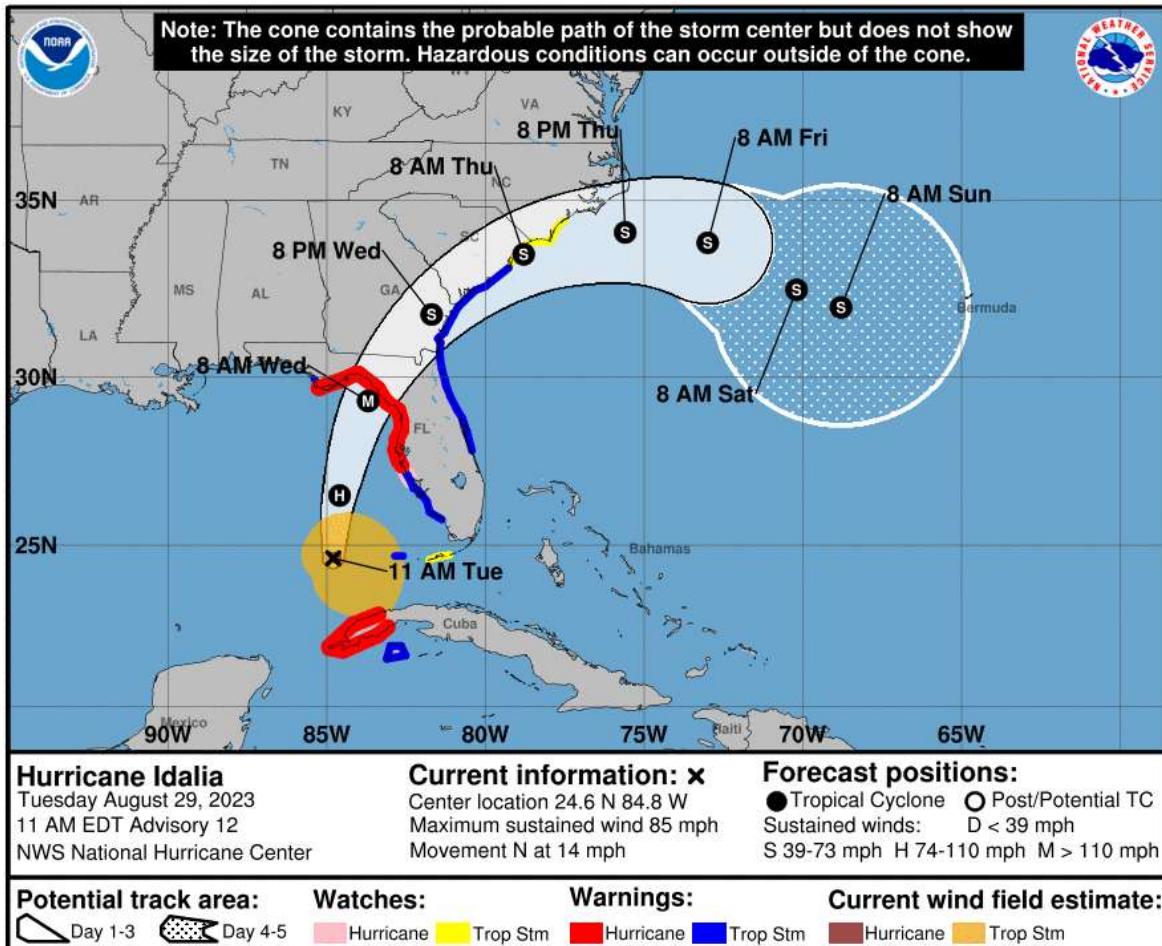


DMS672 Project

Hurricane IDALIA Path Prediction



Installing Requirements

```
In [10]: %pip install geopandas pandas matplotlib
```

Requirement already satisfied: geopandas in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (1.1.1)
Requirement already satisfied: pandas in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (2.3.0)
Requirement already satisfied: matplotlib in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (3.7.0)
Requirement already satisfied: pyogrio>=0.7.2 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from geopandas) (0.11.0)
Requirement already satisfied: packaging in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from geopandas) (22.0)
Requirement already satisfied: shapely>=2.0.0 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from geopandas) (2.1.1)
Requirement already satisfied: pyproj>=3.5.0 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from geopandas) (3.7.1)
Requirement already satisfied: numpy>=1.24 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from geopandas) (1.26.4)
Requirement already satisfied: tzdata>=2022.7 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: cycler>=0.10 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: certifi in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from pyogrio>=0.7.2->geopandas) (2022.12.7)
Requirement already satisfied: six>=1.5 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

Note: you may need to restart the kernel to use updated packages.

In [11]: `#system Level installation, need run as administrator
%pip install "numpy<2" --user`

Requirement already satisfied: numpy<2 in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (1.26.4)
Note: you may need to restart the kernel to use updated packages.

In [12]: `pip install --upgrade bottleneck`

Requirement already satisfied: bottleneck in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (1.5.0)
Requirement already satisfied: numpy in c:\users\abhiraj akhouri\anaconda3\lib\site-packages (from bottleneck) (1.26.4)
Note: you may need to restart the kernel to use updated packages.

Loading Hurricane Info

Note: Warnings due to shapefile layers, fixed later

```
In [13]: import geopandas as gpd
import pandas as pd
import os

# Directory containing all hurricane shapefile ZIPs
data_mining_dir = 'nhc_hurricane_idalia_files/data_mining'

# List all .zip files in the directory
zip_files = [os.path.join(data_mining_dir, f) for f in os.listdir(data_mining_dir) if

# Load all shapefiles into a list of GeoDataFrames
gdf_list = [gpd.read_file(zip_file) for zip_file in zip_files]

# Combine all GeoDataFrames into one
combined_gdf = gpd.GeoDataFrame(pd.concat(gdf_list, ignore_index=True))

# Display columns and first few rows
print("Combined Hurricane Data Columns:", combined_gdf.columns)
print(combined_gdf.head())
```

```
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_001.zip': 'al102023-001_5day_lin' (default), 'al102023-001_5day_pgn', 'al102023-001_5day_pts', 'al102023-001_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_001A.zip': 'al102023-001A_5day_lin' (default), 'al102023-001A_5day_pgn', 'al102023-001A_5day_pts', 'al102023-001A_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_002.zip': 'al102023-002_5day_lin' (default), 'al102023-002_5day_pgn', 'al102023-002_5day_pts', 'al102023-002_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_002A.zip': 'al102023-002A_5day_lin' (default), 'al102023-002A_5day_pgn', 'al102023-002A_5day_pts', 'al102023-002A_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_003.zip': 'al102023-003_5day_lin' (default), 'al102023-003_5day_pgn', 'al102023-003_5day_pts', 'al102023-003_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_003A.zip': 'al102023-003A_5day_lin' (default), 'al102023-003A_5day_pgn', 'al102023-003A_5day_pts', 'al102023-003A_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_004.zip': 'al102023-004_5day_lin' (default), 'al102023-004_5day_pgn', 'al102023-004_5day_pts', 'al102023-004_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_004A.zip': 'al102023-004A_5day_lin' (default), 'al102023-004A_5day_pgn', 'al102023-004A_5day_pts', 'al102023-004A_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_005.zip': 'al102023-005_5day_lin' (default), 'al102023-005_5day_pgn', 'al102023-005_5day_pts', 'al102023-005_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_005A.zip': 'al102023-005A_5day_lin' (default), 'al102023-005A_5day_pgn', 'al102023-005A_5day_pts', 'al102023-005A_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_006.zip': 'al102023-006_5day_lin' (default), 'al102023-006_5day_pgn', 'al102023-006_5day_pts', 'al102023-006_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()  
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_006A.zip': 'al102023-006A_5day_lin' (default), 'al102023-006A_5day_pgn', 'al102023-006A_5day_pts', 'al102023-006A_ww_wwlin'. Specify layer parameter to avoid this warning.  
    result = read_func()
```

```
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_007.zip': 'al102023-007_5day_lin'
(default), 'al102023-007_5day_pgn', 'al102023-007_5day_pts', 'al102023-007_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_007A.zip': 'al102023-007A_5day_li
n' (default), 'al102023-007A_5day_pgn', 'al102023-007A_5day_pts', 'al102023-007A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_008.zip': 'al102023-008_5day_lin'
(default), 'al102023-008_5day_pgn', 'al102023-008_5day_pts', 'al102023-008_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_008A.zip': 'al102023-008A_5day_li
n' (default), 'al102023-008A_5day_pgn', 'al102023-008A_5day_pts', 'al102023-008A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_009.zip': 'al102023-009_5day_lin'
(default), 'al102023-009_5day_pgn', 'al102023-009_5day_pts', 'al102023-009_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_009A.zip': 'al102023-009A_5day_li
n' (default), 'al102023-009A_5day_pgn', 'al102023-009A_5day_pts', 'al102023-009A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_010.zip': 'al102023-010_5day_lin'
(default), 'al102023-010_5day_pgn', 'al102023-010_5day_pts', 'al102023-010_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_010A.zip': 'al102023-010A_5day_li
n' (default), 'al102023-010A_5day_pgn', 'al102023-010A_5day_pts', 'al102023-010A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_011.zip': 'al102023-011_5day_lin'
(default), 'al102023-011_5day_pgn', 'al102023-011_5day_pts', 'al102023-011_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_011A.zip': 'al102023-011A_5day_li
n' (default), 'al102023-011A_5day_pgn', 'al102023-011A_5day_pts', 'al102023-011A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_012.zip': 'al102023-012_5day_lin'
(default), 'al102023-012_5day_pgn', 'al102023-012_5day_pts', 'al102023-012_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_012A.zip': 'al102023-012A_5day_li
n' (default), 'al102023-012A_5day_pgn', 'al102023-012A_5day_pts', 'al102023-012A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
```

```
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_013.zip': 'al102023-013_5day_lin'
(default), 'al102023-013_5day_pgn', 'al102023-013_5day_pts', 'al102023-013_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_013A.zip': 'al102023-013A_5day_li
n' (default), 'al102023-013A_5day_pgn', 'al102023-013A_5day_pts', 'al102023-013A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_014.zip': 'al102023-014_5day_lin'
(default), 'al102023-014_5day_pgn', 'al102023-014_5day_pts', 'al102023-014_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_014A.zip': 'al102023-014A_5day_li
n' (default), 'al102023-014A_5day_pgn', 'al102023-014A_5day_pts', 'al102023-014A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_015.zip': 'al102023-015_5day_lin'
(default), 'al102023-015_5day_pgn', 'al102023-015_5day_pts', 'al102023-015_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_015A.zip': 'al102023-015A_5day_li
n' (default), 'al102023-015A_5day_pgn', 'al102023-015A_5day_pts', 'al102023-015A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_016.zip': 'al102023-016_5day_lin'
(default), 'al102023-016_5day_pgn', 'al102023-016_5day_pts', 'al102023-016_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_016A.zip': 'al102023-016A_5day_li
n' (default), 'al102023-016A_5day_pgn', 'al102023-016A_5day_pts', 'al102023-016A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_017.zip': 'al102023-017_5day_lin'
(default), 'al102023-017_5day_pgn', 'al102023-017_5day_pts', 'al102023-017_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_017A.zip': 'al102023-017A_5day_li
n' (default), 'al102023-017A_5day_pgn', 'al102023-017A_5day_pts', 'al102023-017A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_018.zip': 'al102023-018_5day_lin'
(default), 'al102023-018_5day_pgn', 'al102023-018_5day_pts', 'al102023-018_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_018A.zip': 'al102023-018A_5day_li
n' (default), 'al102023-018A_5day_pgn', 'al102023-018A_5day_pts', 'al102023-018A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
```

```
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_019.zip': 'al102023-019_5day_lin'
(default), 'al102023-019_5day_pgn', 'al102023-019_5day_pts', 'al102023-019_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_019A.zip': 'al102023-019A_5day_li
n' (default), 'al102023-019A_5day_pgn', 'al102023-019A_5day_pts', 'al102023-019A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_020.zip': 'al102023-020_5day_lin'
(default), 'al102023-020_5day_pgn', 'al102023-020_5day_pts', 'al102023-020_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_020A.zip': 'al102023-020A_5day_li
n' (default), 'al102023-020A_5day_pgn', 'al102023-020A_5day_pts', 'al102023-020A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_021.zip': 'al102023-021_5day_li
n' (default), 'al102023-021_5day_pgn', 'al102023-021_5day_pts', 'al102023-021_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_021A.zip': 'al102023-021A_5day_li
n' (default), 'al102023-021A_5day_pgn', 'al102023-021A_5day_pts', 'al102023-021A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_022.zip': 'al102023-022_5day_li
n' (default), 'al102023-022_5day_pgn', 'al102023-022_5day_pts', 'al102023-022_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_022A.zip': 'al102023-022A_5day_li
n' (default), 'al102023-022A_5day_pgn', 'al102023-022A_5day_pts', 'al102023-022A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_023.zip': 'al102023-023_5day_li
n' (default), 'al102023-023_5day_pgn', 'al102023-023_5day_pts', 'al102023-023_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_023A.zip': 'al102023-023A_5day_li
n' (default), 'al102023-023A_5day_pgn', 'al102023-023A_5day_pts', 'al102023-023A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_024.zip': 'al102023-024_5day_li
n' (default), 'al102023-024_5day_pgn', 'al102023-024_5day_pts', 'al102023-024_ww_wwlin'.
Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWa
rning: More than one layer found in 'al102023_5day_024A.zip': 'al102023-024A_5day_li
n' (default), 'al102023-024A_5day_pgn', 'al102023-024A_5day_pts', 'al102023-024A_ww_w
wlin'. Specify layer parameter to avoid this warning.
    result = read_func()
```

```
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_025.zip': 'al102023-025_5day_lin' (default), 'al102023-025_5day_pgn', 'al102023-025_5day_pts', 'al102023-025_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_025A.zip': 'al102023-025A_5day_lin' (default), 'al102023-025A_5day_pgn', 'al102023-025A_5day_pts', 'al102023-025A_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_026.zip': 'al102023-026_5day_lin' (default), 'al102023-026_5day_pgn', 'al102023-026_5day_pts', 'al102023-026_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_027.zip': 'al102023-027_5day_lin' (default), 'al102023-027_5day_pgn', 'al102023-027_5day_pts', 'al102023-027_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_027A.zip': 'al102023-027A_5day_lin' (default), 'al102023-027A_5day_pgn', 'al102023-027A_5day_pts', 'al102023-027A_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_028.zip': 'al102023-028_5day_lin' (default), 'al102023-028_5day_pgn', 'al102023-028_5day_pts', 'al102023-028_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()

Combined Hurricane Data Columns: Index(['STORMNAME', 'STORMTYPE', 'ADVDATE', 'ADVISNUM', 'STORMNUM', 'FCSTPRD', 'BASIN', 'geometry'],  

      dtype='object')
   STORMNAME  STORMTYPE          ADVDATE  ADVISNUM  STORMNUM \
0       Ten        TD  400 PM CDT Sat Aug 26 2023       1     10.0
1       Ten        TD  700 PM CDT Sat Aug 26 2023      1A     10.0
2       Ten        TD 1000 PM CDT Sat Aug 26 2023       2     10.0
3       Ten        TD  100 AM CDT Sun Aug 27 2023      2A     10.0
4       Ten        TD  400 AM CDT Sun Aug 27 2023       3     10.0

   FCSTPRD  BASIN           geometry
0    120.0    AL  LINESTRING (-86.1 21.1, -86.3 21, -86.2 20.9, ...
1    120.0    AL  LINESTRING (-86.2 21.2, -86.3 21, -86.2 20.9, ...
2    120.0    AL  LINESTRING (-86.4 21.1, -86.4 20.9, -86.2 20.9...
3    120.0    AL  LINESTRING (-86.8 20.7, -86.4 20.9, -86.2 20.9...
4    120.0    AL  LINESTRING (-86.8 20.1, -86.5 20.1, -86.3 20.6...

C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_028A.zip': 'al102023-028A_5day_lin' (default), 'al102023-028A_5day_pgn', 'al102023-028A_5day_pts', 'al102023-028A_ww_wwlin'. Specify layer parameter to avoid this warning.
    result = read_func()
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\pyogrio\geopandas.py:275: UserWarning: More than one layer found in 'al102023_5day_029.zip': 'al102023-029_5day_lin' (default), 'al102023-029_5day_pgn', 'al102023-029_5day_pts'. Specify layer parameter to avoid this warning.
    result = read_func()
```

```
In [14]: layers = ['al102023-001_5day_lin', 'al102023-001_5day_pgn', 'al102023-001_5day_pts', 'print(layers)
```

```
[ 'al102023-001_5day_lin', 'al102023-001_5day_pgn', 'al102023-001_5day_pts', 'al102023-001_ww_wwlin' ]
```

```
In [15]: import os
import geopandas as gpd
import fiona
import pandas as pd

# Directory containing all hurricane ZIP shapefiles
data_mining_dir = 'nhc_hurricane_idalia_files/data_mining'
zip_files = [os.path.join(data_mining_dir, f) for f in os.listdir(data_mining_dir) if

# Define base Layer name types
layer_types = ['_5day_lin', '_5day_pgn', '_5day_pts', '_ww_wwlin']

# Function to extract advisory number from filename
def get_advisory_number(zip_filename):
    # Example: 'al102023_5day_001.zip' → '001'
    return zip_filename.split('_')[-1].replace('.zip', '')

# For each Layer type, combine that Layer from all ZIPs
combined_layers = {}

for layer_type in layer_types:
    gdf_list = []
    for zip_file in zip_files:
        zip_name = os.path.basename(zip_file)
        # Construct the correct base prefix: 'al102023-001'
        base_prefix = zip_name.replace('.zip', '').replace('_5day', '').replace('_', '')
        layer_name = base_prefix + layer_type
        try:
            gdf = gpd.read_file(zip_file, layer=layer_name)
            gdf_list.append(gdf)
        except Exception as e:
            print(f"Could not load {layer_name} from {zip_file}: {e}")
    if gdf_list:
        combined_layers[layer_type] = gpd.GeoDataFrame(pd.concat(gdf_list, ignore_index=True))
    else:
        combined_layers[layer_type] = None

# Display each combined GeoDataFrame
for layer_type, gdf in combined_layers.items():
    print(f"\n==== Layer: {layer_type} ===")
    if gdf is not None:
        print(gdf.head())
    else:
        print("No data found for this layer.")
```

Could not load al102023-013_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_013.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-013A_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_013A.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-014_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_014.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-014A_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_014A.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-018_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_018.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-018A_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_018A.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-019_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_019.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-019A_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_019A.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-020_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_020.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-020A_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_020A.zip: IllegalArgumentException: point array must contain 0 or >1 elements

Could not load al102023-029_ww_wwlin from nhc_hurricane_idalia_files/data_mining\al10 2023_5day_029.zip: Layer 'al102023-029_ww_wwlin' could not be opened

==== Layer: _5day_lin ===

	STORMNAME	STORMTYPE	ADVDATE	ADVISNUM	STORMNUM	\
0	Ten	TD	400 PM CDT Sat Aug 26 2023	1	10.0	
1	Ten	TD	700 PM CDT Sat Aug 26 2023	1A	10.0	
2	Ten	TD	1000 PM CDT Sat Aug 26 2023	2	10.0	
3	Ten	TD	100 AM CDT Sun Aug 27 2023	2A	10.0	
4	Ten	TD	400 AM CDT Sun Aug 27 2023	3	10.0	

	FCSTPRD	BASIN	geometry
0	120.0	AL	LINESTRING (-86.1 21.1, -86.3 21, -86.2 20.9, ...)
1	120.0	AL	LINESTRING (-86.2 21.2, -86.3 21, -86.2 20.9, ...)
2	120.0	AL	LINESTRING (-86.4 21.1, -86.4 20.9, -86.2 20.9...)
3	120.0	AL	LINESTRING (-86.8 20.7, -86.4 20.9, -86.2 20.9...)
4	120.0	AL	LINESTRING (-86.8 20.1, -86.5 20.1, -86.3 20.6...)

==== Layer: _5day_pgn ===

	STORMNAME	STORMTYPE	ADVDATE	ADVISNUM	STORMNUM	\
--	-----------	-----------	---------	----------	----------	---

eda

0	Ten	TD	400 PM CDT Sat Aug 26 2023	1	10.0
1	Ten	TD	700 PM CDT Sat Aug 26 2023	1A	10.0
2	Ten	TD	1000 PM CDT Sat Aug 26 2023	2	10.0
3	Ten	TD	100 AM CDT Sun Aug 27 2023	2A	10.0
4	Ten	TD	400 AM CDT Sun Aug 27 2023	3	10.0

FCSTPRD BASIN geometry

0	120.0	AL	POLYGON ((-86.31667 19.91362, -86.2896 19.9098...))
1	120.0	AL	POLYGON ((-86.31667 19.91362, -86.2896 19.9098...))
2	120.0	AL	POLYGON ((-86.39421 20.26079, -86.3663 20.2569...))
3	120.0	AL	POLYGON ((-86.40915 20.27286, -86.38139 20.268...))
4	120.0	AL	POLYGON ((-86.68235 19.68691, -86.65387 19.682...))

== Layer: _5day_pts ==

	ADVDATE	ADVISNUM	BASIN	DATELBL	DVLBL	FCSTPRD	\
0	400 PM CDT Sat Aug 26 2023	1	AL	4:00 PM Sat	D	120.0	
1	400 PM CDT Sat Aug 26 2023	1	AL	1:00 AM Sun	D	120.0	
2	400 PM CDT Sat Aug 26 2023	1	AL	1:00 PM Sun	S	120.0	
3	400 PM CDT Sat Aug 26 2023	1	AL	1:00 AM Mon	S	120.0	
4	400 PM CDT Sat Aug 26 2023	1	AL	1:00 PM Mon	S	120.0	

	FLDATELBL	GUST	LAT	LON	...	STORMNUM	\
0	2023-08-26 1:00 PM Sat CDT	35.0	21.1	-86.1	...	10.0	
1	2023-08-27 1:00 AM Sun CDT	40.0	21.0	-86.3	...	10.0	
2	2023-08-27 1:00 PM Sun CDT	45.0	20.9	-86.2	...	10.0	
3	2023-08-28 1:00 AM Mon CDT	50.0	20.8	-86.1	...	10.0	
4	2023-08-28 1:00 PM Mon CDT	55.0	22.0	-85.7	...	10.0	

	STORMSRC	STORMTYPE	TD	TCDVLP	TAU	TCDIR	TCSPD	\
0	Tropical Cyclone		TD	Tropical Depression	0.0	360.0	0.0	
1	Tropical Cyclone		TD	Tropical Depression	12.0	9999.0	9999.0	
2	Tropical Cyclone		TS	Tropical Storm	24.0	9999.0	9999.0	
3	Tropical Cyclone		TS	Tropical Storm	36.0	9999.0	9999.0	
4	Tropical Cyclone		TS	Tropical Storm	48.0	9999.0	9999.0	

	TIMEZONE	VALIDTIME	geometry
0	CDT	26/1800	POINT (-86.1 21.1)
1	CDT	27/0600	POINT (-86.3 21)
2	CDT	27/1800	POINT (-86.2 20.9)
3	CDT	28/0600	POINT (-86.1 20.8)
4	CDT	28/1800	POINT (-85.7 22)

[5 rows x 24 columns]

== Layer: _ww_wwlin ==

	STORMNAME	STORMTYPE	ADVDATE	ADVISNUM	STORMNUM	FCSTPRD	\
0	Ten	TD	400 PM CDT Sat Aug 26 2023	1	10.0	120.0	
1	Ten	TD	400 PM CDT Sat Aug 26 2023	1	10.0	120.0	
2	Ten	TD	400 PM CDT Sat Aug 26 2023	1	10.0	120.0	
3	Ten	TD	700 PM CDT Sat Aug 26 2023	1A	10.0	120.0	
4	Ten	TD	700 PM CDT Sat Aug 26 2023	1A	10.0	120.0	

	BASIN	TCWW	geometry
0	AL	TWA	LINESTRING (-83.08 22.49, -83.39 22.23, -84.49...)
1	AL	TWA	LINESTRING (-82.52 21.6, -82.66 21.9, -83 21.9...)
2	AL	TWR	LINESTRING (-88.18 21.61, -87.1 21.61, -86.82 ...)
3	AL	TWA	LINESTRING (-83.08 22.49, -83.39 22.23, -84.49...)
4	AL	TWA	LINESTRING (-82.52 21.6, -82.66 21.9, -83 21.9...)

In []:

In []:

Hurricane Visualization by Layers

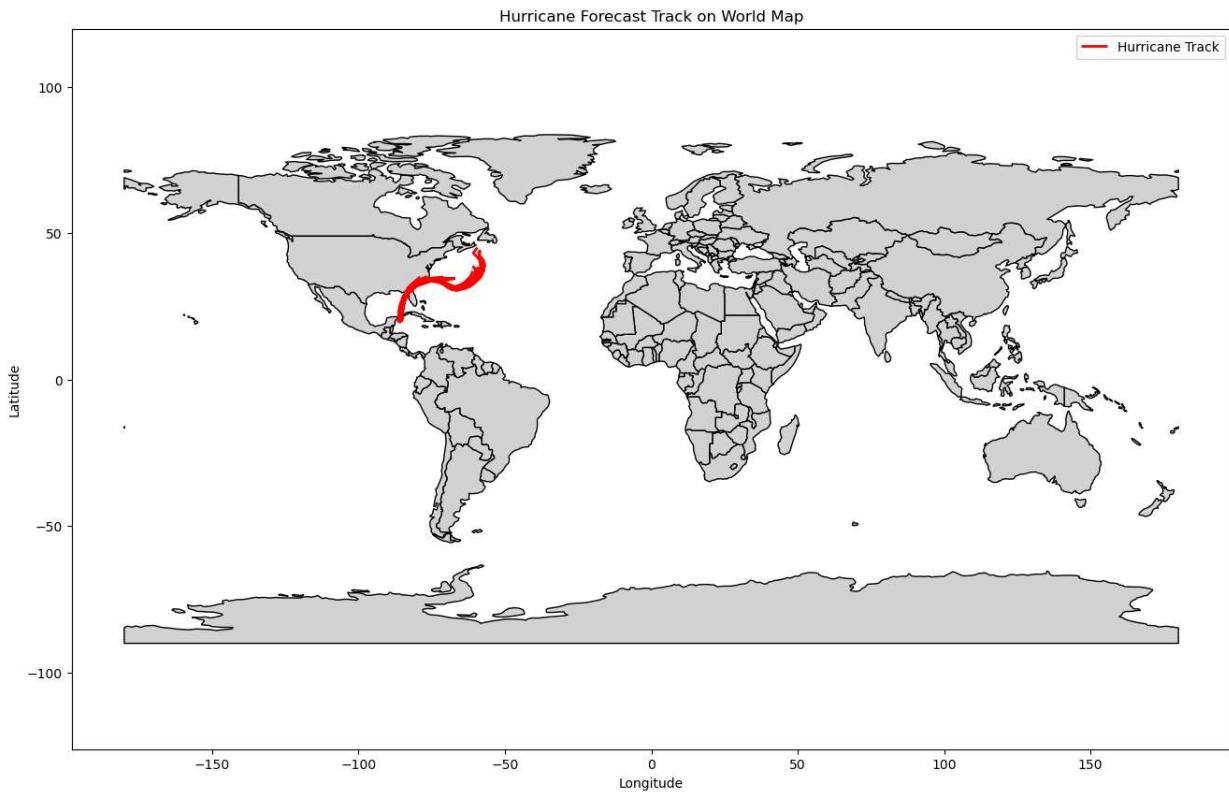
```
In [56]: import geopandas as gpd
import matplotlib.pyplot as plt

# Update the path to where you extracted the shapefile
world = gpd.read_file('world/ne_110m_admin_0_countries.shp')

fig, ax = plt.subplots(figsize=(16, 10))
world.plot(ax=ax, color='lightgrey', edgecolor='black')

# Example: Plot hurricane track (replace with your actual GeoDataFrame)
track_gdf = combined_layers['_5day_lin']
track_gdf.plot(ax=ax, color='red', linewidth=2, label='Hurricane Track')

plt.title('Hurricane Forecast Track on World Map')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.axis('equal')
plt.show()
```



```
In [64]: import geopandas as gpd
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
```

```
# Load the world boundaries shapefile (update path as needed)
world = gpd.read_file('world/ne_110m_admin_0_countries.shp')
```

```

# Hurricane Layers and their display names
layers = {
    '_5day_lin': 'Forecast Track',
    '_5day_pgn': 'Forecast Cone',
    '_5day_pts': 'Forecast Points',
    '_ww_wwlin': 'Watch/Warning Lines'
}

# Assign a unique color to each state
state_colors = {
    'Florida': 'darkgreen',
    'Georgia': 'darkorange',
    'South Carolina': 'purple',
    'North Carolina': 'crimson'
}

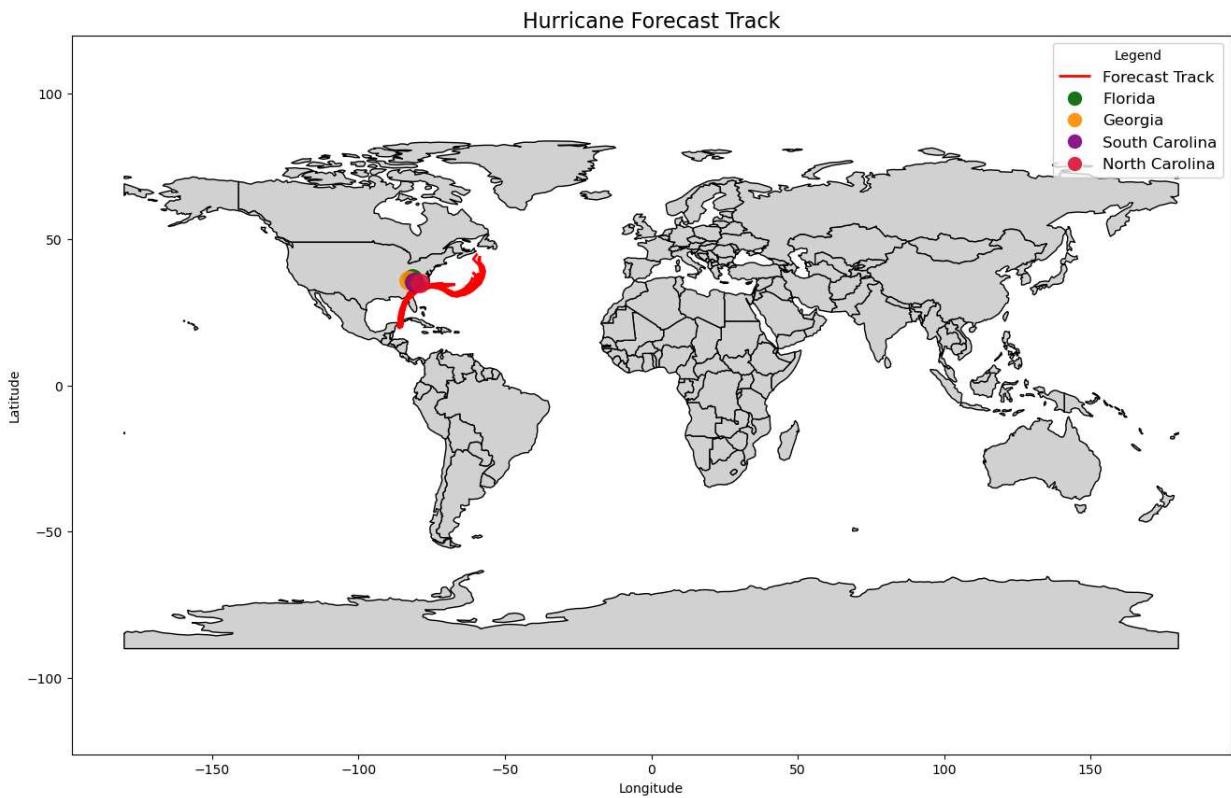
# State marker coordinates (Longitude, Latitude) above the storm region
state_coords_above = {
    'Florida': (-81.5158, 36.5),
    'Georgia': (-82.9001, 36.0),
    'South Carolina': (-80.9450, 35.5),
    'North Carolina': (-79.0193, 35.0)
}

# Create custom Legend handles for states (colored markers, names only in Legend)
state_handles = [
    Line2D([0], [0], marker='o', color='w', label=state,
          markerfacecolor=state_colors[state], markersize=12, alpha=0.9)
    for state in state_colors
]

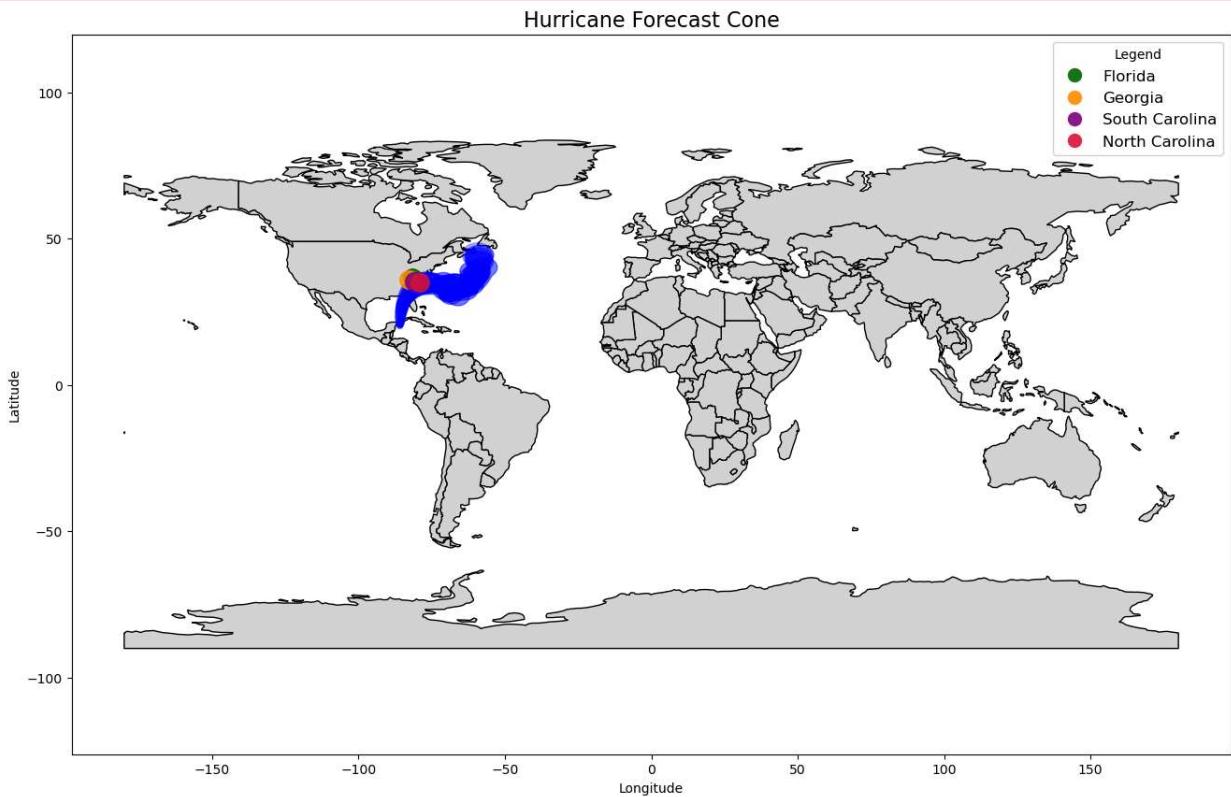
for layer_key, layer_name in layers.items():
    fig, ax = plt.subplots(figsize=(16, 10))
    world.plot(ax=ax, color='lightgrey', edgecolor='black')
    gdf = combined_layers[layer_key]
    if gdf is not None and not gdf.empty:
        if layer_key == '_5day_lin':
            gdf.plot(ax=ax, color='red', linewidth=2, label=layer_name)
        elif layer_key == '_5day_pgn':
            gdf.plot(ax=ax, color='blue', alpha=0.3, edgecolor='blue', label=layer_name)
        elif layer_key == '_5day_pts':
            gdf.plot(ax=ax, color='black', markersize=30, marker='o', label=layer_name)
        elif layer_key == '_ww_wwlin':
            gdf.plot(ax=ax, color='orange', linewidth=2, label=layer_name)
    # Plot state markers above the storm region (no text Labels)
    for state, (lon, lat) in state_coords_above.items():
        ax.plot(lon, lat, marker='o', color=state_colors[state], markersize=14, alpha=0.9)
    # Gather hurricane layer handles for the Legend
    handles, labels = ax.get_legend_handles_labels()
    # Add state handles and names to the Legend (names only in Legend)
    handles += state_handles
    labels += list(state_colors.keys())
    ax.legend(handles, labels, loc='upper right', fontsize=12, title='Legend')
    ax.set_title(f'Hurricane {layer_name}', fontsize=16)
    ax.set_xlabel('Longitude')
    ax.set_ylabel('Latitude')
    ax.set_xlim(-90, -75)
    ax.set_ylim(22, 38)

