CHAPTER 4

SPATIAL DATA VISUALIZATION

Remaining of chapter 3

Time Series Data, Characteristics of Time Data, Visualization time series data, Mapping of time

Time Series Data

A time series is a chronological sequence of observations on a particular variable. Usually the observations are taken at regular intervals (days, months, years), but the sampling could be irregular. A time series analysis consists of steps:

(1) building a model that represents a time series

(2) validating the model proposed

(3) using the model to predict (forecast) future values and/or impute missing values.

If a time series has a regular pattern, then a value of the series should be a function of previous values. The goal of building a time series model is the same as the goal for other types of predictive models which is to create a model such that the error between the predicted value of the target variable and the actual value is as small as possible.

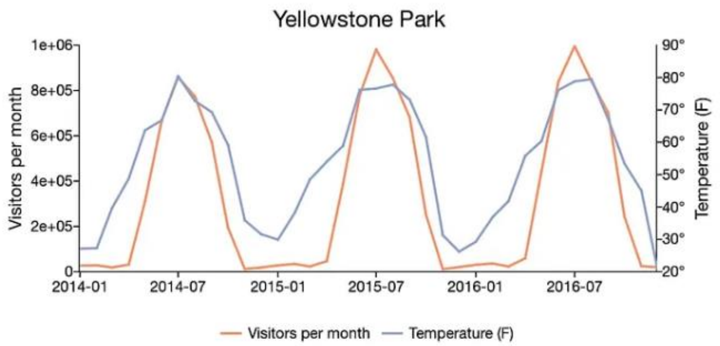
If you wish to take the same churn rate example and want to portray how it has been changing over the years, it would help to **use a line chart, bar chart, scatterplot, sparklines, etc.** that can use the timeline in the X-axis and show how things have been moving.

For example, stocks in stock market.

Time series data is a collection of **observations** (behavior) for a **single subject** (entity) at **different time intervals** (generally equally spaced as in the case of metrics, or unequally spaced as in the case of events).

For example: Max Temperature, Humidity and Wind (all three behaviors) in New York City (single entity) collected on First day of every year (multiple intervals of time)

The relevance of time as an axis makes time series data distinct from other types of data.



Time series can be classified into two different types: stock and flow.

A stock series is a measure of certain attributes at a point in time and can be thought of as "stocktakes". For example, the Monthly Labour Force Survey is a stock measure because it takes stock of whether a person was employed in the reference week.

Flow series are series which are a measure of activity over a given period. For example, surveys of Retail Trade activity. Manufacturing is also a flow measure because a certain amount is produced each day, and then these amounts are summed to give a total value for production for a given reporting period.

The main difference between a stock and a flow series is that flow series can contain effects related to the calendar (trading day effects). Both types of series can still be seasonally adjusted using the same seasonal adjustment process.

Data variations

In time series data, variations can occur sporadically throughout the data:

- **Functional analysis** can pick out the patterns and relationships within the data to identify notable events.
- **Trend analysis** means determining consistent movement in a certain direction. There are two types of trends: deterministic, where we can find the underlying cause, and stochastic, which is random and unexplainable.
- **Seasonal variation** describes events that occur at specific and regular intervals during the course of a year. Serial dependence occurs when data points close together in time tend to be related.

Time series analysis and forecasting models must define the types of data relevant to answering the business question. Once analysts have chosen the relevant data they want to analyze, they choose what types of analysis and techniques are the best fit.

Characteristics of Time Series Data

Time data, also known as temporal data, possesses certain characteristics that distinguish it from other types of data. Understanding these characteristics is essential for effective analysis and visualization of time-based information. Here are the key characteristics of time data:

1. Temporal Order: Time data is inherently ordered chronologically. It represents a sequence of events or measurements recorded at successive time points. The temporal order is crucial as it allows for the analysis of patterns, trends, and changes over time.

2. Time Granularity: Time data can be represented at different levels of granularity, ranging from milliseconds to years, depending on the specific context. The choice of time granularity determines the level of detail and the insights that can be derived from the data.

3. Irreversibility: Time data follows a unidirectional flow and is generally irreversible. Once a specific time point or event has occurred, it cannot be undone or altered. This characteristic is important in understanding the historical nature of time data and its impact on forecasting and prediction.

4. Time Intervals: Time data is often measured and analyzed in specific time intervals, such as seconds, minutes, hours, days, etc. These intervals allow for the aggregation, summarization, and analysis of data over different periods, enabling the identification of patterns and trends at various scales.

5. Seasonality and Cyclicity: Time data often exhibits seasonality and cyclical patterns. Seasonality refers to regular and predictable variations that occur within a specific time period, such as daily, weekly, or yearly patterns. Cyclicity refers to longer-term repetitive patterns that occur over multiple periods, which may or may not follow a fixed duration.

6. Time Zones and Time-Related Adjustments: Time data is influenced by time zones and adjustments like Daylight Saving Time, which can introduce variations and complexities in data analysis and visualization. Taking time zones into account is crucial when dealing with data from different regions or when aligning events across different time zones.

7. Time Lags and Dependencies: Time data can exhibit lags and dependencies, where the value or behavior of a variable at a particular time point may be influenced by past or future events. Understanding and analyzing these dependencies is important for causal analysis, forecasting, and decision-making.

8. Time Series Analysis: Time data often lends itself to time series analysis techniques, which involve studying and modeling the statistical properties, trends, and patterns within the data. Time series analysis methods include trend analysis, seasonality decomposition, autocorrelation analysis, and forecasting models.

Recognizing these characteristics of time data helps in selecting appropriate analysis techniques, designing effective visualizations, and extracting meaningful insights from time-based information.

Visualization Time Series Data

Ways for Time Series Data Visualization

To effectively visualize time series data, various visualization techniques can be employed. Let's explore some popular visualization methods:

1. **Tabular Visualization:** Tabular visualization presents time series data in a structured table format, with each row representing a specific time period and columns representing different variables or measurements. It provides a concise overview of the data but may not capture trends or patterns as effectively as graphical visualizations.
2. **1D Plot of Measurement Times:** This type of visualization represents the measurement times along a one-dimensional axis, such as a timeline. It helps in understanding the temporal distribution of data points and identifying any temporal patterns.

3. **1D Plot of Measurement Values:** A 1D plot of measurement values display the variation in data values over time along a single axis. Line plots and step plots are commonly used techniques for visualizing continuous time series data, while bar charts or dot plots can be used for discrete data.
4. **1D Color Plot of Measurement Values:** In this visualization technique, the variation in measurement values is represented using colors on a one-dimensional axis. It enables the quick identification of high or low values and provides an intuitive overview of the data.
5. **Bubble Plot:** Bubble plots represent time series data using bubbles, where each bubble represents a data point with its size or color encoding a specific measurement value. This visualization method allows the simultaneous representation of multiple variables and their evolution over time.
6. **Scatter Plot:** Scatter plots display the relationship between two variables by plotting data points as individual dots on a Cartesian plane. Time series data can be visualized by representing one variable on the x-axis and another on the y-axis.
7. **Linear Line Plot:** Linear line plots connect consecutive data points with straight lines, emphasizing the trend and continuity of the data over time.
8. **Linear Step Plot:** Linear step plots also connect consecutive data points, but with vertical and horizontal lines, resulting in a stepped appearance. This visualization is useful when tracking changes that occur instantaneously at specific time points.
9. **Linear Smooth Plot:** Linear smooth plots apply a smoothing algorithm to the data, resulting in a continuous curve that captures the overall trend while reducing noise or fluctuations. It helps in visualizing long-term patterns more clearly.
10. **Area Chart:** Area charts fill the area between the line representing the data and the x-axis, emphasizing the cumulative value or distribution over time. They are commonly used to visualize stacked time series data or to show the composition of a variable over time.
11. **Horizon Chart:** Horizon charts condense time series data into a compact, horizontally layered representation. They are particularly useful when comparing multiple time series data on a single chart, optimizing screen space usage.
12. **Bar Chart:** Bar charts represent discrete time series data using rectangular bars, with the height of each bar indicating the value of a specific measurement. They are effective in comparing values between different time periods or categories.
13. **Histogram:** Histograms display the distribution of continuous or discrete time series data by dividing the range of values into equal intervals (bins) and representing the frequency

or count of data points falling within each bin. Here, **Business Intelligence and Visualization training** will help you get mentored by international Tableau, BI, TIBCO, and Data Visualization experts and represent data through insightful visuals.

Time Series Data Visualization Examples

Let us explore some examples of time series data visualizations:

- **Gantt Charts:** Gantt charts are widely used to visualize project schedules or timelines. They display tasks or events along a horizontal timeline, with bars representing the start and end dates of each task. Gantt charts provide a clear overview of project progress, dependencies, and resource allocation over time.
- **Line Graphs:** Line graphs are effective for visualizing continuous time series data. They connect data points with straight lines, allowing us to observe trends, seasonality, or irregularities over time.
- **Heatmap:** Heatmaps represent time series data using color intensity in a grid format. They are useful for visualizing patterns, correlations, or anomalies in multi-dimensional time series data.
- **Map:** Maps can be employed to visualize time series data geographically. By plotting data points on a map, we can observe spatial patterns or changes in variables over time.
- **Stacked Area Charts:** Stacked area charts display the cumulative value or proportion of different variables over time. They are useful for visualizing the composition or contribution of each variable to the total.

Advantages of Visualization of Time Series Data:

1. Pattern Recognition: Visualization allows for the effective identification of patterns, trends, and anomalies in time series data. Visual representations, such as line charts or heatmaps, provide a clear view of how values change over time, making it easier to identify recurring patterns or sudden deviations.

2. Insights into Data Patterns: Time series visualization helps uncover insights into the underlying data patterns. By visualizing data, analysts can observe long-term trends, seasonal variations, cyclic behavior, or irregularities that may not be apparent in raw numerical data alone.

3. Communication and Interpretation: Visualizations enable effective communication and interpretation of time series data to various stakeholders. Visual representations make it easier to convey complex patterns or trends to non-technical audiences, facilitating decision-making and data-driven discussions.

4. Data Exploration and Interactivity: Visualization tools often offer interactive features, allowing

users to explore time series data by zooming in on specific time periods, adjusting time scales, or applying filters. This interactivity enhances data exploration, enabling users to investigate specific segments, outliers, or anomalies within the data.

5. Forecasting and Predictive Analysis: Visualizing time series data helps in forecasting and predictive analysis. By visually examining historical patterns and trends, analysts can make informed predictions about future behavior, aiding in business planning, resource allocation, and decision-making.

6. Temporal Context and Comparison: Visualization provides a temporal context for data analysis. It enables the comparison of multiple time series datasets, highlighting differences, correlations, or synchronized behavior between variables over time. This comparison helps in understanding relationships and making data-driven decisions.

Overall, visualization of time series data offers a powerful means of exploring, interpreting, and communicating temporal patterns and trends. It enhances the understanding of data dynamics, aids in prediction and decision-making, and supports effective communication of insights to diverse audiences.

Chapter 4 started

Spatial data is any type of data that directly or indirectly references a specific geographical area or location. Sometimes called geospatial data or geographic information, spatial data can also numerically represent a physical object in a geographic coordinate system. However, spatial data is much more than a spatial component of a map.

Users can save spatial data in a variety of different formats, as it can also contain more than location-specific data. Analyzing this data provides a better understanding of how each variable impacts individuals, communities, populations, etc.

There are several spatial data types, but the two primary kinds of spatial data are geometric data and geographic data.

Geometric data is a spatial data type that is mapped on a two-dimensional flat surface. An example is the geometric data in floor plans. Google Maps is an application that uses geometric data to provide accurate direction. In fact, it is one of the simplest examples of spatial data in action.

Geographic data is information mapped around a sphere. Most often, the sphere is planet earth. Geographic data highlights the latitude and longitude relationships to a specific object or location. A familiar example of geographic data is a global positioning system.

Georeferencing and geocoding

Similar processes, georeferencing and geocoding, are important aspects of geospatial analysis. Both geocoding and georeferencing involve fitting data into the real world by using appropriate coordinates, but that is where the similarity ends.

Georeferencing concentrates on assigning data coordinates to vectors or rasters. This approach helps accurately model the planet's surface.

Geocoding, on the other hand, provides address and location descriptors. These can include information about cities, states, countries, and so on. Each exact coordinate references a specific location on the earth's surface.

Geospatial mapping can be seen as a process with 4 stages:

1. Data acquisition

The first stage requires the collection of primary and secondary data. The sources and type of data vary and depend on the nature of the phenomenon we work with. They can be acquired, as one of many examples, through the collection of geodetic measurements or through aerial, satellite or drone imagery.

Aerial, satellite and drone imageries are taken with the use of sensors (active or passive) to collect various types of data. Remote sensing can provide us with many kinds of data, such as optical, LIDAR or SAR data. They can be captured, for example, in natural colours or infrared.

2. Data processing and input to databases

The collected data is processed to feed the relevant databases that store information about the raw geometry of objects without any graphic symbols. Data contained in such databases are then passed on to software programmes responsible for analytical functions. Based on them, we can further process these data and carry out spatial analyses.

3. Creating a model and design

The next stage involves creating designs and graphics for the map. In this process, symbols and colours are assigned according to specific cartographic rules, so that objects or places are as recognisable as possible, and the overall perception is understandable to the average user. Also, at this stage one of the most important activities in the map-making process takes place, that is, cartographic generalisation.

4. Publication

The last stage is the publication of the prepared product or a model as an exported map or as a map service that can be embedded in a website.

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

- Scalar fields are mathematical constructs that assign a scalar value to each point in a given space or domain.
- Scalar values represent single numerical quantities like temperature, pressure, or density without a directional component.
- Scalar fields are used to represent continuous data that varies smoothly over space or time.
- They enable the visualization and analysis of complex data patterns and relationships.
- Scalar fields find applications in scientific and engineering disciplines.
- Visualization techniques like contour plots, heatmaps, and color maps are used to depict scalar fields.
- Scalar field visualizations help researchers gain insights into data distribution, trends, and anomalies.
- Scalar fields are used in various fields such as meteorology, geophysics, medical imaging, and computer graphics.
- 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 are lines that connect points of equal value within a scalar field.
- They depict the variation and structure of a scalar quantity over a 2D or 3D space.
- Isocontours are commonly used in topographic terrain maps to represent elevation levels.
- They provide valuable information about the shape, patterns, and relationships within the scalar field.

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()

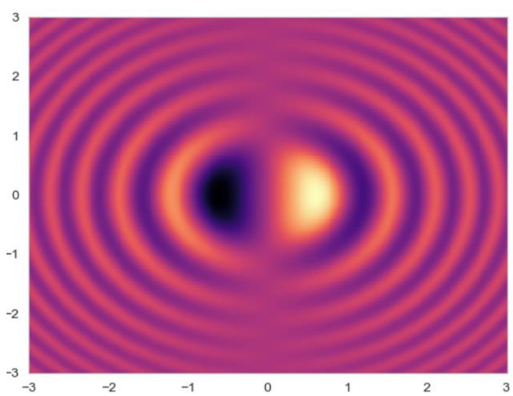
Full Program:

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')

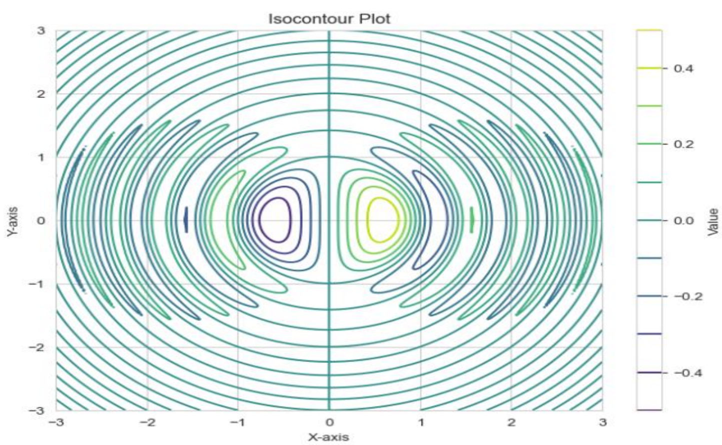
plt.ylabel('Y-axis')

plt.title('Isocontour Plot')

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

Show the plot

plt.show()



<u>Visualizing scalar field using quiver plot</u>

Import necessary libraries

import numpy as np

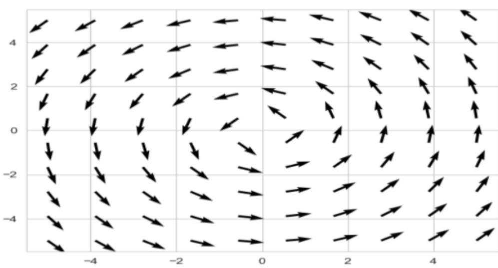
import matplotlib.pyplot as plt

%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 Volumes

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.

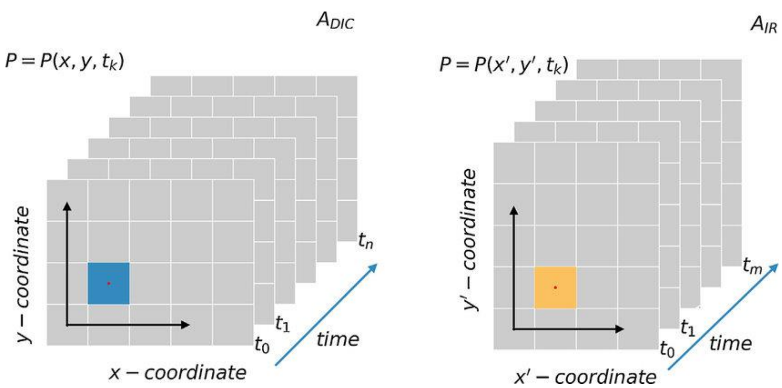
Visualization technique for Scalar Volume

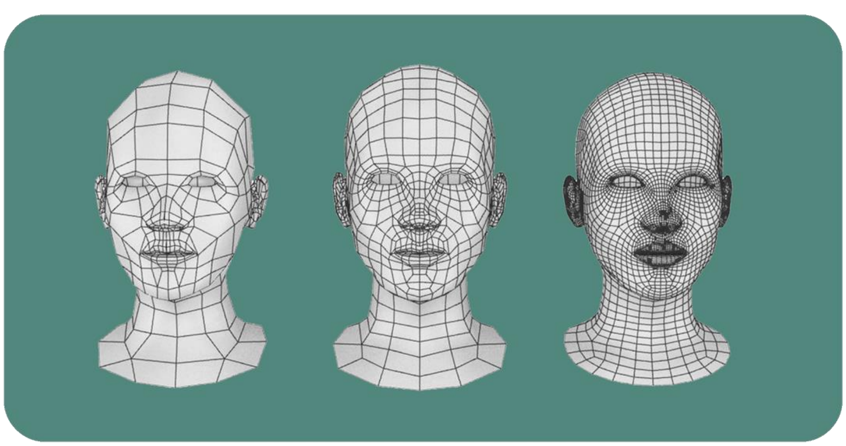
- Isosurface Rendering: Isosurface rendering is a common technique for visualizing scalar volumes. It involves creating surfaces within the volume that enclose regions where the scalar value equals a specified threshold (isosurface). These surfaces are often displayed with shading to highlight shape 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.
- Scalar volumes represent data that is assigned to each point in a three-dimensional space.
- In a scalar volume, each point in space has an associated scalar value, which could represent a physical quantity, such as temperature, pressure, concentration, or any other scalar property.
- These scalar values are usually defined on a regular grid, forming a volumetric dataset.

Data Representation

- Scalar volumes can be represented in various ways, depending on the specific requirements and characteristics of the dataset.
- The most common representation is a regular grid structure, such as a three-dimensional array or a 3D mesh.
- Each element in the grid corresponds to a point in space and stores the scalar value.





Interpolation:

- Interpolation is a mathematical technique to estimate the values of unknown data points that fall in between existing, known data points. This process helps fill in the blanks
- Interpolation means determining a value from the existing values in a given data set. Another way of describing it is the act of inserting or interjecting an intermediate value between two other values.
- Scalar volumes often require interpolation techniques to estimate the scalar values at arbitrary points in space.
- Interpolation allows the generation of a continuous representation of the scalar field, enabling smoother visualization and analysis.

Visualization

- Scalar volumes are visualized using techniques such as volume rendering, contouring, and isosurface extraction.
- Volume rendering creates a 2D image by simulating the passage of light through the volume, producing visual representations that convey information about the scalar values and their distribution.
- Contouring generates 2D curves or surfaces that represent specific scalar values.
- Iso-surface extraction focuses on extracting surfaces that represent specific scalar values, allowing the visualization of regions of interest within the volume.

Analysis:

- Scalar volumes offer a rich set of analysis techniques to extract meaningful insights.
- This includes techniques like feature extraction, segmentation, and classification.
- Feature extraction aims to identify and extract significant structures or patterns within the scalar volume, helping to highlight regions of interest.

Example

- In medical imaging, scalar volumes are used to represent anatomical structures, enabling the visualization and analysis of tissues and organs.
- In fluid dynamics, scalar volumes represent properties such as velocity, pressure, or temperature, aiding in the understanding of fluid flow behavior.

Scalar volumes play a crucial role in data analysis and visualization, allowing researchers and practitioners to study and interpret scalar data in a three-dimensional space. These volumes enable the exploration of complex phenomena and facilitate the extraction of valuable insights from the data.

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()

<u>Topographic Terrain Map:</u>

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.

Topographic terrain Maps(Isocontours)

- By analyzing the spacing, curvature, and arrangement of contour lines, insights can be gained about steepness, slopes, and features of the underlying data.
- Visualization techniques like color-coded contour maps or 3D surface plots are used to represent isocontours.
- Isocontours have applications in geology, hydrology, environmental science, and GIS.

Example

- Suppose we have a dataset representing elevation levels of a mountainous region.
- Isocontours can be plotted to create a topographic map showing the different elevation levels.
- Contour lines connect points of equal elevation, representing values like 1000 meters or 1500 meters.
- The spacing between contour lines indicates steepness, with closely spaced lines indicating a steep slope.
- The curvature and arrangement of contour lines help identify valleys, ridges, and other features.

- Color-coding the contour lines based on elevation values enhances visualization.
- Isocontours aid in understanding the topography, locating high altitude areas, and planning routes.

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

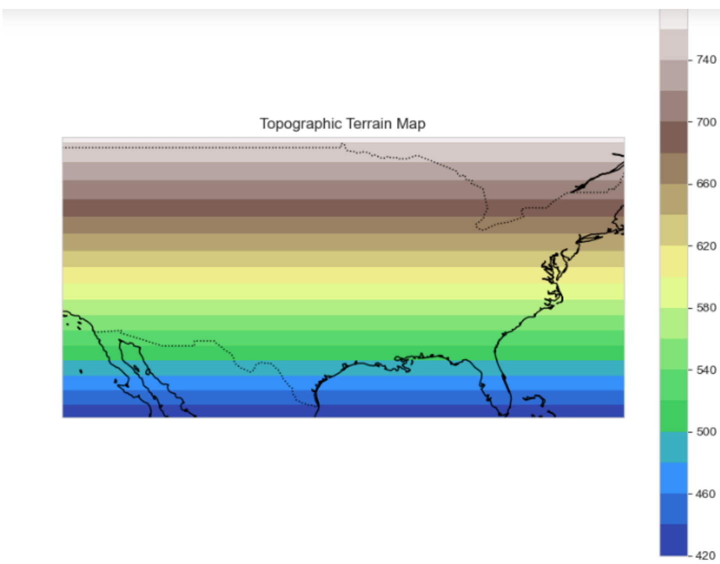
contour = ax.contourf(lon, lat, elevation, levels=20, cmap='terrain', transform=ccrs.PlateCarree())

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()
```



Direct Volume Rendering

- Direct Volume Rendering (DVR) is a powerful technique used in data analysis and visualization
    - to create visual representations of scalar volumes.
    - It allows for the direct exploration and visualization of the volumetric data without the need for intermediate surface extraction or contouring.
- DVR operates by considering the entire volume as a continuous medium and simulating the interaction of light with the scalar values within the volume.

- This technique is particularly useful when dealing with complex and intricate internal structures or when the specific surfaces of interest are not well defined.

Steps:

Ray Casting

- The DVR process starts by casting rays through the volume.
- These rays originate from the viewpoint or camera position and pass through each pixel in the image plane.
- Each ray represents a virtual line of sight through the volume.

Sampling

- As each ray traverses the volume, it samples the scalar values at regular intervals.
- The sampling frequency can vary based on the desired level of detail and computational resources.
- The sampled scalar values are used to determine the appearance and properties of the volume.

Transfer Function

- The transfer function plays a crucial role in DVR.
- It maps the scalar values to optical properties such as color and opacity.
- It defines how the scalar values should be mapped to visual attributes, allowing the user to control the appearance and highlight specific features of interest.
- For example, high scalar values could be associated with brighter colors or higher opacity, while low scalar values might be associated with darker colors or lower opacity.

Compositing

- Once the transfer function assigns optical properties to the sampled scalar values along the ray, the individual samples are blended together to produce the final pixel color.
- The compositing process determines how the optical properties are combined, considering factors such as transparency, occlusion, and lighting.
- Common compositing techniques include alpha blending, maximum intensity projection (MIP), and gradient-based methods.

Shading and Illumination:

- To enhance the perception of depth and shape, shading and illumination techniques can be applied.

- These techniques consider the direction of light sources and the surface normal derived from the gradient of the scalar field.
- Shading can help create a sense of depth and volume, while illumination can provide realistic highlights and shadows.

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.

Maps (Dots)

- Dot maps utilize individual dots to represent specific data points or events.
- Each dot corresponds to a particular location or unit on the map.
- The quantity or value associated with the data point is represented by the number of dots at that location.

Example

- Let's consider an example of a dot map representing store locations.
- Each dot on the map represents a store in a retail chain.
- The density of dots in a particular area indicates the concentration of stores, helping identify areas with high market coverage.

Example 2

- Another application of dot maps is visualizing crime incidents.
- Each dot represents a reported crime event, allowing us to identify crime hotspots and patterns.
- The density and distribution of dots can inform law enforcement strategies and resource allocation.

Maps (Pixels)

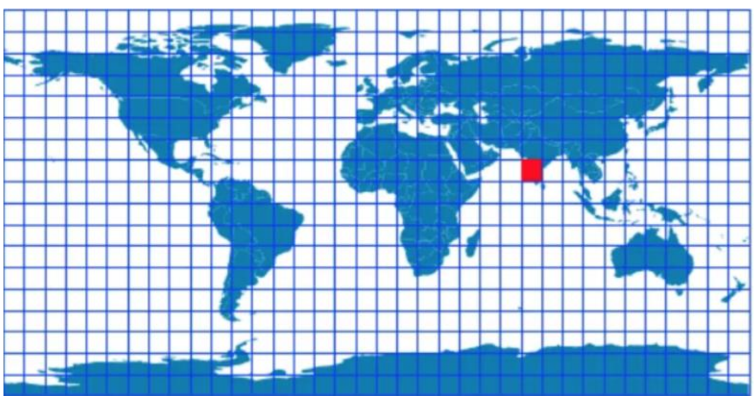
- Pixel maps, also known as raster maps, divide the map into a grid of pixels.
- Each pixel corresponds to a specific location, and its color or intensity represents the associated data value.

Examples – Temperature

- Let's explore a pixel map example using temperature data.
- Each pixel on the map is colored based on the average temperature of a specific area.
- By examining the color gradients, we can identify temperature variations across regions or even identify temperature anomalies.

Example: Population Density

- Another application of pixel maps is representing population density.
- Each pixel's intensity corresponds to the population density of a particular area.
- This allows us to visualize densely populated areas, identify urban centers, and analyze population distribution patterns.



In nutshell,

- Maps are valuable tools for data analysis and visualization, particularly for geographical information.
- Dot maps are suitable for discrete point-based data, such as store locations or crime incidents.
- Pixel maps are ideal for continuous or aggregated data, like temperature or population density.
- By leveraging maps, we can gain insights, identify patterns, and make informed decisions based on spatial data.

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.

Displaying a map using python:

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

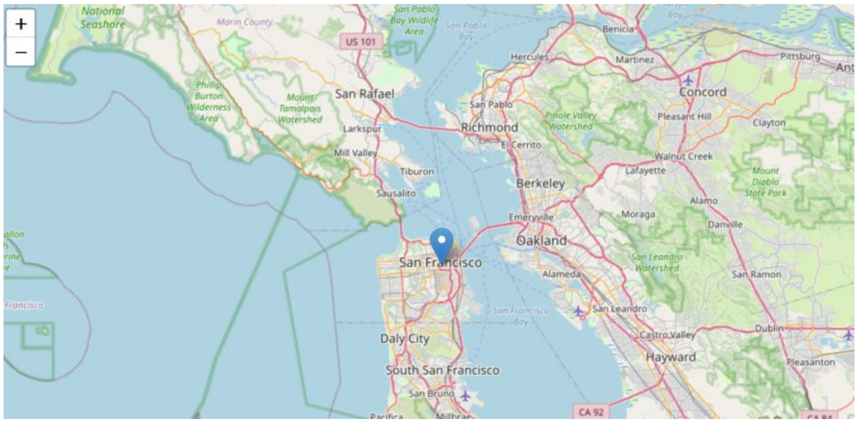
```
import folium

# 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

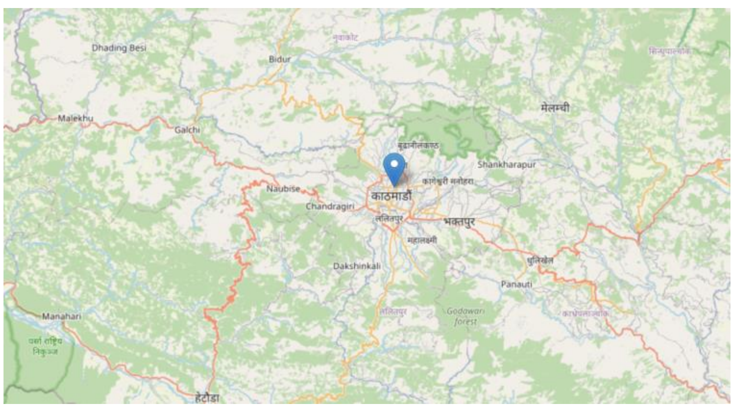


Displaying Kathmandu of Nepal using folium

```
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.

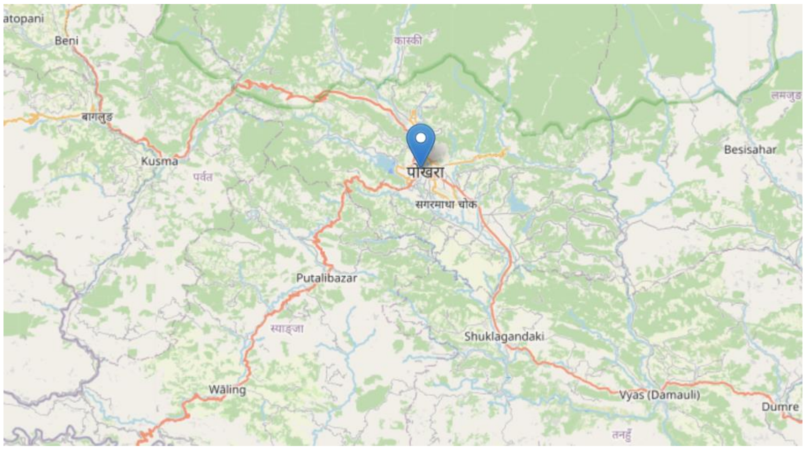
Output:



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 Fields

- vector fields are powerful tools for representing and analyzing spatial data.
- Vector fields represent spatial data using arrows or vectors.
- Each vector has a magnitude and direction, indicating the magnitude and direction of a specific attribute at a particular location.
- 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.

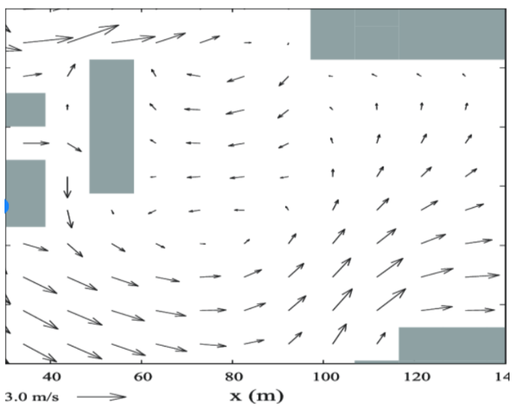
Applications:

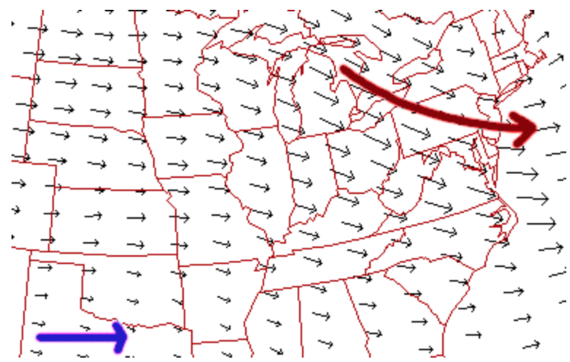
- Fluid Dynamics: Vector fields are commonly used to represent fluid velocity and flow patterns in fluid dynamics simulations.
- Electromagnetism: In physics, vector fields can represent electric and magnetic fields, helping visualize their distribution and behavior.
- Weather and Climate Science: Meteorologists 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.

Example: Wind Pattern

- Let's consider an example of a vector field representing wind patterns.
- Each arrow in the vector field represents the wind direction and magnitude at a specific location.
- By examining the arrows, we can identify prevailing wind patterns, wind speed, and areas of turbulence.





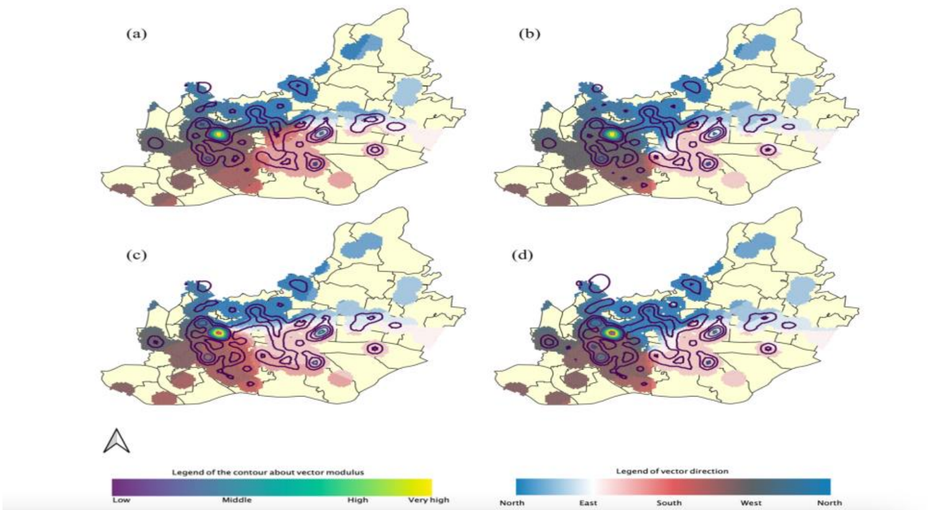
Example: Ocean Currents

- Another application of vector fields is visualizing ocean currents.
- Arrows in the vector field represent the speed and direction of ocean currents at different locations.
- This allows us to study ocean circulation patterns, identify currents' paths, and analyze their impact on marine ecosystems.

Vector Fields with map integration

- Vector fields can be combined with maps to enhance data analysis and visualization.
- By overlaying vector fields on maps, we can correlate spatial patterns and attribute distributions.
- This integration provides a comprehensive understanding of the relationships between geographical features and attribute variations.
- Consider the example of air pollution analysis.

- We can combine a map representing air quality index with a vector field showing wind patterns.
- This integration helps us identify pollutant sources, understand the dispersion of pollutants, and assess their impact on different areas.



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

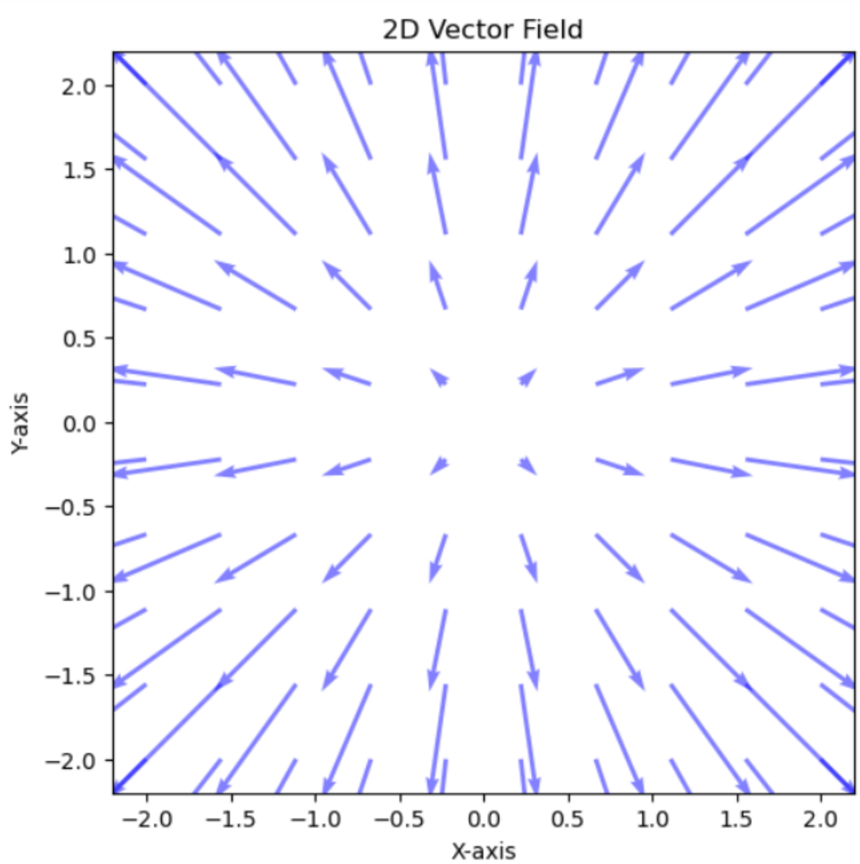
V = Y

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).

Marks

- Marks and channels are fundamental concepts in visual encoding, helping us represent data effectively.
- Marks refer to the visual elements used to represent individual data points or values.
- They can be various shapes, such as points, bars, lines, or areas, depending on the type of data being represented.

Examples

- Points: Representing individual data points on a scatter plot.
- Bars: Representing values of different categories on a bar chart.
- Lines: Representing trends or connections between data points on a line chart or network diagram.
- Areas: Representing regions or distributions on a heat map or choropleth map.

Channels

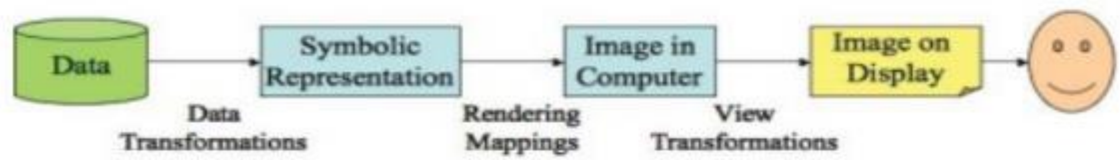
- Channels are the visual properties or attributes used to encode data within marks.
- They define how data values are represented visually, such as position, size, color, shape, and texture.

Examples

- Position: Encoding data values along spatial axes, such as x and y positions on a chart.
- Size: Using variations in size to represent data values, where larger marks indicate higher values.
- Color: Mapping data values to different colors or color gradients to convey information.
- Shape: Using different shapes to represent categories or data groups.
- Texture: Applying different patterns or textures to marks to represent different data values or categories.

The Visualization Process

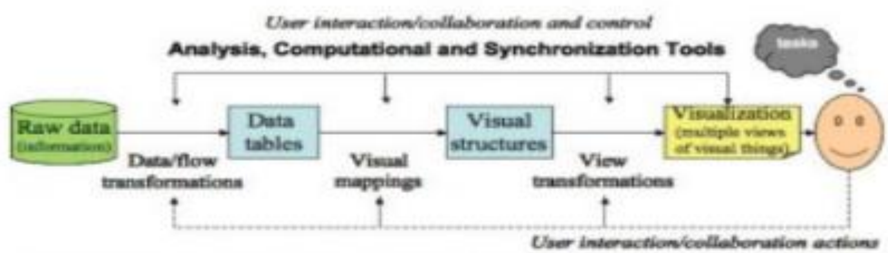
- The process of visualizing directs your subconscious to be aware of the end goal you have in mind.
- The designer of a new visualization most often begins with an analysis of the type of data available for display and of the type of information the viewer hopes to extract from or convey with the display
- The data can come from a wide variety of sources and may be simple or complex in structure



The visualization process at a very high or primitive level view.

The Visualization Pipeline

- The dataffinformation visualization pipeline has some similarities to the graphics pipeline, at least on an abstract level. The stages of this pipeline are as follows:
- Data modeling. The data to be visualized, whether from a file or a database, has to be structured to facilitate its visualization. The name, type, range, and semantics of each attribute or field of a data record must be available in a format that ensures rapid access and easy modification
- Data selection. Similar to clipping, data selection involves identifying the subset of the data that will be potentially visualized. This can occur totally under user control or via algorithmic methods, such as cycling through time slices or automatically detecting features of potential interest to the use



One example of the visualization pipeline. There are many variants, but all transform data into some internal representation within the computer and then use some visual paradigm to display the data on the screen.