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 [2]:
```
# Import required libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

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

# Display the first few rows to confirm successful loading
print("Bike Data:")
print(bike_data.head())

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

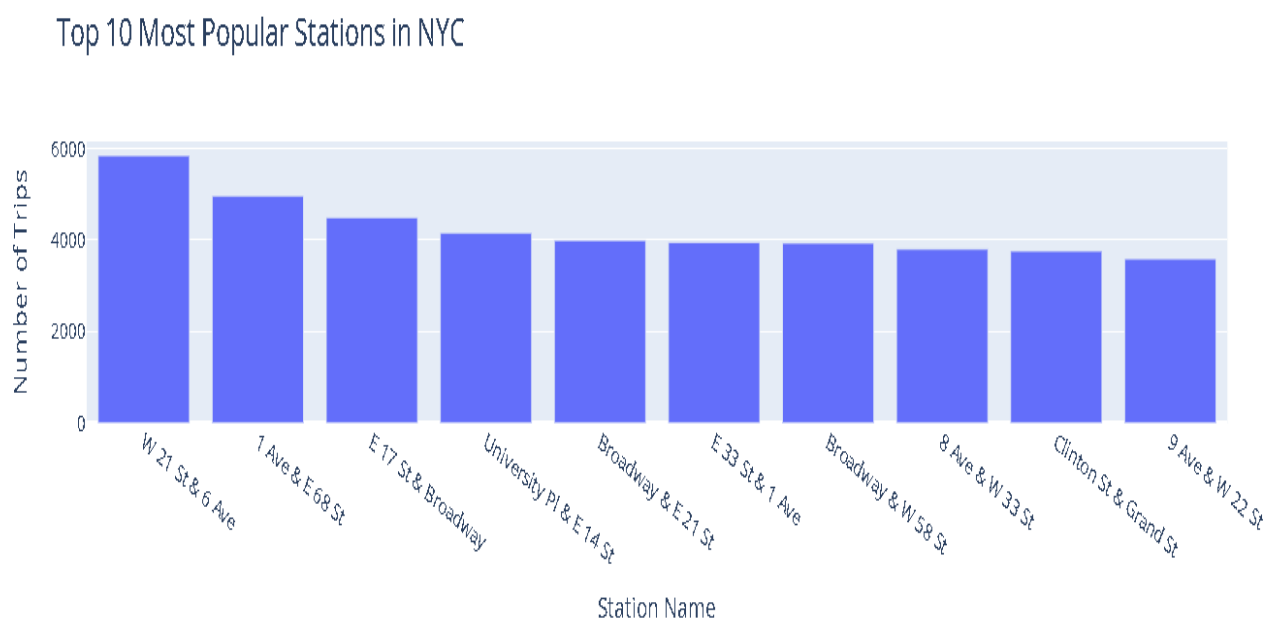
```
Bike Data:
            ride_id Temperature rideable_type started_at ended_at  \
0  BFD29218AB271154          20.8  electric_bike    13:43.4  22:31.5
1  7C953F2FD7BE1302          21.7   classic_bike    30:54.2  41:43.4
2  95893ABD40CED4B8          33.1  electric_bike    52:43.1  06:35.2
3  F853B50772137378          20.2   classic_bike    35:48.2  10:50.5
4  7590ADF834797B4B          34.0   classic_bike    14:23.0  34:57.5

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

  end_station_id  start_lat start_lng     end_lat    end_lng member_casual  \
0        7685.14  40.802117 -73.968181   40.804038 -73.945925        member
1        4488.09  40.673746 -73.985649   40.688489 -73.991160        member
2        6248.06  40.761227 -73.960940   40.745168 -73.986831        member
3        6248.06  40.783964 -73.947167   40.745168 -73.986831        member
4        6248.06  40.749640 -73.988050   40.745168 -73.986831        member

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

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
```

Section 3: Produce a Bar Chart for Most Popular Stations

This step identifies the most popular stations and visualizes them in an interactive bar chart using Plotly.

In [12]:
```
import plotly.express as px

# Find the most popular stations
popular_stations = bike_data['start_station_name'].value_counts().head(10)

# Convert to DataFrame
popular_stations_df = popular_stations.reset_index()
popular_stations_df.columns = ['Station', 'Trips']
```

```
# Plotly Bar Chart
fig = px.bar(popular_stations_df, x='Station', y='Trips',
             title="Top 10 Most Popular Stations in NYC",
             labels={'Station': 'Station Name', 'Trips': 'Number of Trips'})

# Customize layout
fig.update_layout(title_font_size=20, xaxis_title_font_size=15,
                  yaxis_title_font_size=15)

fig.show()
```



Section 4: Create a Dual-Axis Line Chart

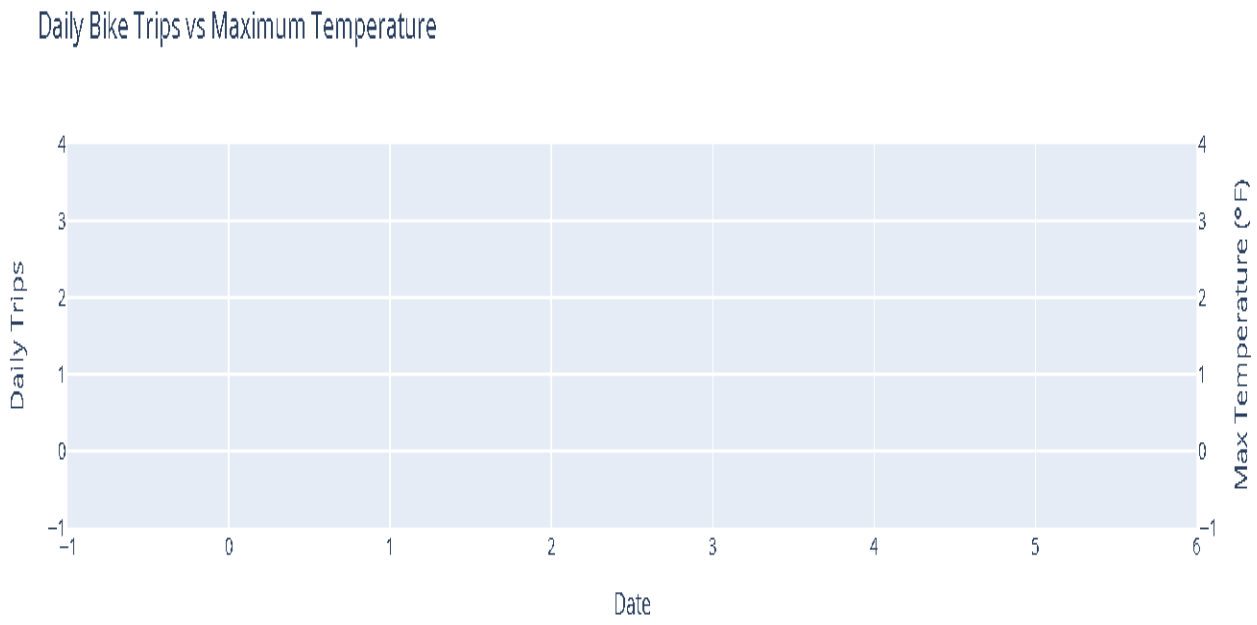
To analyze the relationship between bike trips and maximum temperatures

In [24]:
```
# Check the Merged Data:
print(merged_data.head())
```

```
Empty DataFrame
Columns: [Date, Trips, STATION, DATE, PRCP, TMAX, TMIN]
Index: []
```

In [25]:
```
# Check for NaN Values: NaN values in Trips or TMAX
merged_data = merged_data.dropna(subset=['Trips', 'TMAX'])
```

In [26]:
```
import plotly.graph_objects as go

# Create figure with secondary y-axis
fig = go.Figure()

# Add traces
fig.add_trace(
    go.Scatter(x=merged_data['Date'], y=merged_data['Trips'], name="Daily Trips", yaxis="y1")
)

fig.add_trace(
    go.Scatter(x=merged_data['Date'], y=merged_data['TMAX'], name="Max Temperature", yaxis="y2")
)

# Update layout for dual y-axes
fig.update_layout(
    title="Daily Bike Trips vs Maximum Temperature",
    xaxis=dict(title="Date"),
    yaxis=dict(title="Daily Trips", side="left"),
    yaxis2=dict(title="Max Temperature (°F)", overlaying="y", side="right"),
)

fig.show()
```



Section 5: Create a Streamlit Dashboard

I did my next steps in app.py file

In [ ]: