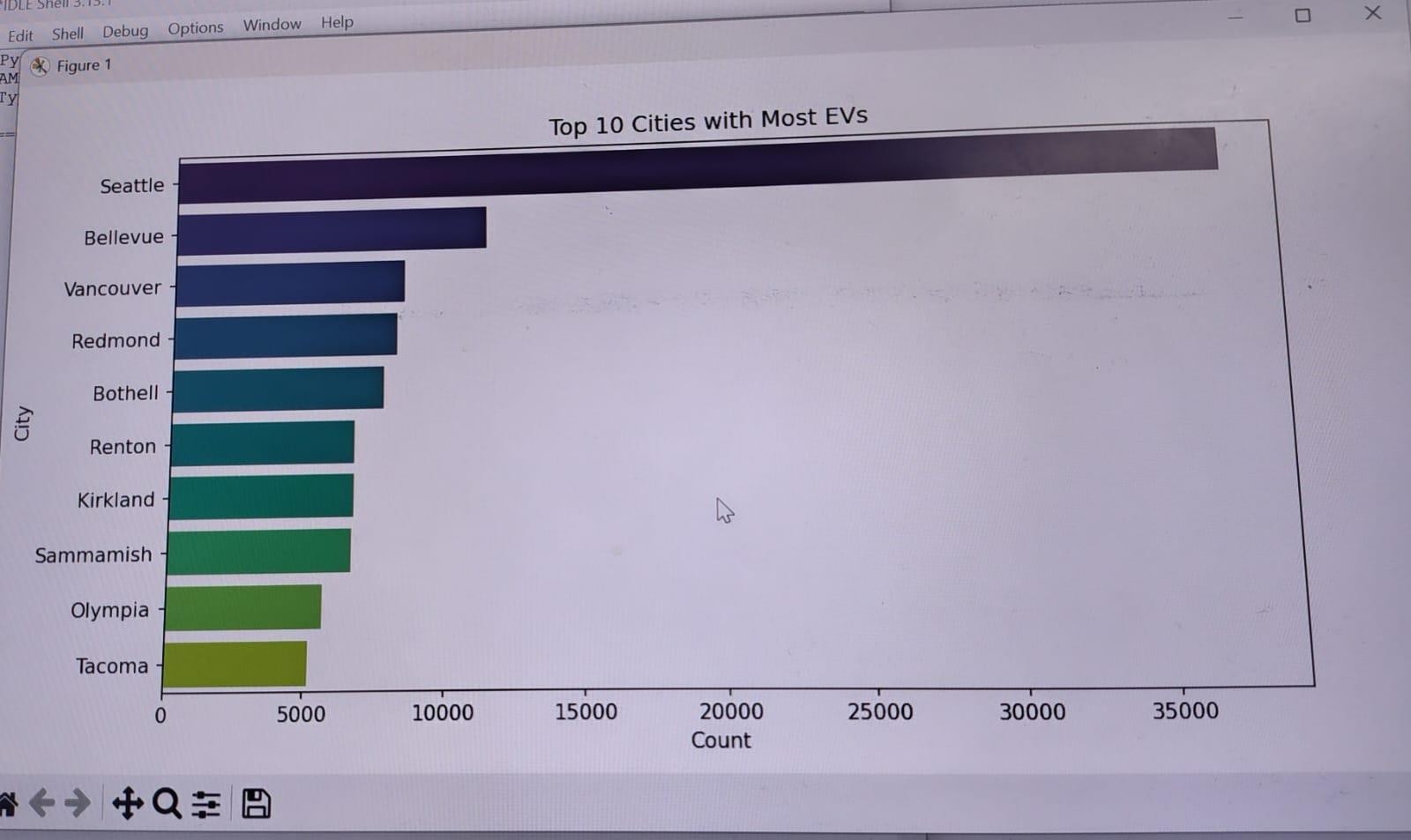
* **Code**: plt.hist(df["PM2.5"], bins=10, color="blue", edgecolor="black")
* **Description**: This histogram shows the frequency distribution of PM2.5 (fine particulate matter) levels in the dataset.
* **Insights**: It indicates how PM2.5 levels are distributed, e.g., whether they are mostly low, high, or clustered around certain values, revealing air quality patterns.



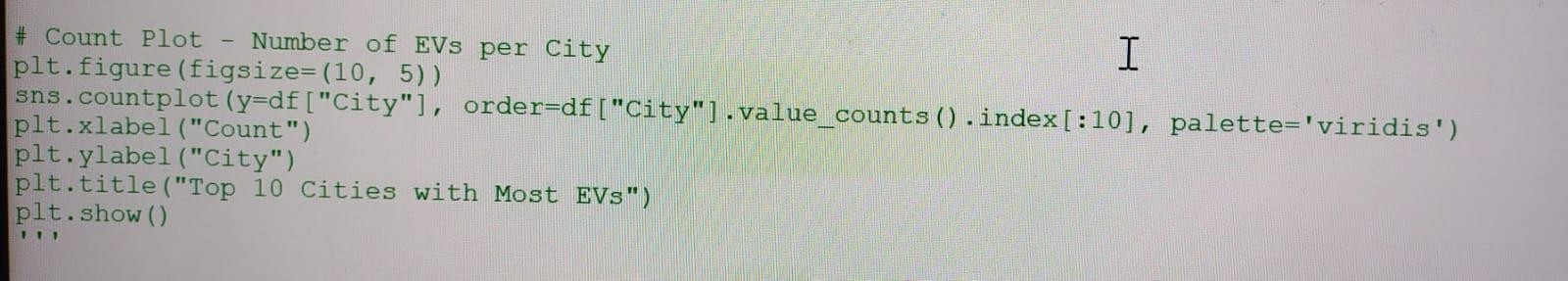
 **Bar Chart - Most Common EV Brands** (Page 7):

* **Code**: df["Make"].value\_counts().head(10).plot(kind='bar', color='orange')
* **Description**: This bar chart displays the top 10 most common EV brands based on their frequency in the dataset.
* **Insights**: It highlights which brands dominate the EV market, e.g., Tesla, Nissan, or Chevrolet, helping identify market leaders.



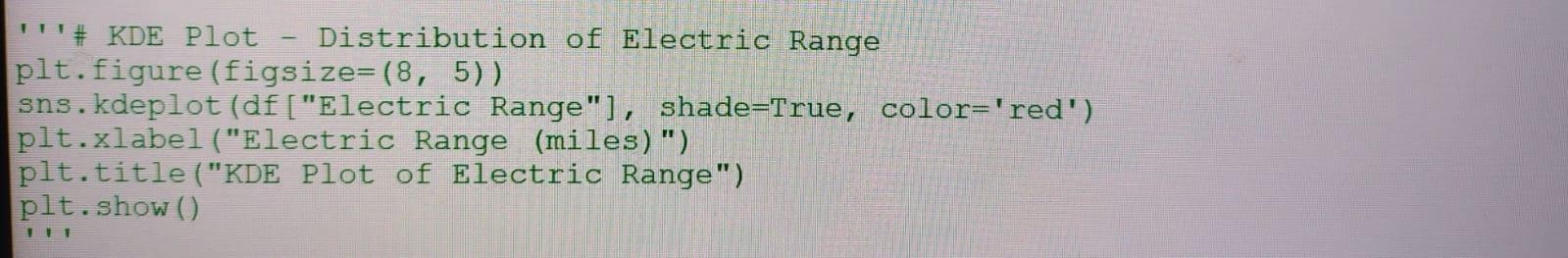
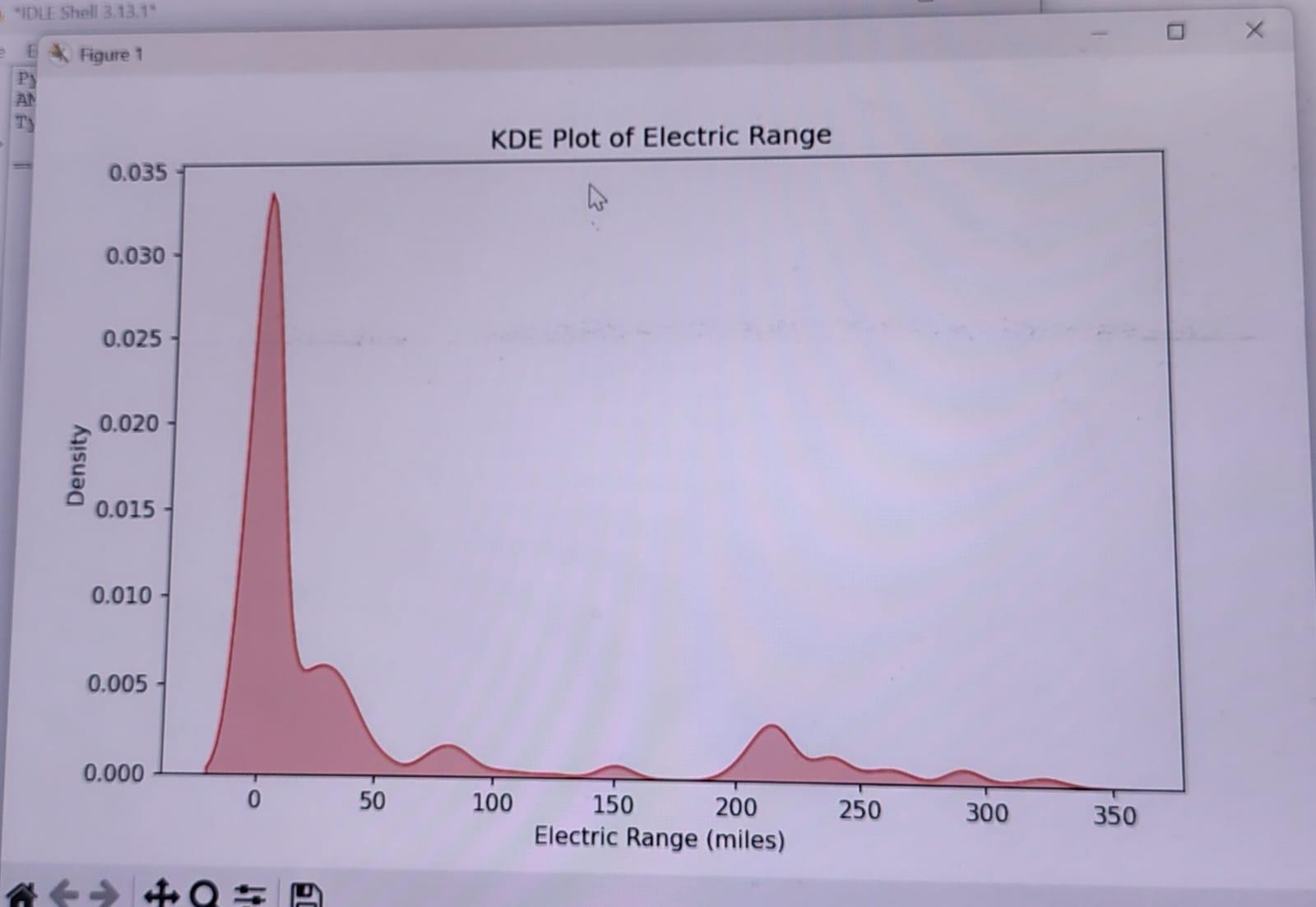
**Count Plot - Number of EVs per City** (Page 7):

* **Code**: sns.countplot(y=df["City"], order=df["City"].value\_counts().index[:10], palette='viridis')
* **Description**: This count plot shows the top 10 cities with the highest number of EVs, with bars representing the count of EVs in each city.
* **Insights**: It indicates geographic hotspots for EV adoption, useful for understanding regional demand and infrastructure needs.



**KDE Plot - Distribution of PM2.5 Levels** (Page 13):

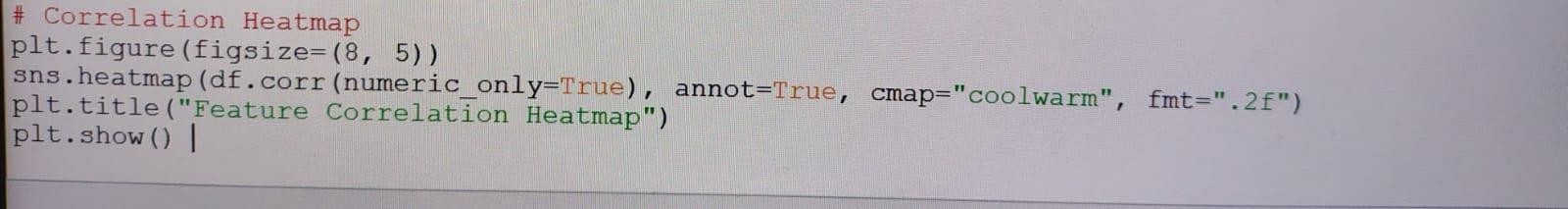
* **Code**: sns.kdeplot(df["PM2.5"], shade=True)
* **Description**: This KDE plot shows the probability density of PM2.5 levels, providing a smooth distribution curve.
* **Insights**: It highlights the likelihood of different PM2.5 levels, complementing the histogram with a continuous perspective.

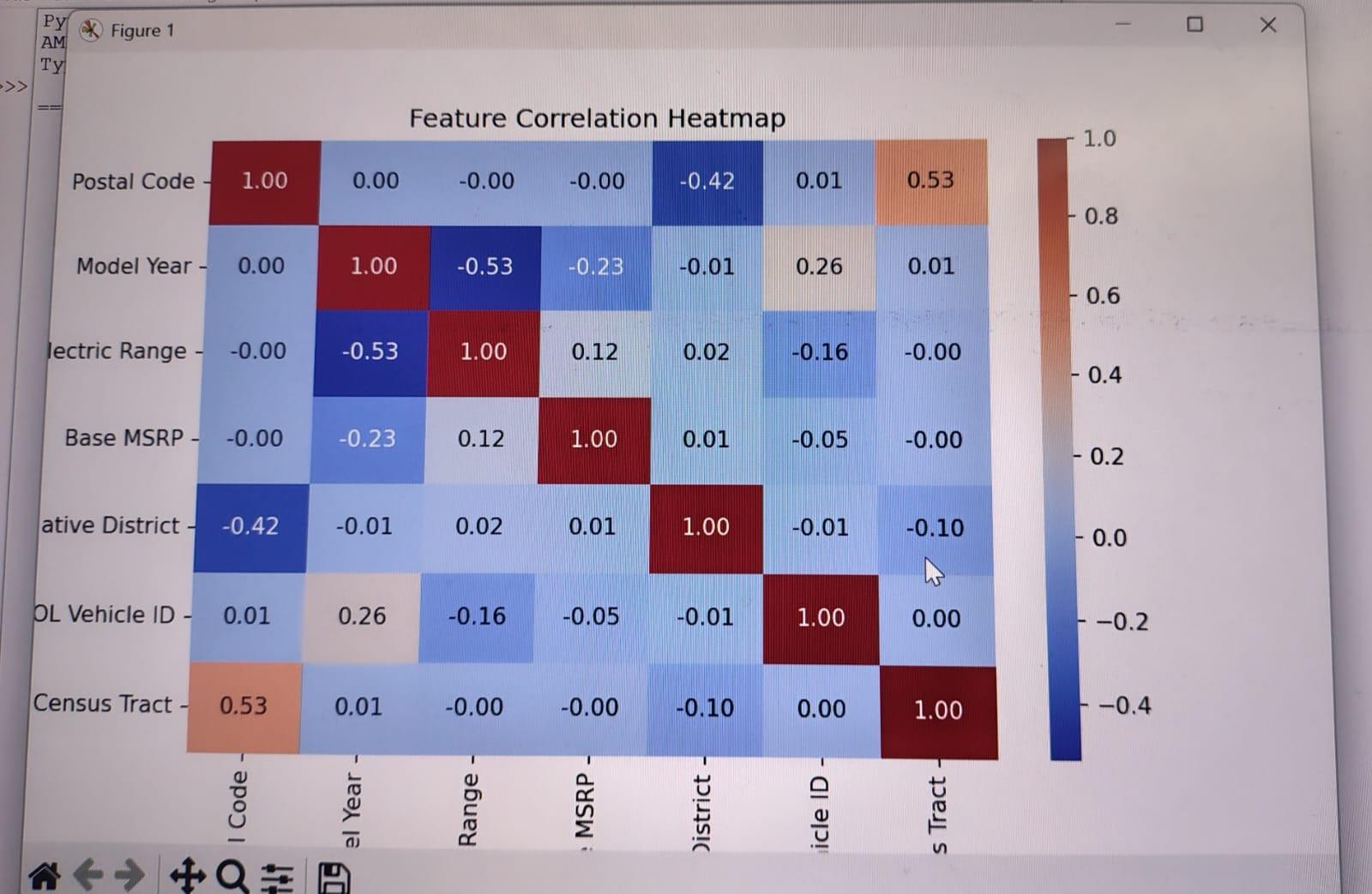


 **Correlation Heatmap** (Page 12):

* **Code**: sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
* **Description**: This heatmap visualizes correlations between air quality indicators (e.g., PM2.5, PM10, NO2, SO2), with annotated correlation coefficients.
* **Insights**: It shows relationships between pollutants, e.g., whether PM2.5 and PM10 are strongly correlated, aiding in understanding pollution sources.







import warnings

warnings.simplefilter(action="ignore", category=FutureWarning)

import numpy as np import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt import scipy.stats as st

# Importing the dataset

df = pd.read\_csv("AirQuality.csv")

# Checking the dimensions of the dataset

#print("Dataset Shape:", df.shape)

# Listing the columns of the dataset #print("Columns:", df.columns)

# Viewing the first five rows of the dataset #print(df.head())

# Viewing the last five rows of the dataset #print(df.tail())

# Viewing all the information about the dataset #print(df.info())

# Descriptive statistics of the dataset #print(df.describe())

# Checking for missing values in the dataset and their total count #print("Missing Values:")

#print(df.isnull().sum())

# Cleaning the dataset by dropping rows with missing values #df.dropna(inplace=True)

# Histogram for a selected pollutant (e.g., PM2.5)

#plt.hist(df["PM2.5"], bins=10, color="blue", edgecolor="black") #plt.xlabel("PM2.5 Levels")

#plt.ylabel("Frequency") #plt.title("Distribution of PM2.5 Levels") #plt.show()

# Line graph to show air quality trends over time (assuming 'Date' column exists) df["Date"] = pd.to\_datetime(df["Date"], errors='coerce')

df.sort\_values("Date", inplace=True) plt.plot(df["Date"], df["PM2.5"], marker='o', linestyle='-') plt.xlabel("Date")

plt.ylabel("PM2.5 Levels")

plt.title("Trend of PM2.5 Levels Over Time") plt.xticks(rotation=45)

plt.show() '''

# Boxplot for pollutant distribution by location (assuming 'Location' column exists) sns.boxplot(x="Location", y="PM2.5", data=df)

plt.title("PM2.5 Levels by Location") plt.xticks(rotation=90)

plt.show()

# Creating a heatmap to visualize the correlation between air quality indicators sns.heatmap(df.corr(), annot=True, cmap="coolwarm")

plt.title("Feature Correlation Heatmap") plt.show()

# Violin plot for PM2.5 distribution by location sns.violinplot(x="Location", y="PM2.5", data=df) plt.title("PM2.5 Levels by Location") plt.xticks(rotation=90)

plt.show()

# Pair plot for selected pollutants

selected\_columns = ["PM2.5", "PM10", "NO2", "SO2"] # Modify based on dataset columns

sns.pairplot(df[selected\_columns]) plt.show()

# KDE plot for the distribution of PM2.5 levels sns.kdeplot(df["PM2.5"], shade=True) plt.title("Distribution of PM2.5 Levels") plt.show()'''