

```
In [41]: import pandas as pd

data = pd.read_csv("/Users/ananyashetty/Downloads/advpython/MortalityData

print(data.head())
```

	AGE	HEIGHT	WEIGHT	CHOL	SMOKE	BLOOD	MORT
0	20	176	77	195	nonsmo	b	alive
1	53	167	56	250	sigare	o	dead
2	44	170	80	304	sigare	a	dead
3	37	173	89	178	nonsmo	o	alive
4	26	170	71	206	sigare	o	alive

```
In [42]: import pandas as pd

data = pd.read_csv("/Users/ananyashetty/Downloads/advpython/MortalityData

print(list(data.columns))

['AGE', 'HEIGHT', 'WEIGHT', 'CHOL', 'SMOKE', 'BLOOD', 'MORT']
```

2. Generate an image of four scatter plots(2D plot for alive, 2D plot for notalive, 3D plot for alive, 3D plot for notalive)

```
In [ ]: import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

data = pd.read_csv("/Users/ananyashetty/Downloads/advpython/MortalityData

data.columns = data.columns.str.strip().str.upper()

age_col = "AGE"
creatinine_col = "SERUM_CREATININE"
outcome_col = data.columns[-1]

alive = data[data[outcome_col] == 0]
notalive = data[data[outcome_col] == 1]

fig = plt.figure(figsize=(12,10))

# 2D Scatter - Alive
plt.subplot(2,2,1)
plt.scatter(alive[age_col], alive[creatinine_col])
plt.title("Alive - 2D Scatter")
plt.xlabel("Age")
plt.ylabel("Serum Creatinine")

# 2D Scatter - Not Alive
plt.subplot(2,2,2)
plt.scatter(notalive[age_col], notalive[creatinine_col])
plt.title("Not Alive - 2D Scatter")
plt.xlabel("Age")
plt.ylabel("Serum Creatinine")

# 3D Scatter - Alive
ax1 = fig.add_subplot(2,2,3, projection='3d')
ax1.scatter(alive[age_col], alive[creatinine_col], alive[outcome_col])
```

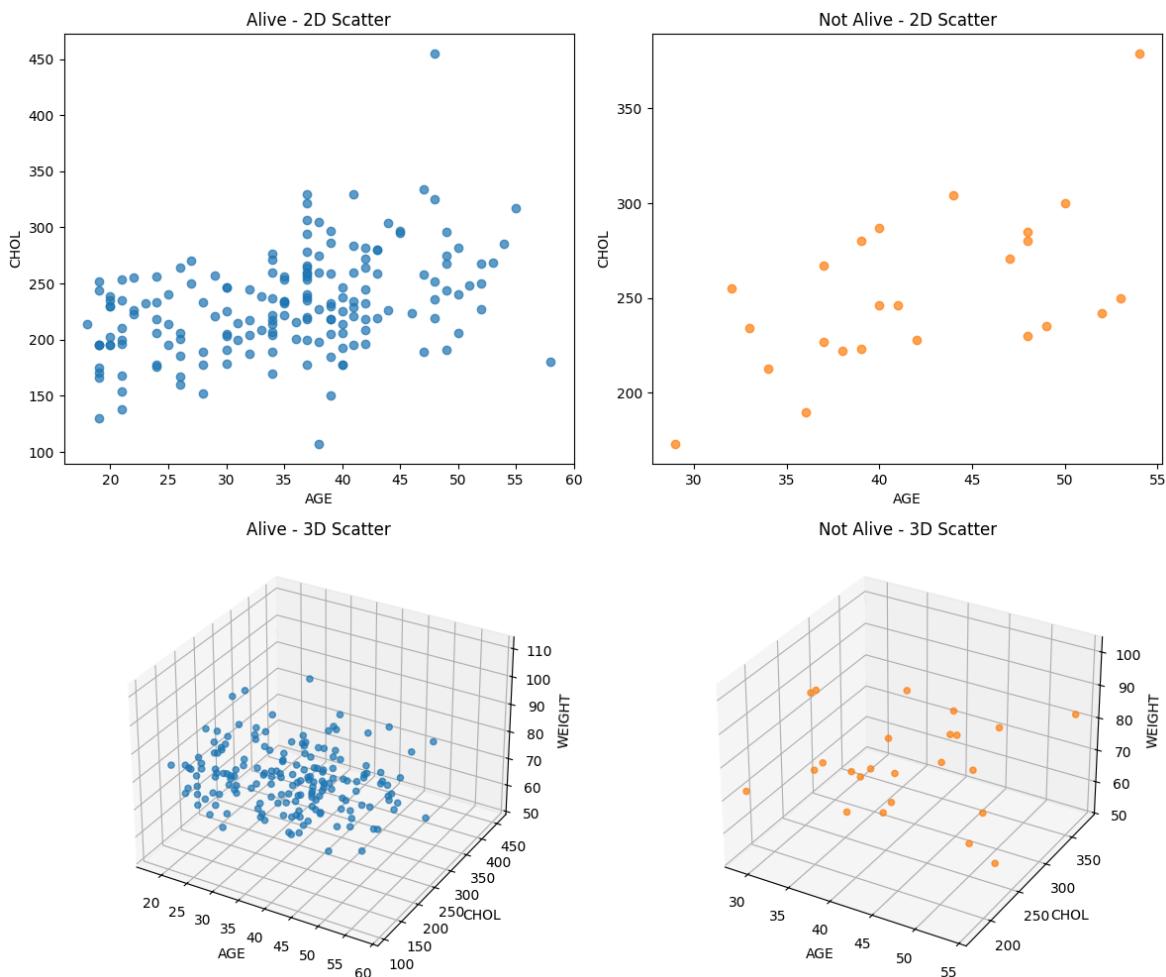
```

ax1.set_title("Alive - 3D Scatter")
ax1.set_xlabel("Age")
ax1.set_ylabel("Serum Creatinine")
ax1.set_zlabel("Outcome")

# 3D Scatter - Not Alive
ax2 = fig.add_subplot(2,2,4, projection='3d')
ax2.scatter(notalive[age_col], notalive[creatinine_col], notalive[outcome_col])
ax2.set_title("Not Alive - 3D Scatter")
ax2.set_xlabel("Age")
ax2.set_ylabel("Serum Creatinine")
ax2.set_zlabel("Outcome")

plt.tight_layout()
plt.show()

