Unit 4

Spatial Data Visualization

Spatial data visualization is the process of representing spatial data (data having geometric coordinates) in a visual way that helps people to understand its meaning. It is used in fields like geography, environmental science, transportation, business etc. It uses geographic or location-based information in various field to analyze, understand and communicate data. some of the ways to visualize spatial data are maps (use symbols to represent on the earth's surface such as points, lines and polygons), heatmaps (use color to represent the intensity of some phenomenon) etc. Spatial data can be collected through various sources including GPS, satellite image, surveys and geospatial sensors. some advantages of spatial data visualization are;

- It can reveal patterns and trends that would not be visible in tabular data
- It can make data more accessible to a wider audience
- It can be used to generate new ideas and insights

Using python spatial data can be visualize using Folium (display variety of maps, including point, maps, heatmaps and choropleth maps), geopandas which combines the power of Pandas and shapely to create and manipulate geospatial data frame and used to visualize in different map form.

Scalar field:

In data visualization, a scalar field is a mathematical concept used to represent data values that are associated with specific positions or points in space. A scalar field assigns a single scalar value (such as a number or color) to each point in a domain, often a two-dimensional or three-dimensional space. Scalar fields are commonly used in various fields, including physics, engineering, geography, and data science, for visualizing and analyzing data.

A scalar field is defined as a function that maps each point in a given domain to a scalar value. Mathematically, if you have a domain represented as D in n-dimensional space ($D \subseteq \mathbb{R}^n$), a scalar field is a function $f: D \rightarrow \mathbb{R}$ that assigns a real number to each point in D. Scalar fields are often visualized using color maps or contour plots. Each data point's scalar value is represented by a color or contour level, making it easy to see how the scalar values vary across the domain.

When visualizing scalar fields, a color map is often used to map scalar values to colors. Common color maps include grayscale, rainbow. Care must be taken to choose an appropriate color map to accurately represent the data without introducing visual artifacts or biases. Scalar fields can be visualized using contour plots, where contour lines connect points with the same scalar value. Contour plots are useful for identifying patterns and boundaries in data.

Application of scalar fields:

Temperature Mapping: Scalar fields can be used to represent temperature distributions in various physical systems, such as weather maps or heat diffusion in materials.

Elevation Mapping: Scalar fields can represent the elevation of terrain, helping create topographic maps.

Data Analysis: In data science, scalar fields are used to visualize data distributions and patterns, such as heatmaps, where color represents data values.

Fluid Dynamics: Scalar fields are employed to visualize various properties in fluid dynamics, like pressure or velocity fields in fluid flow simulations.

Therefore, a scalar field in data visualization is a fundamental concept for representing data values at different points in space using colors or contour lines. It is widely used in various fields to visualize and analyze data distributions, patterns, and properties.

Isocontours:

Isocontours, short for "isoline contours," are lines or curves on a two-dimensional graph or map that connect points with the same constant value. These lines represent the boundaries of regions with equal values of a continuous variable, typically within a geographic or spatial context. Isocontours are commonly used in data visualization, especially in the fields of geography, geology, meteorology, and engineering, to represent various continuous phenomena. Here's how isocontours work and are used in data visualization:

- **Representation of data**: Isocontours are used to visualize data where the variable of interest varies continuously over a spatial domain. This variable could be elevation (topographic maps), temperature, pressure, rainfall, or any other measurable quantity.
- **Contour lines**: Isocontours are typically represented as contour lines on a 2D map or graph. Each contour line connects points with the same value of the variable being visualized. For example, if you're creating a topographic map, each contour line represents a specific elevation, connecting all points at that elevation.
- **Interpolation**: In many cases, data points are not available at every location on the map. Isocontours are created by interpolating between data points to estimate values at non-sample locations. Various interpolation techniques, such as kriging or spline interpolation, may be used to generate these lines smoothly.
- **Spacing**: The spacing between isocontours can vary based on the data and the desired level of detail in the visualization. Closer contours represent a more rapid change in the variable, while wider spacing indicates a gentler gradient.
- **Color filled contours**: In addition to simple contour lines, isocontours can also be represented using color-filled regions. In this case, areas between two consecutive

contour lines are filled with a color gradient to represent the continuous variation of the variable.

Application:

- Topographic Maps: Isocontours are used to represent elevation on topographic maps, allowing viewers to understand the shape and steepness of the terrain.
- Weather Maps: Meteorologists use isocontours to represent variables like temperature, pressure, and precipitation levels on weather maps.
- Geological Maps: Geologists use isocontours to visualize features like rock layers, fault lines, and mineral concentrations.
- Engineering and Environmental Studies: Isocontours are used to study factors like groundwater levels, pollution concentration, or soil properties.

Example on Scalar field and Isocontours

To visualize 2D scalar field and iscontours , we use meshgrid(X, y) function of numpy to generate two dimensional grid (X and Y represent x and y coordinate of mesh grid) and to visualize scalar field we will use pcolormesh() function of matplotlib.

For scalar field visualization:

Step 1: import necessary library

Step 2: creating x and y array and setting limit

x = np.linspace(-3., 3., n)

y = np.linspace(-3., 3., n)

here n = 256

Step 3: converting array into mesh grid

X, Y = np.meshgrid(x, y)

Step 4: Computing extra coordinate Z for scalar field

Z = X * np.sinc(X ** 2 + Y ** 2)

Step 5: plotting scalar field using pcolormesh()

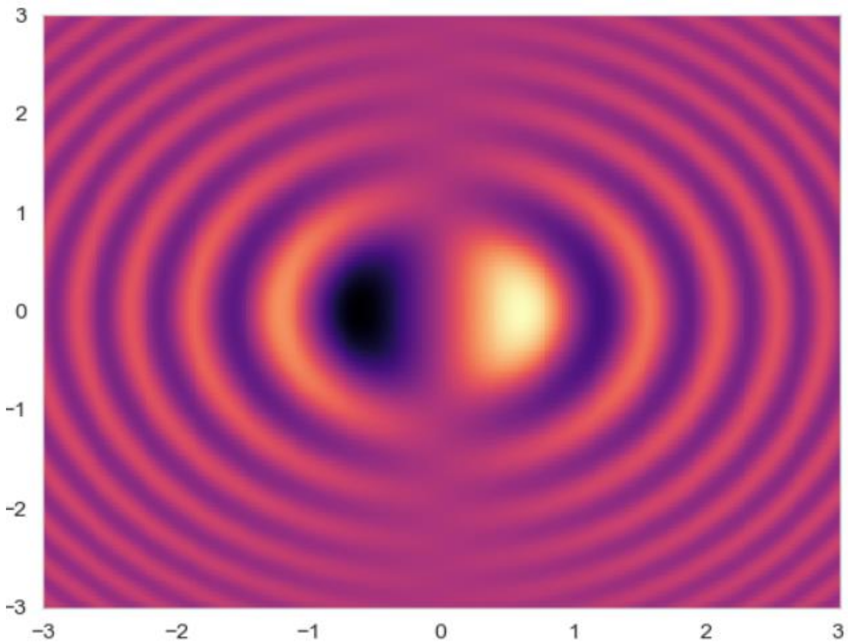
plt.pcolormesh(X, Y, Z, cmap = 'magma')

plt.show()

Full Program:

```
# Imprting necessary libraries

import numpy as np
```

```
from matplotlib import pyplot as plt
import matplotlib.cm as cm
# Setting our linspace limit:
n = 256
# Creating X and Y arrays:
x = np.linspace(-3., 3., n)
y = np.linspace(-3., 3., n)
# Converting arrays into meshgrid:
X, Y = np.meshgrid(x, y)
# Computing Z: 2D Scalar Field
Z = X * np.sinc(X ** 2 + Y ** 2)
# Plotting our 2D Scalar Field using pcolormesh()
plt.pcolormesh(X, Y, Z, cmap = 'magma')
plt.show()
```



Visualizing using isocontours:

Step1: import libraries

import numpy as np

import matplotlib.pyplot as plt

Step 2: create a grid of x and y values

x = np.linspace(-5, 5, 100)

y = np.linspace(-5, 5, 100)

X, Y = np.meshgrid(x, y)

Step 3: create extra variable for isocontours

Z = np.exp(-(X**2 + Y**2) / 5)

Step 4: create a countour plot

plt.figure(figsize=(8, 6))

contours = plt.contour(X, Y, Z, levels=10, cmap='viridis') # Change the number of levels as needed

Add labels and a colorbar

plt.xlabel('X-axis')

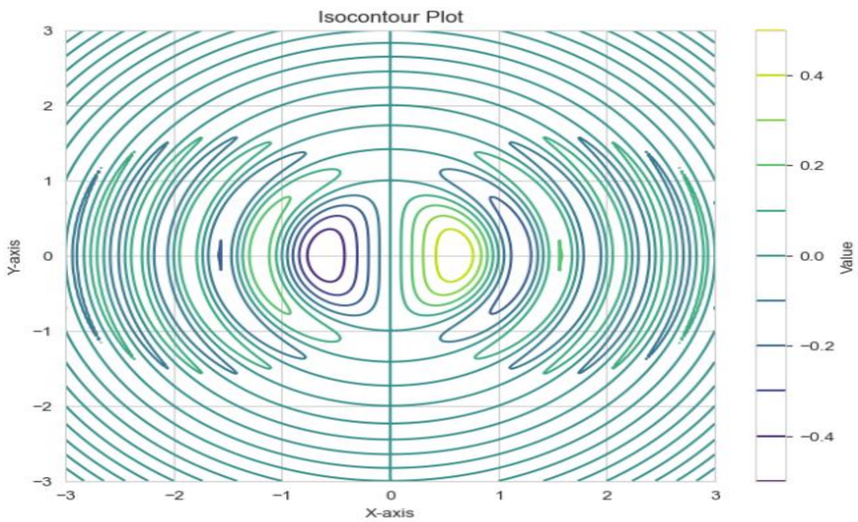
plt.ylabel('Y-axis')

plt.title('Isocontour Plot')

plt.colorbar(contours, label='Value')

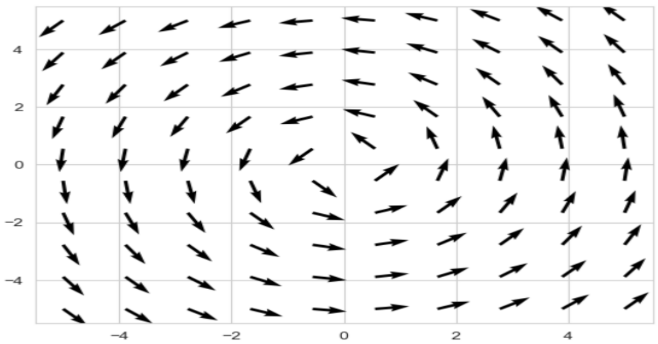
Show the plot

plt.show()



Visualizing scalar field using quiver plot

```
# Import necessary libraries

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

# Creating meshgrid

x,y = np.meshgrid(np.linspace(-5,5,10),np.linspace(-5,5,10))

# Emulating the given equation

u = -y/np.sqrt(x**2 + y**2)

v = x/np.sqrt(x**2 + y**2)

# Quiver plot

plt.quiver(x,y,u,v)

plt.show()
```



Scalar Volume

A scalar volume in data visualization refers to a three-dimensional dataset where scalar values (single values, typically real numbers) are assigned to each point or voxel in a 3D grid or space. Scalar volumes are commonly used in fields such as medical imaging, scientific visualization, and computational simulations to represent a wide range of continuous data, including temperature, density, pressure, concentration, and more. In the context of scalar volumes, each element in the 3D grid is often referred to as a "voxel," which is analogous to a pixel in 2D images. In volume rendering, transfer functions are crucial for mapping scalar values to color and opacity. These functions allow you to emphasize specific features or regions of interest within the scalar volume.

Visualization technique for Scalar Volume

- Isosurface Rendering: Isosurface rendering is a common technique for visualizing scalar volumes. It involves creating surfaces within the volume that enclose regions where the scalar value equals a specified threshold (isosurface). These surfaces are often displayed with shading to highlight shape and structure.
- Volume Rendering: Volume rendering techniques, such as ray casting or texture-based rendering, visualize the entire scalar volume. Different transfer functions map scalar values to color and opacity, allowing you to explore the internal structures and variations within the volume.
- Slicing and Cross-Sections: Slicing the scalar volume to create 2D cross-sections or slices at specific planes (e.g., axial, sagittal, coronal) can provide insights into the internal structure at different depths or orientations.
- Histograms and Statistical Summaries: Visualization techniques can include histograms and statistical summaries of scalar values within the volume to provide an overview of the data distribution and characteristics.

Application:

- Medical Imaging: Scalar volumes are extensively used in medical imaging, such as CT scans, MRI scans, and 3D reconstructions of anatomical structures.
- Scientific Simulations: Scalar volumes are generated in simulations of physical phenomena, like fluid dynamics, heat transfer, and materials science.
- Geospatial Analysis: In geospatial applications, scalar volumes can represent attributes like soil properties, groundwater levels, or seismic data.

Visualizing scalar volume using python:

Visualizing scalar volumes in Python typically involves using specialized libraries like VTK (Visualization Toolkit) or Mayavi. Here's an example using Mayavi, a Python library for 3D scientific data visualization:

Step1: install mayavi by: pip install mayavi

Step2: import libraries:

```
Import numpy as np

From mayavi import mlab
```

Step 3: create 3D grid and scalar data i.e. 3D matrix

```
x, y, z = np.mgrid[-5:5:20j, -5:5:20j, -5:5:20j]

scalar_field = np.sin(np.sqrt(x**2 + y**2 + z**2))
```

Step 4: create mayavi function

```
mlab.figure(size=(800, 600), bgcolor=(1, 1, 1))
```

Step 5: Create volume rendering of scalar field

```
vol = mlab.pipeline.volume(mlab.pipeline.scalar_field(scalar_field), colormap='coolwarm')
```

Step 6: view the volume

```
# Add a color bar

mlab.colorbar(title='Scalar Values')

# View the volume from different angles (optional)

mlab.view(azimuth=45, elevation=45, distance=10, focalpoint=(0, 0, 0))

# Show the Mayavi visualization

mlab.show()
```

Topographic Terrain Map:

A topographic terrain map is a map that shows the elevation of the land surface. It does this by using contour lines, which are lines that connect points of equal elevation. The closer the contour lines are together, the steeper the slope. The colors on the map also represent elevation, with brown being the lowest elevation and green being the highest elevation. Topographic terrain maps are used for a variety of purposes, such as hiking, camping, and military planning. They can also be used to study the Earth's surface and to understand how it has been shaped by geological processes.

Creating a topographic terrain map involves visualizing the elevation or height of the land surface in a geographical area. These maps are used in various applications, including geography, geology, environmental science, and outdoor activities. Some of the feature that are found on topographic terrain map are

- Contour lines: These lines connect points of equal elevation.
- Elevation: The height of the land surface above sea level.

- Slope: The steepness of the land surface.
- Landforms: Natural features on the Earth's surface, such as mountains, valleys, and rivers.
- Roads, trails, and other man-made features.

To create a topographic terrain map in Python, you can use libraries like Matplotlib and Cartopy for plotting and accessing map data.

Step 1: install cartopy by using:

pip install matplotlib numpy cartopy

Step 2: import required libraries

import matplotlib.pyplot as plt

import numpy as np

import cartopy.crs as ccrs

import cartopy.feature as cfeature

Step 3: create elevation data

extent = [-120, -70, 25, 50]

here first value in minimum longitude, second is maximum longitude, third is manimum latitude and last one is maximum latitude

lon = np.linspace(extent[0], extent[1], 100) #array for longitude

lat = np.linspace(extent[2], extent[3], 100) # array for latitude

lon, lat = np.meshgrid(lon, lat) # mesh grid for longitude and latitude

elevation = 1000 * np.sin(np.deg2rad(lat))

Step4: create contour plot

Create a figure and axis

fig, ax = plt.subplots(subplot_kw={'projection': ccrs.PlateCarree()}, figsize=(10, 8))

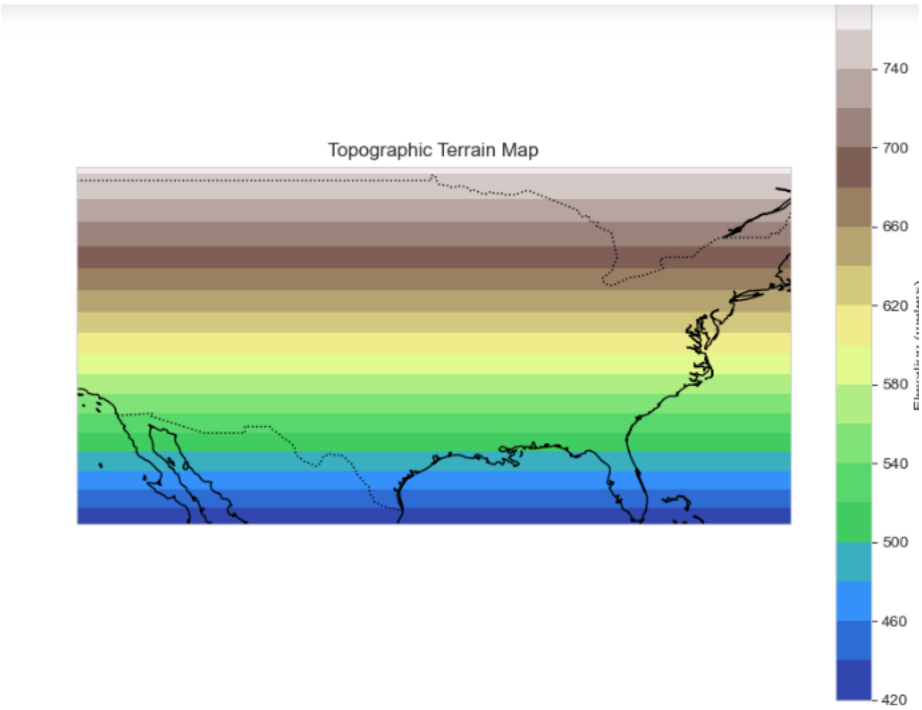
Add terrain data as a contour plot

contour = ax.contourf(lon, lat, elevation, levels=20, cmap='terrain', transform=ccrs.PlateCarree())

Add coastlines and other map features

ax.add_feature(cfeature.COASTLINE)

```
ax.add_feature(cfeature.BORDERS, linestyle=':')

ax.add_feature(cfeature.LAND, edgecolor='k')

ax.add_feature(cfeature.OCEAN, facecolor='lightblue')

# Add a colorbar

cbar = plt.colorbar(contour, ax=ax, label='Elevation (meters)')

# Set title and labels

plt.title('Topographic Terrain Map')

ax.set_xlabel('Longitude')

ax.set_ylabel('Latitude')

# Show the map

plt.show()
```



Map:

Maps play a crucial role in data visualization, especially when dealing with geographic or spatial data. They allow you to present information in a spatial context, making it easier to understand patterns, relationships, and trends. They can be used to show the distribution of data over a geographic area, to identify patterns and trends, and to compare different data sets. Map can be used to:

- To show the distribution of data: Maps can be used to show the distribution of data over a geographic area. For example, a map could be used to show the distribution of population density, crime rates, or income levels.
- To identify patterns and trends: Maps can be used to identify patterns and trends in data. For example, a map could be used to identify areas with high crime rates, or to track the spread of a disease.
- To compare different data sets: Maps can be used to compare different data sets. For example, a map could be used to compare the distribution of population density in different countries, or to track the changes in crime rates over time.
- Geospatial Data Representation: Maps are used to visualize geospatial data, which includes information tied to specific geographic locations. This can include data like population density, weather patterns, land use, and more.

Some common types of maps are:

- Choropleth Maps: These maps use color-coding to represent data values for regions or areas, such as countries, states, or counties. They are commonly used to visualize demographic data, election results, or any data that varies by location.
- Heatmaps: Heatmaps represent data intensity using color gradients. They're used to visualize data concentration or density, such as crime hotspots or population distribution.
- Topographic Maps: These maps depict elevation and terrain features, which are essential for navigation, geology, and environmental studies.
- Flow Maps: Flow maps show the movement of objects, people, or information between locations. They're used in transportation planning, migration studies, and network analysis.
- Custom Overlay Maps: You can create custom maps with overlaid data points, lines, or polygons to visualize specific information in a geographic context.
- Thematic maps: Thematic maps are used to show the distribution of a particular variable, such as population density, crime rates, or income levels.

Following factor should be considered by visualizing map

- The purpose of the map: What information do you want to convey with the map?
- The audience for the map: Who will be viewing the map?
- The data: What data do you have available?
- The map projection: The map projection is the way that the Earth's surface is represented on the map. Different map projections can distort the Earth's surface in different ways, so it is important to choose a projection that is appropriate for the purpose of the map.
- The visual elements: The visual elements of the map, such as the colors, shapes, and symbols, should be used to communicate the data effectively.

Dot and pixel:

In visualization, a dot and a pixel are both used to represent a single piece of data. However, there is a subtle difference between the two terms.

A dot is a mathematical concept that refers to a point in space. It has no size or dimension. A pixel is a physical entity that refers to a single point on a display screen. It has a specific size and dimension.

In visualization, dots are often used to represent data that is continuous, such as the temperature at a particular point in space. Pixels are often used to represent data that is discrete, such as the color of a pixel on a computer screen.

For example, a map of the world could be represented using dots to show the location of cities. The size of the dots could be used to represent the population of the cities. A heat map could be represented using pixels to show the temperature at different points on a map. The color of the pixels could be used to represent the temperature.

The choice of whether to use dots or pixels in visualization depends on the type of data being represented and the desired effect. Some example on dot and pixel used in visualization are:

Dot plot: A dot plot is a simple chart that uses dots to represent the values of a variable. The dots are typically arranged in a vertical or horizontal line, and the size of the dots can be used to represent the magnitude of the values. For example scatter plot

Line chart: A line chart is a chart that uses lines to connect dots that represent the values of a variable at different points in time. Line charts are often used to show trends or changes over time.

Heat map: A heat map is a chart that uses colors to represent the values of a variable. The colors are typically graduated, so that darker colors represent higher values and lighter colors represent lower values. Heat maps are often used to show the distribution of data over a geographic area.

When to use dot map and pixel map:

- Use dot maps when you need to emphasize individual data points, their exact locations, or clustering patterns. This approach is useful for discrete data with specific geographic coordinates.
- Use pixel maps when you have continuous or gridded data that you want to represent with smooth variations in color or intensity. Pixel maps work well for visualizing geographic phenomena that vary continuously across a region, such as elevation, temperature, or population density.

<u>Displaying a map using python:</u>

To show a map, Folium library is used. For creating a map first we have to define longitude and latitude of a place. Following program shows the example on creating map

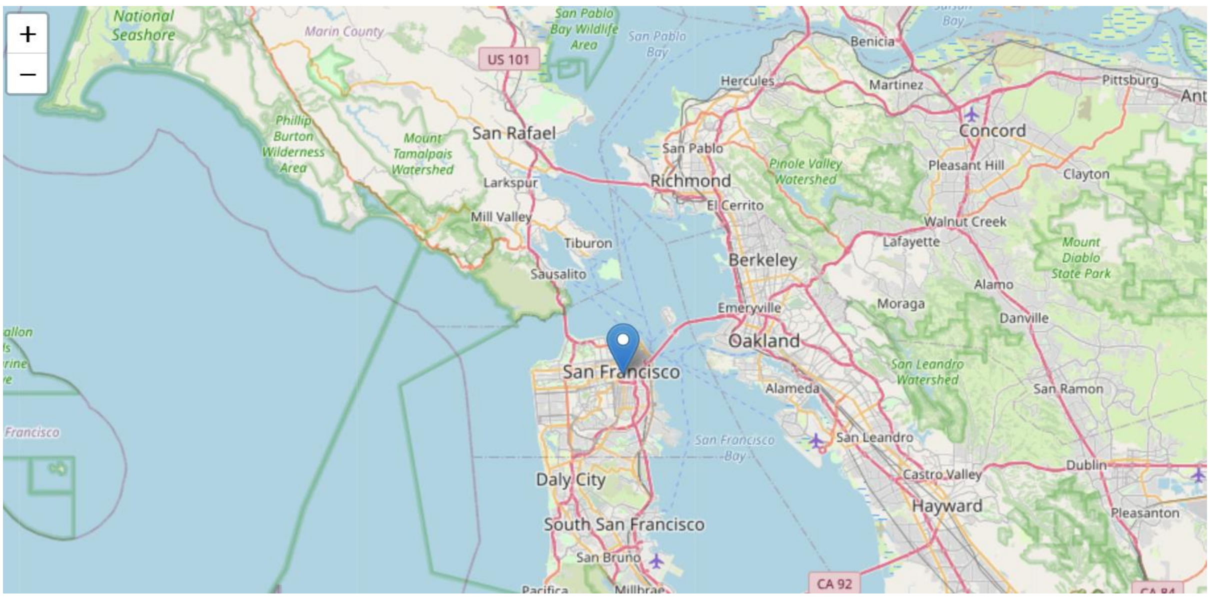
```
import folium

# Create a map centered on a specific location

m = folium.Map(location=[37.7749, -122.4194], zoom_start=10)

# Add a marker with a pop-up label

folium.Marker([37.7749, -122.4194], tooltip='San Francisco').add_to(m)

# Display the map in a Jupyter Notebook

M
```

In above code:

folium.Map(location=[37.7749, -122.4194], zoom_start=10)

=[37.7749, -122.4194] -> refers to longitude and latitude of san Francisco america

Output



Displaying Kathmandu of Nepal using folium

import folium

Create a map centered on a specific location

m = folium.Map(location=[27.7172,85.3240], zoom_start=10)

Add a marker with a pop-up label

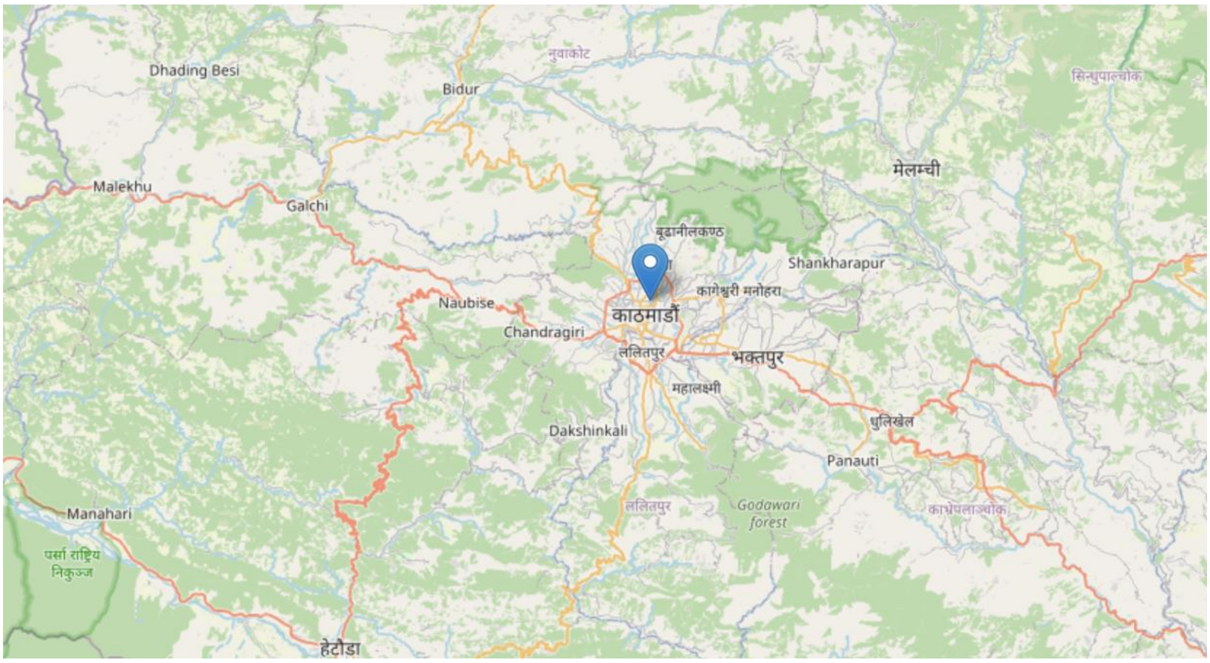
folium.Marker([27.7172,85.3240], tooltip='Nepal').add_to(m)

Display the map in a Jupyter Notebook

M

In code folium.Map(location=[27.7172,85.3240], zoom_start=10), location refers to longitude and latitude of Kathmandu.

Output:



If Pokhara needs to be displayed then use the coordinate of Pokhara. For example

import folium

Create a map centered on a specific location

m = folium.Map(location=[28.2096,83.9856], zoom_start=10)

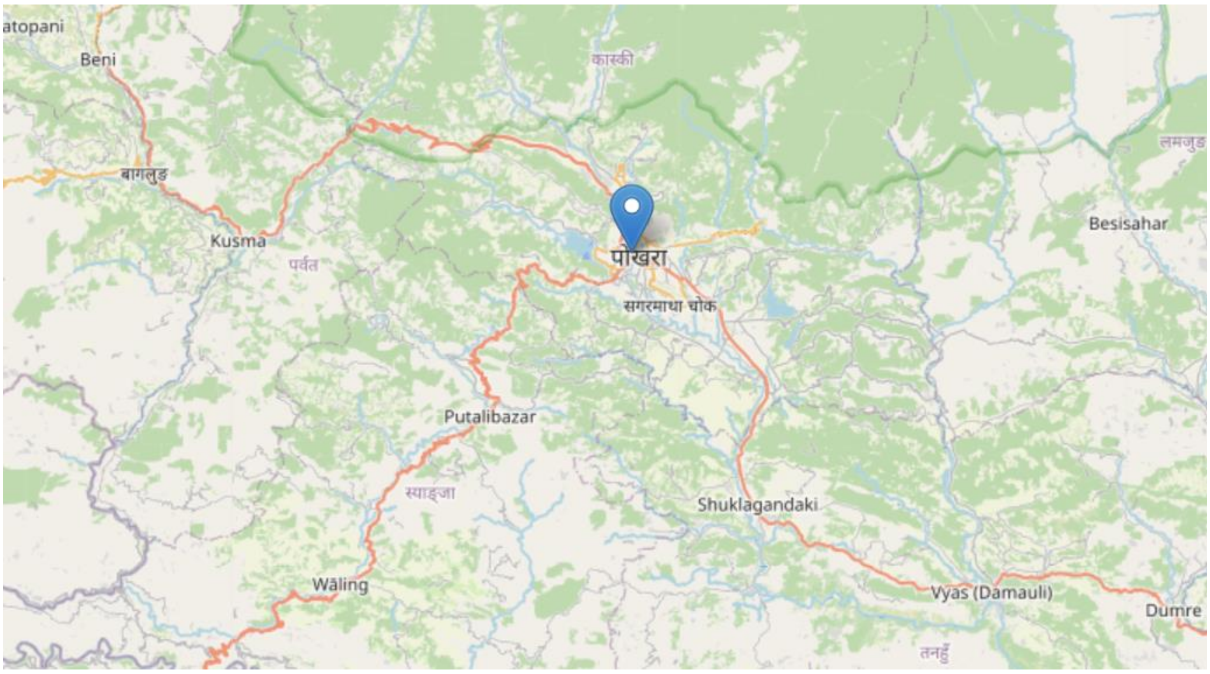
Add a marker with a pop-up label

folium.Marker([28.2096,83.9856], tooltip='Nepal').add_to(m)

Display the map in a Jupyter Notebook

M

Output:



Vector field:

A vector field in data visualization is a mathematical concept used to represent the distribution of vector quantities across space. It involves assigning a vector to each point or location within a specific region or domain. These vectors can represent various physical or abstract quantities, such as velocity, force, displacement, or any other vector quantity of interest. Vector fields are widely used in various fields, including physics, engineering, fluid dynamics, and data science, to visualize and analyze data patterns and spatial relationships.

Applications:

- Fluid Dynamics: Vector fields are commonly used to represent fluid velocity and flow patterns in fluid dynamics simulations.
- Electromagnetism: In physics, vector fields can represent electric and magnetic fields, helping visualize their distribution and behavior.
- Weather and Climate Science: Meteorologists use vector fields to visualize wind patterns and atmospheric circulation.
- Data Analysis: Vector fields can be used in data science to analyze data with spatial dependencies, such as movement patterns, transportation networks, and sensor data.

Vector fields are typically visualized using arrows or line segments, where the direction and length of each arrow represent the magnitude and direction of the vector at that location. Color and opacity can also be used to convey additional information about the vectors. In data science and machine learning, vector fields can be used to analyze complex spatial data, such as network traffic, user movement patterns, or geographic data. Therefore, a vector field in data visualization is a representation of vector quantities across space, often visualized using arrows or line segments. It is a powerful tool for understanding the distribution, direction, and patterns of vector quantities in various fields, providing valuable insights into complex spatial relationships.

Visualizing vector field:

2D vector field can be visualize using matplotlib library with quiver plot.

Step 1: import libraries

```
import matplotlib.pyplot as plt

import numpy as np
```

Step 2: Create array and convert into grid

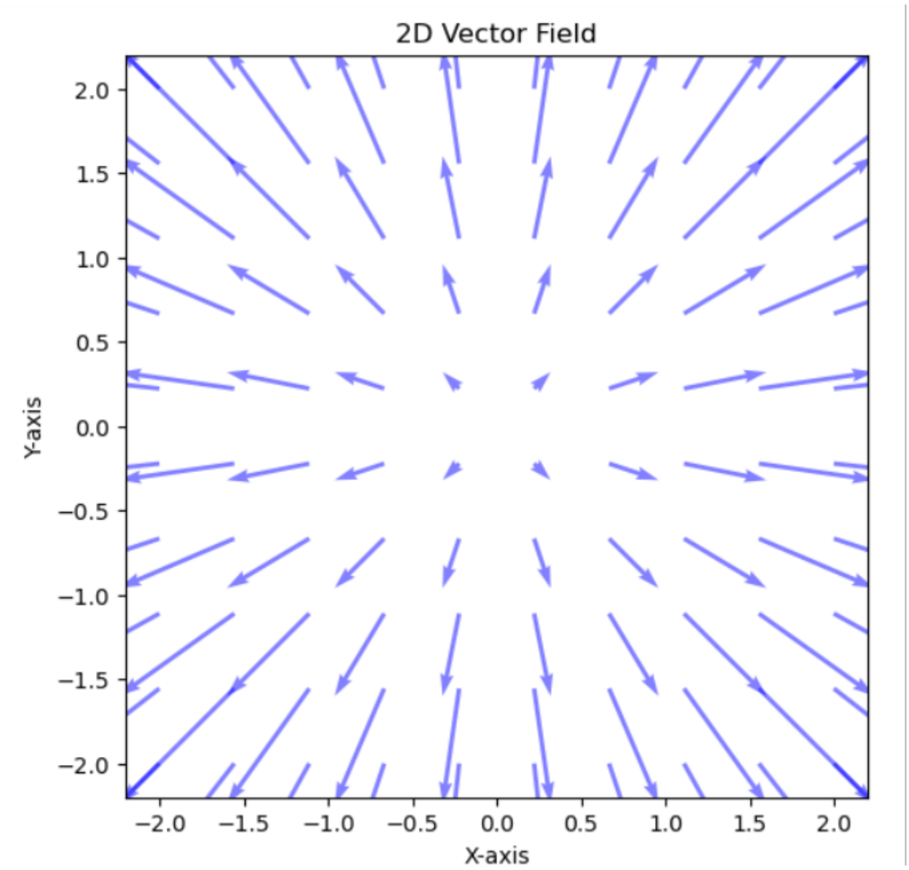
```
# Create a grid of points

x = np.linspace(-2, 2, 10)

y = np.linspace(-2, 2, 10)

X, Y = np.meshgrid(x, y)

# Define vector components

U = X

V = Y
```

Step 3: create a quiver plot

```
# Create a figure and axis

fig, ax = plt.subplots(figsize=(6, 6))

# Create the vector field plot

ax.quiver(X, Y, U, V, scale=10, color='blue', alpha=0.5)

# Customize plot settings (optional)

ax.set_aspect('equal')  # Equal aspect ratio

ax.set_xlabel('X-axis')

ax.set_ylabel('Y-axis')

ax.set_title('2D Vector Field')
```

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

Output



Marks and Channel:

Marks and channel shows how visual elements are used to represent convey information that are closely related to graphics. Marks, also referred to as "visual marks" or "graphical marks," are the basic geometric shapes or symbols used to represent data points or observations in a visualization. Each mark represents a single data point or a group of data points. Common types of marks include points (dots or circles), bars (rectangles or columns), lines, areas (shaded regions), and text labels. Channels, also known as "visual channels" or "encoding channels," are the visual properties or attributes of marks that can be manipulated to convey information about the data. Different channels encode different types of data or characteristics. Channels includes:

- Position: The location of a mark on the x and y axes (e.g., scatter plot).
- Color: The color of a mark, which can represent categorical or continuous data (e.g., heatmaps or color-coded categories).
- Size: The size of a mark, often used to represent quantitative data (e.g., bubble charts).
- Shape: The shape of a mark, often used to differentiate categories (e.g., different shapes for different data groups).

- Opacity: The transparency or opacity of a mark, which can be used to show overlapping data points (e.g., alpha blending in scatter plots).
- Texture: The texture or pattern of a mark's fill, which can be used to encode categorical information.
- Orientation: The rotation angle of a mark, used to represent directional or angular data.
- Connection: The use of lines or arrows to connect marks, indicating relationships or flow.

Example:

In a scatter plot (commonly used for visualizing two variables), the marks are typically points, and the channels used include position (x and y coordinates) to represent data values, and optionally color and size to encode additional information.

In a bar chart, the marks are bars or columns, and the channels used include position (x and y coordinates), color (for categorical distinctions), and height (to represent values).