```

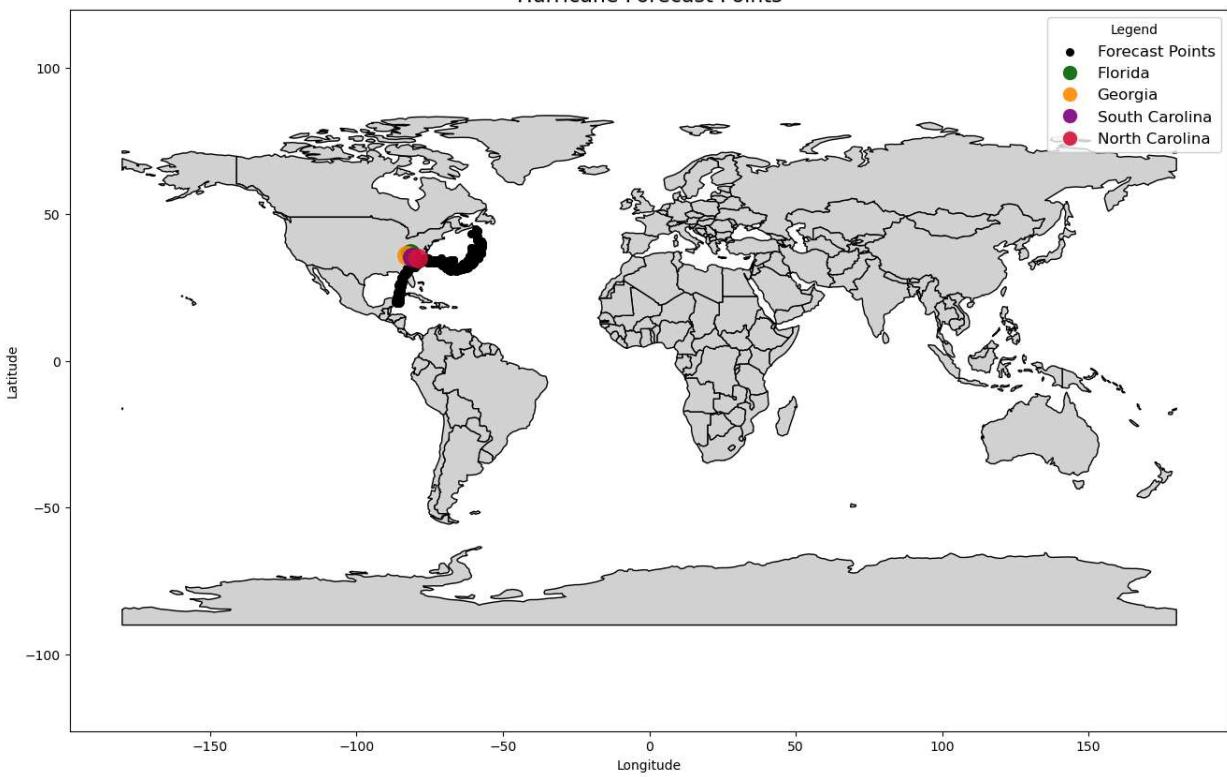
```
ax.axis('equal')
plt.show()
```



```
C:\Users\Abhiraj Akhouri\AppData\Local\Temp\ipykernel_12196\4247969877.py:56: UserWarning: Legend does not support handles for PatchCollection instances.
See: https://matplotlib.org/stable/tutorials/intermediate/legend\_guide.html#implementing-a-custom-legend-handler
    handles, labels = ax.get_legend_handles_labels()
```



Hurricane Forecast Points



Hurricane Watch/Warning Lines



Loading Geo-position Info

```
In [70]: # Paths to each state's census tract shapefile
tract_paths = {
    'Florida': 'geospatial/tl_rd22_12_tract.shp',
    'Georgia': 'geospatial/tl_rd22_13_tract.shp',
    'North Carolina': 'geospatial/tl_rd22_37_tract.shp',
    'South Carolina': 'geospatial/tl_rd22_45_tract.shp'
```

```

}
tract_gdfs = {}
for state, path in tract_paths.items():
    if os.path.exists(path):
        gdf = gpd.read_file(path)
        gdf['State'] = state # Optional: add state name for reference
        tract_gdfs[state] = gdf
        print(f"{state}: {len(gdf)} tracts loaded.")
    else:
        print(f"File not found for {state}: {path}")

```

Florida: 5160 tracts loaded.
 Georgia: 2796 tracts loaded.
 North Carolina: 2672 tracts loaded.
 South Carolina: 1323 tracts loaded.

In [76]:

```

import geopandas as gpd
import pandas as pd

### Use the forecast cone polygons (by advisory/day)
cone_gdf = combined_layers['_5day_pgn']

import pandas as pd
import geopandas as gpd

# Combine all state tract GeoDataFrames into one
all_tracts = gpd.GeoDataFrame(pd.concat(tract_gdfs.values(), ignore_index=True))

all_tracts = all_tracts.to_crs(cone_gdf.crs)

# Prepare a dictionary to hold daily affected tracts
affected_daily = {}
for adv in sorted(cone_gdf['ADVISNUM'].unique()):
    daily_cone = cone_gdf[cone_gdf['ADVISNUM'] == adv]
    affected = gpd.sjoin(all_tracts, daily_cone, how='inner', predicate='intersects')
    affected_daily[adv] = affected

```

In [77]:

```
print(affected_daily[sorted(cone_gdf['ADVISNUM'].unique())[0]].drop(columns='geometry')
```

STATEFP	COUNTYFP	TRACTCE	GEOID	NAME	NAMESAD	MTFCC	\
101	12	131	990000	12131990000	9900	Census Tract 9900	G5020
102	12	103	027703	12103027703	277.03	Census Tract 277.03	G5020
103	12	103	028103	12103028103	281.03	Census Tract 281.03	G5020
104	12	103	028104	12103028104	281.04	Census Tract 281.04	G5020
105	12	103	024408	12103024408	244.08	Census Tract 244.08	G5020

FUNCSTAT	ALAND	AWATER	...	INTPTLON	State index_right	\
101	S	0	659907762	...	-086.1954241	Florida 0
102	S	692549	1715895	...	-082.8482619	Florida 0
103	S	749536	947118	...	-082.7367370	Florida 0
104	S	862800	808476	...	-082.7449430	Florida 0
105	S	1677445	28574	...	-082.6352023	Florida 0

STORMNAME	STORMTYPE	ADVDATE	ADVISNUM	STORMNUM	\
101	Ten	TD 400 PM CDT Sat Aug 26 2023	1	10.0	
102	Ten	TD 400 PM CDT Sat Aug 26 2023	1	10.0	
103	Ten	TD 400 PM CDT Sat Aug 26 2023	1	10.0	
104	Ten	TD 400 PM CDT Sat Aug 26 2023	1	10.0	
105	Ten	TD 400 PM CDT Sat Aug 26 2023	1	10.0	

FCSTPRD	BASIN
101	120.0 AL
102	120.0 AL
103	120.0 AL
104	120.0 AL
105	120.0 AL

[5 rows x 21 columns]

Locating Affected Regions

```
In [79]: import geopandas as gpd
import matplotlib.pyplot as plt

# Load the world boundaries shapefile (update path as needed)
world = gpd.read_file('world/ne_110m_admin_0_countries.shp')

# Example: Visualize affected tracts for a specific advisory (e.g., the first day)
adv = sorted(affected_daily.keys())[0] # Replace with desired advisory number
affected_gdf = affected_daily[adv]

fig, ax = plt.subplots(figsize=(16, 10))
world.plot(ax=ax, color='lightgrey', edgecolor='black', linewidth=0.5)

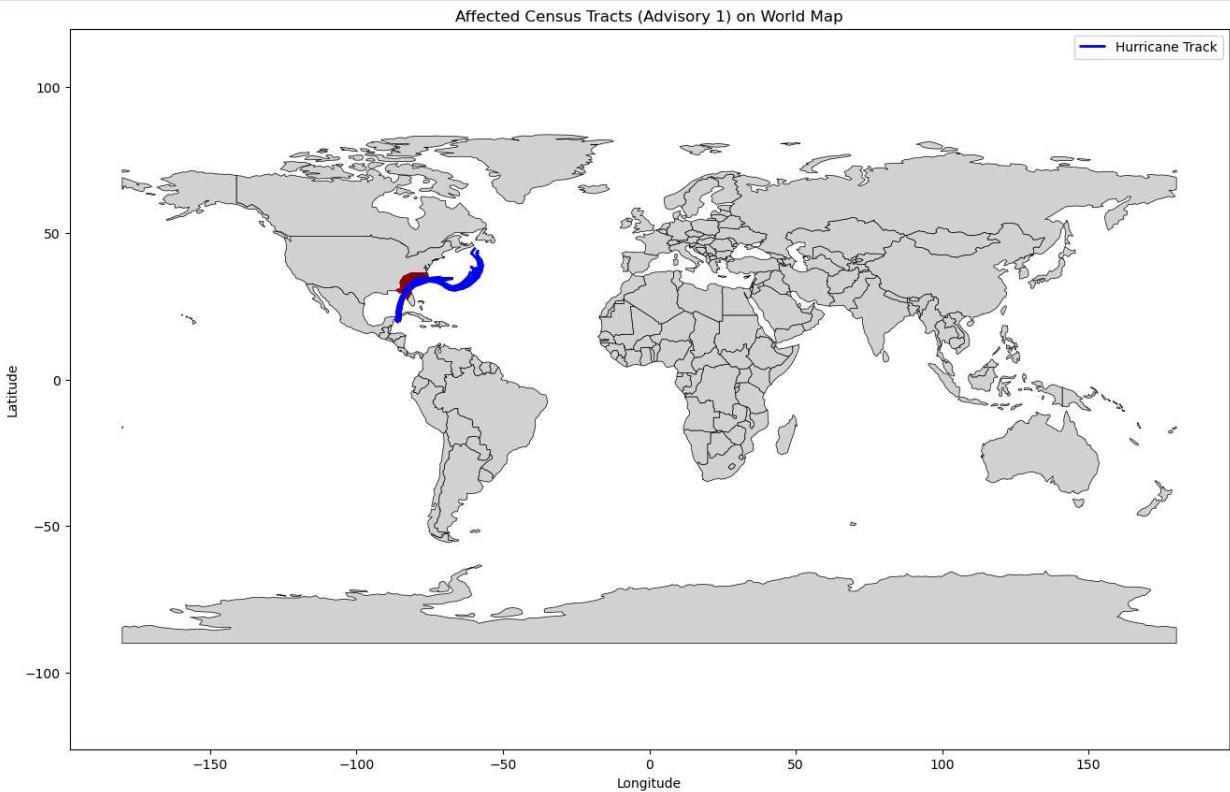
# Plot affected tracts
affected_gdf.plot(ax=ax, color='red', alpha=0.6, edgecolor='darkred', linewidth=0.3, )

# Optionally, overlay hurricane track or cone for context
if '_5day_lin' in combined_layers:
    combined_layers['_5day_lin'].plot(ax=ax, color='blue', linewidth=2, label='Hurricane')

plt.title(f'Affected Census Tracts (Advisory {adv}) on World Map')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.xlim(-100, -60) # Focus on the Western Atlantic/US region
plt.ylim(15, 40)
```

```
plt.axis('equal')
plt.show()
```

C:\Users\Abhiraj Akhouri\AppData\Local\Temp\ipykernel_12196\486793945.py:24: UserWarning: Legend does not support handles for PatchCollection instances.
See: https://matplotlib.org/stable/tutorials/intermediate/legend_guide.html#implementing-a-custom-legend-handler
plt.legend()



Demographic Data Download

```
In [86]: import requests
import pandas as pd

API_KEY = '2e5f752a20bdce7f5fd899d6c37f45d2c79e7216'

state_fips = {
    'Florida': '12',
    'Georgia': '13',
    'North Carolina': '37',
    'South Carolina': '45'
}

def read_variables(filename):
    with open(filename, 'r') as f:
        variables = [line.strip() for line in f if line.strip() and line.strip() != '']
    return variables

variables1 = read_variables('acs_variables.txt')
variables2 = read_variables('acs_variables2.txt')

def download_acs_data(state_name, state_code, variables, api_key):
    variables_api = ','.join(variables)
    url = (
```

```

f"https://api.census.gov/data/2022/acs/acs5"
f"?get={variables_api}&for=tract:{&in=state:{state_code}&key={api_key}}"
)
response = requests.get(url)
if response.status_code == 200:
    try:
        data = response.json()
        df = pd.DataFrame(data[1:], columns=data[0])
        # Reconstruct GEOID
        df['GEOID'] = df['state'] + df['county'] + df['tract']
        return df
    except Exception:
        print(f"Non-JSON response for {state_name}:\n{response.text[:500]}")
        return None
else:
    print(f"Failed for {state_name}: {response.status_code}")
    print(response.text[:500])
    return None

for state, fips in state_fips.items():
    print(f"Downloading data for {state}...")
    df1 = download_acs_data(state, fips, variables1, API_KEY)
    df2 = download_acs_data(state, fips, variables2, API_KEY)
    if df1 is not None and df2 is not None:
        # Merge on GEOID and geography fields (to avoid duplicate columns)
        merge_cols = ['GEOID', 'state', 'county', 'tract']
        merged = pd.merge(df1, df2, on=merge_cols, how='outer', suffixes=('', '_y'))
        # Remove duplicate columns from the second file
        merged = merged.loc[:, ~merged.columns.str.endswith('_y')]
        merged.to_csv(f"{state.replace(' ', '_})_acs5_2022.csv", index=False)
        print(f"Saved {state.replace(' ', '_})_acs5_2022.csv with {len(merged)} rows."
    else:
        print(f"Skipping {state} due to download error.")

```

Downloading data for Florida...
 Saved Florida_acs5_2022.csv with 5160 rows.
 Downloading data for Georgia...
 Saved Georgia_acs5_2022.csv with 2796 rows.
 Downloading data for North Carolina...
 Saved North_Carolina_acs5_2022.csv with 2672 rows.
 Downloading data for South Carolina...
 Saved South_Carolina_acs5_2022.csv with 1323 rows.

In [88]: `import pandas as pd`

```

# File paths for each state
file_paths = [
    'Florida_acs5_2022.csv',
    'Georgia_acs5_2022.csv',
    'North_Carolina_acs5_2022.csv',
    'South_Carolina_acs5_2022.csv'
]

# Read and combine all files
dfs = [pd.read_csv(fp, dtype=str) for fp in file_paths]
acs_df = pd.concat(dfs, ignore_index=True)

# If GEOID is not present, construct it
if 'GEOID' not in acs_df.columns:
    acs_df['GEOID'] = acs_df['state'] + acs_df['county'] + acs_df['tract']

```

```
print(acs_df.head())
```

	NAME	B01001_001E	B01001_003E	\			
0	Census Tract 2.01; Alachua County; Florida	4856	25				
1	Census Tract 2.02; Alachua County; Florida	5464	0				
2	Census Tract 3.01; Alachua County; Florida	4034	64				
3	Census Tract 3.02; Alachua County; Florida	2363	55				
4	Census Tract 4; Alachua County; Florida	5526	62				
	B01001_004E	B01001_005E	B01001_006E	B01001_007E	B01001_008E	B01001_009E	\
0	0	0	44	261	464	688	
1	0	11	15	130	486	312	
2	232	209	18	23	97	71	
3	55	0	15	176	0	0	
4	249	352	47	25	0	47	
	B01001_010E	...	B01001_048E	B01001_049E	B01002_001E	B01001_002E	\
0	327	...	0	9	21.5	2196	
1	612	...	0	0	20.9	1921	
2	124	...	0	51	26.7	2136	
3	0	...	0	37	37.4	1008	
4	0	...	106	0	29.7	2464	
	B01001_026E	B19013_001E	B17001_002E	B19301_001E	B16001_002E	B16004_020E	
0	2660	17727	3121	14313	NaN	0	
1	3543	13045	3072	9396	NaN	0	
2	1898	45227	914	23322	NaN	0	
3	1355	41868	321	27041	NaN	0	
4	3062	50417	1130	25533	NaN	0	

[5 rows x 60 columns]

In [90]: # Merge on GEOFID
eda_df = affected_gdf.merge(acs_df, on='GEOFID', how='left')

Demographics Successfully Integrated with Geo-position

In [100...]
age_labels = {
 # Male age groups
 'B01001_003E': 'M: <5',
 'B01001_004E': 'M: 5-9',
 'B01001_005E': 'M: 10-14',
 'B01001_006E': 'M: 15-17',
 'B01001_007E': 'M: 18-19',
 'B01001_008E': 'M: 20',
 'B01001_009E': 'M: 21',
 'B01001_010E': 'M: 22-24',
 'B01001_011E': 'M: 25-29',
 'B01001_012E': 'M: 30-34',
 'B01001_013E': 'M: 35-39',
 'B01001_014E': 'M: 40-44',
 'B01001_015E': 'M: 45-49',
 'B01001_016E': 'M: 50-54',
 'B01001_017E': 'M: 55-59',
 'B01001_018E': 'M: 60-61',
 'B01001_019E': 'M: 62-64',
 'B01001_020E': 'M: 65-66',
 'B01001_021E': 'M: 67-69',

```
'B01001_022E': 'M: 70-74',
'B01001_023E': 'M: 75-79',
'B01001_024E': 'M: 80-84',
'B01001_025E': 'M: 85+',
# Female age groups
'B01001_027E': 'F: <5',
'B01001_028E': 'F: 5-9',
'B01001_029E': 'F: 10-14',
'B01001_030E': 'F: 15-17',
'B01001_031E': 'F: 18-19',
'B01001_032E': 'F: 20',
'B01001_033E': 'F: 21',
'B01001_034E': 'F: 22-24',
'B01001_035E': 'F: 25-29',
'B01001_036E': 'F: 30-34',
'B01001_037E': 'F: 35-39',
'B01001_038E': 'F: 40-44',
'B01001_039E': 'F: 45-49',
'B01001_040E': 'F: 50-54',
'B01001_041E': 'F: 55-59',
'B01001_042E': 'F: 60-61',
'B01001_043E': 'F: 62-64',
'B01001_044E': 'F: 65-66',
'B01001_045E': 'F: 67-69',
'B01001_046E': 'F: 70-74',
'B01001_047E': 'F: 75-79',
'B01001_048E': 'F: 80-84',
'B01001_049E': 'F: 85+'
}
age_cols = list(age_labels.keys())
for col in age_cols:
    eda_df[col] = pd.to_numeric(eda_df[col], errors='coerce')

# Calculate summary statistics (e.g., sum or mean) for each age group
age_totals = eda_df[age_cols].sum()

# Rename the index to readable captions
age_totals.index = [age_labels.get(col, col) for col in age_totals.index]

# Display as a readable table
print("Affected Population by Age Group:")
print(age_totals)
```

Affected Population by Age Group:

```
M: <5           927670
M: 5-9          974173
M: 10-14        1069890
M: 15-17        635759
M: 18-19        463276
M: 20           237125
M: 21           233976
M: 22-24        650388
M: 25-29        1091505
M: 30-34        1064787
M: 35-39        1029103
M: 40-44        998389
M: 45-49        1003964
M: 50-54        1030893
M: 55-59        1029937
M: 60-61        412301
M: 62-64        576671
M: 65-66        353847
M: 67-69        483477
M: 70-74        701896
M: 75-79        455999
M: 80-84        262221
M: 85+          213896
F: <5           882541
F: 5-9           942799
F: 10-14         1012472
F: 15-17         609409
F: 18-19         446564
F: 20            224083
F: 21            220586
F: 22-24         618076
F: 25-29         1079322
F: 30-34         1087975
F: 35-39         1053594
F: 40-44         1026565
F: 45-49         1032589
F: 50-54         1057399
F: 55-59         1097441
F: 60-61         442633
F: 62-64         642585
F: 65-66         409480
F: 67-69         562693
F: 70-74         812936
F: 75-79         569693
F: 80-84         356799
F: 85+           378074
dtype: int64
```

In [105...]

```
import pandas as pd

# --- Gender ---
eda_df['pct_male'] = eda_df['B01001_002E'] / eda_df['B01001_001E'] * 100
eda_df['pct_female'] = eda_df['B01001_026E'] / eda_df['B01001_001E'] * 100

gender_summary = pd.DataFrame({
    'Gender': ['Male', 'Female'],
    'Mean Percentage (%)': [
        eda_df['pct_male'].mean(),
        eda_df['pct_female'].mean()
    ]
})
```

```

        ]
    })
print("### Gender Distribution\n")
print(gender_summary)

# --- Income and Poverty ---
income_cols = {
    'B19013_001E': 'Median Household Income',
    'B19301_001E': 'Per Capita Income',
    'B17001_002E': 'Population Below Poverty'
}
income_summary = eda_df[list(income_cols.keys())].describe().rename(columns=income_col)
print("\n### Income and Poverty Summary\n")
print(income_summary)

# --- Language Proficiency ---
eda_df['pct_english_only'] = eda_df['B16001_002E'] / eda_df['B01001_001E'] * 100
eda_df['pct_limited_english'] = eda_df['B16004_020E'] / eda_df['B01001_001E'] * 100

language_summary = pd.DataFrame({
    'Language Group': [
        'Speak Only English',
        'Speak English Less Than "Very Well"'
    ],
    'Mean Percentage (%)': [
        eda_df['pct_english_only'].mean(),
        eda_df['pct_limited_english'].mean()
    ]
})
print("\n### Language Proficiency\n")
print(language_summary)

```

Gender Distribution

	Gender	Mean Percentage (%)
0	Male	49.037001
1	Female	50.962999

Income and Poverty Summary

	Median Household Income	Per Capita Income	Population Below Poverty
count	8.294000e+03	8.290000e+03	8294.000000
mean	-1.190678e+07	-6.396280e+06	518.203882
std	8.856372e+07	6.518106e+07	419.170212
min	-6.666667e+08	-6.666667e+08	0.000000
25%	4.789250e+04	2.565200e+04	216.000000
50%	6.281950e+04	3.280000e+04	414.500000
75%	8.473175e+04	4.373300e+04	713.000000
max	2.500010e+05	2.314160e+05	3974.000000

Language Proficiency

	Language Group	Mean Percentage (%)
0	Speak Only English	NaN
1	Speak English Less Than "Very Well"	0.09708

Important Note: Negative values present due invalid entries coded as -66666 etc. need to be removed, language data is corrupt

In [135...]

```
import pandas as pd

# --- Add derived percentage columns ---
eda_df['pct_male'] = eda_df['B01001_002E'] / eda_df['B01001_001E'] * 100
eda_df['pct_female'] = eda_df['B01001_026E'] / eda_df['B01001_001E'] * 100
eda_df['pct_english_only'] = eda_df['B16001_002E'] / eda_df['B01001_001E'] * 100
eda_df['pct_limited_english'] = eda_df['B16004_020E'] / eda_df['B01001_001E'] * 100

# --- Group by State ---
grouped = eda_df.groupby('State')

# --- Age Distribution by State ---
age_summary = grouped[age_cols].sum().rename(columns=age_labels)
print("Age Distribution by State")
print(age_summary.to_string())

# --- Gender Ratio by State ---
gender_summary = grouped[['pct_male', 'pct_female']].mean().rename(
    columns={'pct_male': '% Male', 'pct_female': '% Female'}
)
print("\nGender Ratio by State")
print(gender_summary.to_string())

# --- Income and Poverty by State ---
income_summary = grouped[list(income_cols.keys())].mean().rename(columns=income_cols)
print("\nIncome and Poverty by State")
print(income_summary.to_string())

# --- Language Proficiency by State ---
language_summary = grouped[['pct_english_only', 'pct_limited_english']].mean().rename(
    columns={
        'pct_english_only': '% Speak Only English',
        'pct_limited_english': '% Speak English Less Than \"Very Well\"'
    }
)
print("\nLanguage Proficiency by State")
print(language_summary.to_string())
```

Age Distribution by State

M: <5 M: 5-9 M: 10-14 M: 15-17 M: 18-19 M: 20 M: 21 M: 22-24
 M: 25-29 M: 30-34 M: 35-39 M: 40-44 M: 45-49 M: 50-54 M: 55-59 M: 60-61 M: 62-64 M: 65-66 M: 67-69 M: 70-74 M: 75-79 M: 80-84 M: 85+ F: <5 F: 5-9 F: 10-14 F: 15-17 F: 18-19 F: 20 F: 21 F: 22-24 F: 25-29 F: 30-34 F: 35-39 F: 40-44 F: 45-49 F: 50-54 F: 55-59 F: 60-61 F: 62-64 F: 65-66 F: 67-69 F: 70-74 F: 75-79 F: 80-84 F: 85+

State

Florida	179277	185614	205504	119913	88149	45862	47599	130192
227968	227238	219001	203938	205840	216276	225447	95412	13445
2	86423	120913	182951	133297	79154	66348	172450	180303
113415	87712	47485	47098	126496	218153	221887	213413	201300
05347	217888	245280	98735	148757	98337	137846	206817	154921
97994	105295							
Georgia	300867	320913	353920	210885	144813	74741	71076	207789
345236	333094	325542	316109	316145	317550	308787	117650	15858
2	96332	125615	175727	108179	60872	48915	284932	307870
203366	140581	69770	69894	195882	345470	346796	342247	333328
31578	328592	327060	126823	177470	112062	150246	208356	141969
88545	91407							
North Carolina	301736	314522	343748	206080	153552	78349	76184	215934
353086	341733	328498	323575	330093	338863	332993	133456	18504
5	112538	152280	220206	139525	77905	65116	287668	303494
198382	146368	72776	69118	201152	348899	349969	333512	335237
37930	345723	349773	142362	207711	128476	177143	258164	177190
112834	122596							
South Carolina	145790	153124	166718	98881	76762	38173	39117	96473
165215	162722	156062	154767	151886	158204	162710	65783	9859
2	58554	84669	123012	74998	44290	33517	137491	151132
94246	71903	34052	34476	94546	166800	169323	164422	156700
7734	165196	175328	74713	108647	70605	97458	139599	95613
57426	58776							

Gender Ratio by State

	% Male	% Female
State		
Florida	49.227443	50.772557
Georgia	48.891238	51.108762
North Carolina	49.059135	50.940865
South Carolina	49.037277	50.962723

```

eda_df['pct_male'] = eda_df['B01001_002E'] / eda_df['B01001_001E'] * 100
eda_df['pct_female'] = eda_df['B01001_026E'] / eda_df['B01001_001E'] * 100
eda_df['pct_english_only'] = eda_df['B16001_002E'] / eda_df['B01001_001E'] * 100
eda_df['pct_limited_english'] = eda_df['B16004_020E'] / eda_df['B01001_001E'] * 100

# --- Group by State ---
grouped = eda_df.groupby('State')

# --- Gender Ratio by State ---
gender_summary = grouped[['pct_male', 'pct_female']].mean().rename(
    columns={'pct_male': '% Male', 'pct_female': '% Female'}
)
print("\nGender Ratio by State")
print(gender_summary.to_string())

```

	% Male	% Female
State		
Florida	49.227443	50.772557
Georgia	48.891238	51.108762
North Carolina	49.059135	50.940865
South Carolina	49.037277	50.962723

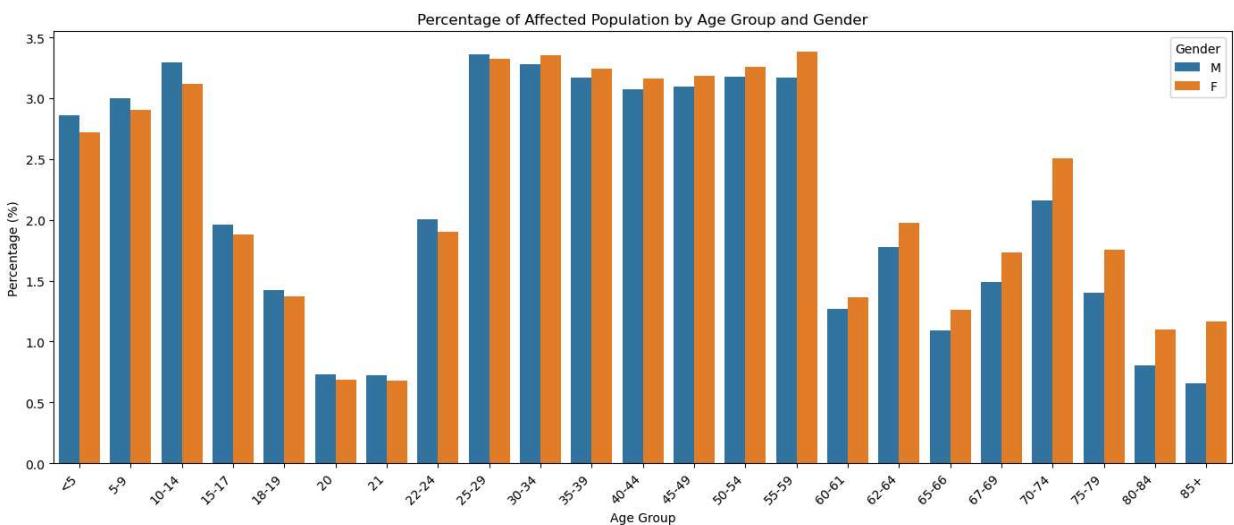
Key Insight:

Female population is higher than male

```

In [111...]: plt.figure(figsize=(14, 6))
sns.barplot(
    data=age_df,
    x='Age',
    y='Percentage',
    hue='Gender',
    order=sorted(age_df['Age'].unique(), key=lambda x: int(x.split('-')[0].replace('<', '')))
)
plt.xticks(rotation=45, ha='right')
plt.title('Percentage of Affected Population by Age Group and Gender')
plt.ylabel('Percentage (%)')
plt.xlabel('Age Group')
plt.legend(title='Gender')
plt.tight_layout()
plt.show()

```



Key Insight:

Adults are equally-distributed & most populous age-group with higher Females in affected regions

In [118...]

```
import numpy as np

# Calculate the average of non-negative values in the column
avg_value = eda_df.loc[eda_df['B01002_001E'] >= 0, 'B01002_001E'].mean()

# Replace negative values with the average (column is cast to float automatically)
eda_df.loc[eda_df['B01002_001E'] < 0, 'B01002_001E'] = avg_value
```

Median Age by Region

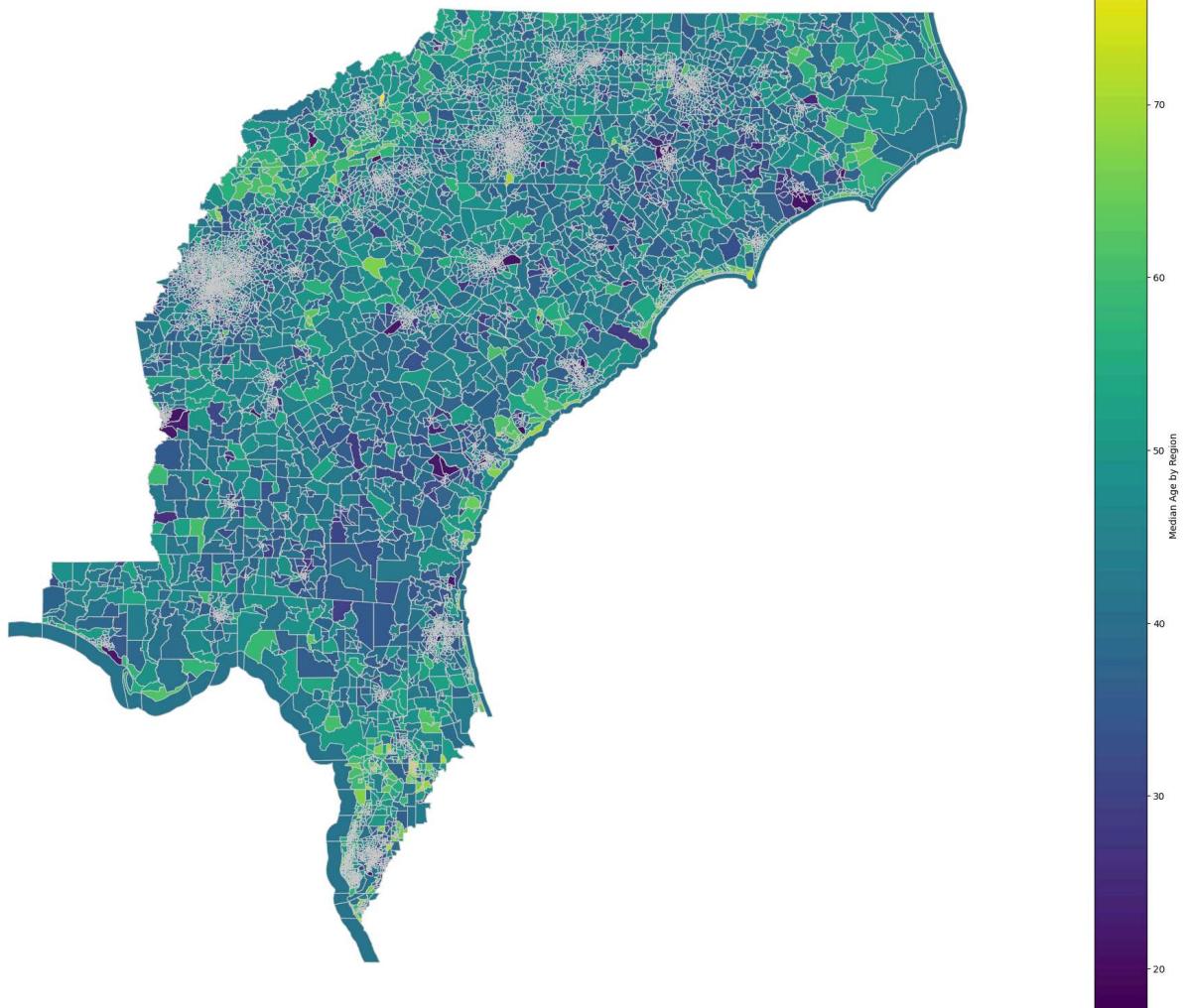
Note: Cone layer used for Hurricane visual

In [124...]

```
import matplotlib.pyplot as plt

# Plotting the median age choropleth with a larger display
fig, ax = plt.subplots(1, 1, figsize=(30, 20)) # Increased from (15, 8) to (20, 12)
eda_df.plot(
    column='B01002_001E',
    cmap='viridis',
    linewidth=0.8,
    ax=ax,
    edgecolor='0.8',
    legend=True,
    legend_kwds={'label': "Median Age by Region", 'orientation': "vertical"}
)
ax.set_title('Median Age by Geographic Region')
ax.axis('off')
plt.show()
```

Median Age by Geographic Region



In [129...]

```
#B19301_001E
import numpy as np

# Calculate the average of non-negative values in the column
avg_value = eda_df.loc[eda_df['B19013_001E'] >= 0, 'B19013_001E'].mean()

# Replace negative values with the average (column is cast to float automatically)
eda_df.loc[eda_df['B19013_001E'] < 0, 'B19013_001E'] = avg_value
```

Average Income by Region

Note: Cone layer used for Hurricane visual

In [130...]

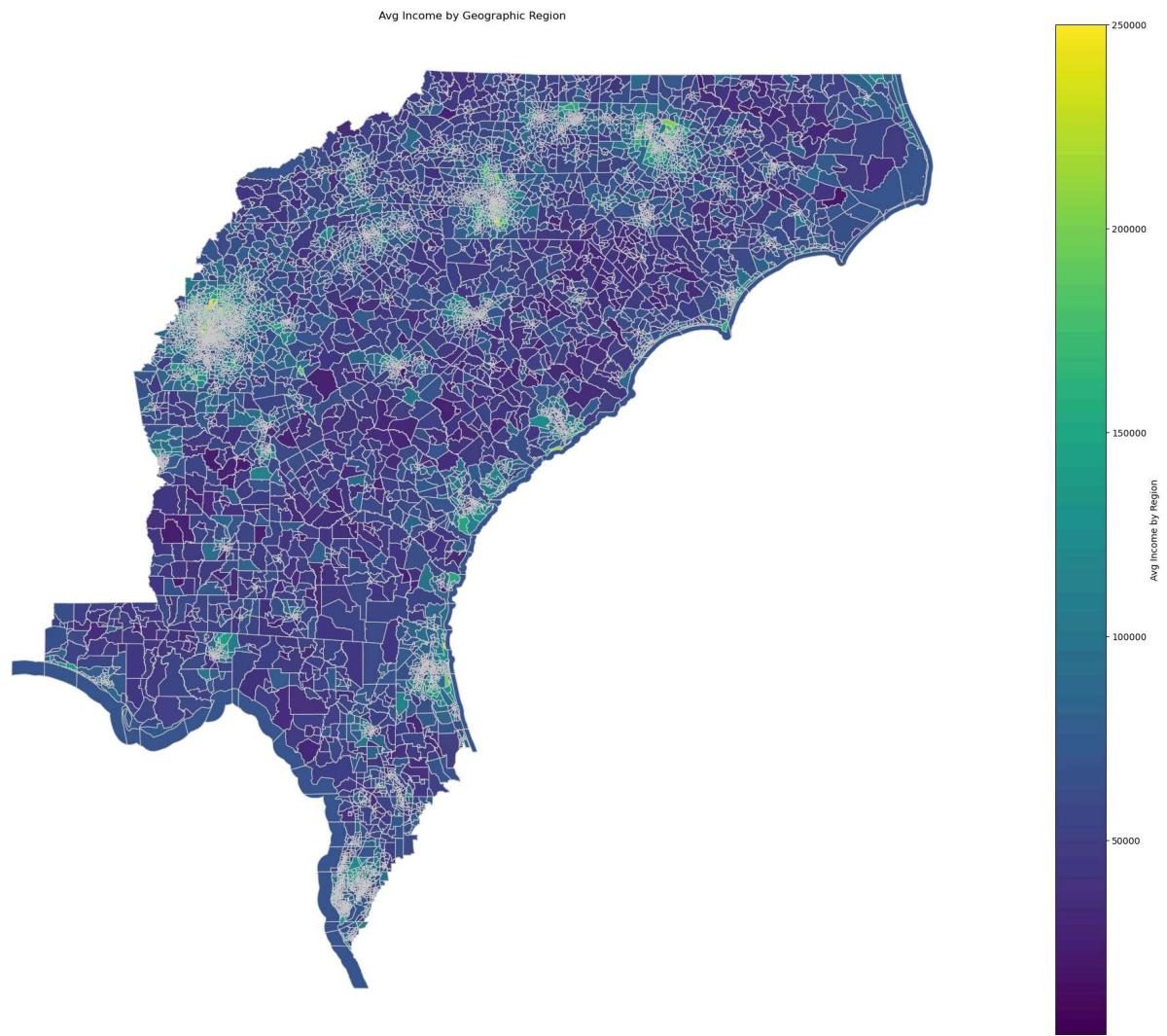
```
import matplotlib.pyplot as plt

# Plotting the avg income choropleth with a larger display
fig, ax = plt.subplots(1, 1, figsize=(30, 20))
eda_df.plot(
    column='B19013_001E',
    cmap='viridis',
    linewidth=0.8,
    ax=ax,
    edgecolor='0.8',
    legend=True,
```

```

        legend_kwds={'label': "Avg Income by Region", 'orientation': "vertical"}
    )
ax.set_title('Avg Income by Geographic Region')
ax.axis('off')
plt.show()

```



In [132...]

```

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# --- Income and Poverty ---
income_cols = {
    'B19013_001E': 'Median Household Income',
    'B19301_001E': 'Per Capita Income',
    'B17001_002E': 'Population Below Poverty'
}

# 1. Replace negative values in income columns with column average
eda_df_clean = eda_df.copy()
for col in income_cols.keys():
    avg = eda_df_clean.loc[eda_df_clean[col] >= 0, col].mean()
    eda_df_clean[col] = eda_df_clean[col].astype(float)
    eda_df_clean.loc[eda_df_clean[col] < 0, col] = avg

# 2. Add state names using FIPS code (first two digits of GEOID)
state_lookup = {
    '01': 'Alabama',
    '02': 'Alaska',
    '04': 'Arizona',
    '05': 'Arkansas',
    '06': 'California',
    '08': 'Colorado',
    '09': 'Connecticut',
    '10': 'Delaware',
    '12': 'Florida',
    '13': 'Georgia',
    '15': 'Hawaii',
    '16': 'Idaho',
    '17': 'Illinois',
    '18': 'Indiana',
    '19': 'Iowa',
    '20': 'Kansas',
    '21': 'Kentucky',
    '22': 'Louisiana',
    '23': 'Maine',
    '24': 'Maryland',
    '25': 'Massachusetts',
    '26': 'Michigan',
    '27': 'Minnesota',
    '28': 'Mississippi',
    '29': 'Missouri',
    '30': 'Montana',
    '31': 'Nebraska',
    '32': 'Nevada',
    '33': 'New Hampshire',
    '34': 'New Jersey',
    '35': 'New Mexico',
    '36': 'New York',
    '37': 'North Carolina',
    '38': 'North Dakota',
    '39': 'Ohio',
    '40': 'Oklahoma',
    '41': 'Oregon',
    '42': 'Pennsylvania',
    '43': 'Rhode Island',
    '44': 'South Carolina',
    '45': 'South Dakota',
    '46': 'Tennessee',
    '47': 'Texas',
    '48': 'Utah',
    '49': 'Vermont',
    '50': 'Virginia',
    '51': 'Washington',
    '52': 'West Virginia',
    '53': 'Wisconsin',
    '54': 'Wyoming'
}

```

```

'12': 'Florida',
'45': 'South Carolina',
'37': 'North Carolina',
'13': 'Georgia'
}
# If GEOFID is not a string, convert it
eda_df_clean['STATE_FIPS'] = eda_df_clean['GEOFID'].astype(str).str[:2]
eda_df_clean['STATE_NAME'] = eda_df_clean['STATE_FIPS'].map(state_lookup)

# 3. Plot 1: Overall distributions
plt.figure(figsize=(16, 5))
for i, col in enumerate(income_cols.keys(), 1):
    plt.subplot(1, 3, i)
    sns.histplot(eda_df_clean[col], kde=True, bins=30)
    plt.title(f'Distribution of {income_cols[col]}')
    plt.xlabel(income_cols[col])
    plt.ylabel('Frequency')
plt.tight_layout()
plt.show()

# 4. Plot 2: Bar plots by selected region
regions = ['Florida', 'South Carolina', 'North Carolina', 'Georgia']
eda_df_regions = eda_df_clean[eda_df_clean['STATE_NAME'].isin(regions)]
```

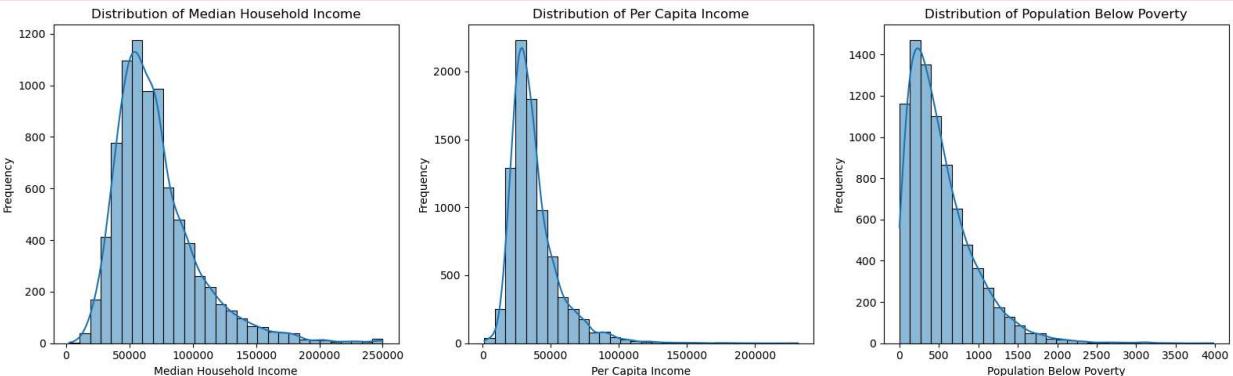
```

plt.figure(figsize=(12, 6))
for i, col in enumerate(income_cols.keys(), 1):
    plt.subplot(1, 3, i)
    sns.barplot(x='STATE_NAME', y=col, data=eda_df_regions, ci=None)
    plt.title(f'{income_cols[col]} by Region')
    plt.xlabel('Region')
    plt.ylabel(income_cols[col])
    plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\seaborn_oldcore.py:1119: Future Warning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.

```

with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: Future Warning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Abhiraj Akhouri\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: Future Warning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):
```



```
C:\Users\Abhiraj Akhouri\AppData\Local\Temp\ipykernel_12196\1161099613.py:48: FutureWarning:
```

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(x='STATE_NAME', y=col, data=eda_df_regions, ci=None)
```

```
C:\Users\Abhiraj Akhouri\AppData\Local\Temp\ipykernel_12196\1161099613.py:48: FutureWarning:
```

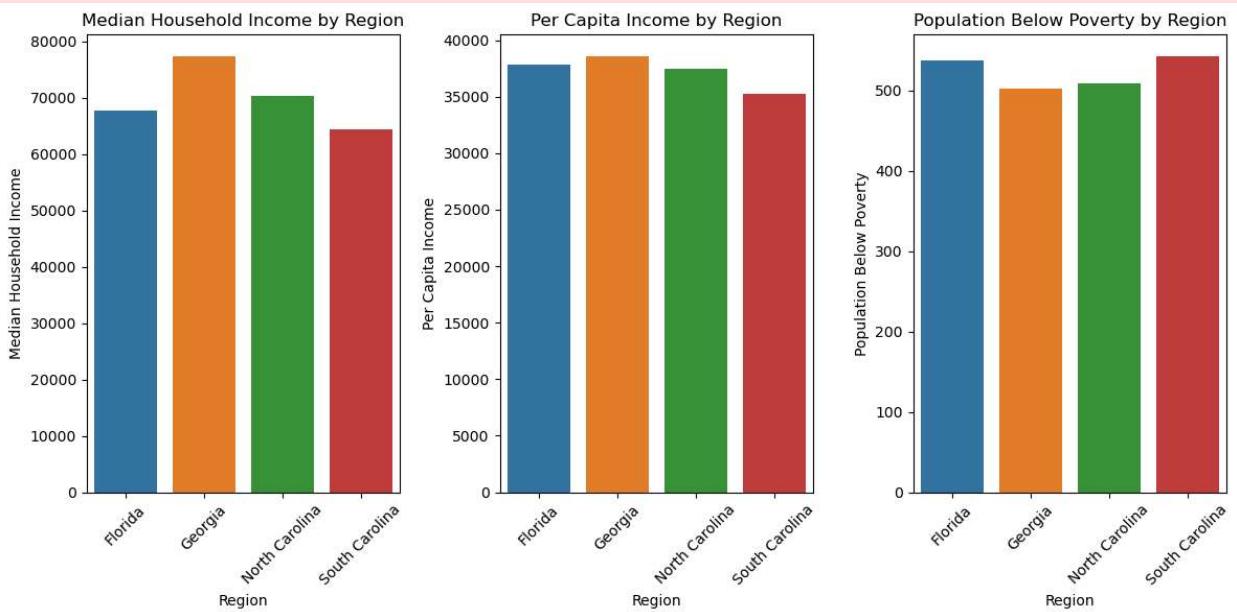
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(x='STATE_NAME', y=col, data=eda_df_regions, ci=None)
```

```
C:\Users\Abhiraj Akhouri\AppData\Local\Temp\ipykernel_12196\1161099613.py:48: FutureWarning:
```

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(x='STATE_NAME', y=col, data=eda_df_regions, ci=None)
```



Key Insight:

Georgian-affected regions pose better income & less poverty, poverty differences are less pronounced than income.

Floridian-affected regions, to surprise, are lesser income WHEN Florida's GDP is 2.5x Georgia's (15th largest global GDP region, 4th US wise)