

Geographic Data Visualization with Basemap

The **Basemap** toolkit in Matplotlib is used for plotting geographic data. It allows you to create high-quality maps by plotting data points, visualizing paths, and rendering geographical boundaries.

Syntax for Basemap

```
from mpl_toolkits.basemap import Basemap
Key Methods in Basemap
```

1. **Basemap()**: Initializes the map with specific projection and geographic boundaries.
 - **projection**: Type of map projection (e.g., 'merc', 'ortho', 'cyl').
 - **llcrnrlat, urcrnrlat**: Latitude bounds.
 - **llcrnrlon, urcrnrlon**: Longitude bounds.
 - **resolution**: Map detail level ('c' for crude, 'l' for low, 'h' for high).
2. **Drawing Features**:
 - **drawcoastlines()**: Draws coastlines.
 - **drawcountries()**: Draws country boundaries.
 - **drawmapboundary()**: Draws a boundary for the map.
 - **fillcontinents()**: Fills continents with color.
 - **drawparallels()**, **drawmeridians()**: Adds latitude and longitude lines.
3. **Plotting Points**:
 - **map.plot()**: Plots points or paths.
 - **map.scatter()**: Plots points on the map.

Installation of Basemap

Basemap needs to be installed separately as it is not included in Matplotlib by default. You can install it with:

```
pip install basemap
```

Example: Visualizing Geographic Data with Basemap

```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt

# Create a map with Mercator projection
plt.figure(figsize=(12, 8))
map = Basemap(projection='merc', llcrnrlat=-60, urcrnrlat=80,
              llcrnrlon=-180, urcrnrlon=180, resolution='c')

# Draw map features
```

```

map.drawcoastlines()
map.drawcountries()
map.fillcontinents(color='lightgreen', lake_color='lightblue')
map.drawmapboundary(fill_color='lightblue')
map.drawparallels(range(-90, 90, 30), labels=[1, 0, 0, 0])
map.drawmeridians(range(-180, 180, 60), labels=[0, 0, 0, 1])

# Add some data points
lats = [37.7749, 48.8566, -33.8688] # Latitudes (San Francisco, Paris, Sydney)
lons = [-122.4194, 2.3522, 151.2093] # Longitudes
map.scatter(lons, lats, latlon=True, s=100, color='red', marker='o', label="Cities")

# Add title and legend
plt.title("Geographic Data with Basemap")
plt.legend(loc='lower left')
plt.show()

```

1. Initializing the Map

- **projection='merc'**: Uses Mercator projection.
- **llcrnrlat, urcrnrlat, llcrnrlon, urcrnrlon**:
 - Defines the latitude and longitude bounds for the map.
 - Covers the globe from -60° to 80° latitude and -180° to 180° longitude.
- **resolution='c'**: Sets the map detail level to crude.

2. Drawing Map Features

- **drawcoastlines()**: Adds coastlines to the map.
- **drawcountries()**: Draws country borders.
- **fillcontinents()**: Colors the continents with lightgreen.
- **drawmapboundary()**: Adds a blue boundary around the map to represent water.
- **drawparallels()** and **drawmeridians()**:
 - Adds grid lines for latitude and longitude at intervals of 30° and 60° .

3. Plotting Geographic Points

- **lats and lons**:
 - Stores the latitude and longitude of cities (San Francisco, Paris, Sydney).
- **map.scatter(lons, lats, latlon=True)**:
 - Plots red markers at the specified coordinates.

4. Title and Legend

- **plt.title()**: Adds a title to the map.
- **plt.legend()**: Creates a legend for the plotted data points.

- A global map with:
 - Green continents and blue water.
 - Latitude and longitude grid lines.
 - Red markers showing the positions of San Francisco, Paris, and Sydney.
 - A legend for the markers.

