Ranjeet Gupta / SC24M138

Lab 6: Vector data processing 2

1. Point in Polygon Analysis

When you have a polygon layer and a point layer - and want to know how many or which of the points fall within the bounds of each polygon, you can use this method of analysis.

In [1]:
```
import geopandas as gpd
#load point and polygon data
point_data = r"C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\Lab-6\l
polygon_data = r"C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\Lab-6

# Load the point (places) and polygon (countries) data
places = gpd.read_file(point_data)
countries = gpd.read_file(polygon_data)

# Ensure both datasets are in WGS 84 (EPSG:4326) coordinate system
places = places.to_crs('epsg:4326')
countries = countries.to_crs('epsg:4326')

print(places)
print(countries.head())
```

```
      scalerank  natscale  labelrank           featurecla  \
0            10         1          8        Admin-1 capital
1            10         1          8        Admin-1 capital
2            10         1          8        Admin-1 capital
3            10         1          8        Admin-1 capital
4            10         1          8        Admin-1 capital
...         ...       ...        ...                    ...
7338          0       600          1        Admin-1 capital
7339          0       600          1        Admin-1 capital
7340          0       600          3        Admin-1 capital
7341          0       600          0        Admin-0 capital
7342          0       600          0  Admin-0 region capital

                       name namepar             namealt  diffascii  \
0     Colonia del Sacramento    None                None          0
1                   Trinidad    None                None          0
2                Fray Bentos    None                None          0
3                  Canelones    None                None          0
4                    Florida    None                None          0
...                      ...     ...                 ...        ...
7338          Rio de Janeiro    None                None          0
7339               São Paulo    None  Sao Paulo|Sio Paulo          0
7340                  Sydney    None                None          0
7341               Singapore    None                None          0
7342               Hong Kong    None                None          0

                  nameascii  adm0cap  ...  rank_max rank_min  geonameid  \
0     Colonia del Sacramento      0.0  ...         7        7  3443013.0
1                   Trinidad      0.0  ...         7        7  3439749.0
2                Fray Bentos      0.0  ...         7        7  3442568.0
3                  Canelones      0.0  ...         6        6  3443413.0
4                    Florida      0.0  ...         7        7  3442585.0
...                      ...      ...  ...       ...      ...        ...
7338          Rio de Janeiro      0.0  ...        14       12  3451190.0
7339               Sao Paulo      0.0  ...        14       14  3448439.0
7340                  Sydney      0.0  ...        12       12  2147714.0
7341               Singapore      1.0  ...        13       12  1880252.0
7342               Hong Kong      0.0  ...        13       12  1819729.0

            meganame         ls_name ls_match checkme min_zoom       ne_id  \
0               None            None        0       0      9.0  1159112629
1               None            None        0       0      9.0  1159112647
2               None            None        0       0      9.0  1159112663
3               None            None        0       0      9.0  1159112679
4               None            None        0       0      7.0  1159112703
...              ...             ...      ...     ...      ...         ...
7338  Rio de Janeiro  Rio de Janeiro        1       0      1.7  1159151619
7339               S       Sao Paolo        1       0      3.0  1159151621
7340          Sydney         Sydney1        1       0      1.7  1159151623
7341       Singapore       Singapore        1       5      2.1  1159151627
7342       Hong Kong       Hong Kong        1       0      3.0  1159151629

                       geometry
0           POINT (-57.84 -34.48)
1         POINT (-56.901 -33.544)
2         POINT (-58.304 -33.139)
3         POINT (-56.284 -34.538)
4         POINT (-56.215 -34.099)
...                         ...
7338  POINT (-43.22697 -22.92308)
```

```
7339  POINT (-46.62697 -23.55673)
7340  POINT (151.18323 -33.91807)
7341    POINT (103.85387 1.29498)
7342   POINT (114.18306 22.30693)

[7343 rows x 39 columns]
   scalerank       featurecla  LABELRANK      SOVEREIGNT SOV_A3  ADM0_DIF  \
0          3  Admin-0 country        5.0     Netherlands    NL1       1.0
1          0  Admin-0 country        3.0     Afghanistan    AFG       0.0
2          0  Admin-0 country        3.0          Angola    AGO       0.0
3          3  Admin-0 country        6.0  United Kingdom    GB1       1.0
4          0  Admin-0 country        6.0         Albania    ALB       0.0

   LEVEL               TYPE        ADMIN ADM0_A3  ...      CONTINENT  \
0    2.0            Country        Aruba     ABW  ...  North America
1    2.0  Sovereign country  Afghanistan     AFG  ...           Asia
2    2.0  Sovereign country       Angola     AGO  ...         Africa
3    2.0         Dependency     Anguilla     AIA  ...  North America
4    2.0  Sovereign country      Albania     ALB  ...         Europe

  REGION_UN        SUBREGION                  REGION_WB NAME_LEN LONG_LEN  \
0  Americas        Caribbean  Latin America & Caribbean      5.0      5.0
1      Asia    Southern Asia                 South Asia     11.0     11.0
2    Africa    Middle Africa         Sub-Saharan Africa      6.0      6.0
3  Americas        Caribbean  Latin America & Caribbean      8.0      8.0
4    Europe  Southern Europe      Europe & Central Asia      7.0      7.0

   ABBREV_LEN  TINY HOMEPART  \
0         5.0   4.0    -99.0
1         4.0 -99.0      1.0
2         4.0 -99.0      1.0
3         4.0 -99.0    -99.0
4         4.0 -99.0      1.0

                                            geometry
0  POLYGON ((-69.99694 12.57758, -69.93639 12.531...
1  POLYGON ((71.0498 38.40866, 71.05714 38.40903,...
2  MULTIPOLYGON (((11.73752 -16.69258, 11.73851 -...
3  MULTIPOLYGON (((-63.03767 18.21296, -63.09952 ...
4  POLYGON ((19.74777 42.5789, 19.74601 42.57993,...

[5 rows x 66 columns]
```

A. Given the locations of all known significant places, we will try to find out which country has had the highest number of important places.

In [2]:

```python
# Perform spatial join to find which points are within each country polygon
# ‘inner’: use intersection of keys from both dfs; retain only left_df geometry
points_in_countries = gpd.sjoin(places, countries, how="inner", predicate="withi
# print(points_in_countries)

# Group by country name to count the number of important places per country
places_count = points_in_countries.groupby('ADMIN').size().reset_index(name='Pla
print(places_count)
```

```python

# Merge with the countries GeoDataFrame to retain the geometries for visualizati
countries = countries.merge(places_count, left_on='ADMIN', right_on='ADMIN', how
print(countries)

# Replace NaN values with 0 for countries with no significant places
countries['Place_Count'] = countries['Place_Count'].fillna(0).astype(int)

# Find the country with the maximum number of important places
max_places_country = countries.loc[countries['Place_Count'].idxmax()]
print(f"The country with the highest number of important places is {max_places_c
              ADMIN  Place_Count
0       Afghanistan           33
1             Aland            1
2           Albania           26
3           Algeria           51
4    American Samoa            1
..              ...          ...
219         Vietnam           60
220  Western Sahara            1
221           Yemen           20
222          Zambia           34
223        Zimbabwe           20

[224 rows x 2 columns]
     scalerank       featurecla  LABELRANK      SOVEREIGNT SOV_A3  ADM0_DIF  \
0            3  Admin-0 country        5.0     Netherlands    NL1       1.0
1            0  Admin-0 country        3.0     Afghanistan    AFG       0.0
2            0  Admin-0 country        3.0          Angola    AGO       0.0
3            3  Admin-0 country        6.0  United Kingdom    GB1       1.0
4            0  Admin-0 country        6.0         Albania    ALB       0.0
..         ...              ...        ...             ...    ...       ...
250          3  Admin-0 country        4.0           Samoa    WSM       0.0
251          0  Admin-0 country        3.0           Yemen    YEM       0.0
252          0  Admin-0 country        2.0    South Africa    ZAF       0.0
253          0  Admin-0 country        3.0          Zambia    ZMB       0.0
254          0  Admin-0 country        3.0        Zimbabwe    ZWE       0.0

     LEVEL               TYPE         ADMIN ADM0_A3  ...  REGION_UN  \
0      2.0            Country         Aruba     ABW  ...   Americas
1      2.0  Sovereign country   Afghanistan     AFG  ...       Asia
2      2.0  Sovereign country        Angola     AGO  ...     Africa
3      2.0         Dependency      Anguilla     AIA  ...   Americas
4      2.0  Sovereign country       Albania     ALB  ...     Europe
..     ...                ...           ...     ...  ...        ...
250    2.0  Sovereign country         Samoa     WSM  ...    Oceania
251    2.0  Sovereign country         Yemen     YEM  ...       Asia
252    2.0  Sovereign country  South Africa     ZAF  ...     Africa
253    2.0  Sovereign country        Zambia     ZMB  ...     Africa
254    2.0  Sovereign country      Zimbabwe     ZWE  ...     Africa

           SUBREGION                   REGION_WB  NAME_LEN LONG_LEN  \
0          Caribbean   Latin America & Caribbean       5.0      5.0
1      Southern Asia                  South Asia      11.0     11.0
2      Middle Africa          Sub-Saharan Africa       6.0      6.0
3          Caribbean   Latin America & Caribbean       8.0      8.0
4    Southern Europe       Europe & Central Asia       7.0      7.0
..               ...                         ...       ...      ...
250        Polynesia         East Asia & Pacific       5.0      5.0
251     Western Asia  Middle East & North Africa       5.0      5.0
252  Southern Africa          Sub-Saharan Africa      12.0     12.0
253   Eastern Africa          Sub-Saharan Africa       6.0      6.0
254   Eastern Africa          Sub-Saharan Africa       8.0      8.0

    ABBREV_LEN  TINY HOMEPART  \
0          5.0   4.0    -99.0
1          4.0 -99.0      1.0
2          4.0 -99.0      1.0
3          4.0 -99.0    -99.0
4          4.0 -99.0      1.0
..         ...   ...      ...
```

```
250        5.0 -99.0      1.0
251        4.0 -99.0      1.0
252        5.0 -99.0      1.0
253        6.0 -99.0      1.0
254        5.0 -99.0      1.0

                                              geometry Place_Count
0    POLYGON ((-69.99694 12.57758, -69.93639 12.531...         1.0
1    POLYGON ((71.0498 38.40866, 71.05714 38.40903,...        33.0
2    MULTIPOLYGON (((11.73752 -16.69258, 11.73851 -...        48.0
3    MULTIPOLYGON (((-63.03767 18.21296, -63.09952 ...         NaN
4    POLYGON ((19.74777 42.5789, 19.74601 42.57993,...        26.0
..                                                 ...         ...
250  MULTIPOLYGON (((-171.57002 -13.93816, -171.564...         1.0
251  MULTIPOLYGON (((53.30824 12.11839, 53.31027 12...        20.0
252  MULTIPOLYGON (((37.86378 -46.94085, 37.83644 -...        72.0
253  POLYGON ((31.11984 -8.61663, 31.14102 -8.60619...        34.0
254  POLYGON ((30.01065 -15.64623, 30.05024 -15.640...        20.0

[255 rows x 67 columns]
The country with the highest number of important places is United States of Ameri
ca with 768 places.
```

B. The point dataset has places with Latitude/Longitude coordinates, choose WGS 84 EPSG:436 as the CRS in the Coordinate Reference System Selector dialog.

