

Section 1: Activate Virtual Environment and Install Required Libraries

```
In [1]: #pip install pandas seaborn matplotlib plotly streamlit
```

Section 2: Import Libraries and Load Data

This step loads our datasets into Pandas DataFrames and checks the first few rows for confirmation.

```
In [7]: import pandas as pd

# Load datasets
bike_data = pd.read_csv('merged_citibike_weather.csv', low_memory=False)
weather_data = pd.read_csv('weather_2022.csv')

# Fix date columns and ensure correct format
bike_data['started_at'] = pd.to_datetime(bike_data['started_at'], errors='coerce')
bike_data['start_date'] = bike_data['started_at'].dt.date
weather_data['DATE'] = pd.to_datetime(weather_data['DATE'], errors='coerce').dt.date

# Display first few rows of both datasets
print("Bike Data:")
print(bike_data.head())

print("\nWeather Data:")
print(weather_data.head())
```

```
Bike Data:
   ride_id  Temperature  rideable_type  started_at  ended_at \
0  BFD29218AB271154      20.8  electric_bike  2024-11-19 13:43:24  22:31.5
1  7C953F2FD7BE1302      21.7   classic_bike                NaT  41:43.4
2  95893ABD40CED4B8      33.1  electric_bike                NaT  06:35.2
3  F853B50772137378      20.2   classic_bike                NaT  10:50.5
4  7590ADF834797B4B      34.0   classic_bike  2024-11-19 14:23:00  34:57.5
```

```
   start_station_name  start_station_id  end_station_name \
0  West End Ave & W 107 St      7650.05  Mt Morris Park W & W 120 St
1           4 Ave & 3 St      4028.04    Boerum Pl\t& Pacific St
2           1 Ave & E 62 St      6753.08           5 Ave & E 29 St
3           2 Ave & E 96 St      7338.02           5 Ave & E 29 St
4           6 Ave & W 34 St      6364.1            5 Ave & E 29 St
```

```
   end_station_id  start_lat  ...  end_lat  end_lng  member_casual \
0      7685.14  40.802117  ...  40.804038  -73.945925         member
1      4488.09  40.673746  ...  40.688489  -73.991160         member
2      6248.06  40.761227  ...  40.745168  -73.986831         member
3      6248.06  40.783964  ...  40.745168  -73.986831         member
4      6248.06  40.749640  ...  40.745168  -73.986831         member
```

```
   year  STATION  DATE  PRCP  TMAX  TMIN  start_date
0  1/21/2022  USW00094728  1/21/2022  0.0  -55.0  -99.0  2024-11-19
1  1/10/2022  USW00094728  1/10/2022  0.0   44.0  -43.0         NaT
2  1/26/2022  USW00094728  1/26/2022  0.0  -21.0  -66.0         NaT
3   1/3/2022  USW00094728   1/3/2022  0.0   28.0  -55.0         NaT
4  1/22/2022  USW00094728  1/22/2022  0.0  -16.0 -105.0  2024-11-19
```

[5 rows x 21 columns]

```
Weather Data:
   STATION  DATE  PRCP  TMAX  TMIN
0  USW00094728  2022-01-01   201   133   100
1  USW00094728  2022-01-02    10   150    28
2  USW00094728  2022-01-03     0    28   -55
3  USW00094728  2022-01-04     0    11   -71
4  USW00094728  2022-01-05    58    83    -5
```

Step 3: Create a Bar Chart for Most Popular Stations

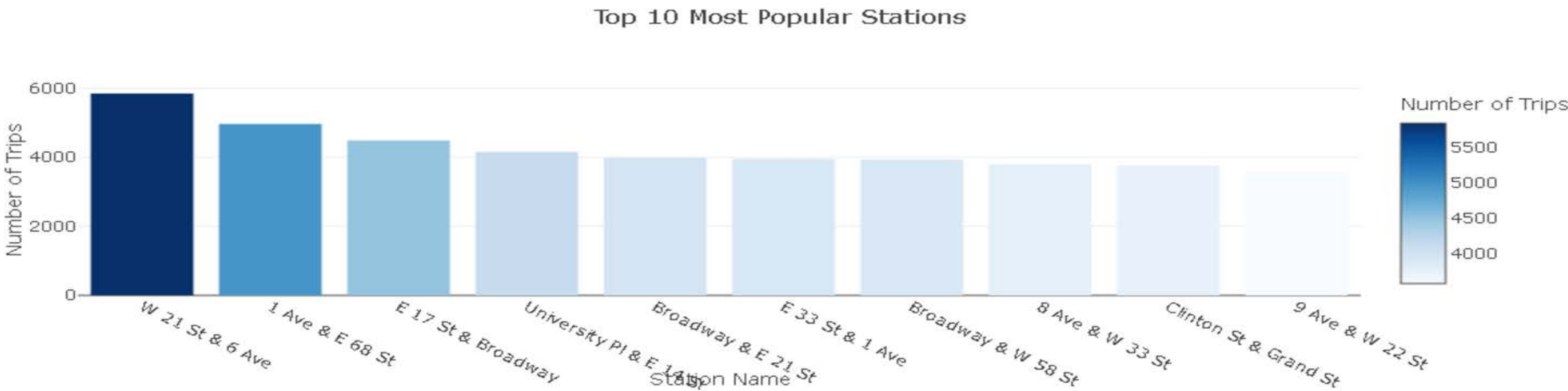
Use Plotly to visualize the most popular stations:

```
In [8]: import plotly.express as px

# Identify the top 10 most popular stations
popular_stations = bike_data['start_station_name'].value_counts().head(10).reset_index()
popular_stations.columns = ['Station', 'Trips']

# Create a bar chart
fig1 = px.bar(popular_stations, x='Station', y='Trips', title="Top 10 Most Popular Stations",
              labels={'Trips': 'Number of Trips', 'Station': 'Station Name'},
              color='Trips', color_continuous_scale='Blues')

fig1.show()
```



Step 4: Create a Dual-Axis Line Chart

Fix the data merging issue and plot a dual-axis line chart:

```
In [9]: # Aggregate daily bike trips
daily_trips = bike_data.groupby('start_date').size().reset_index(name='Trips')
daily_trips.columns = ['Date', 'Trips']

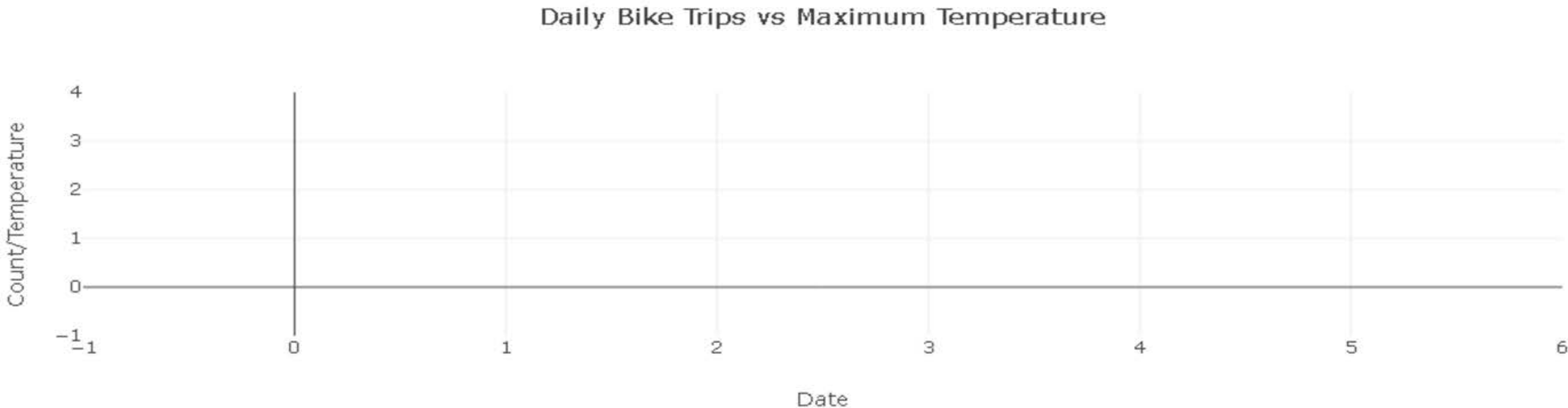
# Merge datasets and handle potential mismatches
merged_data = pd.merge(daily_trips, weather_data, left_on='Date', right_on='DATE', how='inner')

# Ensure merged_data is not empty
if merged_data.empty:
    print("Data merging issue detected. Check for date mismatches.")
else:
    print("Merged data preview:")
    print(merged_data.head())

# Create dual-axis line chart
fig2 = px.line(merged_data, x='Date', y=['Trips', 'TMAX'],
              labels={'value': 'Count/Temperature', 'variable': 'Metric'},
              title="Daily Bike Trips vs Maximum Temperature")

fig2.show()
```

Data merging issue detected. Check for date mismatches.



I did further steps in my_dashbaord.py file

In []: