Analyzing EV Charging Infrastructure: A Data-Driven Approach

APPLICATION (PYTHON)

Group 2

Rohan Verma

Priyansh Nileshbhai Vagadiya

617 459 9214

857 339 8070

verma.rohan@northeastern.edu

vagadiya.p@northeastern.edu

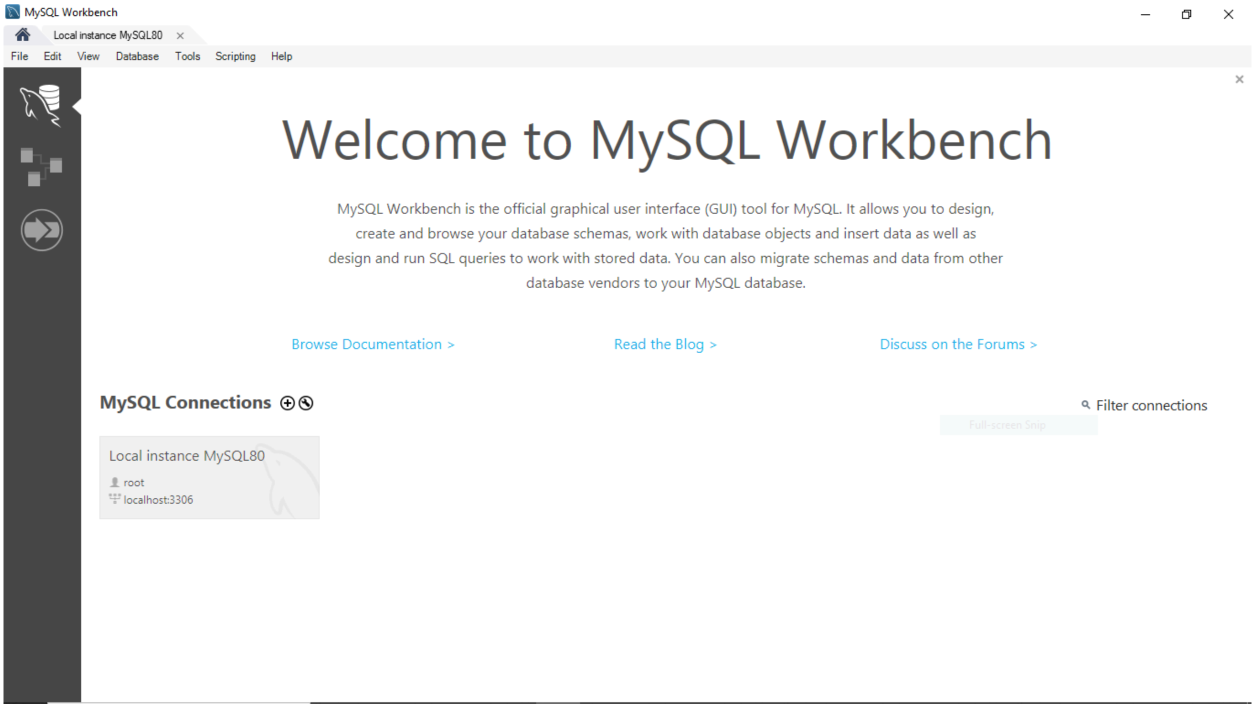
Percentage of Effort Contributed by Student1 : 50%

Percentage of Effort Contributed by Student2 : 50%

Signature of Student 1 : Rohan Verma

Signature of Student 2 : Priyansh Nileshbhai Vagadiya

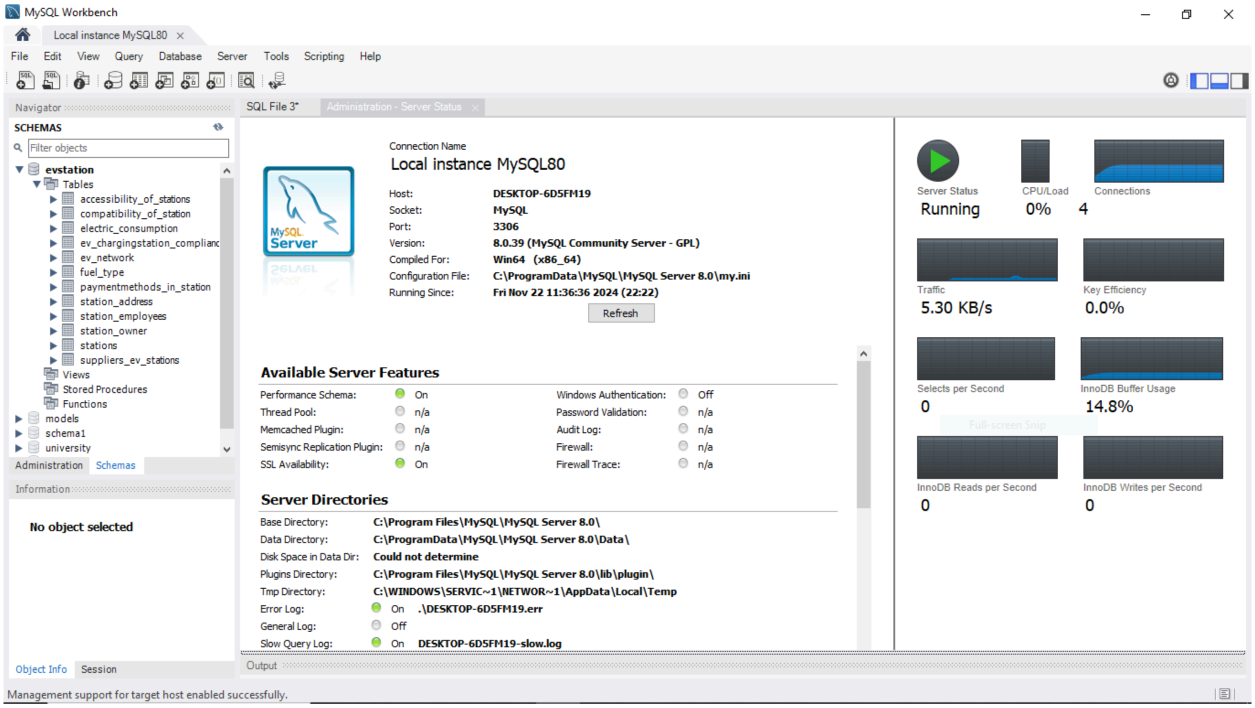
Our MySQL Workbench



Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Local instance MySQL80
root
localhost:3306



- evstation
  - Tables
    - accessibility_of_stations
    - compatibility_of_station
    - electric_consumption
    - ev_chargingstation_complianc
    - ev_network
    - fuel_type
    - paymentmethods_in_station
    - station_address
    - station_employees
    - station_owner
    - stations
    - suppliers_ev_stations
  - Views
  - Stored Procedures
  - Functions
- models
- schema1
- university

No object selected

Connection Name
Local instance MySQL80

```
Host:               DESKTOP-6D5FM19
Socket:             MySQL
Port:               3306
Version:            8.0.39 (MySQL Community Server - GPL)
Compiled For:       Win64  (x86_64)
Configuration File: C:\ProgramData\MySQL\MySQL Server 8.0\my.ini
Running Since:      Fri Nov 22 11:36:36 2024 (22:22)
```

Available Server Features

```
Performance Schema:           ● On     Windows Authentication:  ○ Off
Thread Pool:                  ○ n/a    Password Validation:     ○ n/a
Memcached Plugin:             ○ n/a    Audit Log:               ○ n/a
Semisync Replication Plugin:  ○ n/a    Firewall:                ○ n/a
SSL Availability:             ● On     Firewall Trace:          ○ n/a
```

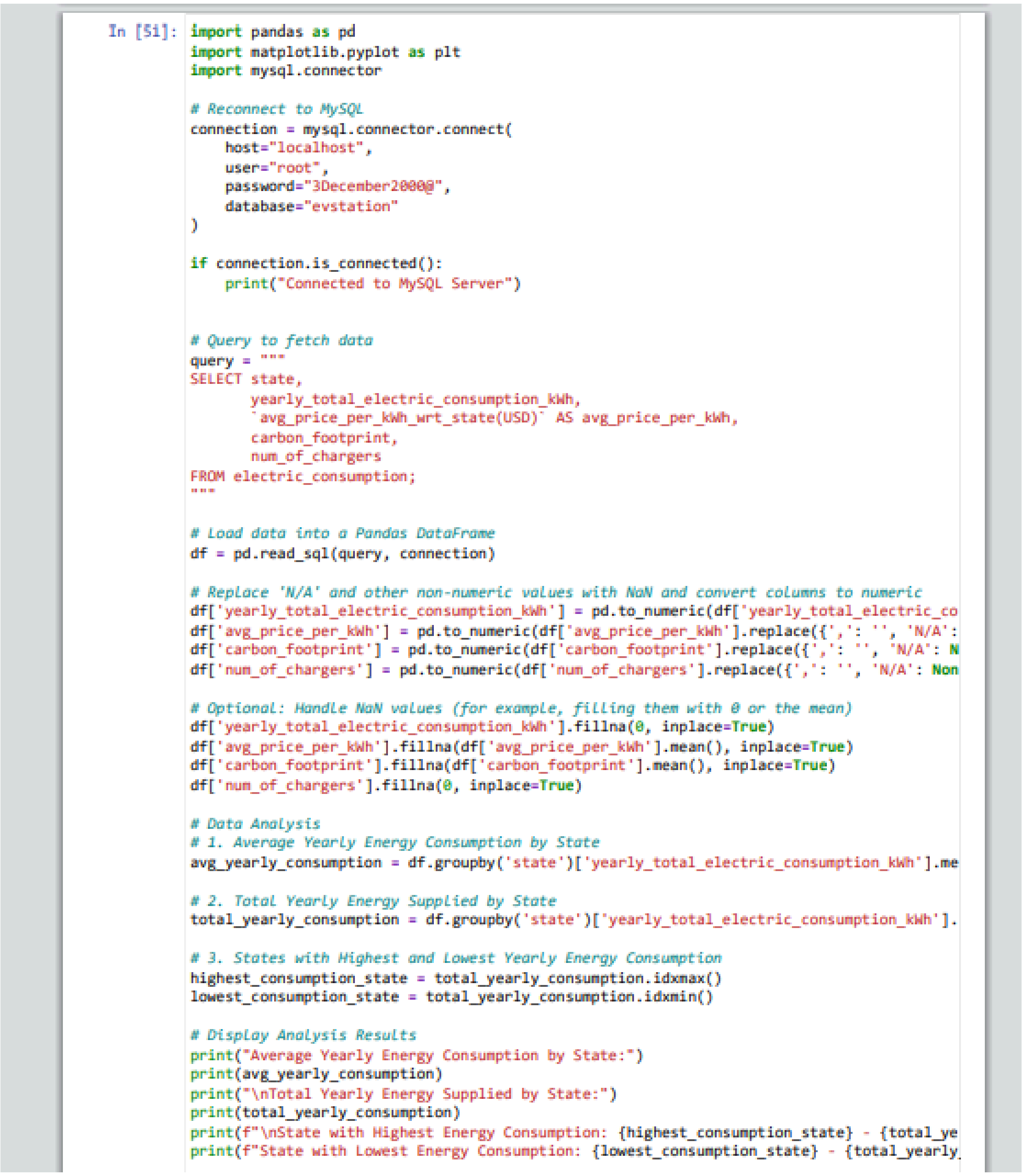
Server Directories

```
Base Directory:          C:\Program Files\MySQL\MySQL Server 8.0\
Data Directory:          C:\ProgramData\MySQL\MySQL Server 8.0\Data\
Disk Space in Data Dir:  Could not determine
Plugins Directory:       C:\Program Files\MySQL\MySQL Server 8.0\lib\plugin\
Tmp Directory:           C:\WINDOWS\SERVIC~1\NETWOR~1\AppData\Local\Temp
Error Log:               ● On   .\DESKTOP-6D5FM19.err
General Log:             ○ Off
Slow Query Log:          ● On   DESKTOP-6D5FM19-slow.log
```

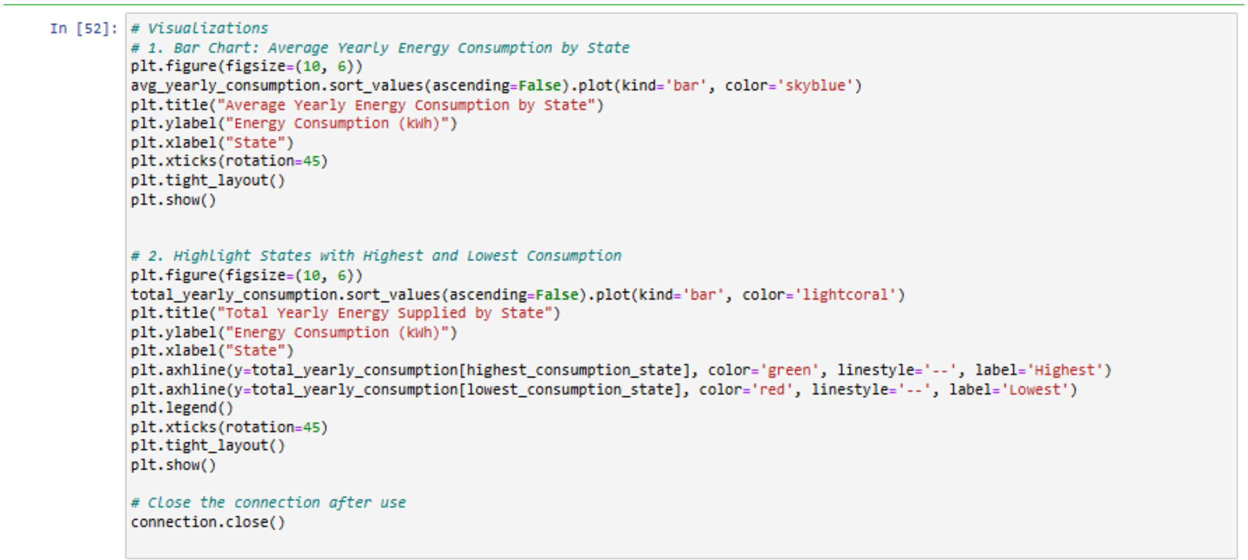
```
Server Status          CPU/Load    Connections
Running                0%          4

Traffic                Key Efficiency
5.30 KB/s              0.0%

Selects per Second     InnoDB Buffer Usage
0                      14.8%

InnoDB Reads per Second    InnoDB Writes per Second
0                          0
```

Management support for target host enabled successfully.

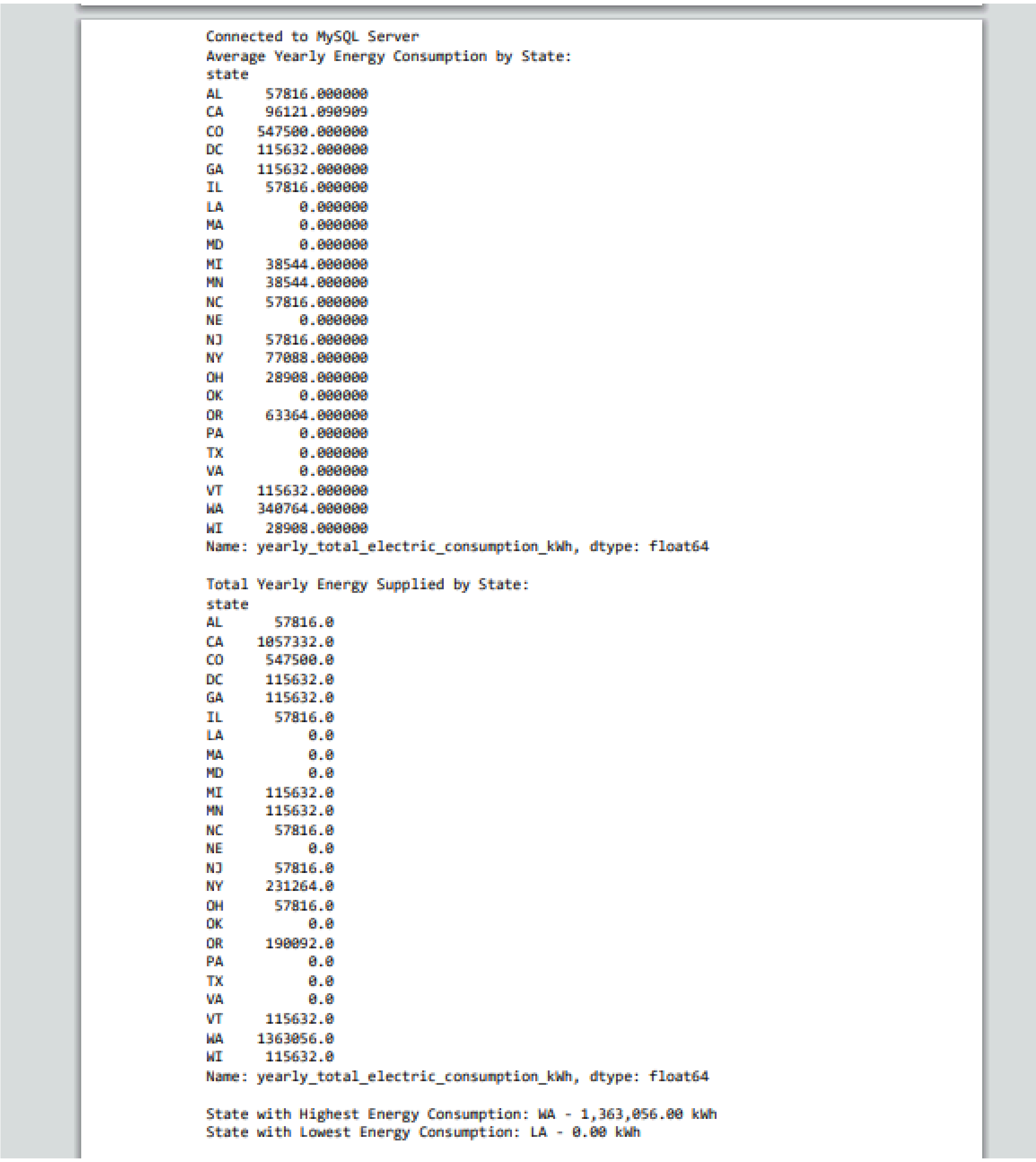
APPLICATION 1



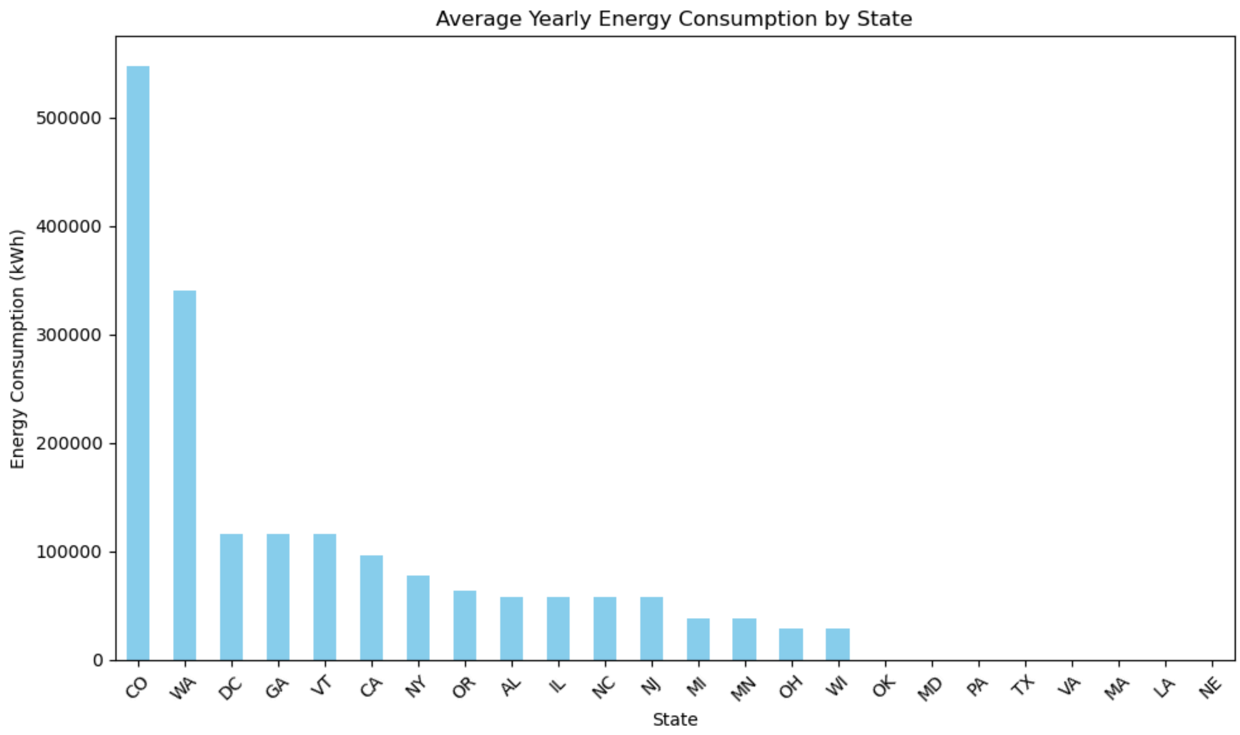
```
In [51]: import pandas as pd
         import matplotlib.pyplot as plt
         import mysql.connector

         # Reconnect to MySQL
         connection = mysql.connector.connect(
             host="localhost",
             user="root",
             password="3December2000@",
             database="evstation"
         )

         if connection.is_connected():
             print("Connected to MySQL Server")


         # Query to fetch data
         query = """
         SELECT state,
                yearly_total_electric_consumption_kWh,
                `avg_price_per_kWh_wrt_state(USD)` AS avg_price_per_kWh,
                carbon_footprint,
                num_of_chargers
         FROM electric_consumption;
         """

         # Load data into a Pandas DataFrame
         df = pd.read_sql(query, connection)

         # Replace 'N/A' and other non-numeric values with NaN and convert columns to numeric
         df['yearly_total_electric_consumption_kWh'] = pd.to_numeric(df['yearly_total_electric_co
         df['avg_price_per_kWh'] = pd.to_numeric(df['avg_price_per_kWh'].replace({',': '', 'N/A':
         df['carbon_footprint'] = pd.to_numeric(df['carbon_footprint'].replace({',': '', 'N/A': N
         df['num_of_chargers'] = pd.to_numeric(df['num_of_chargers'].replace({',': '', 'N/A': Non

         # Optional: Handle NaN values (for example, filling them with 0 or the mean)
         df['yearly_total_electric_consumption_kWh'].fillna(0, inplace=True)
         df['avg_price_per_kWh'].fillna(df['avg_price_per_kWh'].mean(), inplace=True)
         df['carbon_footprint'].fillna(df['carbon_footprint'].mean(), inplace=True)
         df['num_of_chargers'].fillna(0, inplace=True)

         # Data Analysis
         # 1. Average Yearly Energy Consumption by State
         avg_yearly_consumption = df.groupby('state')['yearly_total_electric_consumption_kWh'].me

         # 2. Total Yearly Energy Supplied by State
         total_yearly_consumption = df.groupby('state')['yearly_total_electric_consumption_kWh'].

         # 3. States with Highest and Lowest Yearly Energy Consumption
         highest_consumption_state = total_yearly_consumption.idxmax()
         lowest_consumption_state = total_yearly_consumption.idxmin()

         # Display Analysis Results
         print("Average Yearly Energy Consumption by State:")
         print(avg_yearly_consumption)
         print("\nTotal Yearly Energy Supplied by State:")
         print(total_yearly_consumption)
         print(f"\nState with Highest Energy Consumption: {highest_consumption_state} - {total_ye
         print(f"State with Lowest Energy Consumption: {lowest_consumption_state} - {total_yearly_
```

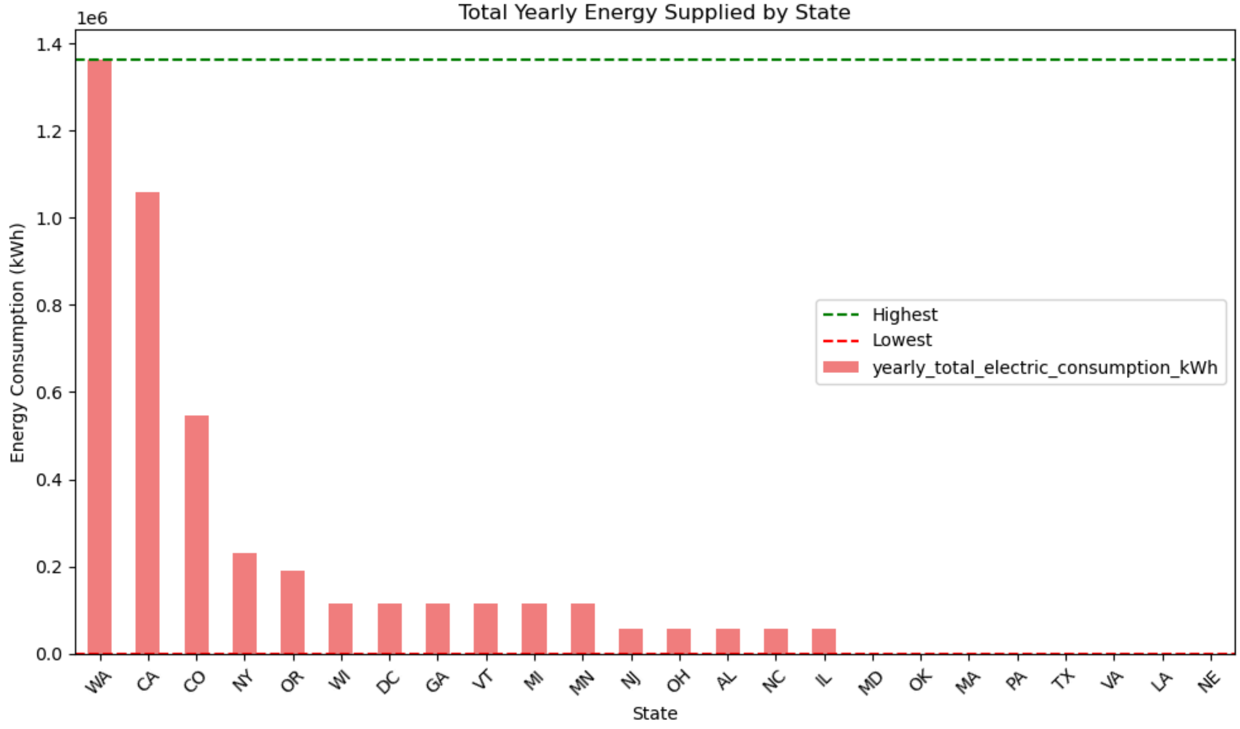


```
In [52]: # Visualizations
         # 1. Bar Chart: Average Yearly Energy Consumption by State
         plt.figure(figsize=(10, 6))
         avg_yearly_consumption.sort_values(ascending=False).plot(kind='bar', color='skyblue')
         plt.title("Average Yearly Energy Consumption by State")
         plt.ylabel("Energy Consumption (kWh)")
         plt.xlabel("State")
         plt.xticks(rotation=45)
         plt.tight_layout()
         plt.show()


         # 2. Highlight States with Highest and Lowest Consumption
         plt.figure(figsize=(10, 6))
         total_yearly_consumption.sort_values(ascending=False).plot(kind='bar', color='lightcoral')
         plt.title("Total Yearly Energy Supplied by State")
         plt.ylabel("Energy Consumption (kWh)")
         plt.xlabel("State")
         plt.axhline(y=total_yearly_consumption[highest_consumption_state], color='green', linestyle='--', label='Highest')
         plt.axhline(y=total_yearly_consumption[lowest_consumption_state], color='red', linestyle='--', label='Lowest')
         plt.legend()
         plt.xticks(rotation=45)
         plt.tight_layout()
         plt.show()

         # Close the connection after use
         connection.close()
```

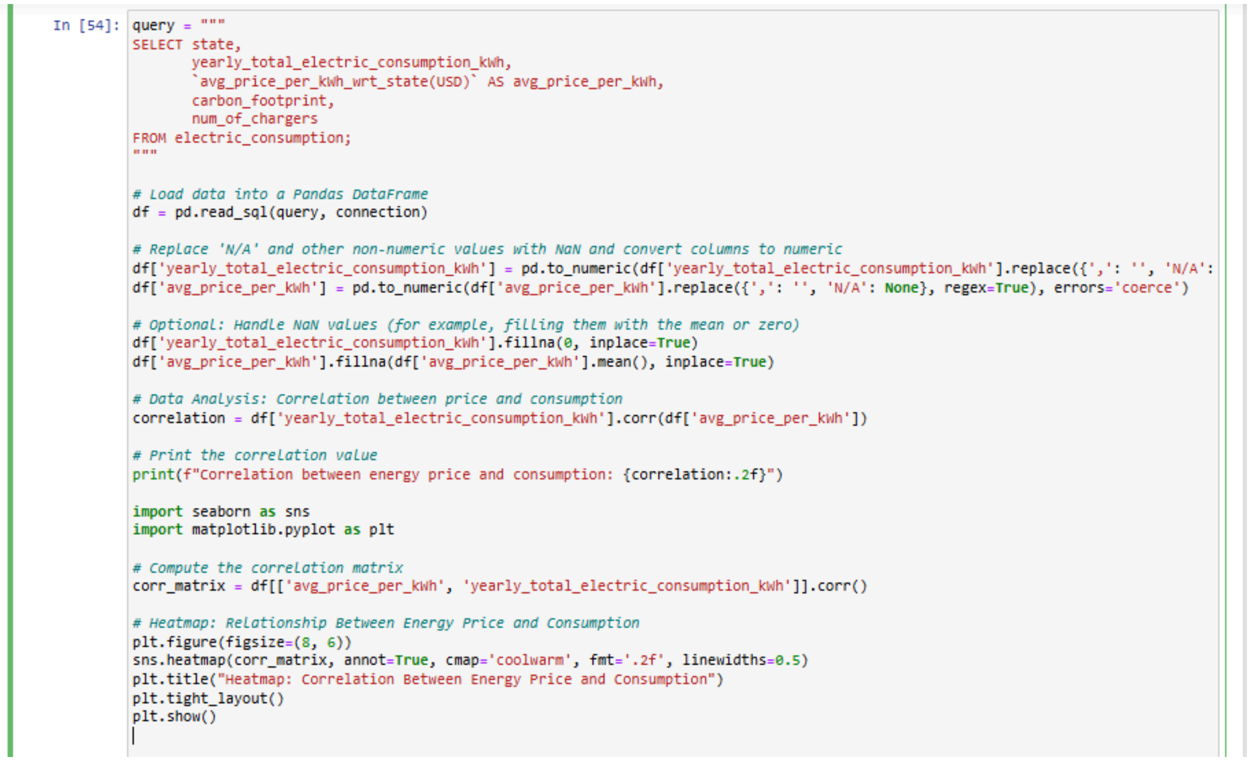


```
Connected to MySQL Server
Average Yearly Energy Consumption by State:
state
AL     57816.000000
CA     96121.090909
CO    547500.000000
DC    115632.000000
GA    115632.000000
IL     57816.000000
LA         0.000000
MA         0.000000
MD         0.000000
MI     38544.000000
MN     38544.000000
NC     57816.000000
NE         0.000000
NJ     57816.000000
NY     77088.000000
OH     28908.000000
OK         0.000000
OR     63364.000000
PA         0.000000
TX         0.000000
VA         0.000000
VT    115632.000000
WA    340764.000000
WI     28908.000000
Name: yearly_total_electric_consumption_kWh, dtype: float64

Total Yearly Energy Supplied by State:
state
AL      57816.0
CA    1057332.0
CO     547500.0
DC     115632.0
GA     115632.0
IL      57816.0
LA          0.0
MA          0.0
MD          0.0
MI     115632.0
MN     115632.0
NC      57816.0
NE          0.0
NJ      57816.0
NY     231264.0
OH      57816.0
OK          0.0
OR     190092.0
PA          0.0
TX          0.0
VA          0.0
VT     115632.0
WA    1363056.0
WI     115632.0
Name: yearly_total_electric_consumption_kWh, dtype: float64

State with Highest Energy Consumption: WA - 1,363,056.00 kWh
State with Lowest Energy Consumption: LA - 0.00 kWh
```


Average Yearly Energy Consumption by State
Energy Consumption (kWh)
500000
400000
300000
200000
100000
0
CO
WA
DC
GA
VT
CA
NY
OR
AL
IL
NC
NJ
MI
MN
OH
WI
OK
MD
PA
TX
VA
MA
LA
NE
State

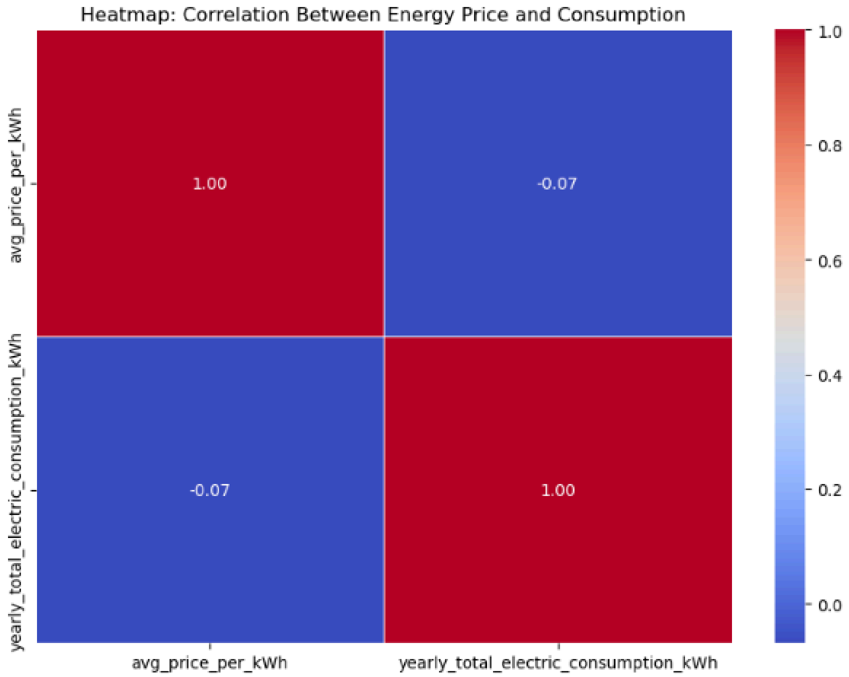

Total Yearly Energy Supplied by State
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Energy Consumption (kWh)
Highest
Lowest
yearly_total_electric_consumption_kWh
WA
CA
CO
NY
OR
WI
DC
GA
VT
MI
MN
NJ
OH
AL
NC
IL
MD
OK
MA
PA
TX
VA
LA
NE
State

APPLICATION 2

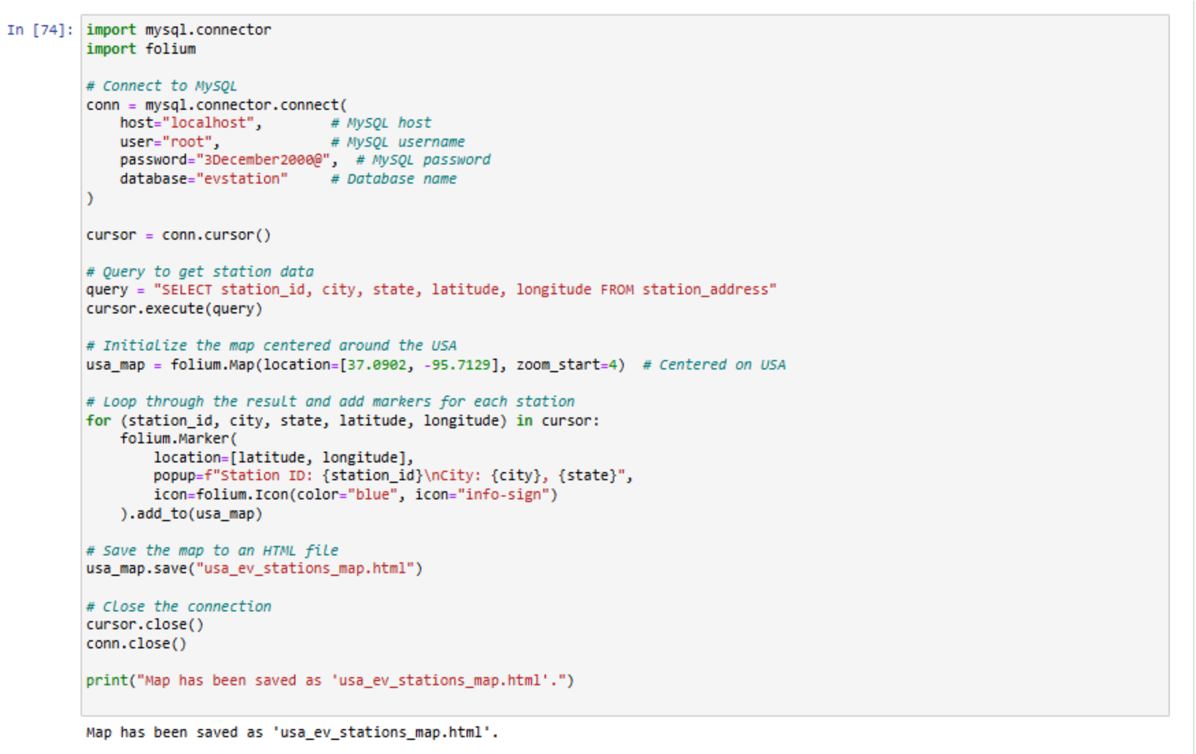


```
In [54]: query = """
SELECT state,
       yearly_total_electric_consumption_kWh,
       `avg_price_per_kWh_wrt_state(USD)` AS avg_price_per_kWh,
       carbon_footprint,
       num_of_chargers
FROM electric_consumption;
"""

# Load data into a Pandas DataFrame
df = pd.read_sql(query, connection)

# Replace 'N/A' and other non-numeric values with NaN and convert columns to numeric
df['yearly_total_electric_consumption_kWh'] = pd.to_numeric(df['yearly_total_electric_consumption_kWh'].replace({',': '', 'N/A':
df['avg_price_per_kWh'] = pd.to_numeric(df['avg_price_per_kWh'].replace({',': '', 'N/A': None}, regex=True), errors='coerce')

# Optional: Handle NaN values (for example, filling them with the mean or zero)
df['yearly_total_electric_consumption_kWh'].fillna(0, inplace=True)
df['avg_price_per_kWh'].fillna(df['avg_price_per_kWh'].mean(), inplace=True)

# Data Analysis: Correlation between price and consumption
correlation = df['yearly_total_electric_consumption_kWh'].corr(df['avg_price_per_kWh'])

# Print the correlation value
print(f"Correlation between energy price and consumption: {correlation:.2f}")

import seaborn as sns
import matplotlib.pyplot as plt

# Compute the correlation matrix
corr_matrix = df[['avg_price_per_kWh', 'yearly_total_electric_consumption_kWh']].corr()

# Heatmap: Relationship Between Energy Price and Consumption
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title("Heatmap: Correlation Between Energy Price and Consumption")
plt.tight_layout()
plt.show()
```

```
Correlation between energy price and consumption: -0.07
```



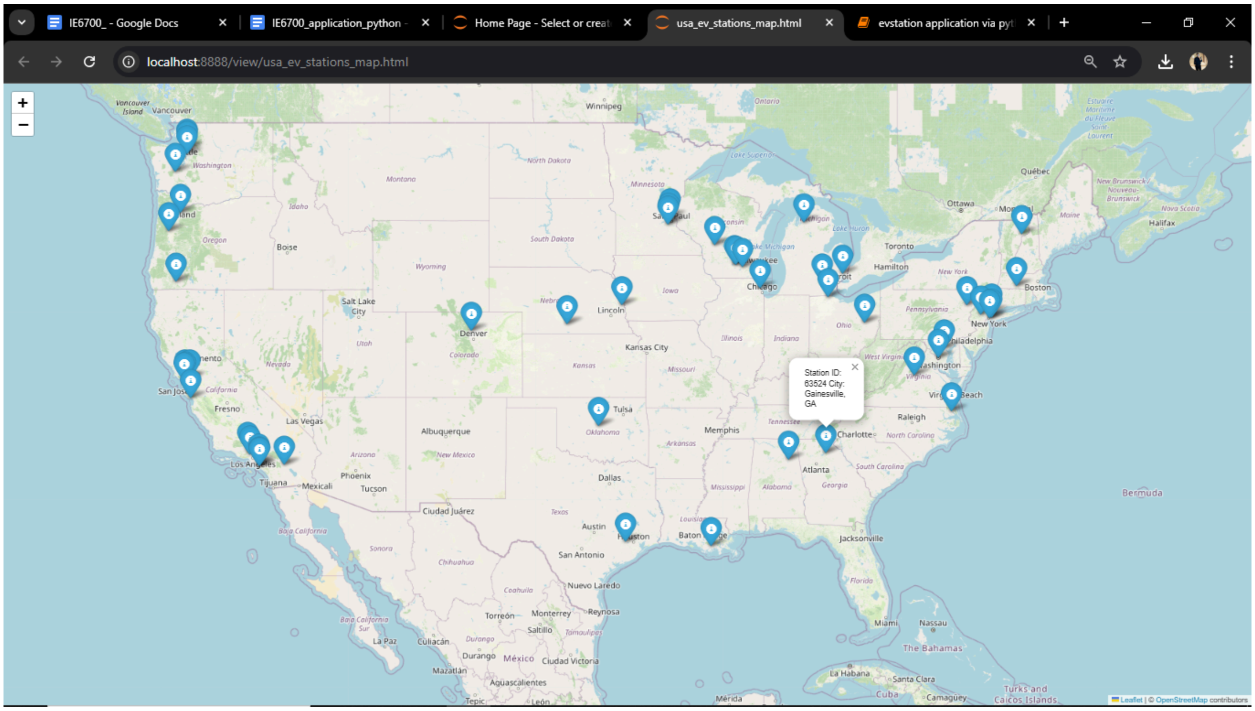
APPLICATION 3



In [74]:

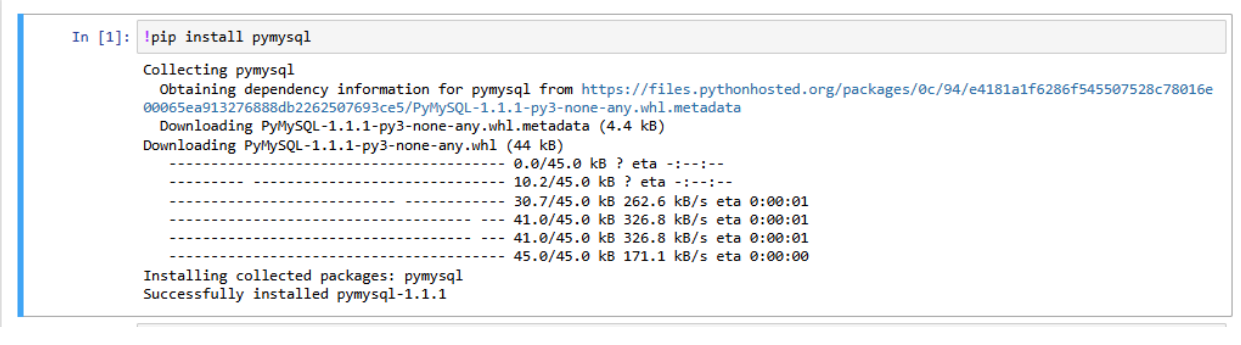
```python
import mysql.connector
import folium

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",        # MySQL host
    user="root",             # MySQL username
    password="3December2000@",  # MySQL password
    database="evstation"     # Database name
)

cursor = conn.cursor()

# Query to get station data
query = "SELECT station_id, city, state, latitude, longitude FROM station_address"
cursor.execute(query)

# Initialize the map centered around the USA
usa_map = folium.Map(location=[37.0902, -95.7129], zoom_start=4)  # Centered on USA

# Loop through the result and add markers for each station
for (station_id, city, state, latitude, longitude) in cursor:
    folium.Marker(
        location=[latitude, longitude],
        popup=f"Station ID: {station_id}\nCity: {city}, {state}",
        icon=folium.Icon(color="blue", icon="info-sign")
    ).add_to(usa_map)

# Save the map to an HTML file
usa_map.save("usa_ev_stations_map.html")

# Close the connection
cursor.close()
conn.close()

print("Map has been saved as 'usa_ev_stations_map.html'.")
```

```
Map has been saved as 'usa_ev_stations_map.html'.
```

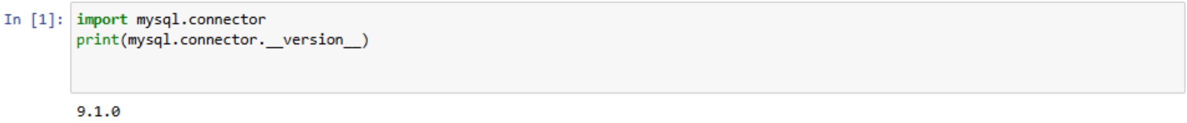


Station ID: 63524 City: Gainesville, GA

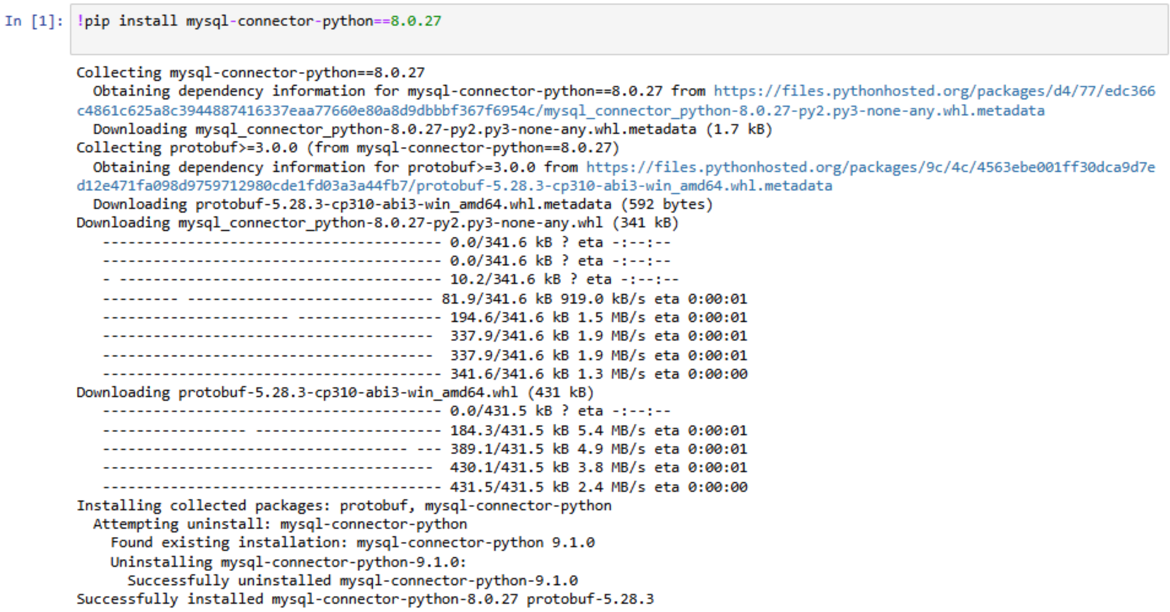
ALL PYTHON LIBRARIES INSTALLED



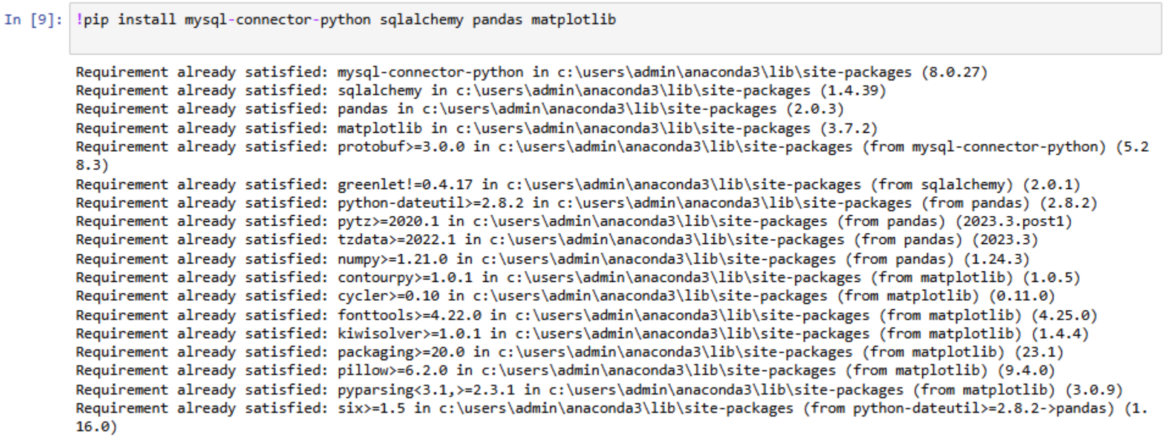
```
In [1]: !pip install pymysql

        Collecting pymysql
          Obtaining dependency information for pymysql from https://files.pythonhosted.org/packages/0c/94/e4181a1f6286f545507528c78016e
        00065ea913276888db2262507693ce5/PyMySQL-1.1.1-py3-none-any.whl.metadata
          Downloading PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
        Downloading PyMySQL-1.1.1-py3-none-any.whl (44 kB)
           ---------------------------------------- 0.0/45.0 kB ? eta -:--:--
           --------- ------------------------------ 10.2/45.0 kB ? eta -:--:--
           -------------------------- ------------- 30.7/45.0 kB 262.6 kB/s eta 0:00:01
           ------------------------------------ --- 41.0/45.0 kB 326.8 kB/s eta 0:00:01
           ------------------------------------ --- 41.0/45.0 kB 326.8 kB/s eta 0:00:01
           ---------------------------------------- 45.0/45.0 kB 171.1 kB/s eta 0:00:00
        Installing collected packages: pymysql
        Successfully installed pymysql-1.1.1
```



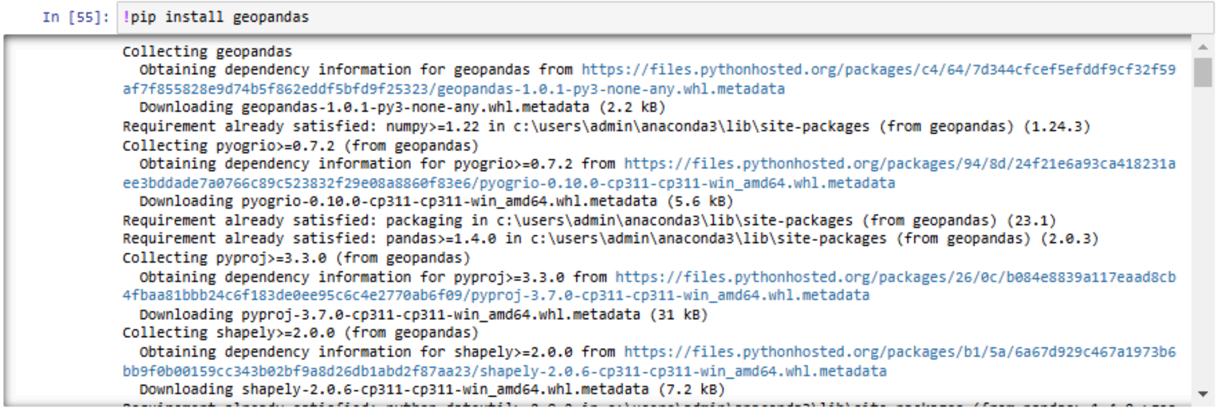
```
In [1]: import mysql.connector
        print(mysql.connector.__version__)

        9.1.0
```



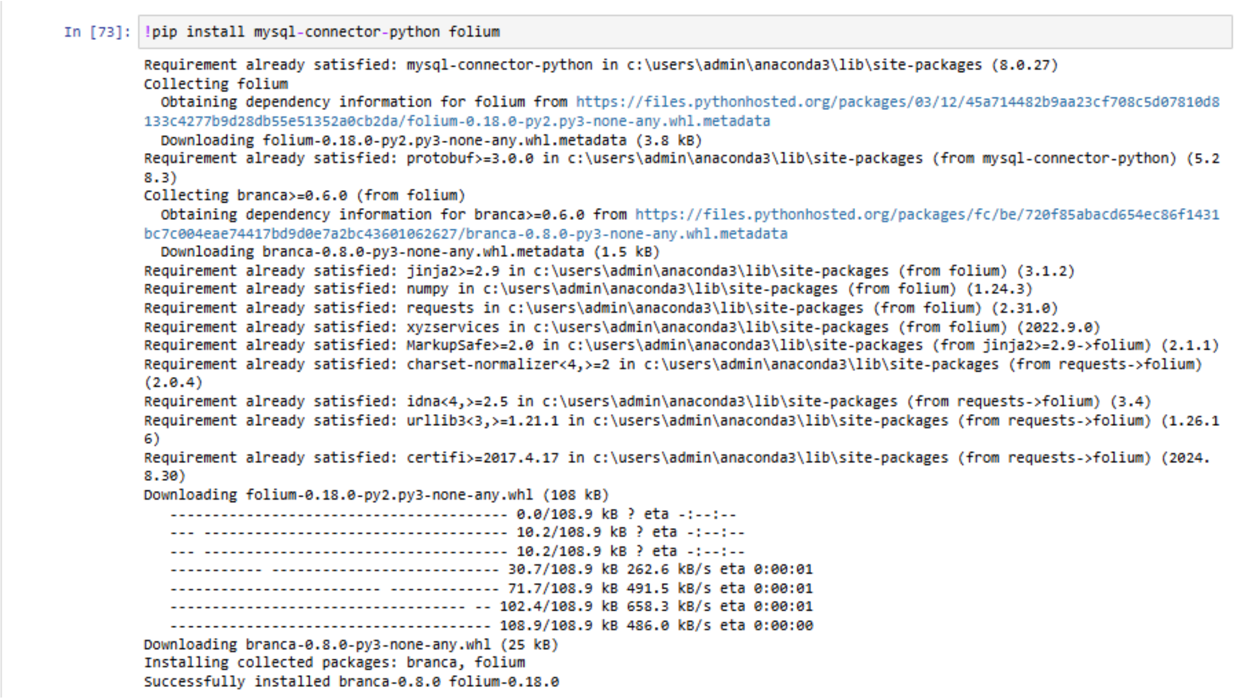
```
In [1]: !pip install mysql-connector-python==8.0.27

        Collecting mysql-connector-python==8.0.27
          Obtaining dependency information for mysql-connector-python==8.0.27 from https://files.pythonhosted.org/packages/d4/77/edc366
        c4861c625a8c3944887416337eaa77660e80a8d9dbbbf367f6954c/mysql_connector_python-8.0.27-py2.py3-none-any.whl.metadata
          Downloading mysql_connector_python-8.0.27-py2.py3-none-any.whl.metadata (1.7 kB)
        Collecting protobuf>=3.0.0 (from mysql-connector-python==8.0.27)
          Obtaining dependency information for protobuf>=3.0.0 from https://files.pythonhosted.org/packages/9c/4c/4563ebe001ff30dca9d7e
        d12e471fa098d9759712980cde1fd03a3a44fb7/protobuf-5.28.3-cp310-abi3-win_amd64.whl.metadata
          Downloading protobuf-5.28.3-cp310-abi3-win_amd64.whl.metadata (592 bytes)
        Downloading mysql_connector_python-8.0.27-py2.py3-none-any.whl (341 kB)
           ---------------------------------------- 0.0/341.6 kB ? eta -:--:--
           ---------------------------------------- 0.0/341.6 kB ? eta -:--:--
           - -------------------------------------- 10.2/341.6 kB ? eta -:--:--
           --------- ------------------------------ 81.9/341.6 kB 919.0 kB/s eta 0:00:01
           ---------------------- ----------------- 194.6/341.6 kB 1.5 MB/s eta 0:00:01
           ---------------------------------------  337.9/341.6 kB 1.9 MB/s eta 0:00:01
           ---------------------------------------  337.9/341.6 kB 1.9 MB/s eta 0:00:01
           ---------------------------------------- 341.6/341.6 kB 1.3 MB/s eta 0:00:00
        Downloading protobuf-5.28.3-cp310-abi3-win_amd64.whl (431 kB)
           ---------------------------------------- 0.0/431.5 kB ? eta -:--:--
           ----------------- ---------------------- 184.3/431.5 kB 5.4 MB/s eta 0:00:01
           ------------------------------------ --- 389.1/431.5 kB 4.9 MB/s eta 0:00:01
           ---------------------------------------  430.1/431.5 kB 3.8 MB/s eta 0:00:01
           ---------------------------------------- 431.5/431.5 kB 2.4 MB/s eta 0:00:00
        Installing collected packages: protobuf, mysql-connector-python
          Attempting uninstall: mysql-connector-python
            Found existing installation: mysql-connector-python 9.1.0
            Uninstalling mysql-connector-python-9.1.0:
              Successfully uninstalled mysql-connector-python-9.1.0
        Successfully installed mysql-connector-python-8.0.27 protobuf-5.28.3
```



```
In [9]: !pip install mysql-connector-python sqlalchemy pandas matplotlib

Requirement already satisfied: mysql-connector-python in c:\users\admin\anaconda3\lib\site-packages (8.0.27)
Requirement already satisfied: sqlalchemy in c:\users\admin\anaconda3\lib\site-packages (1.4.39)
Requirement already satisfied: pandas in c:\users\admin\anaconda3\lib\site-packages (2.0.3)
Requirement already satisfied: matplotlib in c:\users\admin\anaconda3\lib\site-packages (3.7.2)
Requirement already satisfied: protobuf>=3.0.0 in c:\users\admin\anaconda3\lib\site-packages (from mysql-connector-python) (5.2
8.3)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\admin\anaconda3\lib\site-packages (from sqlalchemy) (2.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\admin\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\admin\anaconda3\lib\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\admin\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\admin\anaconda3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\admin\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.
16.0)
```



```
In [55]: !pip install geopandas

Collecting geopandas
  Obtaining dependency information for geopandas from https://files.pythonhosted.org/packages/c4/64/7d344cfcef5efddf9cf32f59
af7f855828e9d74b5f862eddf5bfd9f25323/geopandas-1.0.1-py3-none-any.whl.metadata
  Downloading geopandas-1.0.1-py3-none-any.whl.metadata (2.2 kB)
Requirement already satisfied: numpy>=1.22 in c:\users\admin\anaconda3\lib\site-packages (from geopandas) (1.24.3)
Collecting pyogrio>=0.7.2 (from geopandas)
  Obtaining dependency information for pyogrio>=0.7.2 from https://files.pythonhosted.org/packages/94/8d/24f21e6a93ca418231a
ee3bddade7a0766c89c523832f29e08a8860f83e6/pyogrio-0.10.0-cp311-cp311-win_amd64.whl.metadata
  Downloading pyogrio-0.10.0-cp311-cp311-win_amd64.whl.metadata (5.6 kB)
Requirement already satisfied: packaging in c:\users\admin\anaconda3\lib\site-packages (from geopandas) (23.1)
Requirement already satisfied: pandas>=1.4.0 in c:\users\admin\anaconda3\lib\site-packages (from geopandas) (2.0.3)
Collecting pyproj>=3.3.0 (from geopandas)
  Obtaining dependency information for pyproj>=3.3.0 from https://files.pythonhosted.org/packages/26/0c/b084e8839a117eaad8cb
4fbaa81bbb24c6f183de0ee95c6c4e2770ab6f09/pyproj-3.7.0-cp311-cp311-win_amd64.whl.metadata
  Downloading pyproj-3.7.0-cp311-cp311-win_amd64.whl.metadata (31 kB)
Collecting shapely>=2.0.0 (from geopandas)
  Obtaining dependency information for shapely>=2.0.0 from https://files.pythonhosted.org/packages/b1/5a/6a67d929c467a1973b6
bb9f0b00159cc343b02bf9a8d26db1abd2f87aa23/shapely-2.0.6-cp311-cp311-win_amd64.whl.metadata
  Downloading shapely-2.0.6-cp311-cp311-win_amd64.whl.metadata (7.2 kB)
```



```
In [73]: !pip install mysql-connector-python folium

Requirement already satisfied: mysql-connector-python in c:\users\admin\anaconda3\lib\site-packages (8.0.27)
Collecting folium
  Obtaining dependency information for folium from https://files.pythonhosted.org/packages/03/12/45a714482b9aa23cf708c5d07810d8
133c4277b9d28db55e51352a0cb2da/folium-0.18.0-py2.py3-none-any.whl.metadata
  Downloading folium-0.18.0-py2.py3-none-any.whl.metadata (3.8 kB)
Requirement already satisfied: protobuf>=3.0.0 in c:\users\admin\anaconda3\lib\site-packages (from mysql-connector-python) (5.2
8.3)
Collecting branca>=0.6.0 (from folium)
  Obtaining dependency information for branca>=0.6.0 from https://files.pythonhosted.org/packages/fc/be/720f85abacd654ec86f1431
bc7c004eae74417bd9d0e7a2bc43601062627/branca-0.8.0-py3-none-any.whl.metadata
  Downloading branca-0.8.0-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: jinja2>=2.9 in c:\users\admin\anaconda3\lib\site-packages (from folium) (3.1.2)
Requirement already satisfied: numpy in c:\users\admin\anaconda3\lib\site-packages (from folium) (1.24.3)
Requirement already satisfied: requests in c:\users\admin\anaconda3\lib\site-packages (from folium) (2.31.0)
Requirement already satisfied: xyzservices in c:\users\admin\anaconda3\lib\site-packages (from folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\admin\anaconda3\lib\site-packages (from jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\admin\anaconda3\lib\site-packages (from requests->folium)
(2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\admin\anaconda3\lib\site-packages (from requests->folium) (1.26.1
6)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests->folium) (2024.
8.30)
Downloading folium-0.18.0-py2.py3-none-any.whl (108 kB)
   ---------------------------------------- 0.0/108.9 kB ? eta -:--:--
   --- ------------------------------------ 10.2/108.9 kB ? eta -:--:--
   --- ------------------------------------ 10.2/108.9 kB ? eta -:--:--
   ----------- --------------------------- 30.7/108.9 kB 262.6 kB/s eta 0:00:01
   ------------------------- ------------- 71.7/108.9 kB 491.5 kB/s eta 0:00:01
   ----------------------------------- -- 102.4/108.9 kB 658.3 kB/s eta 0:00:01
   -------------------------------------- 108.9/108.9 kB 486.0 kB/s eta 0:00:00
Downloading branca-0.8.0-py3-none-any.whl (25 kB)
Installing collected packages: branca, folium
Successfully installed branca-0.8.0 folium-0.18.0
```