In [3]:
```python
# Load the dataset
places = gpd.read_file(point_data)

# Reprojects the data into the WGS 84 CRS if it’s in a different CRS.
places = places.to_crs("EPSG:4326")
```

C. Open countries vector layer

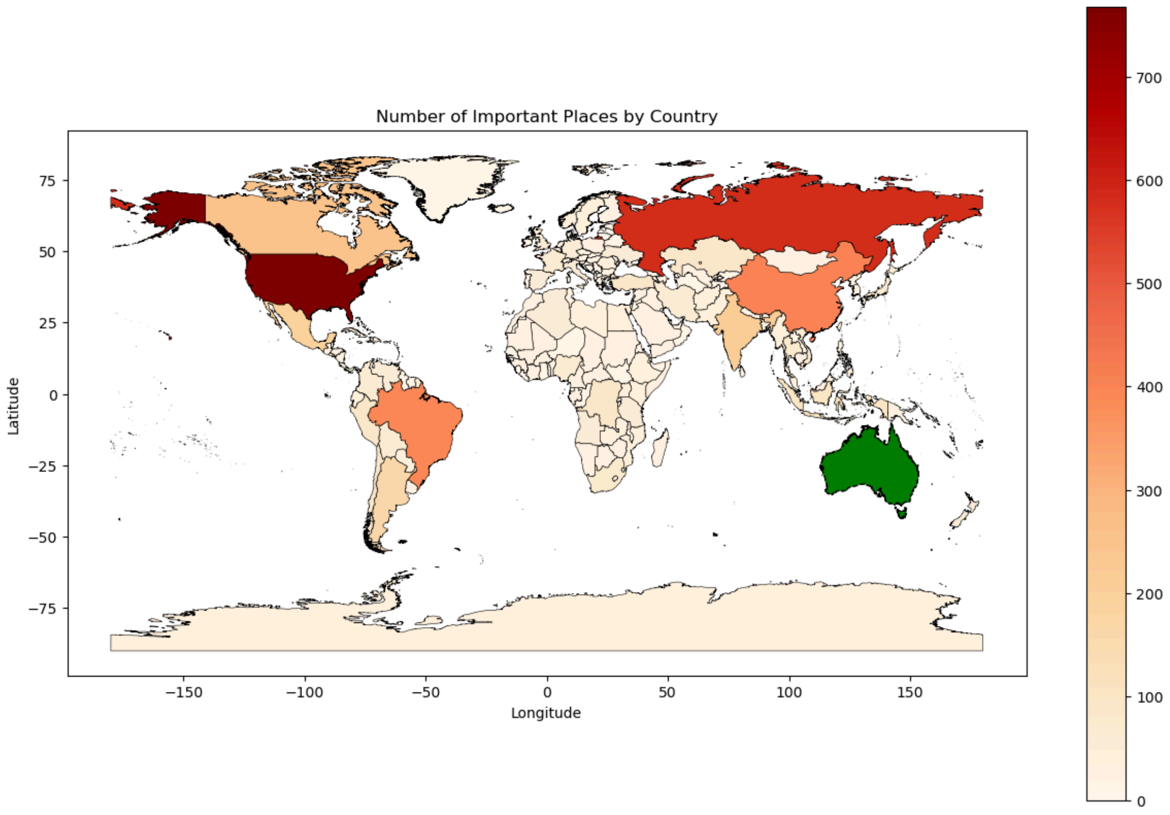
In [4]:
```python
# Display the first few rows of the dataset
print(countries.head())
```

```
   scalerank      featurecla  LABELRANK     SOVEREIGNT SOV_A3  ADM0_DIF  \
0          3  Admin-0 country        5.0    Netherlands    NL1       1.0
1          0  Admin-0 country        3.0    Afghanistan    AFG       0.0
2          0  Admin-0 country        3.0         Angola    AGO       0.0
3          3  Admin-0 country        6.0 United Kingdom    GB1       1.0
4          0  Admin-0 country        6.0        Albania    ALB       0.0

   LEVEL              TYPE       ADMIN ADM0_A3  ...  REGION_UN  \
0    2.0           Country       Aruba     ABW  ...   Americas
1    2.0  Sovereign country  Afghanistan     AFG  ...       Asia
2    2.0  Sovereign country      Angola     AGO  ...     Africa
3    2.0        Dependency    Anguilla     AIA  ...   Americas
4    2.0  Sovereign country     Albania     ALB  ...     Europe

         SUBREGION                  REGION_WB  NAME_LEN LONG_LEN ABBREV_LEN  \
0        Caribbean  Latin America & Caribbean       5.0      5.0        5.0
1    Southern Asia                 South Asia      11.0     11.0        4.0
2    Middle Africa         Sub-Saharan Africa       6.0      6.0        4.0
3        Caribbean  Latin America & Caribbean       8.0      8.0        4.0
4  Southern Europe      Europe & Central Asia       7.0      7.0        4.0

   TINY HOMEPART                                           geometry  \
0   4.0    -99.0  POLYGON ((-69.99694 12.57758, -69.93639 12.531...
1 -99.0      1.0  POLYGON ((71.0498 38.40866, 71.05714 38.40903,...
2 -99.0      1.0  MULTIPOLYGON (((11.73752 -16.69258, 11.73851 -...
3 -99.0    -99.0  MULTIPOLYGON (((-63.03767 18.21296, -63.09952 ...
4 -99.0      1.0  POLYGON ((19.74777 42.5789, 19.74601 42.57993,...

  Place_Count
0           1
1          33
2          48
3           0
4          26

[5 rows x 67 columns]
```

D. Using point in polygon vector analysis find the number of important places in each country. Colour code accordingly (You can highlight your favourite country)

In [5]:
```
import matplotlib.pyplot as plt
favourite_country = 'Australia'
# Plot the countries with color coding by the number of important places

fig, ax = plt.subplots(1, 1, figsize=(15, 10))
# Color map by the number of important places
countries.plot(column='Place_Count', cmap='OrRd', linewidth=0.4, ax=ax, edgecolo

# Highlight favorite country
countries[countries['ADMIN'] == favourite_country].plot(ax=ax, color="green", ed

# Add title and other plot settings
```

```
ax.set_title("Number of Important Places by Country")
ax.set_xlabel("Longitude")
ax.set_ylabel("Latitude")
plt.show()
```



In [ ]:

```

```

2. Spatial Querying

A. Spatial query to find all cities that are within 10 kms of a river.

In [6]:

```
rivers_data = r"C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\Lab-6\

# Load the cities and rivers GeoDataFrames
cities = gpd.read_file(point_data).to_crs("EPSG:32643")  # Replace with appropri
rivers = gpd.read_file(rivers_data).to_crs("EPSG:32643")  # Same CRS as cities

# and project to a suitable CRS (e.g., UTM for accurate distance, EPSG:32643 for

# Create a 10,000-meter (10 km) buffer around each river
rivers_buffer = rivers.buffer(10000)

# Convert buffer to a GeoDataFrame for spatial operations
rivers_buffer_gdf = gpd.GeoDataFrame(rivers, geometry=rivers_buffer, crs=rivers.
print(rivers_buffer_gdf, "\n")

# Spatial join: find cities that intersects with the rivers buffer
cities_near_rivers = gpd.sjoin(cities, rivers_buffer_gdf, how="inner", predicate
print(cities_near_rivers, "\n")
print(cities_near_rivers.columns, "\n")

# Display cities near rivers
print(cities_near_rivers[['name_left', 'geometry']])  # place the actual name fi

# # Optional: plot the cities and rivers for visualization
# import matplotlib.pyplot as plt

# fig, ax = plt.subplots(figsize=(10, 10))
# rivers.plot(ax=ax, color="blue", label="Rivers")
# rivers_buffer_gdf.plot(ax=ax, color="lightblue", alpha=0.5, label="10 km Buffe
# cities_near_rivers.plot(ax=ax, color="red", label="Cities within 10 km")
# ax.legend()
# plt.show()
```

```
                dissolve  scalerank       featurecla            name  \
0                 0River        1.0            River  Irrawaddy Delta   
1     1001Lake Centerline        9.0  Lake Centerline        Tonle Sap   
2               1001River        9.0            River        Tonle Sap   
3     1002Lake Centerline        9.0  Lake Centerline          Sheksna   
4               1002River        9.0            River          Sheksna   
...                   ...        ...              ...              ...   
1450  2049Lake Centerline       10.0  Lake Centerline             Ohau   
1451             219River        6.0            River               Po   
1452             178River        5.0            River            Loire   
1453             178River        5.0            River            Loire   
1454              303Drau        7.0            River             Drau   

     name_alt  rivernum             note  min_zoom   name_en  min_label  ...  \
0        None         0             None       2.0  Irrawaddy        3.0  ...   
1        None      1001             None       7.1       None        8.1  ...   
2        None      1001             None       7.1       None        8.1  ...   
3        None      1002             None       7.1    Sheksna        8.1  ...   
4        None      1002             None       7.1    Sheksna        8.1  ...   
...       ...       ...              ...       ...        ...        ...  ...   
1450     None      2049             None       7.2       Ohau        8.2  ...   
1451     None       219   Version 4 edit       5.0         Po        6.0  ...   
1452     None    178000   Changed in 2.0       4.7      Loire        5.7  ...   
1453     None       178   Changed in 4.0       4.7      Loire        5.7  ...   
1454    Drava       303             None       6.0      Drava        7.0  ...   

        name_pl        name_pt            name_ru    name_sv          name_tr  \
0       Irawadi   Rio IrauÃ¡di    Ð�Ñ�Ð°Ð²Ð°Ð´Ð¸  Irrawaddy  Ä°ravadi Nehri   
1     Tonle Sap           None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None             None   
2     Tonle Sap           None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None             None   
3       Szeksna           None    Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna             None   
4       Szeksna           None    Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna             None   
...         ...            ...             ...        ...              ...   
1450       None           None            None       None             None   
1451        Pad         Rio PÃ³            Ð�Ð¾         Po         Po Nehri   
1452      Loara     Rio Loire       Ð�Ñ�Ð°Ñ�Ð°      Loire      Loire Nehri   
1453      Loara     Rio Loire       Ð�Ñ�Ð°Ñ�Ð°      Loire      Loire Nehri   
1454      Drawa     Rio Drava       Ð�Ñ�Ð°Ð²Ð°      Drava            Drava   

                name_vi           name_zh wdid_score      ne_id  \
0     SÃ´ng Ayeyarwaddy  ä¼�æ´�ç�¦åº�æ±�          2  1159109417   
1                  None              None          4  1159109429   
2                  None              None          4  1159109445   
3                  None  è��å��æ�¯ç´�æ²³          4  1159109447   
4                  None  è��å��æ�¯ç´�æ²³          4  1159109461   
...                 ...               ...        ...        ...   
1450               None              None          4  1159129657   
1451         SÃ´ng Po          æ³¢æ²³          4  1159129663   
1452      SÃ´ng Loire      å�¢ç�¦å°�æ²³          4  1159129671   
1453      SÃ´ng Loire      å�¢ç�¦å°�æ²³          4  1159129677   
1454               None      å¾·æ��ç�¦æ²³          5  1159129685   

                                               geometry  
0     MULTIPOLYGON (((2626820.774 1940760.031, 26267...  
1     POLYGON ((3735875.977 1668968.192, 3736305.948...  
2     POLYGON ((3860138.032 1554810.586, 3859705.598...  
3     POLYGON ((-1559957.077 7164826.652, -1559090.0...  
4     POLYGON ((-1518116.889 7150895.351, -1517786.0...  
...                                                 ...  
1450  POLYGON ((6153347.591 -10590043.95, 6153087.04...  
1451  POLYGON ((-4548785.052 7661687.108, -4549413.0...
1452                                               None
1453  POLYGON ((-4674939.259 7950052.14, -4676042.69...
1454  POLYGON ((-4008520.476 7401613.758, -4004420.6...

[1455 rows x 35 columns]

      scalerank_left  natscale  labelrank  featurecla_left  name_left  \
24                10         1          5  Admin-1 capital    Yên Bái
26                10         1          5  Admin-1 capital  Thái Bình
27                10         1          5  Admin-1 capital    Tuy Hòa
30                10         1          5  Admin-1 capital   Cao Lãnh
33                10         1          5  Admin-1 capital  Vĩnh Long
...              ...       ...        ...              ...        ...
7331               0       600          3  Admin-0 capital      Cairo
7332               0       600          1  Admin-1 capital   Shanghai
7335               0       600          3  Admin-0 capital      Paris
7337               0       600          1  Admin-1 capital    Kolkata
7339               0       600          1  Admin-1 capital  São Paulo

       namepar               namealt  diffascii  nameascii  adm0cap  ...  \
24        None                  None          0    Yen Bai      0.0  ...
26        None                  None          0  Thai Binh      0.0  ...
27        None                  None          0    Tuy Hoa      0.0  ...
30        None                  None          0   Cao Lanh      0.0  ...
33        None                  None          0  Vinh Long      0.0  ...
...        ...                   ...        ...        ...      ...  ...
7331      None           Al-Qahirah          0      Cairo      1.0  ...
7332      None                  None          0   Shanghai      0.0  ...
7335      None                  None          0      Paris      1.0  ...
7337  Calcutta                  None          0    Kolkata      0.0  ...
7339      None  Sao Paulo|Sio Paulo          0  Sao Paulo      0.0  ...

         name_nl         name_pl       name_pt        name_ru       name_sv  \
24    Rode Rivier  Rzeka Czerwona  Rio Vermelho  Ð¥Ð¾Ð½Ð³Ñ�Ð°  RÃ¶da floden
26    Rode Rivier  Rzeka Czerwona  Rio Vermelho  Ð¥Ð¾Ð½Ð³Ñ�Ð°  RÃ¶da floden
27           None            None          None           None          None
30         Mekong          Mekong   Rio Mekong  Ð�ÐµÐºÐ¾Ð½Ð³        Mekong
33           None            None          None           None          None
...           ...             ...           ...            ...           ...
7331         Nijl             Nil      Rio Nilo         Ð�Ð¸Ð»         Nilen
7332         None            None          None           None          None
7335        Seine         Sekwana      Rio Sena       Ð¡ÐµÐ½Ð°         Seine
7337       Ganges          Ganges    Rio Ganges      Ð�Ð°Ð½Ð³        Ganges
7339       TietÃª           TietÃª    Rio TietÃª    Ð¢Ð¸ÐµÑ�Ðµ    Rio TietÃª

          name_tr        name_vi    name_zh wdid_score ne_id_right
24    KÄ±zÄ±l Nehir  SÃ´ng Há»�ng      çº¢æ²³          5  1159116785
26    KÄ±zÄ±l Nehir  SÃ´ng Há»�ng      çº¢æ²³          5  1159116785
27           None           None       None          0  1159111003
30         Mekong     MÃª KÃ´ng  æ¹�å�¬æ²³          4  1159121023
33           None           None       None          0  1159113739
...           ...            ...        ...        ...         ...
7331          Nil     SÃ´ng Nin  å°¼ç½�æ²³          4  1159121589
7332         None           None       None          0  1159123017
7335    Sen Nehri    SÃ´ng Seine  å¡�çº³æ²³          4  1159112177
7337   Ganj Nehri  SÃ´ng Háº±ng     æ��æ²³          4  1159122643
7339         None           None  é�µç�¹æ²³          4  1159125573

[1941 rows x 74 columns]
```

```
Index(['scalerank_left', 'natscale', 'labelrank', 'featurecla_left',
       'name_left', 'namepar', 'namealt', 'diffascii', 'nameascii', 'adm0cap',
       'capalt', 'capin', 'worldcity', 'megacity', 'sov0name', 'sov_a3',
       'adm0name', 'adm0_a3', 'adm1name', 'iso_a2', 'note_left', 'latitude',
       'longitude', 'changed', 'namediff', 'diffnote', 'pop_max', 'pop_min',
       'pop_other', 'rank_max', 'rank_min', 'geonameid', 'meganame', 'ls_name',
       'ls_match', 'checkme', 'min_zoom_left', 'ne_id_left', 'geometry',
       'index_right', 'dissolve', 'scalerank_right', 'featurecla_right',
       'name_right', 'name_alt', 'rivernum', 'note_right', 'min_zoom_right',
       'name_en', 'min_label', 'label', 'wikidataid', 'name_ar', 'name_bn',
       'name_de', 'name_es', 'name_fr', 'name_el', 'name_hi', 'name_hu',
       'name_id', 'name_it', 'name_ja', 'name_ko', 'name_nl', 'name_pl',
       'name_pt', 'name_ru', 'name_sv', 'name_tr', 'name_vi', 'name_zh',
       'wdid_score', 'ne_id_right'],
      dtype='object')

      name_left                            geometry
24      Yên Bái     POINT (3695475.625 2729239.517)
26    Thái Bình      POINT (3896857.741 2610661.08)
27      Tuy Hòa       POINT (4436861.3 1739688.494)
30     Cao Lãnh     POINT (4013112.929 1341089.401)
33    Vĩnh Long      POINT (4057373.447 1318655.57)
...         ...                                 ...
7331      Cairo    POINT (-3909845.463 4288553.943)
7332   Shanghai     POINT (5124053.991 4581810.359)
7335      Paris    POINT (-4212632.342 8373560.534)
7337    Kolkata     POINT (1879181.673 2549954.605)
7339  São Paulo  POINT (-6179656.865 -15580440.684)

[1941 rows x 2 columns]
```

B. Load the river lake line data and cities populated places data. Explore the attribute table

In [7]:
```python
# Check the first few rows to explore the attribute tables
print("Rivers/Lakes Data Attributes:")
print(rivers.head())  # Display the first few rows of the river
print("\nRivers/Lakes Data Columns:")
print(rivers.columns)  # Display all column names for an overview

print("\nCities/Populated Places Data Attributes:")
print(cities.head())  # Display the first few rows of the cities
print("\nCities/Populated Places Data Columns:")
print(cities.columns)  # Display all column names for an overview
```

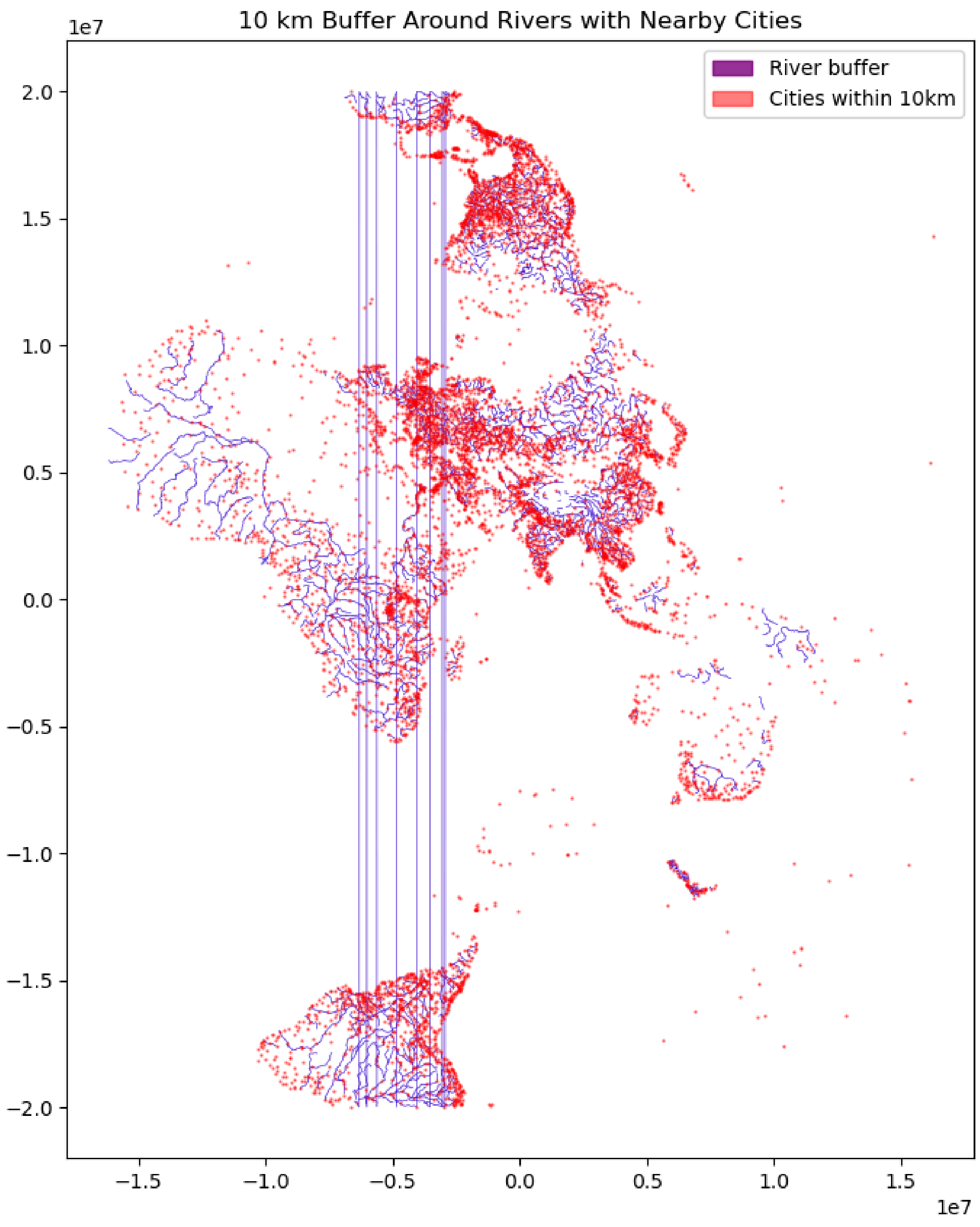
```
Rivers/Lakes Data Attributes:
              dissolve  scalerank       featurecla              name name_alt  \
0                0River        1.0            River  Irrawaddy Delta     None
1  1001Lake Centerline        9.0  Lake Centerline        Tonle Sap     None
2             1001River        9.0            River        Tonle Sap     None
3  1002Lake Centerline        9.0  Lake Centerline          Sheksna     None
4             1002River        9.0            River          Sheksna     None

   rivernum  note  min_zoom    name_en  min_label  ...    name_pl  \
0         0  None       2.0  Irrawaddy        3.0  ...    Irawadi
1      1001  None       7.1       None        8.1  ...  Tonle Sap
2      1001  None       7.1       None        8.1  ...  Tonle Sap
3      1002  None       7.1    Sheksna        8.1  ...    Szeksna
4      1002  None       7.1    Sheksna        8.1  ...    Szeksna

        name_pt           name_ru    name_sv         name_tr  \
0  Rio IrauÃ¡di    Ð�Ñ�Ð°Ð²Ð°Ð´Ð¸  Irrawaddy  Ä°ravadi Nehri
1          None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None            None
2          None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None            None
3          None      Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna            None
4          None      Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna            None

            name_vi          name_zh wdid_score       ne_id  \
0  SÃ´ng Ayeyarwaddy  ä¼�æ´�ç�¦åº�æ±�          2  1159109417
1              None             None          4  1159109429
2              None             None          4  1159109445
3              None  è��å��æ�¯ç´�æ²³          4  1159109447
4              None  è��å��æ�¯ç´�æ²³          4  1159109461

                                            geometry
0  MULTILINESTRING ((2705619.301 2094592.654, 270...
1  MULTILINESTRING ((3731406.707 1659438.944, 373...
2  LINESTRING (3858032.783 1568315.839, 3858536.1...
3  LINESTRING (-1556543.92 7154971.481, -1554135....
4  LINESTRING (-1510524.362 7144405.677, -1509386...

[5 rows x 35 columns]

Rivers/Lakes Data Columns:
Index(['dissolve', 'scalerank', 'featurecla', 'name', 'name_alt', 'rivernum',
       'note', 'min_zoom', 'name_en', 'min_label', 'label', 'wikidataid',
       'name_ar', 'name_bn', 'name_de', 'name_es', 'name_fr', 'name_el',
       'name_hi', 'name_hu', 'name_id', 'name_it', 'name_ja', 'name_ko',
       'name_nl', 'name_pl', 'name_pt', 'name_ru', 'name_sv', 'name_tr',
       'name_vi', 'name_zh', 'wdid_score', 'ne_id', 'geometry'],
      dtype='object')

Cities/Populated Places Data Attributes:
   scalerank  natscale  labelrank       featurecla                    name  \
0         10         1          8  Admin-1 capital  Colonia del Sacramento
1         10         1          8  Admin-1 capital                Trinidad
2         10         1          8  Admin-1 capital             Fray Bentos
3         10         1          8  Admin-1 capital               Canelones
4         10         1          8  Admin-1 capital                 Florida

  namepar namealt  diffascii               nameascii  adm0cap  ...  rank_max  \
0    None    None          0  Colonia del Sacramento      0.0  ...         7
1    None    None          0                Trinidad      0.0  ...         7
2    None    None          0             Fray Bentos      0.0  ...         7
3    None    None          0               Canelones      0.0  ...         6
```

```
4    None    None          0                 Florida     0.0  ...         7

  rank_min  geonameid  meganame ls_name ls_match checkme min_zoom       ne_id  \
0        7  3443013.0      None    None        0       0      9.0  1159112629
1        7  3439749.0      None    None        0       0      9.0  1159112647
2        7  3442568.0      None    None        0       0      9.0  1159112663
3        6  3443413.0      None    None        0       0      9.0  1159112679
4        7  3442585.0      None    None        0       0      7.0  1159112703

                             geometry
0  POINT (-3969096.441 -14975071.773)
1  POINT (-4131395.315 -15029487.368)
2   POINT (-4018601.143 -15163527.45)
3  POINT (-4117648.235 -14871923.154)
4  POINT (-4158131.122 -14919756.839)

[5 rows x 39 columns]

Cities/Populated Places Data Columns:
Index(['scalerank', 'natscale', 'labelrank', 'featurecla', 'name', 'namepar',
       'namealt', 'diffascii', 'nameascii', 'adm0cap', 'capalt', 'capin',
       'worldcity', 'megacity', 'sov0name', 'sov_a3', 'adm0name', 'adm0_a3',
       'adm1name', 'iso_a2', 'note', 'latitude', 'longitude', 'changed',
       'namediff', 'diffnote', 'pop_max', 'pop_min', 'pop_other', 'rank_max',
       'rank_min', 'geonameid', 'meganame', 'ls_name', 'ls_match', 'checkme',
       'min_zoom', 'ne_id', 'geometry'],
      dtype='object')
```

C. To do buffer analysis, the layers must be in projected Coordinate Reference System (CRS). as the buffer distance would be in m or km

In [8]:
```
# Choose an appropriate projected CRS (e.g., UTM Zone 43N for India, EPSG:32643)
projected_crs = "EPSG:32643"   # Adjust as needed for your region

# Reproject to the chosen projected CRS
rivers_projected = rivers.to_crs(projected_crs)
cities_projected = cities.to_crs(projected_crs)

# Create a 10 km buffer around the rivers
rivers_buffer = rivers_projected.buffer(10000)  # 10,000 meters (10 km)

# # Plot to visualize
# fig, ax = plt.subplots(figsize=(10, 10))
# rivers_buffer.plot(ax=ax, color="blue", edgecolor="black", alpha=0.5, label= '

# cities_projected.plot(ax=ax, color="red", markersize=1.5, label= 'Cities withi
# ax.set_title("10 km Buffer Around Rivers with Nearby Cities")
# ax.legend()
# plt.show()

# Plot to visualize
import matplotlib.patches as mpatches

fig, ax = plt.subplots(figsize=(10, 10))
```

```
rivers_buffer.plot(ax=ax, color="purple", edgecolor="blue", linewidth=0.15, alph
cities_projected.plot(ax=ax, color="red", alpha=0.5, markersize=0.5, label='Citi

# Create custom legend entries
buffer_patch = mpatches.Patch(color="purple", alpha=0.8, label="River buffer")
city_patch = mpatches.Patch(color="red", alpha=0.5, label="Cities within 10km")

# Add the custom legend
ax.legend(handles=[buffer_patch, city_patch])
ax.set_title("10 km Buffer Around Rivers with Nearby Cities")
plt.show()
```



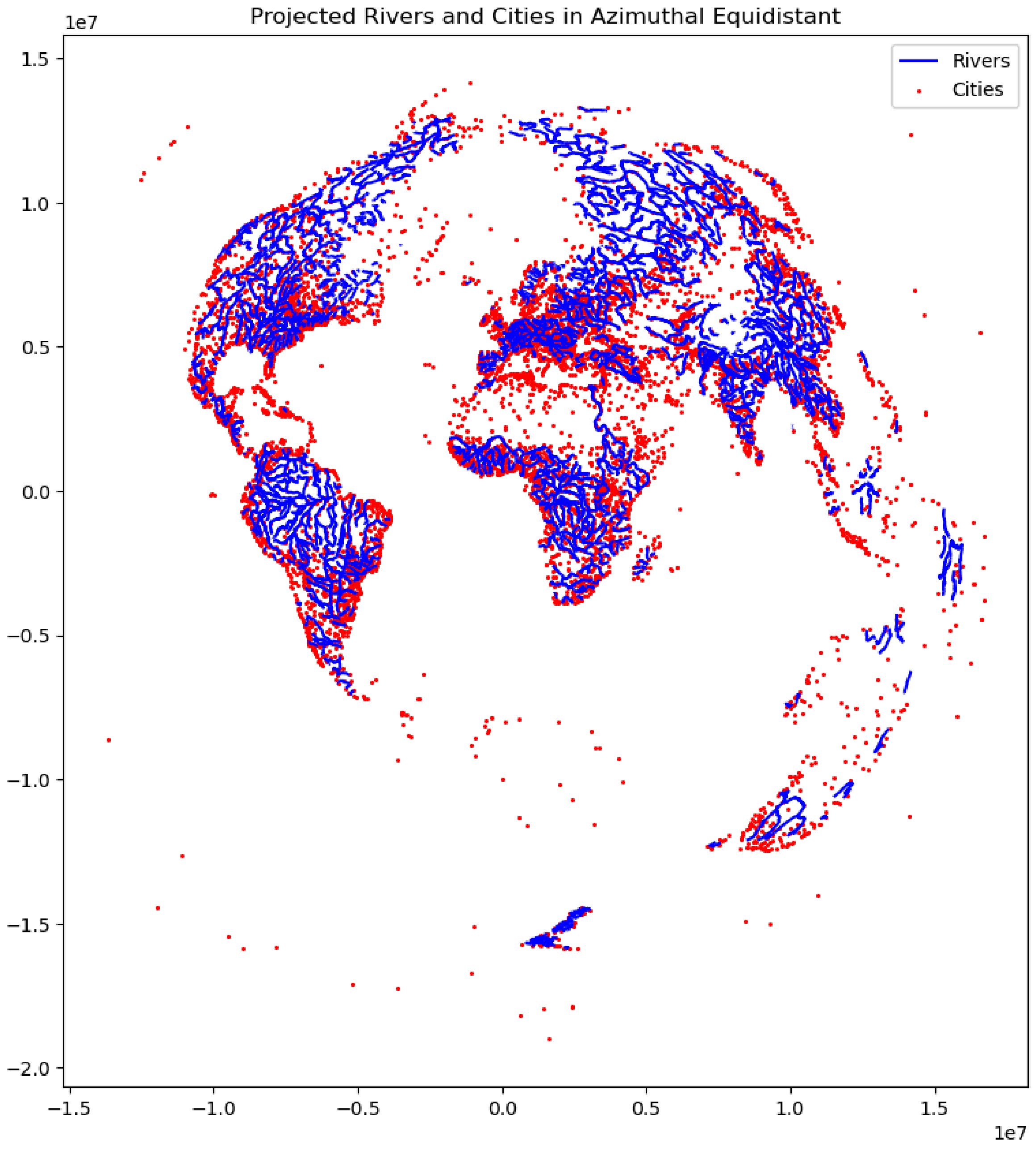
D. Convert the files from geographic coordinate system to projected

coordinate system. Project both the datsets to World_Azimuthal_Equidistant. For creating buffers, an Azimuthal Equidistant projection would be best suited as radial distances around the centre of the projection are accurate

In [9]:

```python
# Check the CRS of the loaded data
print(rivers.crs)
print(cities.crs)

from pyproj import CRS

# Define the World Azimuthal Equidistant projection with a PROJ string
world_ae_proj = CRS("+proj=aeqd +lat_0=0 +lon_0=0 +datum=WGS84 +units=m +no_defs

# Reproject rivers and cities to World Azimuthal Equidistant
rivers_ae = rivers.to_crs(world_ae_proj)
cities_ae = cities.to_crs(world_ae_proj)

# print(rivers_ae)

# Verify the new CRS
print(f"Rivers dataset CRS after projection: {rivers_ae.crs}")
print(f"Cities dataset CRS after projection: {cities_ae.crs}")

# Optional: Visualize the projected data
fig, ax = plt.subplots(figsize=(10, 10))
rivers_ae.plot(ax=ax, color='blue', label='Rivers')
cities_ae.plot(ax=ax, color='red', markersize=1.5, label='Cities')
ax.set_title("Projected Rivers and Cities in Azimuthal Equidistant")
ax.legend()
plt.show()
```

```
EPSG:32643
EPSG:32643
Rivers dataset CRS after projection: +proj=aeqd +lat_0=0 +lon_0=0 +datum=WGS84 +u
nits=m +no_defs +type=crs
Cities dataset CRS after projection: +proj=aeqd +lat_0=0 +lon_0=0 +datum=WGS84 +u
nits=m +no_defs +type=crs
```

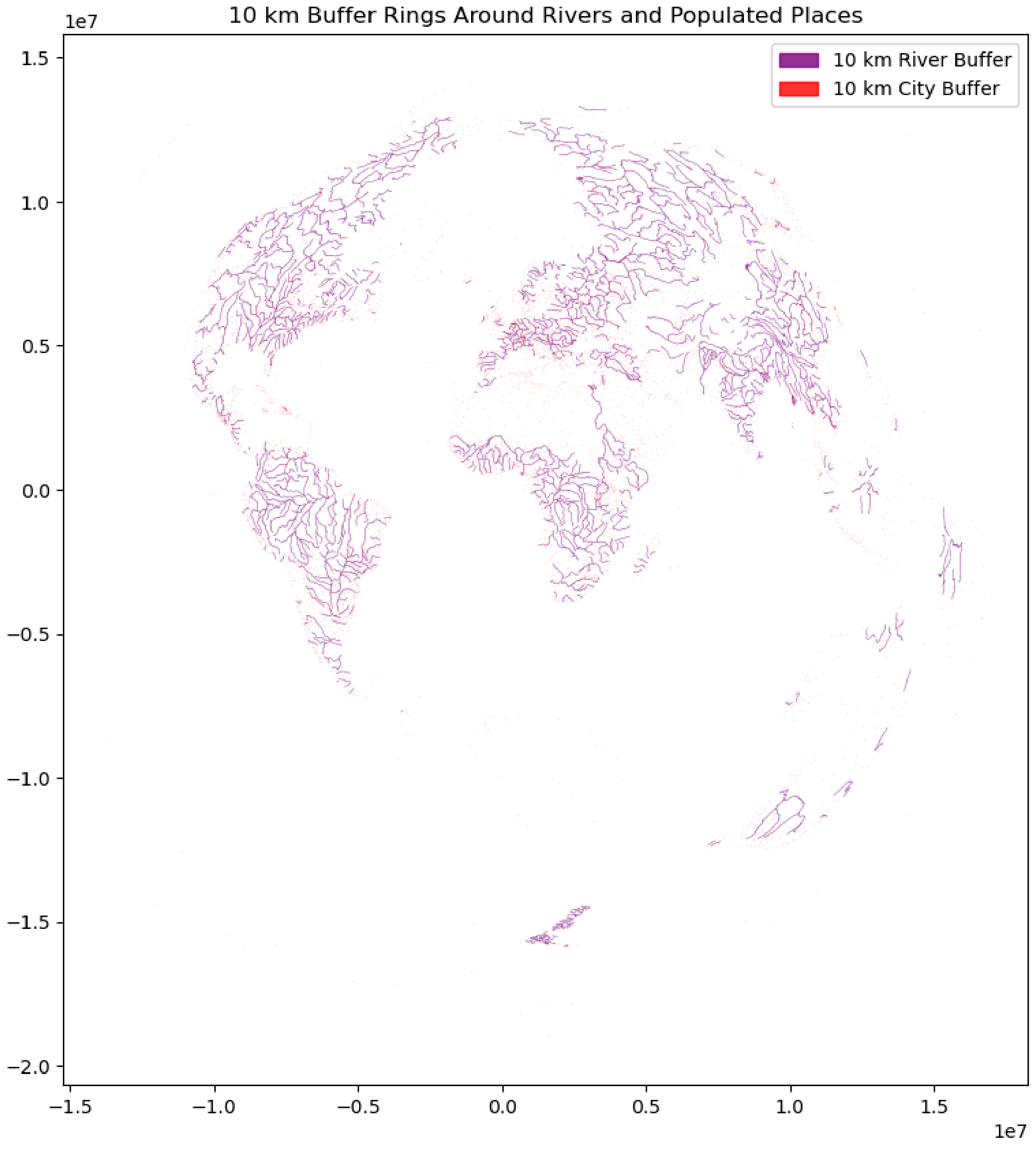


E. create buffer rings around the rivers and populated place. Use buffer distance as 10000 (10 Km)

In [10]:
```
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches


# Create a 10 km buffer around the rivers and cities in the projected coordinate
rivers_buffer = rivers_ae.buffer(10000)  # 10,000 meters = 10 km
cities_buffer = cities_ae.buffer(10000)  # 10,000 meters = 10 km

# Convert these buffered objects into GeoDataFrames for easier handling
rivers_buffer_gdf = gpd.GeoDataFrame(rivers_ae, geometry=rivers_buffer, crs=rive
cities_buffer_gdf = gpd.GeoDataFrame(cities_ae, geometry=cities_buffer, crs=citi
print(rivers_buffer_gdf)
print(cities_buffer_gdf)
```

```
# UserWarning: Legend does not support handles for PatchCollection instances. Fr
# # Plotting the buffer rings around rivers and populated places
# fig, ax = plt.subplots(figsize=(10, 10))

# # Plot river buffers first (could be more visually prominent)
# rivers_buffer_gdf.plot(ax=ax, color="cyan", edgecolor="black", linewidth=0.15,

# # Plot city buffers on top (may overlap)
# cities_buffer_gdf.plot(ax=ax, color="orange", edgecolor="black", linewidth=0.1

# ax.set_title("10 km Buffer Rings Around Rivers and Populated Places")
# ax.legend()
# plt.show()


# Plotting the buffer rings around rivers and populated places
fig, ax = plt.subplots(figsize=(10, 10))

# Plot river buffers first (could be more visually prominent)
rivers_buffer_gdf.plot(ax=ax, color="purple", alpha= 0.8)  # No label here

# Plot city buffers on top (may overlap)
cities_buffer_gdf.plot(ax=ax, color="red", alpha=0.8)  # No label here


# Create custom legend entries
river_patch = mpatches.Patch(color="purple", alpha=0.8, label="10 km River Buffe
city_patch = mpatches.Patch(color="red", alpha=0.8, label="10 km City Buffer")

# Add the custom legend
ax.legend(handles=[river_patch, city_patch])
ax.set_title("10 km Buffer Rings Around Rivers and Populated Places")
plt.show()
```

```
                 dissolve  scalerank       featurecla            name  \
0                  0River        1.0            River  Irrawaddy Delta
1     1001Lake Centerline        9.0  Lake Centerline        Tonle Sap
2                1001River        9.0            River        Tonle Sap
3     1002Lake Centerline        9.0  Lake Centerline          Sheksna
4                1002River        9.0            River          Sheksna
...                   ...        ...              ...              ...
1450  2049Lake Centerline       10.0  Lake Centerline             Ohau
1451              219River        6.0            River               Po
1452              178River        5.0            River            Loire
1453              178River        5.0            River            Loire
1454               303Drau        7.0            River             Drau

     name_alt  rivernum            note  min_zoom    name_en  min_label  ...  \
0        None         0            None       2.0  Irrawaddy        3.0  ...
1        None      1001            None       7.1       None        8.1  ...
2        None      1001            None       7.1       None        8.1  ...
3        None      1002            None       7.1    Sheksna        8.1  ...
4        None      1002            None       7.1    Sheksna        8.1  ...
...       ...       ...             ...       ...        ...        ...  ...
1450     None      2049            None       7.2       Ohau        8.2  ...
1451     None       219  Version 4 edit       5.0         Po        6.0  ...
1452     None    178000  Changed in 2.0       4.7      Loire        5.7  ...
1453     None       178  Changed in 4.0       4.7      Loire        5.7  ...
1454    Drava       303            None       6.0      Drava        7.0  ...

        name_pl       name_pt          name_ru    name_sv         name_tr  \
0       Irawadi  Rio IrauÃ¡di   Ð�Ñ�Ð°Ð²Ð°Ð´Ð¸  Irrawaddy  Ä°ravadi Nehri
1     Tonle Sap          None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None            None
2     Tonle Sap          None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None            None
3       Szeksna          None     Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna            None
4       Szeksna          None     Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna            None
...         ...           ...              ...        ...             ...
1450       None          None             None       None            None
1451        Pad        Rio PÃ³             Ð�Ð¾         Po        Po Nehri
1452      Loara    Rio Loire      Ð�Ñ�Ð°Ñ�Ð°      Loire     Loire Nehri
1453      Loara    Rio Loire      Ð�Ñ�Ð°Ñ�Ð°      Loire     Loire Nehri
1454      Drawa    Rio Drava      Ð�Ñ�Ð°Ð²Ð°      Drava           Drava

                 name_vi          name_zh wdid_score      ne_id  \
0     SÃ´ng Ayeyarwaddy  ä¼�æ´�ç�¦åº�æ±�          2  1159109417
1                   None             None          4  1159109429
2                   None             None          4  1159109445
3                   None  è��å��æ�¯ç´�æ²³          4  1159109447
4                   None  è��å��æ�¯ç´�æ²³          4  1159109461
...                  ...              ...        ...         ...
1450                None             None          4  1159129657
1451           SÃ´ng Po           æ³¢æ²³          4  1159129663
1452        SÃ´ng Loire      å�¢ç�¦å°�æ²³          4  1159129671
1453        SÃ´ng Loire      å�¢ç�¦å°�æ²³          4  1159129677
1454                None      å¾·æ��ç�¦æ²³          5  1159129685

                                               geometry
0     MULTIPOLYGON (((10138551.54 3049624.556, 10138...
1     POLYGON ((11183514.629 2708469.621, 11184239.3...
2     POLYGON ((11322060.743 2548275.499, 11321592.1...
3     POLYGON ((2523610.596 6878389.198, 2523565.166...
4     POLYGON ((2491759.669 6980840.76, 2491289.873 ...
...                                                 ...
1450  POLYGON ((2535034.246 -14768011.839, 2534683.4...
1451  POLYGON ((634107.252 4970775.329, 635297.987 4...
1452                                               None
1453  POLYGON ((247148.376 5224039.482, 246788.137 5...
1454  POLYGON ((1051399.884 5219971.477, 1055899.439...

[1455 rows x 35 columns]
      scalerank  natscale  labelrank            featurecla  \
0            10         1          8        Admin-1 capital
1            10         1          8        Admin-1 capital
2            10         1          8        Admin-1 capital
3            10         1          8        Admin-1 capital
4            10         1          8        Admin-1 capital
...         ...       ...        ...                    ...
7338          0       600          1        Admin-1 capital
7339          0       600          1        Admin-1 capital
7340          0       600          3        Admin-1 capital
7341          0       600          0        Admin-0 capital
7342          0       600          0  Admin-0 region capital

                         name namepar              namealt  diffascii  \
0     Colonia del Sacramento    None                 None          0
1                   Trinidad    None                 None          0
2                Fray Bentos    None                 None          0
3                  Canelones    None                 None          0
4                    Florida    None                 None          0
...                      ...     ...                  ...        ...
7338          Rio de Janeiro    None                 None          0
7339               São Paulo    None  Sao Paulo|Sio Paulo          0
7340                  Sydney    None                 None          0
7341               Singapore    None                 None          0
7342               Hong Kong    None                 None          0

                   nameascii  adm0cap  ...  rank_max rank_min  geonameid  \
0     Colonia del Sacramento      0.0  ...         7        7  3443013.0
1                   Trinidad      0.0  ...         7        7  3439749.0
2                Fray Bentos      0.0  ...         7        7  3442568.0
3                  Canelones      0.0  ...         6        6  3443413.0
4                    Florida      0.0  ...         7        7  3442585.0
...                      ...      ...  ...       ...      ...        ...
7338          Rio de Janeiro      0.0  ...        14       12  3451190.0
7339               Sao Paulo      0.0  ...        14       14  3448439.0
7340                  Sydney      0.0  ...        12       12  2147714.0
7341               Singapore      1.0  ...        13       12  1880252.0
7342               Hong Kong      0.0  ...        13       12  1819729.0

             meganame         ls_name ls_match checkme min_zoom       ne_id  \
0                None            None        0       0      9.0  1159112629
1                None            None        0       0      9.0  1159112647
2                None            None        0       0      9.0  1159112663
3                None            None        0       0      9.0  1159112679
4                None            None        0       0      7.0  1159112703
...               ...             ...      ...     ...      ...         ...
7338  Rio de Janeiro  Rio de Janeiro        1       0      1.7  1159151619
7339               S       Sao Paolo        1       0      3.0  1159151621
7340          Sydney         Sydney1        1       0      1.7  1159151623
7341       Singapore       Singapore        1       5      2.1  1159151627
7342       Hong Kong       Hong Kong        1       0      3.0  1159151629

                                               geometry
0     POLYGON ((-5524358.943 -4466465.068, -5524407....
1     POLYGON ((-5486417.74 -4327207.242, -5486465.8...
2     POLYGON ((-5636960.417 -4310525.118, -5637008....
3     POLYGON ((-5379035.389 -4435724.561, -5379083....
4     POLYGON ((-5395047.121 -4379773.605, -5395095....
...                                                ...
7338  POLYGON ((-4523570.96 -2782837.179, -4523619.1...
7339  POLYGON ((-4859015.78 -2903590.001, -4859063.9...
7340  POLYGON ((8838153.443 -12379152.889, 8838105.2...
7341  POLYGON ((11567431.615 268596.058, 11567383.46...
7342  POLYGON ((11408055.84 5122887.034, 11408007.68...

[7343 rows x 39 columns]
```

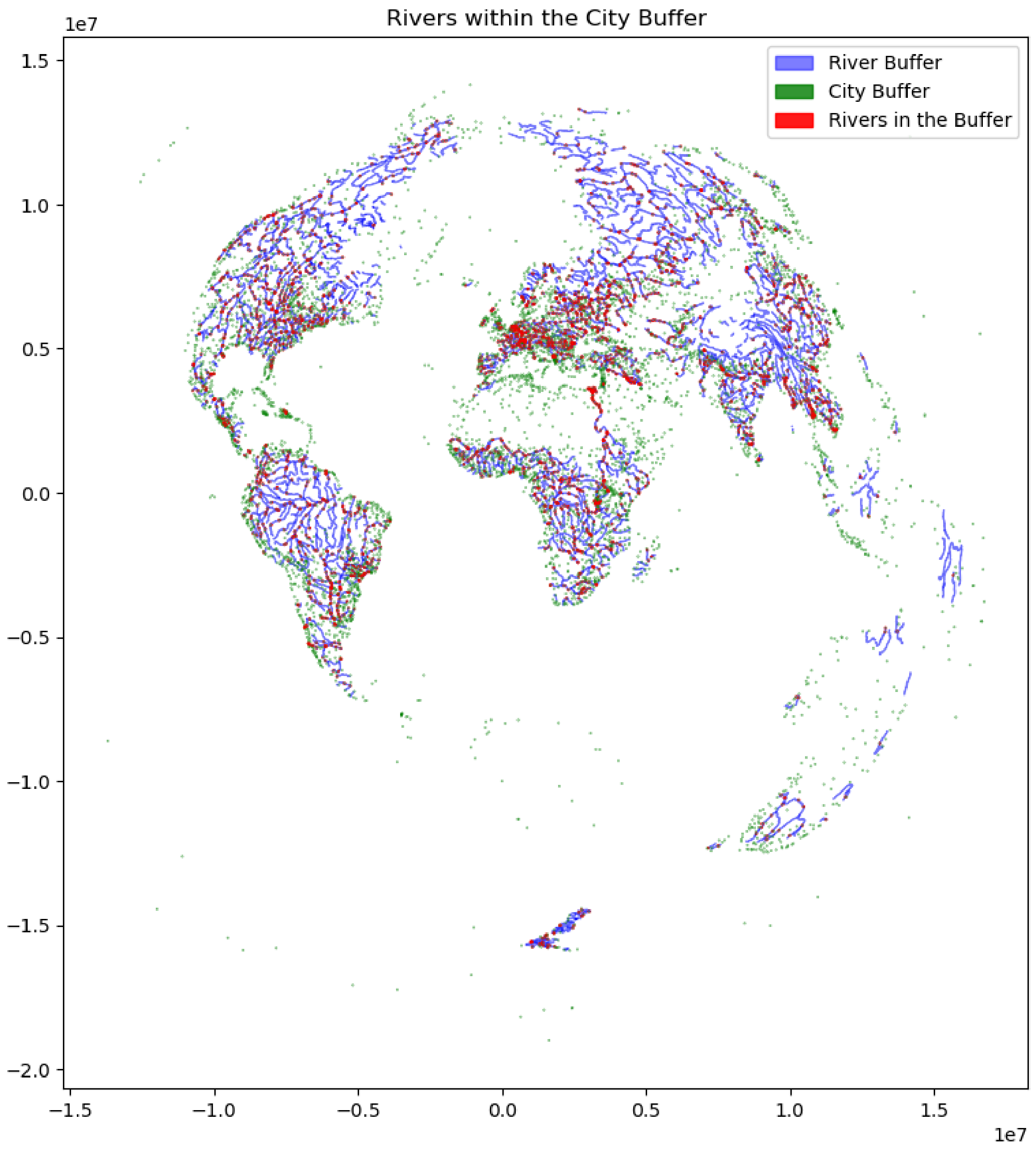


F. Now select only the rivers from the buffer created that intersect

In [11]:
```
# Perform spatial intersection to get only the parts of the rivers that intersec
rivers_within_buffer = gpd.overlay(rivers_buffer_gdf, cities_buffer_gdf, how='in
# Display the resulting GeoDataFrame to check the selected rivers
print(rivers_within_buffer)
print(rivers_buffer_gdf)

# Visualize the selected rivers within the  buffer
fig, ax = plt.subplots(figsize=(10, 10))

# Plot the river buffer
rivers_buffer_gdf.plot(ax=ax, color="lightblue", edgecolor='blue', linewidth=0.7

# Plot the original cities buffer to show the area of intersection
cities_buffer_gdf.plot(ax=ax, color="green", edgecolor="green", linewidth=1, alp

# Plot the rivers within the city buffer
rivers_within_buffer.plot(ax=ax, color="red", edgecolor="red", linewidth=1.5, al

# Custom Legend with mpatches
river_patch = mpatches.Patch(color="blue", alpha=0.5, label="River Buffer")
cities_patch = mpatches.Patch(color="green", alpha=0.8, label="City Buffer" )
intersected_rivers = mpatches.Patch(color="red", alpha=0.9, label="Rivers in the

# Add legend and title
ax.set_title("Rivers within the City Buffer")
ax.legend(handles=[river_patch, cities_patch, intersected_rivers])
plt.show()
```

```
      dissolve  scalerank_1 featurecla_1          name_1 name_alt  rivernum  \
0       0River          1.0        River  Irrawaddy Delta     None         0
1       0River          1.0        River  Irrawaddy Delta     None         0
2       0River          1.0        River  Irrawaddy Delta     None         0
3       0River          1.0        River  Irrawaddy Delta     None         0
4       0River          1.0        River  Irrawaddy Delta     None         0
...        ...          ...          ...             ...      ...       ...
2446  178River          5.0        River           Loire     None       178
2447  178River          5.0        River           Loire     None       178
2448  178River          5.0        River           Loire     None       178
2449   303Drau          7.0        River            Drau    Drava       303
2450   303Drau          7.0        River            Drau    Drava       303

              note_1  min_zoom_1    name_en  min_label  ... rank_max rank_min  \
0               None         2.0  Irrawaddy        3.0  ...        9        7
1               None         2.0  Irrawaddy        3.0  ...        7        7
2               None         2.0  Irrawaddy        3.0  ...        3        3
3               None         2.0  Irrawaddy        3.0  ...        9        9
4               None         2.0  Irrawaddy        3.0  ...       10        9
...              ...         ...        ...        ...  ...      ...      ...
2446  Changed in 4.0         4.7      Loire        5.7  ...        8        7
2447  Changed in 4.0         4.7      Loire        5.7  ...       10        9
2448  Changed in 4.0         4.7      Loire        5.7  ...        7        7
2449            None         6.0      Drava        7.0  ...        9        8
2450            None         6.0      Drava        7.0  ...        8        8

      geonameid meganame        ls_name ls_match checkme min_zoom_2  \
0     1314042.0     None       Letpadan        1       0        6.7
1     1289828.0     None         Wakema        1       0        6.1
2     1315244.0     None        Labutta        1       0        6.7
3     1325211.0     None       Hinthada        1       2        6.7
4     1328421.0     None        Pathein        1       0        6.7
...         ...      ...            ...      ...     ...        ...
2446  2983362.0     None         Roanne        1       0        7.0
2447  2980291.0     None  Saint-Etienne        1       0        6.7
2448  2990474.0     None         Nevers        1       0        6.7
2449  3195506.0     None        Maribor        1       2        6.7
2450  2774326.0     None     Klagenfurt        1       0        6.1

         ne_id_2                                           geometry
0     1159145911  POLYGON ((10109298.54 3257113.535, 10109334.13...
1     1159145943  POLYGON ((10132978.362 3033470.506, 10132961.1...
2     1159145949  POLYGON ((10111513.814 2922930.967, 10110561.2...
3     1159145963  POLYGON ((10102793.021 3225143.052, 10102462.1...
4     1159145967  POLYGON ((10081986.441 3042426.302, 10082510.3...
...          ...                                                ...
2446  1159129807  POLYGON ((343682.726 5109674.409, 343676.896 5...
2447  1159139913  POLYGON ((379670.909 5045906.978, 379717.343 5...
2448  1159141991  POLYGON ((279956.838 5206598.25, 279812.843 52...
2449  1159135827  POLYGON ((1338559.469 5199156.147, 1337948.925...
2450  1159145087  POLYGON ((1214679.971 5210510.032, 1217423.536...

[2451 rows x 73 columns]
                dissolve  scalerank       featurecla             name  \
0                 0River        1.0            River  Irrawaddy Delta
1     1001Lake Centerline        9.0  Lake Centerline        Tonle Sap
2              1001River        9.0            River        Tonle Sap
3     1002Lake Centerline        9.0  Lake Centerline          Sheksna
4              1002River        9.0            River          Sheksna
...                  ...        ...              ...              ...
```

```
1450  2049Lake Centerline      10.0  Lake Centerline            Ohau
1451             219River        6.0            River              Po
1452             178River        5.0            River           Loire
1453             178River        5.0            River           Loire
1454              303Drau        7.0            River            Drau

     name_alt  rivernum            note  min_zoom    name_en  min_label  ...  \
0        None         0            None       2.0  Irrawaddy        3.0  ...
1        None      1001            None       7.1       None        8.1  ...
2        None      1001            None       7.1       None        8.1  ...
3        None      1002            None       7.1    Sheksna        8.1  ...
4        None      1002            None       7.1    Sheksna        8.1  ...
...       ...       ...             ...       ...        ...        ...  ...
1450     None      2049            None       7.2       Ohau        8.2  ...
1451     None       219  Version 4 edit       5.0         Po        6.0  ...
1452     None    178000  Changed in 2.0       4.7      Loire        5.7  ...
1453     None       178  Changed in 4.0       4.7      Loire        5.7  ...
1454    Drava       303            None       6.0      Drava        7.0  ...

        name_pl       name_pt           name_ru    name_sv         name_tr  \
0       Irawadi  Rio IrauÃ¡di    Ð�Ñ�Ð°Ð²Ð°Ð´Ð¸  Irrawaddy  Ä°ravadi Nehri
1     Tonle Sap          None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None            None
2     Tonle Sap          None  Ð¢Ð¾Ð½Ð»ÐµÑ�Ð°Ð¿       None            None
3       Szeksna          None      Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna            None
4       Szeksna          None      Ð¨ÐµÐºÑ�Ð½Ð°    Sjeksna            None
...         ...           ...               ...        ...             ...
1450       None          None              None       None            None
1451        Pad        Rio PÃ³              Ð�Ð¾         Po        Po Nehri
1452      Loara    Rio Loire        Ð�Ñ�Ð°Ñ�Ð°      Loire     Loire Nehri
1453      Loara    Rio Loire        Ð�Ñ�Ð°Ñ�Ð°      Loire     Loire Nehri
1454      Drawa    Rio Drava        Ð�Ñ�Ð°Ð²Ð°      Drava           Drava

                name_vi           name_zh wdid_score       ne_id  \
0     SÃ´ng Ayeyarwaddy  ä¼�æ´�ç�¦åº�æ±�          2  1159109417
1                  None              None          4  1159109429
2                  None              None          4  1159109445
3                  None  è��å��æ�¯ç´�æ²³          4  1159109447
4                  None  è��å��æ�¯ç´�æ²³          4  1159109461
...                 ...               ...        ...         ...
1450               None              None          4  1159129657
1451          SÃ´ng Po            æ³¢æ²³          4  1159129663
1452       SÃ´ng Loire      å�¢ç�¦å°�æ²³          4  1159129671
1453       SÃ´ng Loire      å�¢ç�¦å°�æ²³          4  1159129677
1454               None      å¾·æ��ç�¦æ²³          5  1159129685

                                               geometry
0     MULTIPOLYGON (((10138551.54 3049624.556, 10138...
1     POLYGON ((11183514.629 2708469.621, 11184239.3...
2     POLYGON ((11322060.743 2548275.499, 11321592.1...
3     POLYGON ((2523610.596 6878389.198, 2523565.166...
4     POLYGON ((2491759.669 6980840.76, 2491289.873 ...
...                                                 ...
1450  POLYGON ((2535034.246 -14768011.839, 2534683.4...
1451  POLYGON ((634107.252 4970775.329, 635297.987 4...
1452                                               None
1453  POLYGON ((247148.376 5224039.482, 246788.137 5...
1454  POLYGON ((1051399.884 5219971.477, 1055899.439...

[1455 rows x 35 columns]
```

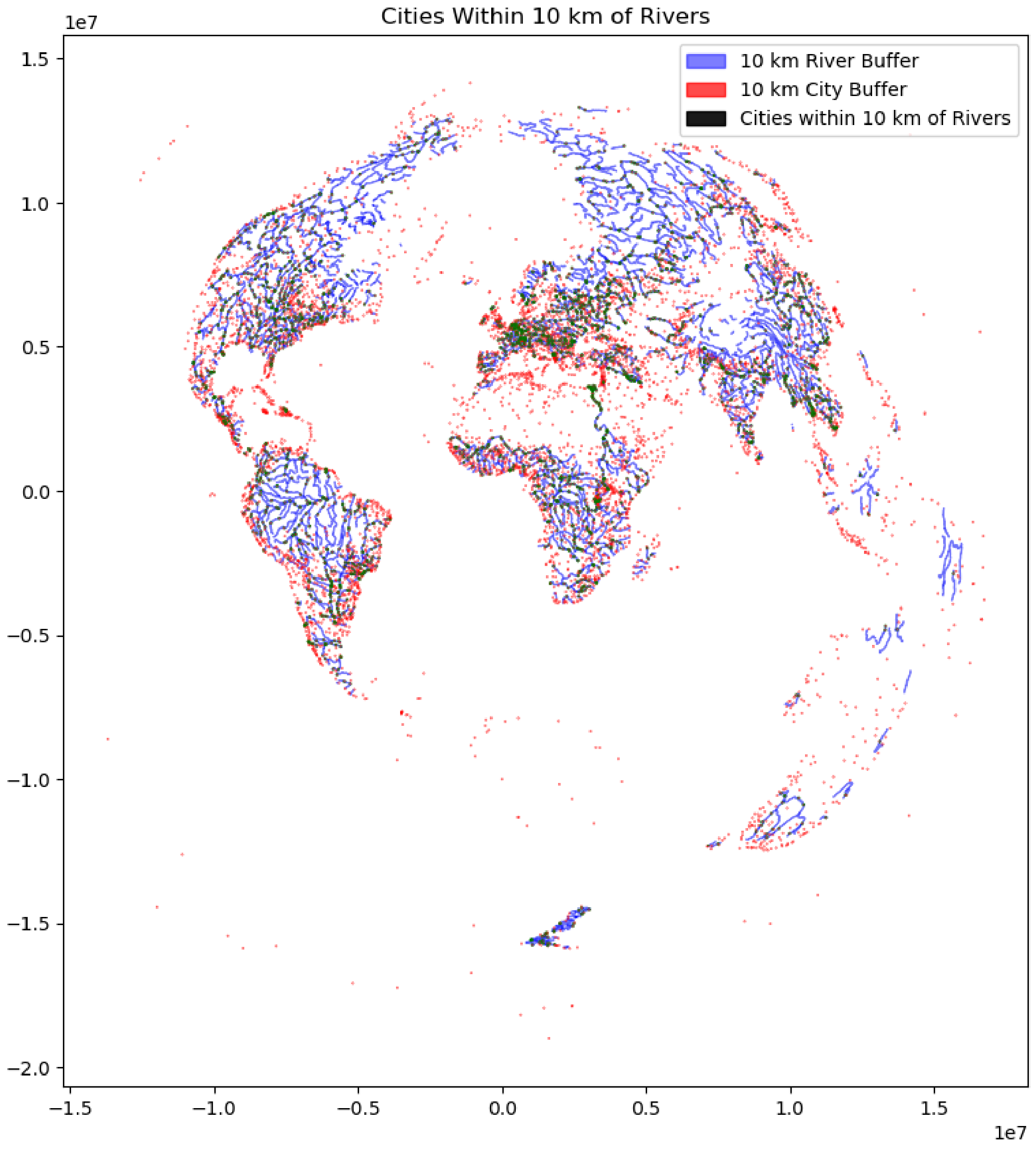


G. select features from the buffered places that intersect with the buffered river lines.

In [39]:

```
import geopandas as gpd
import matplotlib.pyplot as plt

# Perform the spatial intersection between city buffers and river buffers
cities_near_rivers = gpd.overlay(cities_buffer_gdf, rivers_buffer_gdf, how='inte

# Plot to visualize the intersected features
fig, ax = plt.subplots(figsize=(10, 10))

# Plot the river buffer
rivers_buffer_gdf.plot(ax=ax, color="lightblue", edgecolor='blue', linewidth=0.7

# Plot the city buffer
```

```
cities_buffer_gdf.plot(ax=ax, color="red", edgecolor='red', linewidth=1, alpha=0

# Plot only the intersecting features (cities near rivers)
cities_near_rivers.plot(ax=ax, color="black", edgecolor='green', linewidth=1.5,

# UserWarning: Legend does not support handles for PatchCollection instances.
# # Add legend and title
# ax.legend()
# ax.set_title("Cities Within 10 km of Rivers")
# plt.show()

# Custom legend with mpatches
river_patch = mpatches.Patch(color="blue", alpha=0.5, label="10 km River Buffer"
city_patch = mpatches.Patch(color="red", alpha=0.7, label="10 km City Buffer")
intersection_patch = mpatches.Patch(color="black", alpha=0.9, label="Cities with

# Add custom legend
ax.legend(handles=[river_patch, city_patch, intersection_patch])
ax.set_title("Cities Within 10 km of Rivers")
plt.show()

# print the intersected GeoDataFrame
print(cities_near_rivers)
print(cities_buffer_gdf)
```



Cities Within 10 km of Rivers
1e7
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1e7
10 km River Buffer
10 km City Buffer
Cities within 10 km of Rivers

```
      scalerank_1  natscale  labelrank     featurecla_1      name_1   namepar  \
0              10         1          5  Admin-1 capital     Yên Bái      None   
1              10         1          5  Admin-1 capital   Thái Bình      None   
2              10         1          5  Admin-1 capital    Tuy Hòa      None   
3              10         1          5  Admin-1 capital    Cao Lãnh      None   
4              10         1          5  Admin-1 capital  Truc Giang      None   
...           ...       ...        ...              ...         ...       ...   
2446            0       600          1  Admin-1 capital    Shanghai      None   
2447            0       600          3  Admin-0 capital       Paris      None   
2448            0       600          3  Admin-0 capital       Paris      None   
2449            0       600          1  Admin-1 capital     Kolkata  Calcutta   
2450            0       600          1  Admin-1 capital   São Paulo      None   

                  nameaslt  diffascii   nameascii  adm0cap  ...  \
0                     None          0     Yen Bai      0.0  ...   
1                     None          0   Thai Binh      0.0  ...   
2                     None          0     Tuy Hoa      0.0  ...   
3                     None          0    Cao Lanh      0.0  ...   
4                     None          0  Truc Giang      0.0  ...   
...                    ...        ...         ...      ...  ...   
2446                  None          0    Shanghai      0.0  ...   
2447                  None          0       Paris      1.0  ...   
2448                  None          0       Paris      1.0  ...   
2449                  None          0     Kolkata      0.0  ...   
2450  Sao Paulo|Sio Paulo          0   Sao Paulo      0.0  ...   

             name_pl        name_pt       name_ru       name_sv       name_tr  \
0     Rzeka Czerwona  Rio Vermelho  Ð¥Ð¾Ð½Ð³Ñ�Ð°  RÃ¶da floden  KÄ±zÄ±l Nehir   
1     Rzeka Czerwona  Rio Vermelho  Ð¥Ð¾Ð½Ð³Ñ�Ð°  RÃ¶da floden  KÄ±zÄ±l Nehir   
2               None          None          None          None          None   
3             Mekong   Rio Mekong  Ð�ÐµÐºÐ¾Ð½Ð³        Mekong        Mekong   
4               None          None          None          None          None   
...              ...           ...           ...           ...           ...   
2446            None          None          None          None          None   
2447           Marna    Rio Marne     Ð�Ð°Ñ�Ð½Ð°         Marne   Marne Nehri   
2448         Sekwana     Rio Sena      Ð¡ÐµÐ½Ð°         Seine     Sen Nehri   
2449          Ganges   Rio Ganges     Ð�Ð°Ð½Ð³        Ganges    Ganj Nehri   
2450          TietÃª    Rio TietÃª    Ð¢Ð¸ÐµÑ�Ðµ    Rio TietÃª          None   

            name_vi     name_zh wdid_score     ne_id_2  \
0     SÃ´ng Há»�ng      çº¢æ²³          5  1159116785   
1     SÃ´ng Há»�ng      çº¢æ²³          5  1159116785   
2              None        None          0  1159111003   
3         MÃª KÃ´ng  æ¹�å�¬æ²³          4  1159121023   
4              None        None          0  1159113739   
...             ...         ...        ...         ...   
2446           None        None          0  1159123017   
2447   SÃ´ng Marne  é©¬æ�©æ²³          4  1159110541   
2448   SÃ´ng Seine  å¡�çº³æ²³          4  1159112177   
2449  SÃ´ng Háº±ng      æ��æ²³          4  1159122643   
2450           None  é�µç�¹æ²³          4  1159125573   

                                               geometry  
0     POLYGON ((10695369.377 4391711.374, 10695225.3...  
1     POLYGON ((10932426.633 4231744.893, 10931767.5...  
2     POLYGON ((11769956.218 2891404.917, 11769812.2...  
3     POLYGON ((11529816.651 2205405.833, 11529672.6...  
4     POLYGON ((11601201.356 2174161.458, 11600542.3...  
...                                                 ...  
2446  POLYGON ((10577469.109 7502299.273, 10576884.5...  
2447  POLYGON ((203498.335 5415218.185, 203354.341 5...
2448  POLYGON ((203498.335 5415218.185, 203354.341 5...
2449  POLYGON ((9095777.263 3747144.52, 9094853.276 ...
2450  POLYGON ((-4859063.932 -2904570.173, -4859207....

[2451 rows x 73 columns]
      scalerank  natscale  labelrank             featurecla  \
0            10         1          8        Admin-1 capital
1            10         1          8        Admin-1 capital
2            10         1          8        Admin-1 capital
3            10         1          8        Admin-1 capital
4            10         1          8        Admin-1 capital
...         ...       ...        ...                    ...
7338          0       600          1        Admin-1 capital
7339          0       600          1        Admin-1 capital
7340          0       600          3        Admin-1 capital
7341          0       600          0        Admin-0 capital
7342          0       600          0  Admin-0 region capital

                       name namepar             namealt  diffascii  \
0     Colonia del Sacramento    None                None          0
1                   Trinidad    None                None          0
2                Fray Bentos    None                None          0
3                  Canelones    None                None          0
4                    Florida    None                None          0
...                      ...     ...                 ...        ...
7338          Rio de Janeiro    None                None          0
7339               São Paulo    None  Sao Paulo|Sio Paulo          0
7340                  Sydney    None                None          0
7341               Singapore    None                None          0
7342               Hong Kong    None                None          0

                  nameascii  adm0cap  ...  rank_max rank_min  geonameid  \
0     Colonia del Sacramento      0.0  ...         7        7  3443013.0
1                   Trinidad      0.0  ...         7        7  3439749.0
2                Fray Bentos      0.0  ...         7        7  3442568.0
3                  Canelones      0.0  ...         6        6  3443413.0
4                    Florida      0.0  ...         7        7  3442585.0
...                      ...      ...  ...       ...      ...        ...
7338          Rio de Janeiro      0.0  ...        14       12  3451190.0
7339               Sao Paulo      0.0  ...        14       14  3448439.0
7340                  Sydney      0.0  ...        12       12  2147714.0
7341               Singapore      1.0  ...        13       12  1880252.0
7342               Hong Kong      0.0  ...        13       12  1819729.0

            meganame         ls_name ls_match checkme min_zoom       ne_id  \
0               None            None        0       0      9.0  1159112629
1               None            None        0       0      9.0  1159112647
2               None            None        0       0      9.0  1159112663
3               None            None        0       0      9.0  1159112679
4               None            None        0       0      7.0  1159112703
...              ...             ...      ...     ...      ...         ...
7338  Rio de Janeiro  Rio de Janeiro        1       0      1.7  1159151619
7339               S       Sao Paolo        1       0      3.0  1159151621
7340          Sydney         Sydney1        1       0      1.7  1159151623
7341       Singapore       Singapore        1       5      2.1  1159151627
7342       Hong Kong       Hong Kong        1       0      3.0  1159151629

                                               geometry
0     POLYGON ((-5524358.943 -4466465.068, -5524407....
1     POLYGON ((-5486417.74 -4327207.242, -5486465.8...
2     POLYGON ((-5636960.417 -4310525.118, -5637008....
3     POLYGON ((-5379035.389 -4435724.561, -5379083....
4     POLYGON ((-5395047.121 -4379773.605, -5395095....
...                                                ...
7338  POLYGON ((-4523570.96 -2782837.179, -4523619.1...
7339  POLYGON ((-4859015.78 -2903590.001, -4859063.9...
7340  POLYGON ((8838153.443 -12379152.889, 8838105.2...
7341  POLYGON ((11567431.615 268596.058, 11567383.46...
7342  POLYGON ((11408055.84 5122887.034, 11408007.68...

[7343 rows x 39 columns]
```

H. Save this as a kml file and view on Google earth

In [43]:
```
print(cities_near_rivers.columns)
print(cities_near_rivers)

# # Save the rivers within the city buffer as a KML file
# output_kml_path = r"C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\

# # Save to KML
# cities_near_rivers.to_file(output_kml_path, driver='KML')
# print(f"File saved as {output_kml_path}. You can now open this file in Google
#                                              # BY using the above code we get th
#                                              # have values that cannot be fully

# Selecting essential columns for KML export
selected_columns = ['name_1', 'nameascii', 'adm0name', 'iso_a2', 'pop_max', 'pop
cities_in_river_buffer = cities_near_rivers[selected_columns]

# Save to KML
output_kml_path = r"C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\La
cities_in_river_buffer.to_file(output_kml_path, driver='KML')
print(f"File saved as {output_kml_path}. You can now open this file in Google Ea
Index(['scalerank_1', 'natscale', 'labelrank', 'featurecla_1', 'name_1',
       'namepar', 'namealt', 'diffascii', 'nameascii', 'adm0cap', 'capalt',
       'capin', 'worldcity', 'megacity', 'sov0name', 'sov_a3', 'adm0name',
       'adm0_a3', 'adm1name', 'iso_a2', 'note_1', 'latitude', 'longitude',
       'changed', 'namediff', 'diffnote', 'pop_max', 'pop_min', 'pop_other',
       'rank_max', 'rank_min', 'geonameid', 'meganame', 'ls_name', 'ls_match',
       'checkme', 'min_zoom_1', 'ne_id_1', 'dissolve', 'scalerank_2',
       'featurecla_2', 'name_2', 'name_alt', 'rivernum', 'note_2',
       'min_zoom_2', 'name_en', 'min_label', 'label', 'wikidataid', 'name_ar',
       'name_bn', 'name_de', 'name_es', 'name_fr', 'name_el', 'name_hi',
       'name_hu', 'name_id', 'name_it', 'name_ja', 'name_ko', 'name_nl',
       'name_pl', 'name_pt', 'name_ru', 'name_sv', 'name_tr', 'name_vi',
       'name_zh', 'wdid_score', 'ne_id_2', 'geometry'],
      dtype='object')
      scalerank_1  natscale  labelrank     featurecla_1      name_1   namepar  \
0              10         1          5  Admin-1 capital     Yên Bái      None   
1              10         1          5  Admin-1 capital   Thái Bình      None   
2              10         1          5  Admin-1 capital     Tuy Hòa      None   
3              10         1          5  Admin-1 capital    Cao Lãnh      None   
4              10         1          5  Admin-1 capital  Truc Giang      None   
...           ...       ...        ...              ...         ...       ...   
2446            0       600          1  Admin-1 capital    Shanghai      None   
2447            0       600          3  Admin-0 capital       Paris      None   
2448            0       600          3  Admin-0 capital       Paris      None   
2449            0       600          1  Admin-1 capital     Kolkata  Calcutta   
2450            0       600          1  Admin-1 capital   São Paulo      None   

                  namealt  diffascii   nameascii  adm0cap  ...  \
0                    None          0     Yen Bai      0.0  ...   
1                    None          0   Thai Binh      0.0  ...   
2                    None          0     Tuy Hoa      0.0  ...   
3                    None          0    Cao Lanh      0.0  ...   
4                    None          0  Truc Giang      0.0  ...   
...                   ...        ...         ...      ...  ...   
2446                 None          0    Shanghai      0.0  ...   
2447                 None          0       Paris      1.0  ...   
2448                 None          0       Paris      1.0  ...   
2449                 None          0     Kolkata      0.0  ...   
2450  Sao Paulo|Sio Paulo          0   Sao Paulo      0.0  ...   

             name_pl       name_pt        name_ru        name_sv        name_tr  \
0     Rzeka Czerwona  Rio Vermelho  Ð¥Ð¾Ð½Ð³Ñ⍰Ð°  RÃ¶da floden  KÄ±zÄ±l Nehir   
1     Rzeka Czerwona  Rio Vermelho  Ð¥Ð¾Ð½Ð³Ñ⍰Ð°  RÃ¶da floden  KÄ±zÄ±l Nehir   
2               None          None           None           None           None   
3             Mekong   Rio Mekong  Ð⍰ÐµÐºÐ¾Ð½Ð³         Mekong         Mekong   
4               None          None           None           None           None   
...              ...           ...            ...            ...            ...   
2446            None          None           None           None           None   
2447           Marna    Rio Marne    Ð⍰Ð°Ñ⍰Ð½Ð°          Marne    Marne Nehri   
2448         Sekwana     Rio Sena      Ð¡ÐµÐ½Ð°          Seine      Sen Nehri   
2449          Ganges   Rio Ganges      Ð⍰Ð°Ð½Ð³         Ganges     Ganj Nehri   
2450          TietÃª    Rio TietÃª    Ð¢Ð¸ÐµÑ⍰Ðµ     Rio TietÃª           None   

          name_vi    name_zh wdid_score     ne_id_2  \
0     SÃ´ng Há»⍰ng     çº¢æ²³          5  1159116785   
1     SÃ´ng Há»⍰ng     çº¢æ²³          5  1159116785   
2             None       None          0  1159111003   
3        MÃª KÃ´ng  æ¹⍰å⍰¬æ²³          4  1159121023   
4             None       None          0  1159113739   
...            ...        ...        ...         ...   
```

```
2446          None       None           0  1159123017
2447   SÃ´ng Marne  é©¬æ�©æ²³           4  1159110541
2448   SÃ´ng Seine  å¡�çº³æ²³           4  1159112177
2449  SÃ´ng Háº±ng       æ��æ²³           4  1159122643
2450          None  é�µç�¹æ²³           4  1159125573

                                              geometry
0     POLYGON ((10695369.377 4391711.374, 10695225.3...
1     POLYGON ((10932426.633 4231744.893, 10931767.5...
2     POLYGON ((11769956.218 2891404.917, 11769812.2...
3     POLYGON ((11529816.651 2205405.833, 11529672.6...
4     POLYGON ((11601201.356 2174161.458, 11600542.3...
...                                                ...
2446  POLYGON ((10577469.109 7502299.273, 10576884.5...
2447  POLYGON ((203498.335 5415218.185, 203354.341 5...
2448  POLYGON ((203498.335 5415218.185, 203354.341 5...
2449  POLYGON ((9095777.263 3747144.52, 9094853.276 ...
2450  POLYGON ((-4859063.932 -2904570.173, -4859207....

[2451 rows x 73 columns]
File saved as C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\Lab-6\cit
ies_in_river_buffer.kml. You can now open this file in Google Earth.
```

Just for visual representation of kml file by load to the OSM map by using folium

In [45]:
```python
import folium
from folium import plugins
import geopandas as gpd

# Read the KML file using GeoPandas
gdf = gpd.read_file(r"C:\Users\Ranjeet Gupta\Downloads\Scientific Computing Lab\

# Create a base map
m = folium.Map(location=[gdf.geometry.centroid.y.mean(), gdf.geometry.centroid.x

# Add KML layer to the map
folium.GeoJson(gdf).add_to(m)

# Display the map in the notebook
m
```

```
C:\Users\Ranjeet Gupta\AppData\Local\Temp\ipykernel_22988\301996777.py:9: UserWar
ning: Geometry is in a geographic CRS. Results from 'centroid' are likely incorre
ct. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS before t
his operation.

  m = folium.Map(location=[gdf.geometry.centroid.y.mean(), gdf.geometry.centroid.
x.mean()], zoom_start=2)
```

```
Out[45]: Make this Notebook Trusted to load map: File -> Trust Notebook
```

```
In [ ]:
```