## Step 1: Activate the Virtual Environment and Install Libraries

In [20]:
```
# pip install pandas matplotlib seaborn jupyterlab
```

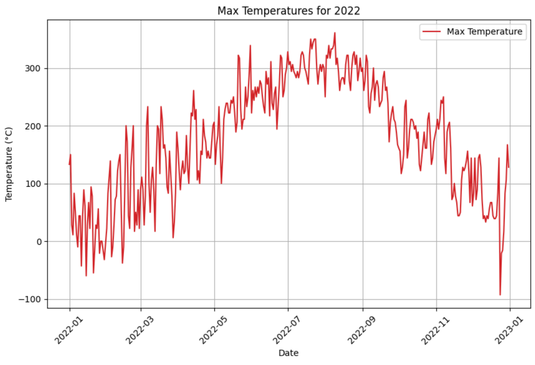
## Step 2: Import Libraries and Load Dataset

In [41]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load data
bike_data = pd.read_csv('merged_citibike_weather.csv')  # Replace with the actual file path
weather_data = pd.read_csv('weather_2022.csv')  # Replace with the actual file path

# Display the first few rows to ensure the data loaded correctly
print(bike_data.head())
print(weather_data.head())
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_17660\2188774527.py:6: DtypeWarning: Columns (6,8) have mixed types. Specify dtype option on import or set low_memory=False.
  bike_data = pd.read_csv('merged_citibike_weather.csv')  # Replace with the actual file path
            ride_id  Temperature  rideable_type started_at  ended_at  \
0  BFD29218AB271154         20.8  electric_bike    13:43.4  22:31.5
1  7C953F2FD7BE13O2         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  \
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  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
      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
```

## 3. Create a Line Plot of the Temperatures for 2022

In [42]:
```
# Convert 'DATE' column to datetime
weather_data['DATE'] = pd.to_datetime(weather_data['DATE'])

# Filter the data for the year 2022
weather_2022 = weather_data[weather_data['DATE'].dt.year == 2022]

# Plot the temperatures (TMAX) for 2022
plt.figure(figsize=(10, 6))
plt.plot(weather_2022['DATE'], weather_2022['TMAX'], label='Max Temperature', color='tab:red')
plt.title('Max Temperatures for 2022')
plt.xlabel('Date')
plt.ylabel('Temperature (°C)')
plt.xticks(rotation=45)
plt.grid(True)
plt.legend()
plt.show()
```




## 4. Create a Column with the Count of Trips per Day and Merge it with the Weather DataFrame

In [58]:
```
# Convert 'start_date' to datetime
bike_data['start_date'] = pd.to_datetime(bike_data['start_date'])

# Count trips per day
trips_per_day = bike_data.groupby('start_date').size().reset_index(name='trip_count')

# Merge the trips_per_day dataframe with the weather_2022 dataframe on 'DATE'
df_merged = pd.merge(weather_2022, trips_per_day, how='left', left_on='DATE', right_on='start_date')

# Display the merged dataframe
df_merged.head()
```

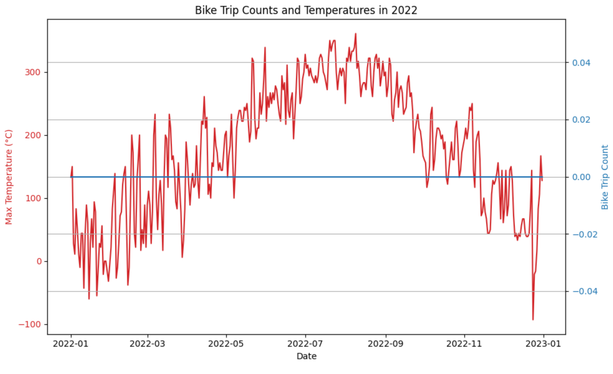
Out[58]:

| | ride_id | Temperature | rideable_type | started_at | ended_at | start_station_name | start_station_id | end_station_name | end_station_id | start_lat | ... | end_lng | member_casual | year | STATION | DATE | PRCP | TMAX | TMIN | start_date | trip_count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | BFD29218AB271154 | 20.8 | electric_bike | 2024-11-14 13:43:24 | 22:31.5 | West End Ave & W 107 St | 7650.05 | Mt Morris Park W & W 120 St | 7685.14 | 40.802117 | ... | -73.945925 | member | 1/21/2022 | USW00094728 | 2022-01-21 | 0.0 | -55.0 | -99.0 | NaT | NaN |
| 1 | 7590ADF834797B4B | 34.0 | classic_bike | 2024-11-14 14:23:00 | 34:57.5 | 6 Ave & W 34 St | 6364.1 | 5 Ave & E 29 St | 6248.06 | 40.749640 | ... | -73.986831 | member | 1/22/2022 | USW00094728 | 2022-01-22 | 0.0 | -16.0 | -105.0 | NaT | NaN |
| 2 | 621225A86D88489F | 39.5 | electric_bike | 2024-11-14 08:37:00 | 26:01.9 | 6 Ave & W 34 St | 6364.1 | Allen St & Rivington St | 5414.06 | 40.749640 | ... | -73.989978 | member | 1/6/2022 | USW00094728 | 2022-01-06 | 0.0 | 50.0 | 11.0 | NaT | NaN |
| 3 | F3D0C298E2EBC08A | 23.4 | classic_bike | 2024-11-14 16:37:54 | 20:59.7 | 6 Ave & W 34 St | 6364.1 | 5 Ave & E 29 St | 6248.06 | 40.749640 | ... | -73.986831 | member | 1/17/2022 | USW00094728 | 2022-01-17 | 295.0 | 67.0 | 11.0 | NaT | NaN |
| 4 | AC3302FDC2B1E054 | 32.3 | electric_bike | 2024-11-14 22:58:30 | 29:34.3 | Cleveland Pl & Spring St | 5492.05 | Broadway & E 14 St | 5905.12 | 40.722104 | ... | -73.990741 | member | 1/25/2022 | USW00094728 | 2022-01-25 | 0.0 | 56.0 | -21.0 | NaT | NaN |

5 rows × 22 columns

## 5. Create a line chart of bike trip counts and temperatures plotted on a dual axis

In [56]:
```
# Create the plot with two y-axes
fig, ax1 = plt.subplots(figsize=(10, 6))

# Plot temperature on the primary y-axis
ax1.set_xlabel('Date')
ax1.set_ylabel('Max Temperature (°C)', color='tab:red')
ax1.plot(df_merged['DATE'], df_merged['TMAX'], color='tab:red', label='Max Temperature')
ax1.tick_params(axis='y', labelcolor='tab:red')

# Create the secondary y-axis for bike trip counts
ax2 = ax1.twinx()
ax2.set_ylabel('Bike Trip Count', color='tab:blue')
ax2.plot(df_merged['DATE'], df_merged['trip_count'], color='tab:blue', label='Bike Trip Count')
ax2.tick_params(axis='y', labelcolor='tab:blue')

# Title and grid
plt.title('Bike Trip Counts and Temperatures in 2022')
fig.tight_layout()  # Ensures everything fits without overlap
plt.grid(True)

# Show the plot
plt.show()
```




## 6. Markdown Explanation for Code

## Explanation:

Plotting Temperatures: We used matplotlib's plot() function to visualize the maximum temperatures (TMAX) for the year 2022 as a time series. The DATE column was converted to a datetime format, and the data was filtered for the year 2022.

Counting Bike Trips: The bike trips are grouped by the start_date, and the count is calculated using the groupby() function. The resulting counts are merged with the weather data on the DATE column.

Dual-Axis Plot: We created a dual-axis plot using matplotlib. The primary axis (left) shows the temperatures, and the secondary axis (right) shows the bike trip counts. The method twinx() was used to create the second axis, allowing two different y-scales for comparison.

In [ ]:
```
```