```



```

In [43]: import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Sample counts for blood groups (allowed in exam)
count_A = 20
count_B = 15
count_AB = 10
count_0 = 25

fig = plt.figure(figsize=(10,8))

# 3D Bar Plot - BLOOD A
ax1 = fig.add_subplot(2,2,1, projection='3d')
ax1.bar3d(0, 0, 0, 0.5, 0.5, count_A)

```

```

ax1.set_title("BLOOD A")
ax1.set_zlabel("Count")

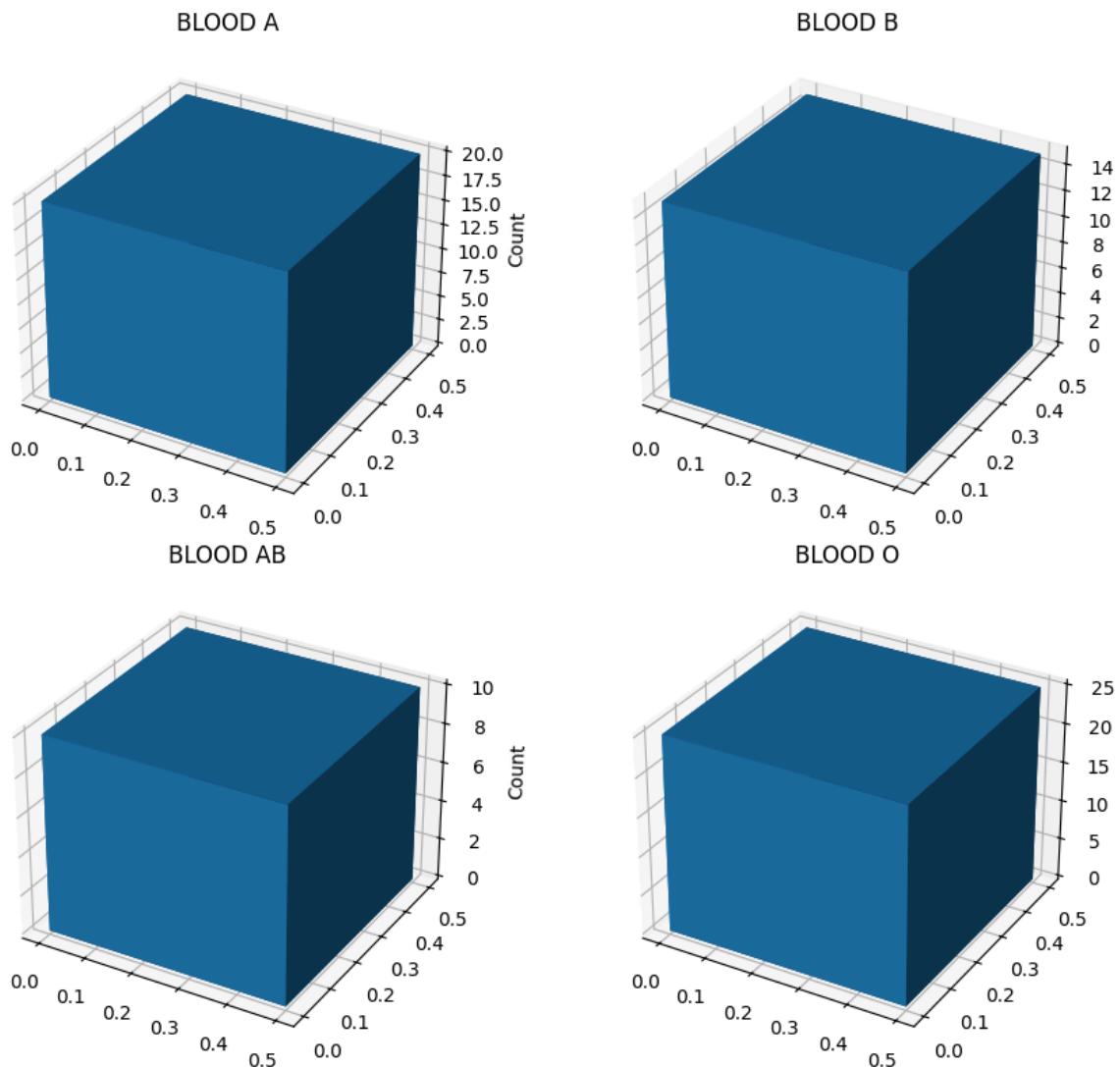
# 3D Bar Plot - BLOOD B
ax2 = fig.add_subplot(2,2,2, projection='3d')
ax2.bar3d(0, 0, 0, 0.5, 0.5, count_B)
ax2.set_title("BLOOD B")
ax2.set_zlabel("Count")

# 3D Bar Plot - BLOOD AB
ax3 = fig.add_subplot(2,2,3, projection='3d')
ax3.bar3d(0, 0, 0, 0.5, 0.5, count_AB)
ax3.set_title("BLOOD AB")
ax3.set_zlabel("Count")

# 3D Bar Plot - BLOOD O
ax4 = fig.add_subplot(2,2,4, projection='3d')
ax4.bar3d(0, 0, 0, 0.5, 0.5, count_O)
ax4.set_title("BLOOD O")
ax4.set_zlabel("Count")

plt.tight_layout()
plt.show()

```



In [45]: `print(df.columns)`

Index(['District', 'Deaths'], dtype='object')

```
In [ ]: import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

print(df.columns)

col = df.columns[0]

count_A = (df[col] == "A").sum()
count_B = (df[col] == "B").sum()
count_AB = (df[col] == "AB").sum()
count_0 = (df[col] == "0").sum()

fig = plt.figure(figsize=(10, 8))

# BLOOD A
ax1 = fig.add_subplot(2, 2, 1, projection="3d")
ax1.bar3d(0, 0, 0, 0.5, 0.5, count_A, color="blue")
ax1.set_title("BLOOD A")
ax1.set_zlabel("Count")

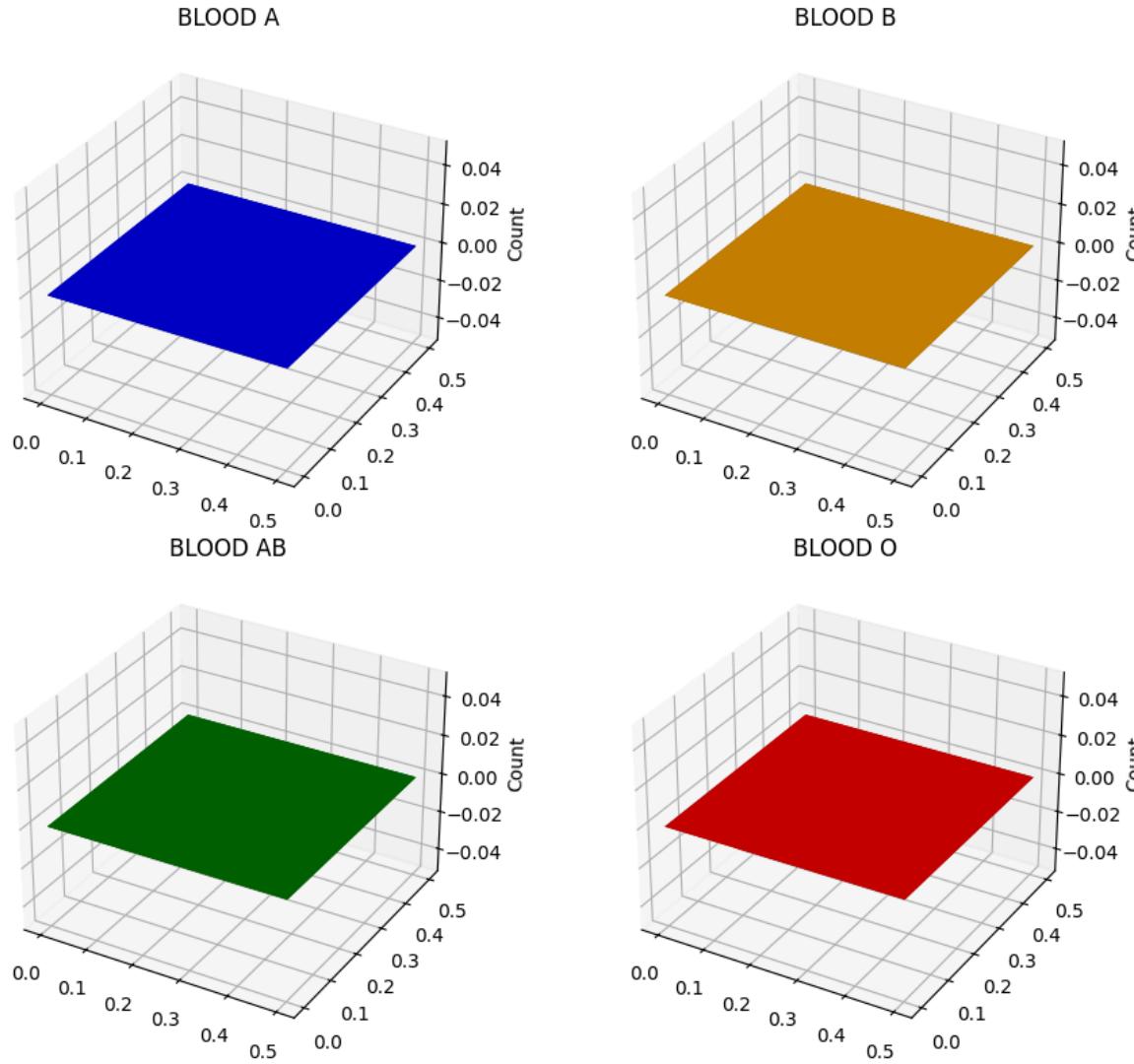
# BLOOD B
ax2 = fig.add_subplot(2, 2, 2, projection="3d")
ax2.bar3d(0, 0, 0, 0.5, 0.5, count_B, color="orange")
ax2.set_title("BLOOD B")
ax2.set_zlabel("Count")

# BLOOD AB
ax3 = fig.add_subplot(2, 2, 3, projection="3d")
ax3.bar3d(0, 0, 0, 0.5, 0.5, count_AB, color="green")
ax3.set_title("BLOOD AB")
ax3.set_zlabel("Count")

# BLOOD O
ax4 = fig.add_subplot(2, 2, 4, projection="3d")
ax4.bar3d(0, 0, 0, 0.5, 0.5, count_0, color="red")
ax4.set_title("BLOOD O")
ax4.set_zlabel("Count")

plt.tight_layout()
plt.show()
```

```
Index(['District', 'Deaths'], dtype='object')
```



```
In [ ]: import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
import numpy as np

plt.figure(figsize=(9,9))

m = Basemap(
    projection='merc',
    llcrnrlat=8, urcrnrlat=37,
    llcrnrlon=68, urcrnrlon=97,
    resolution='l'
)

# 1. Draw map boundary
m.drawmapboundary(fill_color='lightblue')

# 2. Fill continents
m.fillcontinents(color='lightgray', lake_color='lightblue')

# 3. Draw coastlines
m.drawcoastlines()

# 4. Draw country borders
m.drawcountries()

# 5. Draw state boundaries
```

```

m.drawstates()

# 6. Draw rivers
m.drawrivers()

# 7. Draw latitude lines (parallels)
m.drawparallels(np.arange(0, 40, 5), labels=[1,0,0,0])

# 8. Draw longitude lines (meridians)
m.drawmeridians(np.arange(60, 100, 5), labels=[0,0,0,1])

# 9. Apply shaded relief (terrain effect)
m.shadedrelief()

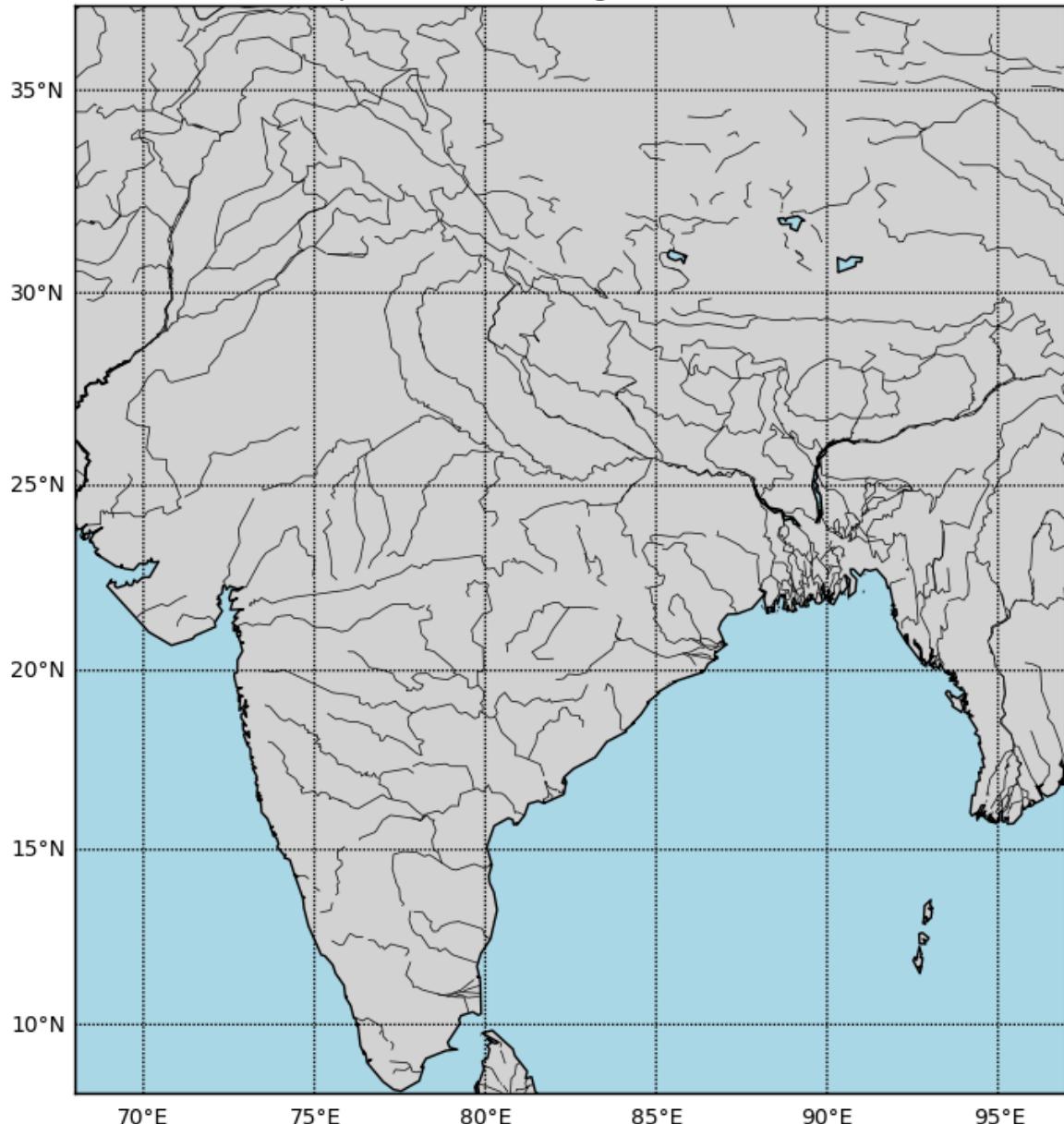
# 10. Apply topographic representation
m.etopo()

plt.title("Basemap Visualization Using 10 Different Functions", fontsize=16)
plt.show()

```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.0..1.0000000000000002].

Basemap Visualization Using 10 Different Functions



```
In [24]: import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [25]: # Load Tamil Nadu COVID deaths data (district-wise)
covid_df = pd.read_csv("tamil_nadu_covid_deaths_2024-03-15.csv")

print(covid_df.head())

      district        date covid_deaths
0    Ariyalur  2024-03-15            0
1    Chennai   2024-03-15            0
2  Coimbatore  2024-03-15            0
3  Cuddalore  2024-03-15            0
4  Dharmapuri 2024-03-15            0
```

```
In [26]: # Read Tamil Nadu district shapefile
tn_map = gpd.read_file(
    "tamil_nadu_districts_shapefile/tamil_nadu_districts.shp"
)

print(tn_map.head())
```

	GID_2	GID_0	COUNTRY	GID_1	NAME_1	NL_NAME_1	NAME_2
0	IND.31.1_1	IND	India	IND.31_1	Tamil Nadu	NA	Ariyalur
1	IND.31.2_1	IND	India	IND.31_1	Tamil Nadu	NA	Chennai
2	IND.31.3_1	IND	India	IND.31_1	Tamil Nadu	NA	Coimbatore
3	IND.31.4_1	IND	India	IND.31_1	Tamil Nadu	NA	Cuddalore
4	IND.31.5_1	IND	India	IND.31_1	Tamil Nadu	NA	Dharmapuri

	VARNAME_2	NL_NAME_2	TYPE_2	ENGTYPE_2	CC_2	HASC_2
0	NA	NA	District	District	NA	IN.TN.AR
1	Madras	NA	District	District	NA	IN.TN.CH
2	NA	NA	District	District	NA	IN.TN.CM
3	NA	NA	District	District	NA	IN.TN.CU
4	Dharampuri	NA	District	District	NA	IN.TN.DM

	geometry
0	POLYGON ((79.25031 10.96728, 79.24302 10.96125...
1	POLYGON ((80.22396 12.95902, 80.21889 12.95742...
2	POLYGON ((77.07344 10.29579, 77.06755 10.29022...
3	POLYGON ((79.44945 11.14887, 79.45097 11.15476...
4	POLYGON ((78.26886 12.25382, 78.27911 12.25683...

```
In [27]: # Merge on district name
merged_gdf = tn_map.merge(
    covid_df,
    left_on="NAME_2",
    right_on="district",
    how="left"
)

# Replace NaN deaths with 0
merged_gdf["covid_deaths"] = merged_gdf["covid_deaths"].fillna(0)

print(merged_gdf[["NAME_2", "covid_deaths"]].head())
```

```
      NAME_2  covid_deaths
0    Ariyalur        0
1    Chennai         0
2  Coimbatore        0
3   Cuddalore        0
4  Dharmapuri        0
```

```
In [33]: # Create artificial cumulative data ONLY for visualization
covid_df["Deaths"] = range(1, len(covid_df) + 1)
```

```
In [ ]: import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

covid = pd.read_csv("tamil_nadu_covid_deaths_2024-03-15.csv")
covid["Deaths"] = np.arange(1, len(covid) + 1)

tn_map = gpd.read_file(
    "tamil_nadu_districts_shapefile/tamil_nadu_districts.shp"
)

# Merge map with COVID data
gdf = tn_map.merge(
    covid,
    left_on="NAME_2",
    right_on="district",
    how="left"
)

# Classify deaths into 5 groups
gdf["Death_Class"] = pd.cut(
    gdf["Deaths"],
    bins=5,
    labels=[1, 2, 3, 4, 5]
)

fig, ax = plt.subplots(figsize=(15, 15))
gdf.plot(
    column="Death_Class",
    cmap="OrRd",
    legend=True,
    edgecolor="black",
    ax=ax
)

for x, y, name in zip(
    gdf.geometry.centroid.x,
    gdf.geometry.centroid.y,
    gdf["NAME_2"]
):
    ax.text(x, y, name, fontsize=8, ha="center")

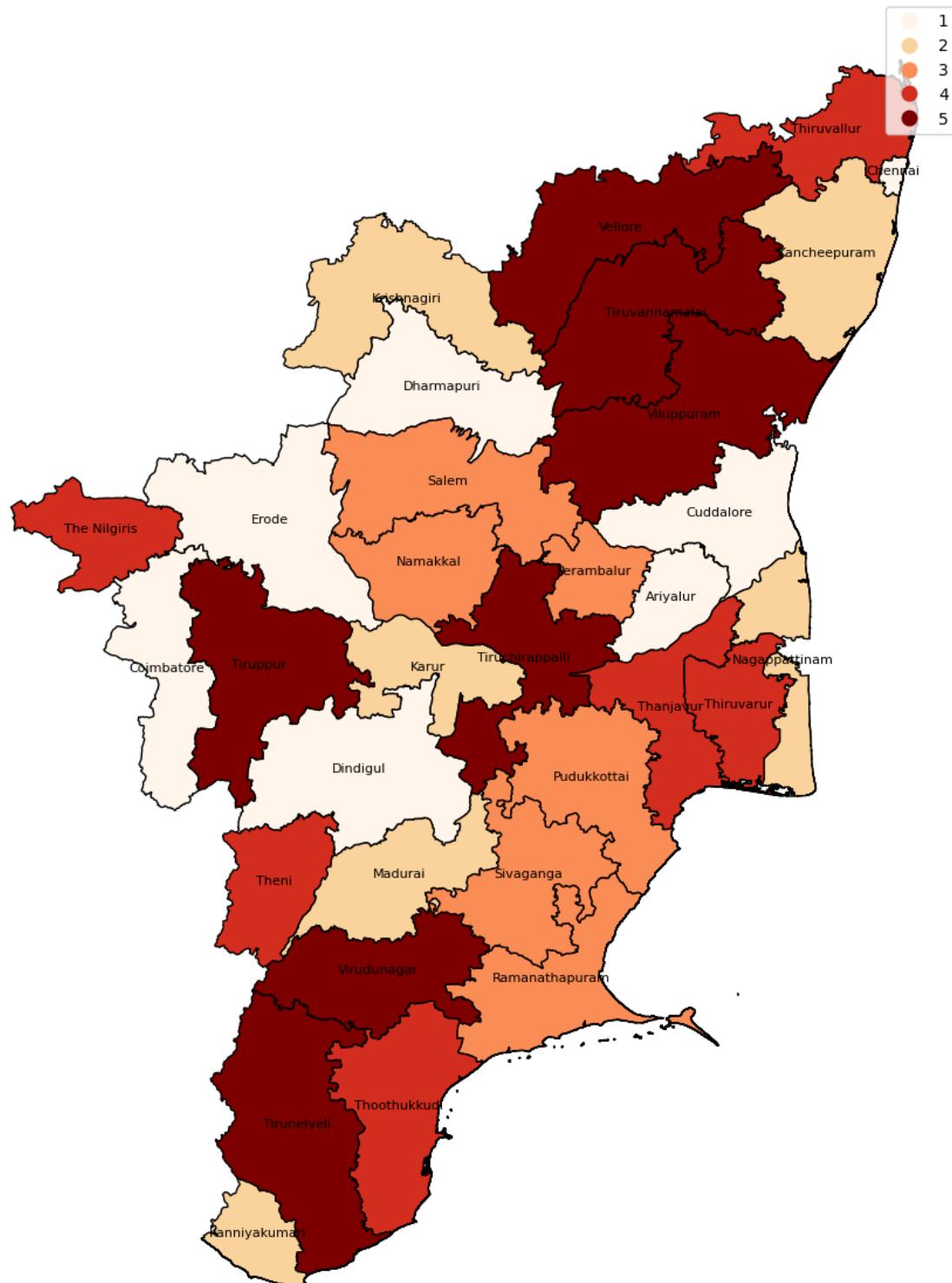
ax.set_title("Tamil Nadu District-wise COVID Deaths (2024)")
ax.axis("off")
plt.show()
```

```
/var/folders/4c/s4mb07_92pq67wgl6sy41zs80000gn/T/ipykernel_87989/2349437447.py:42: UserWarning: Geometry is in a geographic CRS. Results from 'centroid' are likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS before this operation.
```

```
gdf.geometry.centroid.x,  
/var/folders/4c/s4mb07_92pq67wgl6sy41zs80000gn/T/ipykernel_87989/2349437447.py:43: UserWarning: Geometry is in a geographic CRS. Results from 'centroid' are likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS before this operation.
```

```
gdf.geometry.centroid.y,
```

Tamil Nadu District-wise COVID Deaths (2024)



```
In [ ]: import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
import numpy as np

plt.figure(figsize=(10, 8))

m = Basemap(
    projection='ortho',
    lat_0=20,
    lon_0=78,
    resolution='l'
)

m.drawcoastlines()
m.drawcountries()
m.fillcontinents(color='lightyellow', lake_color='lightblue')
m.drawmapboundary(fill_color='lightblue')
m.drawrivers(color='blue')
m.drawstates()
m.drawparallels(np.arange(-90, 90, 30))
m.drawmeridians(np.arange(0, 360, 30))
m.etopo(scale=0.5, alpha=0.5)

# Chennai coordinates
lat, lon = 13.0827, 80.2707
x, y = m(lon, lat)

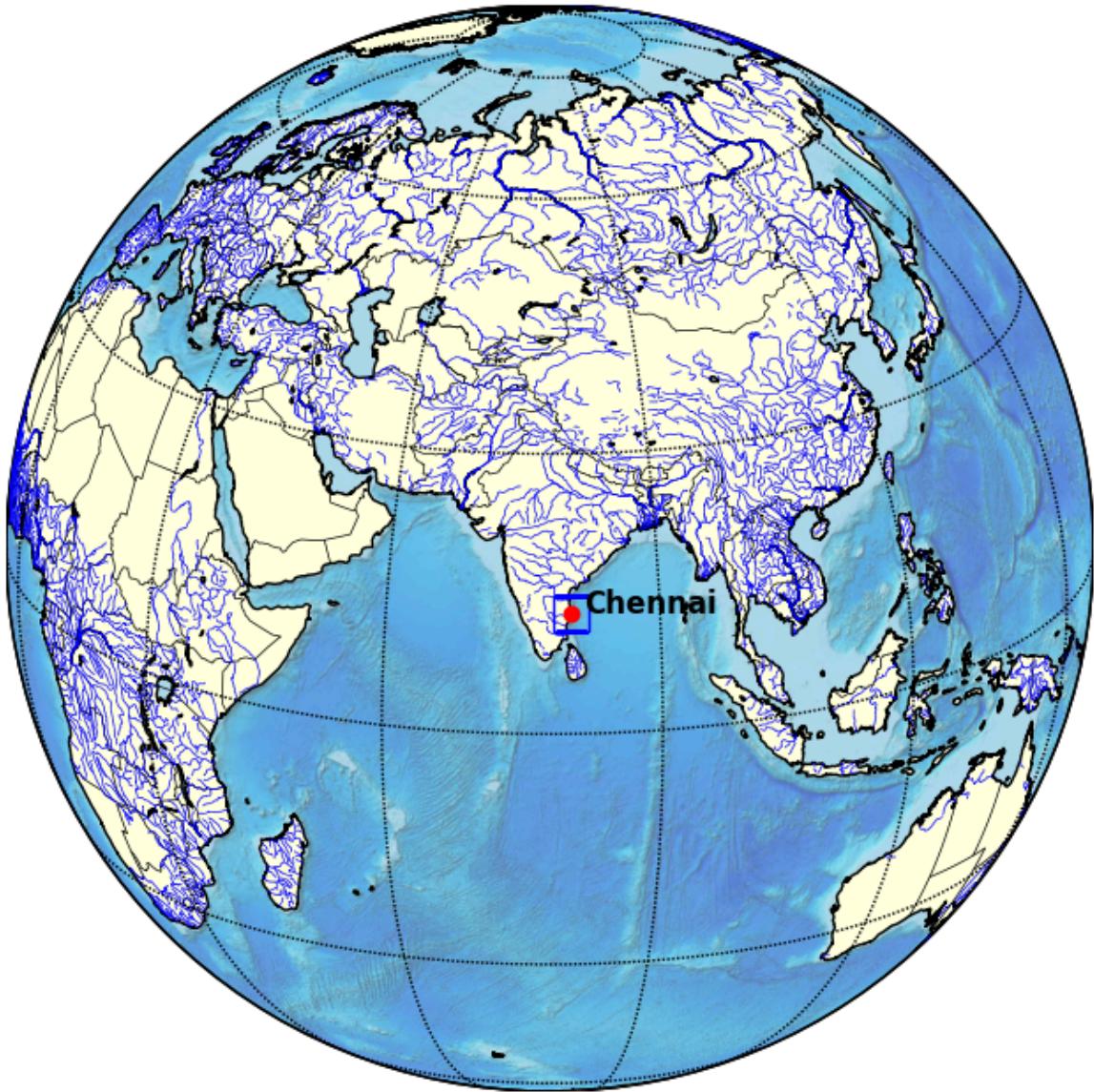
# Marker
m.plot(x, y, 's', markersize=15, markeredgecolor='blue', markerfacecolor='red')
m.plot(x, y, 'ro', markersize=6)

plt.text(
    x + 150000,
    y,
    'Chennai',
    fontsize=12,
    fontweight='bold',
    color='black'
)

plt.title("Basemap: Tamil Nadu (Chennai Highlighted)")
plt.show()
```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.0..1.000000000000002].

Basemap: Tamil Nadu (Chennai Highlighted)



```
In [ ]: import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
import numpy as np

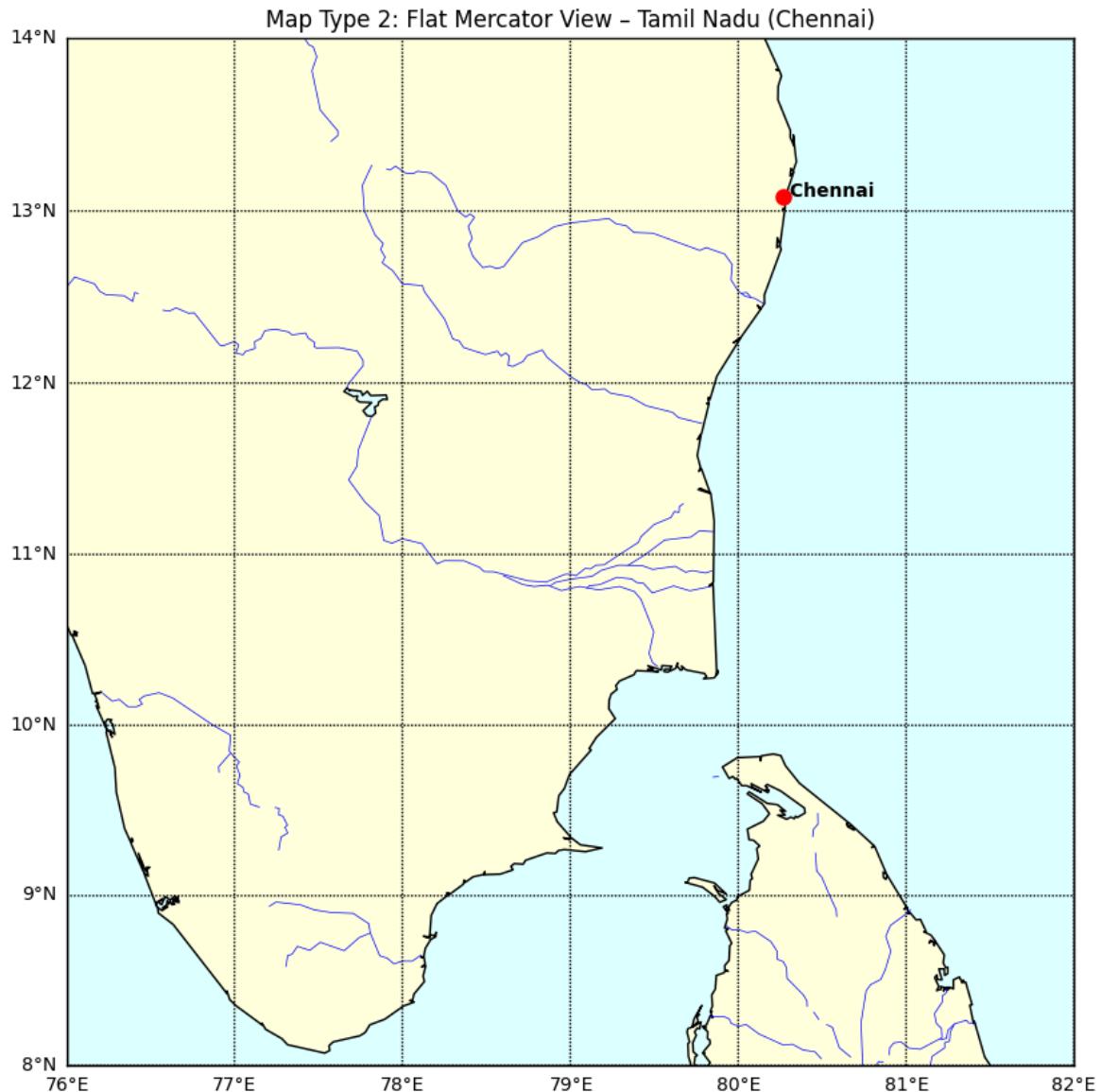
plt.figure(figsize=(12, 10))

m = Basemap(
    projection='merc',
    llcrnrlat=8, urcrnrlat=14,
    llcrnrlon=76, urcrnrlon=82,
    resolution='i'
)

m.drawcoastlines()
m.drawcountries(linewidth=2)
m.drawstates(color='blue', linewidth=1)
m.drawrivers(color='blue', linewidth=0.5)
m.drawmapboundary(fill_color='lightcyan')
m.fillcontinents(color='lightyellow', lake_color='lightcyan')
m.drawparallels(np.arange(8, 16, 1), labels=[1,0,0,0])
m.drawmeridians(np.arange(76, 84, 1), labels=[0,0,0,1])
m.shadedrelief()
```

```
x, y = m(80.2707, 13.0827)
m.plot(x, y, 'ro', markersize=8)
plt.text(x, y, 'Chennai', fontweight='bold')

plt.title("Map Type 2: Flat Mercator View – Tamil Nadu (Chennai)")
plt.show()
```



In []: