Assignment Report

Data Visualization

Thejas V 20201ISE0054

Vikram 20201ISE0055

Shreyas S 20201ISE0085

VishalSingh Rajpurohit 20201ISE0075

Aiyaan Hasan 20201ISE0092

Avinash Shinde 20201ISE0081

Shivam Kumar 20201ISE0080

Berlin Gangamma 20201ISE0071

Grantha Kushalappa 20201ISE0048

Introduction

Data visualization is the graphical representation of information and data. By using visual elements such as charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data[1]. In the contemporary digital age, where vast amounts of data are generated every second, the ability to interpret and communicate this data effectively has become crucial [2]. Data visualization not only aids in making sense of complex data sets but also enhances the decision-making process by presenting data in a clear and actionable format.[3]

The roots of data visualization can be traced back to the 17th century, with early examples like John Snow's cholera map in 1854 and Florence Nightingale's coxcomb charts in the 1850s [4]. These historical examples demonstrate the long-standing importance of visualizing data to understand and solve problems. In the modern era, advancements in technology have transformed data visualization from static images into dynamic and interactive experiences. Modern tools and software, such as Tableau, Power BI, and D3.js, empower users to create sophisticated visualizations that can be easily shared and interpreted across various platforms.

Data visualization serves multiple purposes. It helps in simplifying complex data sets, making large amounts of information comprehensible at a glance. This simplification is crucial in various fields, from business analytics to scientific research, where understanding data quickly and accurately can lead to better insights and decisions. Furthermore, data visualization is essential in storytelling, allowing data to tell a story that highlights key messages and insights that might be overlooked in text-based data presentations.

One of the most compelling aspects of data visualization is its ability to reveal insights that are not immediately obvious. Patterns, correlations, and trends can emerge through visual representation, enabling analysts and decision-makers to identify opportunities and risks that might not be evident through raw data alone. Effective visualizations leverage principles of design and cognitive science to present data in ways that enhance comprehension, retention, and decision-making. They use color, size, shape, and motion to highlight relationships and trends, making complex information more accessible and understandable.

The role of data visualization has become even more critical in an era dominated by big data and analytics. Organizations across various sectors, including business, healthcare, science, and public policy, rely heavily on data-driven insights to inform their strategies and operations. In business, data visualization helps in identifying market trends, customer behaviors, and operational efficiencies. In healthcare, it aids in tracking disease outbreaks, patient outcomes, and treatment efficacy. In science, visualizations are used to explore research data, uncover new findings, and communicate results to the broader community. In public policy, they help in understanding and addressing societal issues, from economic trends to environmental impacts.

As the volume of data continues to grow exponentially, so does the need for effective data visualization techniques. The increasing complexity of data requires innovative approaches to visualization, combining the latest technological advancements with robust analytical methods. This report will explore the fundamental concepts of data visualization, examine the tools and techniques used by professionals, and highlight best practices for creating impactful visualizations. By understanding these key elements, we can better harness the power of data to drive innovation and inform decisions.

In summary, data visualization is more than just a tool for making data look attractive; it is a critical process for interpreting and communicating information in a way that is both insightful and actionable. As we delve deeper into the world of data visualization, we will uncover the principles that make visualizations effective, the technologies that enable their creation, and the practical applications that demonstrate their value across various domains.

Book Structure

1.Introduction

- Introduction
- Book Structure

2.From Graphics to Visualization

- A simple Example
- Graphics-Rendering Basics
- Rendering the height plot
- Texture mapping
- Transparency and Blending
- Conclusion

3.Data Representation

- Continuous Data
- Sampled Data
- Discrete Data
- Cell Types
- Grid Types
- Attributes
- Computing Derivatives of sampled Data
- Implementation
- Advanced data representation
- Conclusions

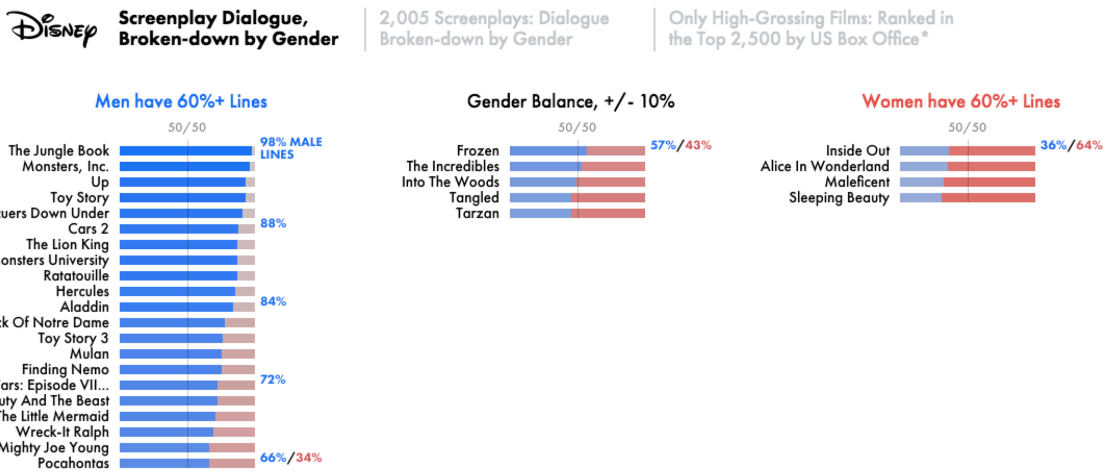
4. The visualization pipeline

- Conceptual Perspective
- Implementation perspective
- Algorithm classification
- Conclusions

From Graphics to Visualization

Need for Visualization: Data visualization has come a long way from its early beginnings in graphics and computer-aided design. The term "data visualization" was first coined in the 1980s, but it wasn't until the advent of digital technologies that the field began to take shape.[5] This article explores the journey from simple graphics to sophisticated data visualizations, highlighting key milestones and innovations that have shaped the field into what it is today.

A Simple Example



The above visualization shows a bar chart comparing the percentage of dialogue spoken by male and female characters in around 2,000 films.

Each bar represents a different film. The height of the blue bars indicates the percentage of dialogue spoken by male characters, while the height of the orange bars shows the percentage spoken by female characters.

The films are sorted from left to right in descending order based on the percentage of dialogue spoken by male characters.

The visualization clearly shows that in the majority of films, male characters dominate the dialogue, with their bars being much taller than the female character bars.

Only a few films have bars of similar height for both genders, indicating more equal dialogue distribution.

This data visualization effectively highlights the significant gender imbalance in dialogue representation across a large sample of popular films.

In summary, the visualization uses a simple bar chart to compare the percentage of dialogue spoken by male and female characters in thousands of films. It reveals a clear disparity, with male characters having significantly more dialogue than female characters in most movies.

Here, Data visualization played a crucial role in highlighting the significant gender imbalance in dialogue representation across a large sample of popular films.[6] The visualization effectively conveyed the disparity in dialogue distribution between male and female characters, making it easier to understand and analyze the data. This representation not only aids in understanding and analyzing data but also enhances retention and facilitates communication of complex information effectively

Graphics Rendering:

Rainbow Tip: Graphics rendering is a fundamental aspect of computer graphics that involves the process of generating images from models by means of computer programs. It encompasses various techniques and algorithms to create visual representations of objects in a virtual environment.[7]

Importance of Graphics Rendering

Graphics rendering plays a crucial role in fields like video games, animation, virtual reality, and simulations. It is essential for creating realistic and immersive visual experiences that engage users and enhance the quality of digital content.

Techniques in Graphics Rendering

1. Rasterization: Rasterization is a common technique that involves converting vector graphics into raster images composed of pixels. It is widely used in real-time rendering for its efficiency in rendering complex scenes.

2. Ray Tracing: Ray tracing is a more advanced rendering technique that simulates the behavior of light rays in a scene to generate highly realistic images with accurate lighting and reflections. While computationally intensive, ray tracing produces photorealistic results.

3. Global Illumination: Global illumination algorithms aim to simulate the complex interactions of light in a scene, including indirect lighting, reflections, and refractions. This technique enhances the realism of rendered images by accounting for light bouncing off surfaces.

Components of Graphics Rendering

1. Geometry Processing: Geometry processing involves transforming and manipulating geometric data to define the shape and structure of objects in a scene. This step includes tasks like mesh generation, tessellation, and deformation.

2. Shading: Shading is the process of determining the color and appearance of surfaces in a rendered image. It involves applying textures, materials, and lighting models to simulate how light interacts with objects.

3. Rendering Pipeline: The rendering pipeline is a series of stages through which geometric data passes to be transformed into a final image. It includes stages like vertex processing, rasterization, fragment shading, and frame buffer operations.

Applications of Graphics Rendering

1. Video Games: Graphics rendering is essential for creating the immersive visuals in video games, including realistic environments, characters, and special effects.

2. Film and Animation: In the film industry, graphics rendering is used to produce visual effects, animations, and CGI sequences that enhance storytelling and visual appeal.

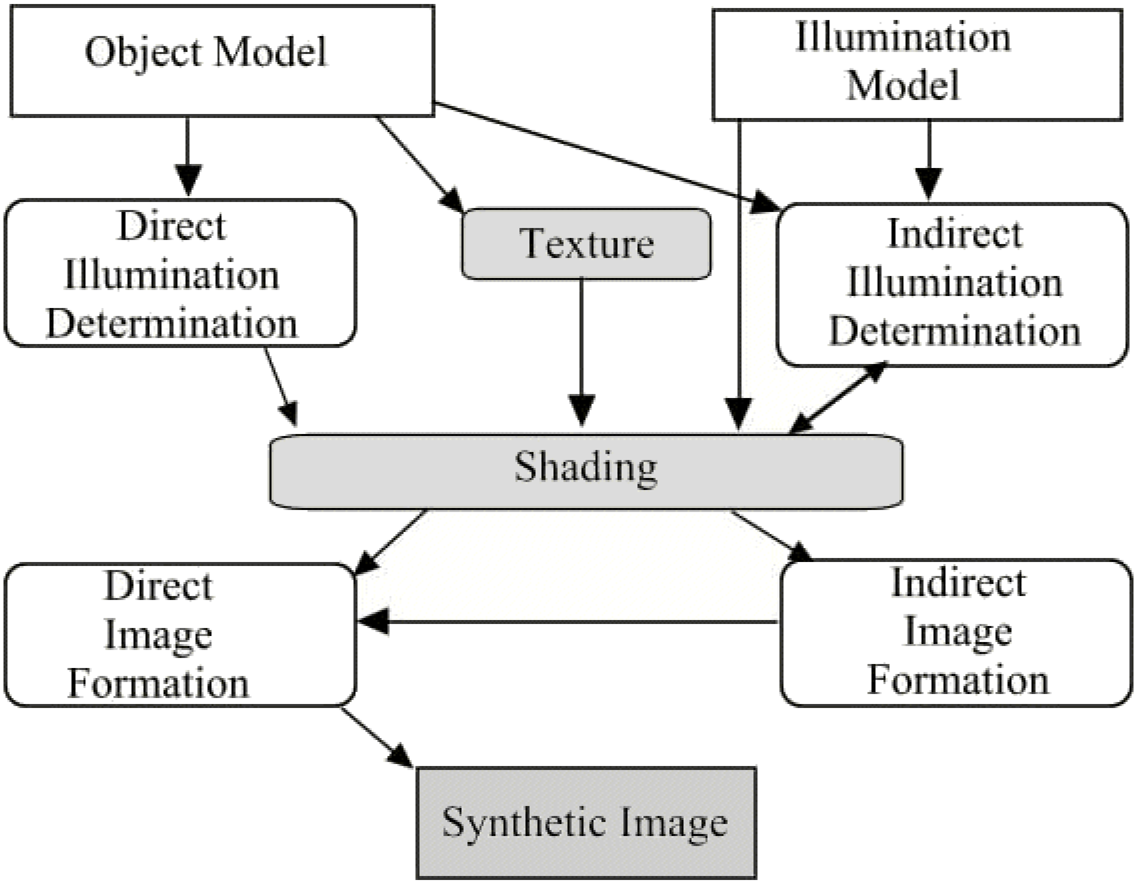
3. Architectural Visualization: Graphics rendering is employed in architectural design to create realistic 3D visualizations of buildings, interiors, and landscapes for planning and presentation purposes

Future Trends in Graphics Rendering

Advancements in graphics rendering technology continue to push the boundaries of visual realism and interactivity. Emerging trends include real-time ray tracing, machine learning-driven rendering techniques, and the integration of virtual and augmented reality for immersive experiences.[8]

In conclusion, graphics rendering is a multifaceted field that underpins the creation of visually compelling digital content across various industries, driving innovation and pushing the boundaries of visual storytelling and user engagement.

Graphic Rendering Techniques



There are several types of graphics rendering techniques used in computer graphics:

Rasterization
Rasterization is a common technique that converts vector graphics into raster images composed of pixels. It is widely used in real-time rendering for its efficiency in rendering complex scenes.

Ray Tracing
Ray tracing is an advanced rendering technique that simulates the behavior of light rays in a scene to generate highly realistic images with accurate lighting and reflections. While computationally intensive, ray tracing produces photorealistic results.

Scanline
The scanline algorithm rapidly creates images by processing a row-based rather than pixel-based basis. It only processes the area within the camera's field of view, making it efficient for real-time rendering.

Z-Buffer
The Z-Buffer is a technique that computes and stores the depth value of each pixel to determine which objects are in front of others. It is often used in conjunction with the Depth of Field effect to create natural blurring of the background.

Shading and Lighting
Shading and lighting techniques manipulate light and dark levels using software effects to simulate how light interacts with surfaces. They allow artists to craft the visual atmosphere with a high level of control over the final aesthetic.

Texture/Bump Mapping
Texture and bump mapping techniques add realistic surface details by applying textures and simulating small-scale surface variations. They enhance the visual realism of rendered objects.

Radiosity
Radiosity is a method that simulates light reflection from diffuse surfaces, where every pixel has its own color and produces or absorbs light. It generates photorealistic results by accounting for indirect lighting.

The choice of rendering technique depends on factors such as scene complexity, computational resources, realism vs. speed, artistic vision, interactivity requirements, and memory constraints.[9]

Ray Casting and Ray Tracing

Ray Casting:

Definition: Ray casting involves creating a ray and calculating what it hits or if it hits a particular object in the scene.

Speed: Ray casting is faster than ray tracing because it stops the first time it hits an object and uses a standard lighting model to determine incoming illumination.

Development: Ray casting was developed in the early 1980s and was successfully exploited in the ground-breaking 3D shooter, Wolfenstein 3D, in 1992.

Basic Algorithm: Ray casting is considered the most basic of computer graphics rendering algorithms and uses the geometric algorithm of ray.

Rendering: Ray casting is well-used in early real-time 3D video games due to its high speed of calculation.

Ray Tracing:

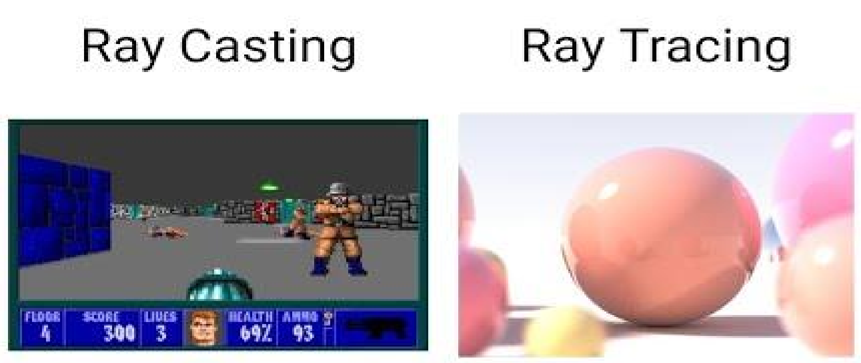
Definition: Ray tracing involves creating a ray and tracing its path as it reflects/refracts through the scene to determine what light might arrive at its source from the other direction.[10]

Realism: Ray tracing can generate a very high degree of visual realism, higher than typical scanline rendering methods, but it uses more compute cycles.

Features: Ray tracing can simulate a wide variety of optical effects like reflection and refraction, making it suitable for applications requiring highly accurate reflections and shadows.

Applications: Ray tracing is used for still images, film and television visual effects, but not yet in real-time applications like video games where speed is critical.

Key Differences:
Speed vs. Accuracy: Ray casting is faster but lacks advanced features like reflections and refractions, while ray tracing is slower but offers higher visual realism.[11]
Recursive Nature: Ray tracing is recursive, involving the propagation of many other rays to determine illumination accurately, while ray casting stops at the first hit without recursion.



Rendering the Height Field Plot

Reading Rainbow Tip: Rendering a height field plot involves visualizing data in a way that represents height information on a surface.

Definition and Applications

- Height Field Plots: Height field plots are used in various scientific fields, especially in geographic data visualization, to display maps with height information combined with a second scalar field.[12]
- Applications: These plots are valuable for representing terrain data, vegetation distribution, or any data where height variations are significant.

Techniques and Description

- Scalar Index Notation: This technique enables the display of two scalar fields over a 2D domain. One field is shown as a surface net, while the other is displayed as a shaded contour map draped over the surface.[13]
- Mapping: The mapping technique allows for the visualization of height information combined with a second scalar field, making it useful for representing complex data sets.

Examples and Visualization

- Example: An example of a height field plot could involve using different hues to represent vegetation types in various regions.
- Visualization: The visualization of height field plots typically involves creating a surface plot where the hue or shading indicates specific attributes related to the data being represented.

Techniques in Height Field Plot Rendering

- Rasterization: Rasterization techniques may be used to convert height field data into visual representations composed of pixels for display.
- Shading and Lighting: Shading and lighting effects can enhance the visual appeal of height field plots by simulating how light interacts with the terrain or surface being visualized.

Importance and Applications

- Importance: Height field plots are crucial for visualizing spatial data, especially in fields like geography, geology, environmental science, and remote sensing.
- Applications: These plots find applications in creating topographic maps, analyzing terrain features, studying vegetation patterns, and understanding geographical data.[14]

In summary, rendering height field plots involves creating visual representations that display height information on a surface, making them valuable tools for visualizing geographic and scientific data in a meaningful and informative way.[15]

Textured Mapping

1. Diffuse Mapping: Applies color and texture to the surface, giving the base appearance.
2. Normal Mapping: Simulates surface details by affecting how light interacts with the surface, without adding extra geometry.
3. Specular Mapping: Defines the shininess or reflectivity of a surface.
4. Bump Mapping: Creates the illusion of bumps, dents, or roughness on a surface by modifying the apparent depth.

Techniques in Texture Mapping

1. UV Mapping: Establishes the relationship between 2D texture coordinates (UV coordinates) and the 3D model's vertices, ensuring accurate and distortion-free application of textures.
2. Mipmapping: Creates a series of progressively lower-resolution versions of a texture to reduce aliasing and improve rendering performance as objects move or change size.
3. Multitexturing: Applies multiple levels of texturing on a single polygon, including detailed-based textures and microtextures, to increase fidelity.
4. Texture Filtering: Improves the quality of the textured model by reducing artifacts like aliasing.
5. Texture Streaming: Loads appropriate mipmap levels into memory based on the draw distance, optimizing texture quality and performance.

Texture Mapping in Animation

Texture mapping plays a crucial role in enhancing the visual quality of 3D models in animation. It allows artists to add surface details, colors, patterns, and other attributes that make the models look more lifelike and natural.

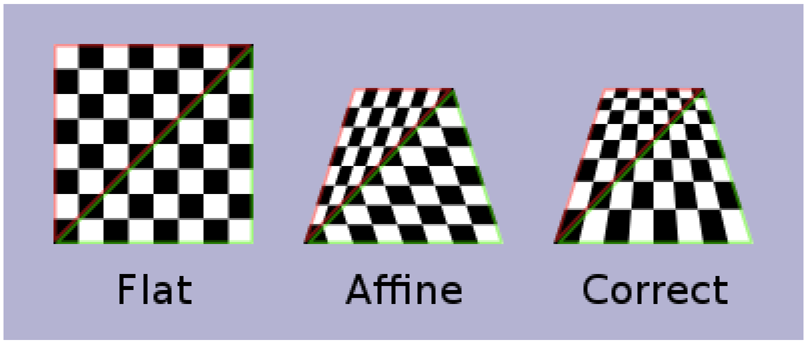
Benefits of Texture Mapping

- Increased Realism: Texture mapping significantly enhances the visual appearance and realism of 3D models by adding detailed surface textures and colors.
- Reduced Geometry: Texture mapping enables the addition of high-frequency detail without increasing the geometric complexity of the model.

- Improved Performance: Techniques like mipmapping and texture streaming optimize texture quality and rendering performance, especially in real-time applications.

In summary, texture mapping is a powerful tool in computer graphics that enables the application of detailed surface appearances to 3D models, enhancing realism, reducing geometry, and improving performance through various techniques and methods.

Transparency Blending



Transparency Blending

Transparency Blending in Computer Graphics

Transparency blending, also known as alpha blending or alpha compositing, is a technique used in computer graphics to combine images or objects with varying levels of transparency to create the appearance of partial or full transparency. It involves mixing the colors of overlapping pixels based on their alpha values, which represent the opacity or transparency of each pixel.[16]

Alpha Channel

The alpha channel is an additional channel associated with each pixel that stores the transparency information. The alpha value ranges from 0 to 1, where 0 represents fully transparent and 1 represents fully opaque. Pixels with alpha values between 0 and 1 are considered partially transparent.

Blending Equations

The most common blending equation used in transparency blending is the "over" operator, which combines the source (foreground) and destination (background) pixels based on their alpha values. The equation is as follows:

$$\$C_{final} = C_s\alpha_s + C_d(1 - \alpha_s)$$

Where:
- $C_{final} is the final color of the pixel
- $C_s is the source (foreground) color
- $C_d is the destination (background) color
- $\alpha_s is the source (foreground) alpha value

This equation ensures that the final color is a weighted average of the source and destination colors, based on their respective alpha values.

Blending Modes

In addition to the "over" operator, there are various other blending modes or blend functions that can be used to achieve different visual effects. Some common blending modes include:

- Multiply: Darkens the image by multiplying the source and destination colors
- Screen: Lightens the image by inverting the source and destination colors, multiplying them, and then inverting the result
- Overlay: Combines the source and destination colors based on the destination color

The choice of blending mode depends on the desired visual effect and the specific application of transparency blending.[17]

Challenges and Limitations

One of the main challenges in transparency blending is the order in which objects are rendered.[19] Since the final color of a pixel depends on the colors and alpha values of all the overlapping objects, the rendering order can significantly affect the final result. To ensure correct transparency, objects are typically sorted from back to front or front to back before rendering.

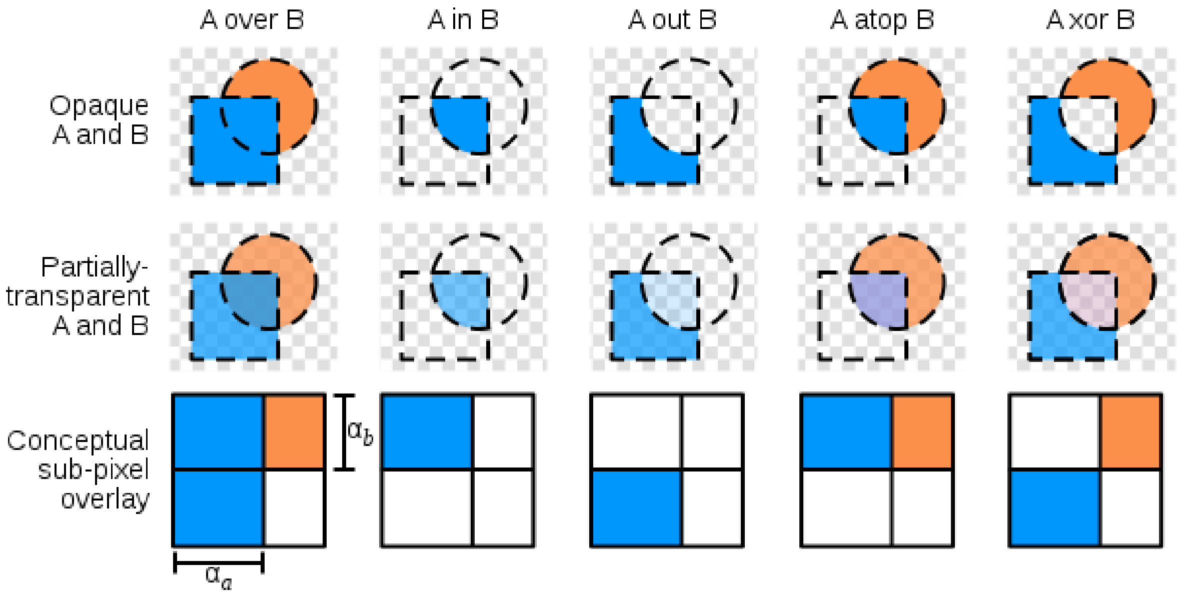
Another limitation is the potential for artifacts or visual glitches, such as z-fighting, which can occur when objects with similar depth values are rendered. This can be mitigated by reducing the depth of the view frustum and carefully modeling distant geometry.

Applications

Transparency blending has numerous applications in computer graphics, including:

- Rendering semi-transparent objects like glass, water, or smoke
- Compositing computer-generated images with live footage in film and visual effects
- Overlaying user interface elements with varying levels of transparency
- Simulating atmospheric effects like fog or haze

In summary, transparency blending is a fundamental technique in computer graphics that enables the combination of images or objects with varying levels of transparency to create realistic and visually appealing results.[18] By leveraging the alpha channel and blending equations, developers can achieve a wide range of transparency effects and visual styles.



Conclusion

The Evolution of Data Visualization: From Graphics to Sophisticated Representations

Data visualization has come a long way since its early beginnings in simple graphics and computer-aided design. The field has undergone a remarkable transformation, evolving from basic techniques to sophisticated methods that enable the effective communication of complex information.

In the past, data visualization relied heavily on static images and diagrams, often lacking interactivity and limited in its ability to convey dynamic information. However, with the rapid advancements in computer graphics and digital technologies, the field has expanded exponentially.

One of the most significant milestones in the evolution of data visualization was the invention of computer graphics and 3D computer graphics. These developments revolutionized the way data could be represented, allowing for the creation of highly detailed and interactive visualizations. The introduction of animation further enhanced the ability to convey information that changes over time, such as in educational animations or timelines.

Today, data visualization has a wide range of applications across various disciplines, including science, education, engineering, medicine, and interactive multimedia.[20] Researchers in the field of computer graphics and visualization are continuously pushing the boundaries of what is possible, exploring new techniques and algorithms to create more effective and engaging visualizations.

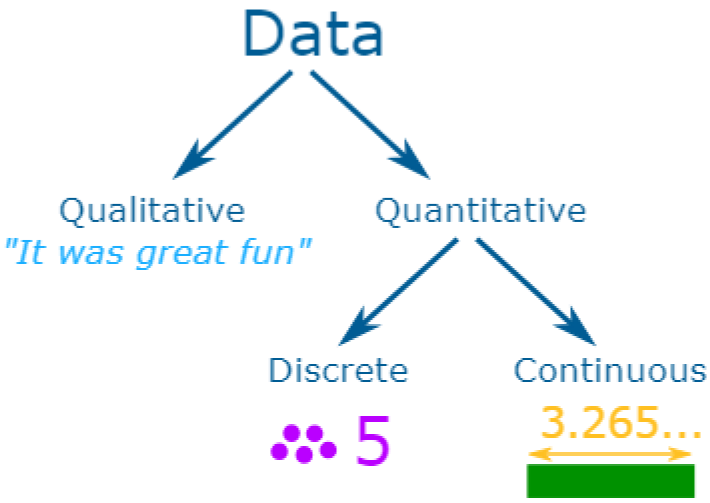
One of the key areas of focus in modern data visualization is scientific visualization, which involves the use of interactive, sensory representations to reinforce cognition, hypothesis building, and reasoning. By transforming and selecting data from simulations or experiments, scientific visualization enables researchers to gain deeper insights and make more informed decisions.

Another important aspect of data visualization is the distinction between abstract and model-based visualizations. Abstract visualizations focus on conceptual constructs, while model-based visualizations either place overlays of data on real or digitally constructed images of reality, or make a digital construction of a real object directly from the scientific data.

As data visualization continues to evolve, researchers are exploring new frontiers, such as virtual reality and augmented reality, to create even more immersive and engaging experiences. These technologies have the potential to revolutionize the way we interact with and understand data, opening up new possibilities for exploration and discovery.

In conclusion, the journey from simple graphics to sophisticated data visualizations has been a remarkable one, marked by significant advancements in computer graphics, animation, and digital technologies. Today, data visualization is an essential tool for communicating complex information, facilitating decision-making, and driving innovation across a wide range of fields. As the field continues to evolve, we can expect to see even more remarkable developments in the years to come, pushing the boundaries of what is possible in the world of data representation and communication.

Data Representation



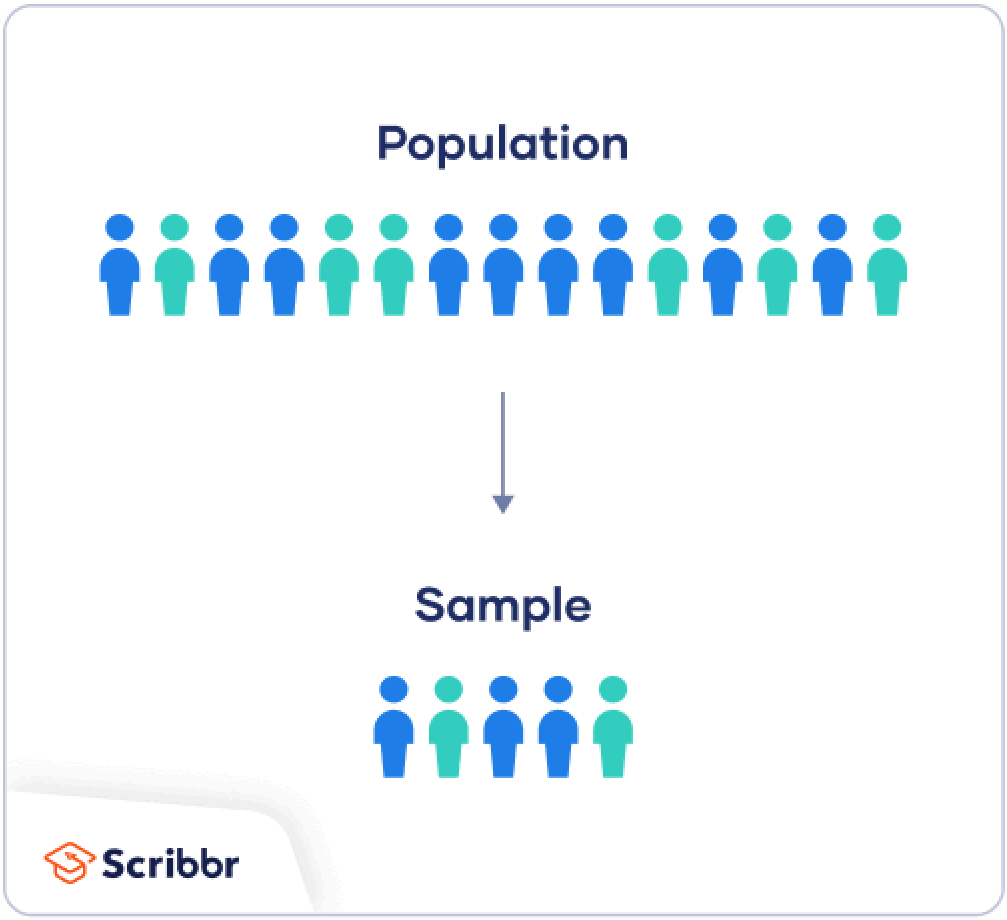
Continuous Data

Definition:Continuous data is the **data that can be of any value**. Over time, some continuous data can change. It may take any numeric value, within a potential value range of finite or infinite. The continuous data can be broken down into fractions and decimals, i.e. according to measurement accuracy, it can be significantly subdivided into smaller sections.

Example:
1.Height - continuous physical measure
2.Volume- continuous spatial measure
3.Temperature-continuous Thermal measure
4.weight - continuous mass measure
5.Time - continuous temporal measure

Sampled Data

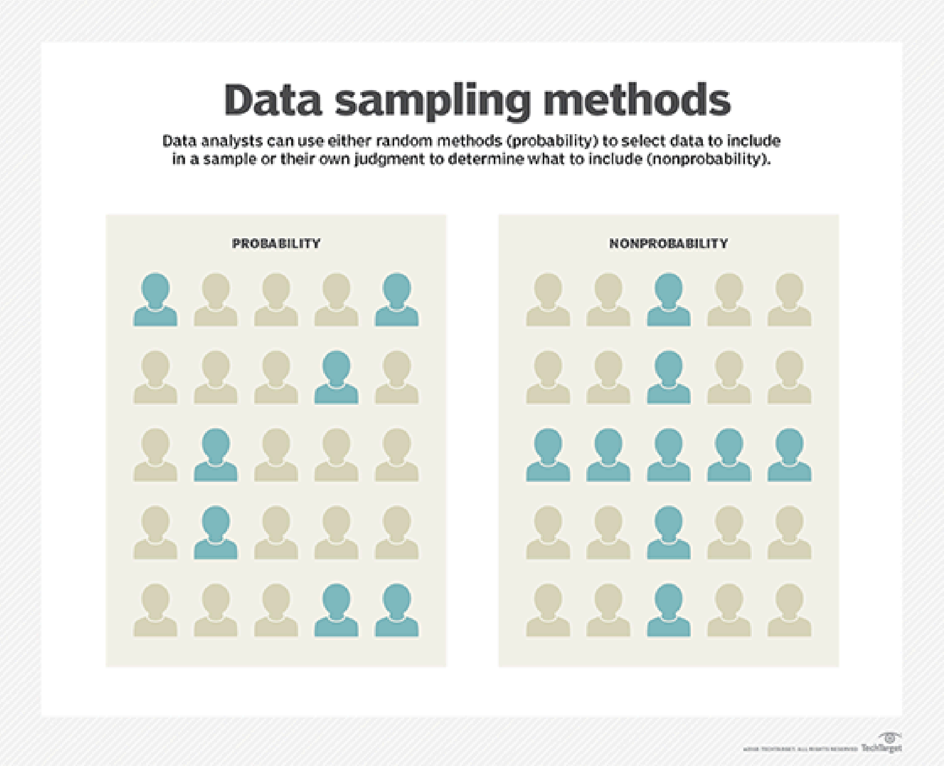
Definition: In statistics, quality assurance, and survey methodology, sampling is the selection of a subset or a statistical sample of individuals from within a statistical population to estimate characteristics, data collected from sampling is called “Sampled Data”.



When you conduct research about a group of people, it's rarely possible to collect data from every person in that group. Instead, you select a sample. The sample is the group of individuals who will actually participate in the research.

To draw valid conclusions from your results, you have to carefully decide how you will select a sample that is representative of the group as a whole. This is called a sampling method. There are two primary types of sampling methods that you can use in your research:

- Probability sampling involves random selection, allowing you to make strong statistical inferences about the whole group.
- Non-probability sampling involves non-random selection based on convenience or other criteria, allowing you to easily collect data.



Data sampling methods

Data analysts can use either random methods (probability) to select data to include in a sample or their own judgment to determine what to include (nonprobability).

Discrete Datasets

Definition: Data that can only take certain values is called discrete data or discrete values. This is data that can be counted and has a limited number of values. It usually comes in the form of whole numbers or integers. These values must fit into certain categories and can't be broken into smaller parts.

Examples:
Discrete data examples include the following:

- The size of your department’s workforce.
- How many new clients you brought on board in the previous quarter?
- How many items are currently kept in stock?

<u>Difference between Discrete data and Continuous data</u>

Discrete Data	**Continuous Data**
Takes particular countable values.	Takes any measured value within a given range.
Discrete data is information that has noticeable gaps between values.	Continuous data is information that occurs in a continuous series.
Discrete data is made up of discrete or distinct values.	Directly in opposition, continuous data includes any value that falls inside a range.
Discrete data can be counted.	Continuous data is quantifiable.
Bar graphs are a visual representation of discrete data.	Continuous data are graphically represented using a histogram.
Ungrouped frequency distribution refers to the tabulation of discrete data against a single value.	The tabulation of continuous data performed against a set of values is called grouped frequency distribution.

For discrete data, a classification like 10-19, 20-29,…, etc., are non-overlapping or mutually inclusive.	For continuous data, classifications such as 10-20, 20-30, etc., overlap or are mutually exclusive.
The discrete function graph exhibits a distinct point that is nonetheless disconnected.	A broken line connects the points on a continuous function graph.
Examples of frequent discrete data include the number of students, children, shoe size, and so forth.	Some common continuous data types are height, weight, time, temperature, age, etc.

Cell Types

-(https://jupyter-notebook.readthedocs.io/en/stable/notebook.html)-

Code cells

A code cell allows you to edit and write new code, with full syntax highlighting and tab completion. The programming language you use depends on the kernel, and the default kernel (IPython) runs Python code.

When a code cell is executed, code that it contains is sent to the kernel associated with the notebook. The results that are returned from this computation are then displayed in the notebook as the cell's output. The output is not limited to text, with many other possible forms of output are also possible, including matplotlib figures and HTML tables (as used, for example, in the pandas data analysis package). This is known as IPython's rich display capability.

There are **three types of cells**: code cells, markdown cells, and raw cells. Every cell starts off being a code cell, but its type can be changed by using a drop-down on the toolbar (which will be "Code", initially), or via keyboard shortcuts.

Markdown cells

You can document the computational process in a literate way, alternating descriptive text with code, using rich text. In IPython this is accomplished by marking up text with the Markdown language. The corresponding cells are called Markdown cells. The Markdown language provides a simple way to perform this text markup, that is, to specify which parts of the text should be emphasized (italics), bold, form lists, etc.

If you want to provide structure for your document, you can use markdown headings. Markdown headings consist of 1 to 6 hash # signs # followed by a space and the title of your section. The markdown heading will be converted to a clickable link for a section of the notebook. It is also used as a hint when exporting to other document formats, like PDF.

When a Markdown cell is executed, the Markdown code is converted into the corresponding formatted rich text. Markdown allows arbitrary HTML code for formatting.

Within Markdown cells, you can also include mathematics in a straightforward way, using standard LaTeX notation: $...$ for inline mathematics and $$...$$ for displayed mathematics. When the Markdown cell is executed, the LaTeX portions are automatically rendered in the HTML output as equations with high quality typography. This is made possible by MathJax, which supports a large subset of LaTeX functionality

Standard mathematics environments defined by LaTeX and AMS-LaTeX (the amsmath package) also work, such as \begin{equation}...\end{equation}, and \begin{align}...\end{align}. New LaTeX macros may be defined using standard methods, such as \newcommand, by placing them anywhere between math delimiters in a Markdown cell. These definitions are then available throughout the rest of the IPython session.

Raw cells

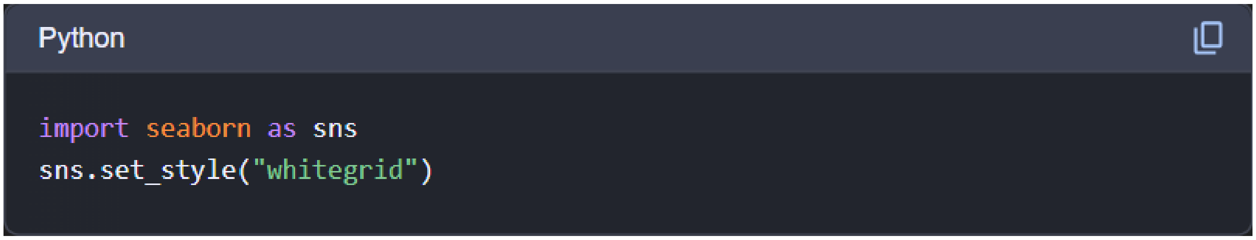
Raw cells provide a place in which you can write output directly. Raw cells are not evaluated by the notebook. When passed through [nbconvert], raw cells arrive in the destination format unmodified. For example, you can type full LaTeX into a raw cell, which will only be rendered by LaTeX after conversion by nbconvert.

Grid Types:

Here are some types of grids in Python Seaborn:

- Whitegrid: This is the default grid style in Seaborn. It has a light gray background with white grid lines.
- Darkgrid: This grid style has a dark gray background with black grid lines.
- Ticks: This grid style only shows the tick marks on the axes, and no grid lines.
- No grid: This grid style does not show any grid lines or tick marks.

You can change the grid style of a Seaborn plot by using the set_style() function. For example, to change the grid style to whitegrid, you would use the following code:

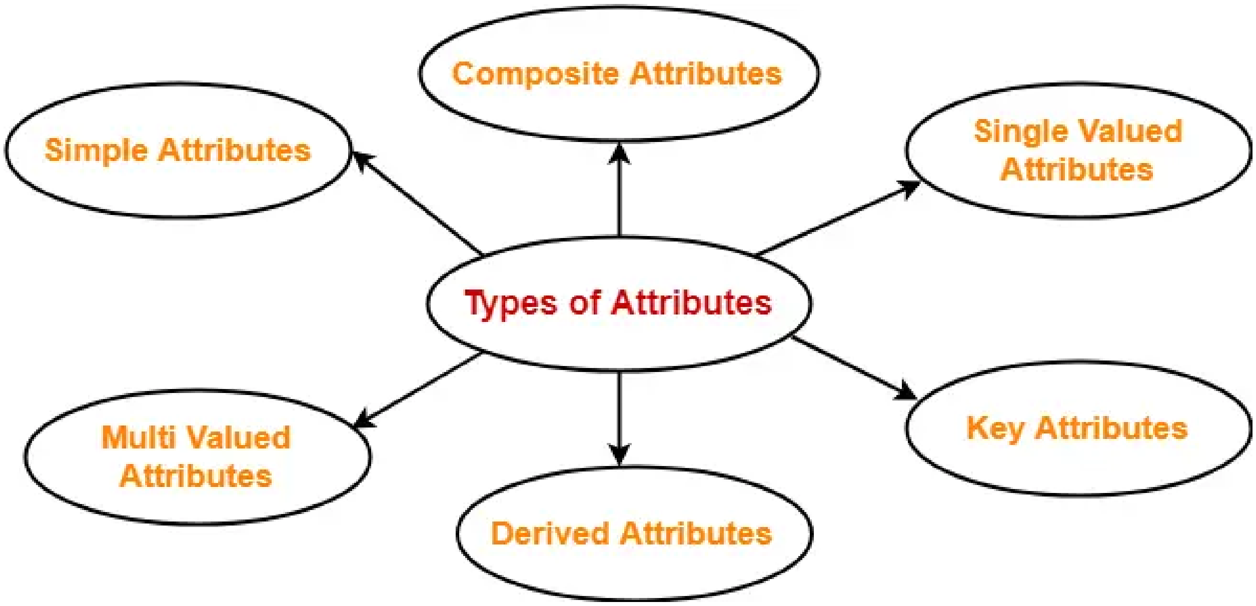


```
import seaborn as sns
sns.set_style("whitegrid")
```

Attributes

-(https://www.statistics.com/glossary/attribute/#:~:text=In%20data%20analysis%20or%20data,red%2C%20yellow%2C%20green).

Definition: In data analysis or data mining, an attribute is a characteristic or feature that is measured for each observation (record) and can vary from one observation to another. It might measured in continuous values (e.g. time spent on a web site), or in categorical values (e.g. red, yellow, green).



Computing Derivatives of Sampled Data

Computing derivatives of sampled data refers to the process of estimating the rate of change (i.e., the derivative) of a function based on discrete data points rather than a continuous function. In practical terms, this means you have a set of data points sampled at specific intervals and you want to find out how the values change with respect to changes in the independent variable (often time or space).

Here are the key concepts involved:

1. Sampled Data

Sampled data are discrete data points collected from measurements or observations at specific intervals. For example, you might have temperature readings taken every hour or stock prices recorded at the end of each trading day.

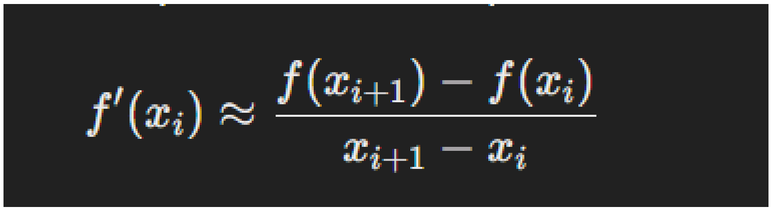
2. Derivative

The derivative of a function at a point measures the rate at which the function's value changes as the independent variable changes. For continuous functions, this is typically done using calculus. For discrete data, this involves estimating the derivative based on the sampled values.

3. Finite Differences

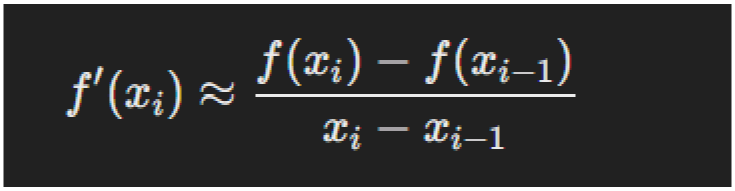
A common method for estimating derivatives from discrete data is using finite difference methods. There are several types:

- Forward Difference: Uses the current point and the next point.



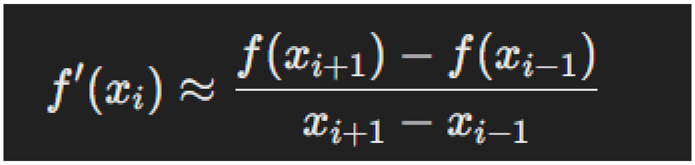
$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

- Backward Difference: Uses the current point and the previous point.



$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

- Central Difference: Uses the average of the forward and backward differences.



$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}$$

4. Higher-Order Derivatives

To compute higher-order derivatives (like the second derivative), finite difference methods can be extended. For example, the second derivative using central differences is:

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{(x_{i+1} - x_i)^2}$$

5. Smoothing

When data are noisy, directly computing derivatives can amplify the noise. Smoothing techniques, such as fitting a polynomial to the data or using moving averages, can help reduce noise before computing the derivative.

Applications

- Engineering: Estimating velocity and acceleration from position data.
- Economics: Determining the rate of change in economic indicators.
- Science: Analyzing rates of change in experimental data, like reaction rates in chemistry or growth rates in biology.

Implementation

Suppose you have temperature readings taken every hour: \(T = [20, 22, 23, 24, 26, 27]\) degrees Celsius at times \(t = [0, 1, 2, 3, 4, 5]\) hours. To estimate the rate of change of temperature at \(t = 2\) hours using central difference:

$$\left.\frac{dT}{dt}\right|_{t=2} \approx \frac{T(3) - T(1)}{3 - 1} = \frac{24 - 22}{2} = 1 \text{ degree/hour}$$

In summary, computing derivatives of sampled data involves using numerical methods to estimate the rate of change of a variable based on discrete measurements. This is a crucial technique in many scientific and engineering applications where continuous monitoring is impractical.

Advanced data Representation

Advanced data representation involves sophisticated techniques and methods for visualizing and interpreting complex data sets. These methods go beyond simple charts and graphs to offer deeper insights, facilitate understanding of intricate relationships, and support decision-making processes in various fields. Here are some advanced data representation techniques:

1. **Multivariate Visualizations**

   - **Scatter Plot Matrix (SPLOM)**: Displays scatter plots for every pair of variables in a dataset, allowing for the examination of relationships between multiple variables simultaneously.
   - **Parallel Coordinates**: Uses parallel axes to represent multiple variables, with each line representing an observation across all axes.
   - **Heatmaps**: Represent data values in a matrix format using color gradients to highlight variations and patterns across multiple dimensions.

2. **Geospatial Visualizations**

   - **Choropleth Maps**: Use color coding to represent data values across geographic regions, often used in demographics and epidemiology.
   - **Heat Maps (Geospatial)**: Overlay color-coded data points on maps to show density and intensity of data, commonly used in crime mapping and real estate.
   - **Cartograms**: Distort the size of geographic regions to represent data values, such as population or GDP.

3. **Network and Graph Visualizations**

- **Node-Link Diagrams**: Visualize relationships between entities using nodes (representing entities) and links (representing relationships), used in social network analysis and computer networks.
- **Force-Directed Graphs**: Position nodes using algorithms that simulate physical forces to reveal clusters and relationships in networks.
- **Adjacency Matrices**: Represent relationships between entities in a matrix format, highlighting connections and patterns.

4. **Dimensionality Reduction Techniques**

- **Principal Component Analysis (PCA)**: Reduces the number of dimensions in a dataset while preserving as much variance as possible, often visualized in 2D or 3D scatter plots.
- **t-Distributed Stochastic Neighbor Embedding (t-SNE)**: Reduces high-dimensional data to two or three dimensions for visualization, particularly useful for exploring high-dimensional datasets.
- **UMAP (Uniform Manifold Approximation and Projection)**: Similar to t-SNE but often faster and better at preserving the global structure of data.

5. **Interactive and Dynamic Visualizations**

- **Dashboards**: Combine multiple interactive visualizations into a single interface, allowing users to explore and analyze data dynamically.
- **Animated Visualizations**: Use motion to show changes over time or to highlight specific patterns and trends.
- **Drill-Down Capabilities**: Enable users to click on visual elements to reveal more detailed data, supporting exploratory data analysis.

6. **Advanced Statistical Visualizations**

- **Box Plots and Violin Plot**: Display the distribution of data points, highlighting medians, quartiles, and outliers, with violin plots adding density information.

- **Density Plots**: Show the distribution of a continuous variable, often overlaid on histograms or used in place of them.
- **Ridgeline Plots**: Show distributions of multiple groups or time points as overlapping density plots, useful for comparing distributions across categories or over time.

7. **3D and Immersive Visualizations**

- **3D Scatter Plots and Surface Plots**: Visualize three-dimensional data points or surfaces, often used in scientific and engineering applications.
- **Virtual Reality (VR) and Augmented Reality (AR)**: Create immersive environments for data exploration, allowing users to interact with data in three-dimensional space.

8. **Time Series and Temporal Visualizations**

- **Time Series Line Chart**: Show data points at successive time intervals, often with annotations and trend lines.
- **Gantt Charts**: Visualize project schedules and timelines, showing the start and end dates of tasks and their relationships.
- **Streamgraphs**: Display changes in data over time, with varying thicknesses representing different categories or values.

9.**Infographics and Storytelling**

- **Narrative Visualizations**: Combine text, images, and visualizations to tell a data-driven story, often used in journalism and presentations.
- Infographics: Use a mix of visuals and text to convey complex information in an easily digestible format, focusing on key messages and insights.
-

10. **Hybrid Visualizations**

- **Matrix Plots**: Combine different types of plots (e.g., scatter, line, bar) in a matrix layout to show relationships between multiple variables.
- **Small Multiple**: Use a series of similar graphs or charts to compare different subsets of data, allowing for easy comparison across categories or time periods.

These advanced data representation techniques enable deeper insights and more effective communication of complex data, making them invaluable in fields such as data science, analytics, and research.

Visualization Pipeline

With information overload becoming the norm, our capacity to glean meaning from data is paramount. Data visualization steps in as a hero, translating raw numbers into clear and compelling visuals. These visuals expose patterns, trends, and connections that would be difficult, if not impossible, to grasp from spreadsheets alone. The secret sauce behind impactful visualizations is the visualization pipeline. This pipeline acts as a roadmap, guiding data through a series of steps – from its initial import to its final transformation into an insightful graphic. This book offers a deep dive into the visualization pipeline, exploring not only the core concepts but also the practical tools and algorithms that make it tick. By understanding the pipeline's inner workings, you'll be empowered to create visualizations that illuminate the stories hidden within your data.

Conceptual Perspective

Understanding the Essence of Visualization: A Symphony of Mind, Eye, and Data

Data visualization is both an art and a science. To create truly impactful visualizations, we need to understand the delicate interplay between human perception, cognition, and the principles of visual design. This chapter delves into the theoretical foundation of visualization, exploring this fascinating dance between data, perception, and cognition.

Unveiling the Secrets of Perception: We'll embark on a journey into the human mind, examining the psychological and neurological mechanisms that shape how we see the world:

- **Color Theory:** We'll explore how colors hold immense power, influencing our emotional response, guiding our attention, and even impacting how we interpret information within a visualization. Understanding color theory equips you to leverage the emotional weight of colors to evoke specific reactions in your audience.
- **Visual Attention:** Not all elements within a visualization are created equal. We'll delve into the science of visual attention, understanding how our eyes naturally focus on specific areas within a visual. By strategically placing key data points and using visual cues, you can ensure your audience sees the most important information.
- **Spatial Reasoning:** Our brains are wired to understand the world in three dimensions. This chapter explores how we perceive spatial relationships within visualizations, including concepts like proximity, size, and position. By effectively utilizing spatial relationships, you can create visualizations that are not only informative but also intuitive for viewers to navigate.

The Science of Cognition: Building Blocks for Effective Design: We'll bridge the gap between perception and how we think, exploring cognitive theories that provide the foundation for strong visual design:

- **Gestalt Principles:** These principles explain how our brains inherently group and organize visual elements. Understanding these principles allows you to create visualizations that are clear, concise, and easy for viewers to comprehend.
- **Cognitive Load:** Every visualization demands a certain amount of mental effort from the viewer. We'll explore the concept of cognitive load, helping you design visualizations that minimize mental strain and maximize information absorption.
- **Information Processing:** Our brains process information in a specific way. This chapter dives into how we process visual data, including concepts like pattern recognition and short-term memory limitations. By understanding these processes, you can tailor your visualizations to effectively communicate your message.

Data, Perception, and Cognition: A Powerful Trio: Through engaging examples and thought-provoking discussions, we'll explore how data, perception, and cognition work together. By understanding this interplay, you'll gain a deeper understanding of the power of visualization and its ability to transform complex information into clear and impactful messages.

Implementation Perspective - From Theory to Action

With a firm grasp of the theoretical underpinnings of visualization, we now embark on the exciting journey of bringing those concepts to life. This chapter delves into the practical aspects of the visualization pipeline, guiding you through the technical steps that transform ideas into interactive and informative visuals.

Building the Foundation: Data Acquisition and Preprocessing

The quality of your data directly impacts the effectiveness of your visualization. Here, we'll explore the crucial steps that prepare your data for visualization glory:

- **The Importance of Data Quality:** We'll discuss the concept of data quality and its various dimensions, such as accuracy, completeness, and consistency. Techniques for identifying and addressing data quality issues will be explored, ensuring your visualizations are built upon a solid foundation.
- **Data Cleaning in Action:** No data set is perfect. This section equips you with the skills to tackle common data issues like missing values, outliers, and inconsistencies. We'll explore practical techniques for data cleaning using popular data manipulation libraries like Pandas (Python) or dplyr (R).
- **Data Transformation: Shaping Your Data for Insights:** Raw data often needs some massaging before it can be effectively visualized. We'll delve into data transformation techniques like aggregation, filtering, and normalization, empowering you to reshape your data to reveal hidden patterns and trends.

From Data to Visuals: Rendering and Interaction Design

Once your data is prepped and ready, it's time to translate it into a visual masterpiece. This section explores the technical aspects of bringing your visualizations to life:

- **Data Encoding: The Art of Mapping Data to Visual Properties:** Data encoding refers to the process of translating data attributes into visual properties like color, position, or size. We'll explore different encoding techniques and how they can be used to effectively represent various data types.
- **Visualization Grammars: A Structured Approach to Visualization Design:** Visualization grammars provide a powerful framework for creating visualizations.

We'll explore popular visualization libraries like Vega-Lite and explore how they leverage grammar concepts to simplify the visualization creation process.

- **Rendering Engines: The Power Behind the Pixels:** The final step in the rendering pipeline involves translating the visualization specification into actual pixels on the screen. We'll discuss different rendering engines and their capabilities, ensuring you choose the right tool for the job.

Engaging Your Audience: Interaction Design Principles

Static visualizations can be informative, but interactive visualizations take engagement to a whole new level. This section explores the principles of interaction design, allowing viewers to explore and manipulate your visualizations:

- **Understanding User Input:** We'll delve into different ways users can interact with your visualizations, including mouse clicks, hover events, and touch gestures.
- **Event Handling: Responding to User Interaction:** Learn how to capture and respond to user input, enabling your visualizations to react dynamically and provide users with deeper insights into the data.
- **Animation: Bringing Your Visualization to Life:** Strategic use of animation can enhance user experience and draw attention to specific elements within your visualization. We'll explore different animation techniques and how to implement them effectively.

Choosing the Right Tools for the Job

The modern visualization landscape offers a plethora of tools and technologies to choose from. This section equips you to navigate this vast ecosystem and select the most suitable tools for your project:

- **Programming Languages: The Building Blocks of Visualization:** We'll discuss the strengths and weaknesses of popular programming languages used for visualization, including Python, R, and JavaScript. Understanding the unique characteristics of each language will help you make informed decisions for your project.
- **Visualization Libraries: Powerhouses of Pre-built Components:** Delve into the world of popular visualization libraries like D3.js, Matplotlib, and Seaborn. We'll explore the rich set of pre-built components and functionalities offered by these libraries, allowing you to create sophisticated visualizations with relative ease.

- **Frameworks and Platforms: Streamlining the Visualization Process:** Frameworks and platforms like Tableau, Power BI, and QlikView offer a user-friendly interface and pre-built functionalities for data exploration and visualization creation. We'll explore the advantages and limitations of these tools, helping you decide if they're the right fit for your needs.

Algorithm Classification - The Invisible Code Behind Stunning Visuals

Effective data visualization isn't just about aesthetics; it's about harnessing the power of algorithms to transform raw data into clear and insightful visuals. This chapter unveils the hidden world of algorithms that underpin visualization, empowering you to understand the magic behind the curtain.

The Pillars of Visualization: Essential Algorithms

Data can be overwhelming, and effective visualization often hinges on strategic data manipulation. Here, we'll delve into the core algorithms that form the building blocks of many visualizations:

- **Data Aggregation: Unveiling Patterns Through Consolidation:** Data aggregation involves grouping and summarizing data points to reveal underlying patterns and trends. We'll explore techniques like counting, summing, averaging, and more, along with practical examples using popular data manipulation libraries like Pandas (Python) or dplyr (R). Discover how to leverage aggregation to create visualizations like histograms, bar charts, and heatmaps.
- **Data Filtering: Focusing on What Matters Most:** Not all data points are created equal. Data filtering allows you to hone in on specific subsets of your data, highlighting areas of interest and removing extraneous information. We'll explore various filtering techniques, including filtering by attribute (e.g., selecting customers from a specific region), filtering by range (e.g., focusing on sales figures within a certain timeframe), and filtering by pattern (e.g., identifying data points that deviate from the norm). Learn how to implement filters to create focused visualizations that cater to specific audiences or answer targeted questions.
- **Clustering: Unveiling Hidden Groups Within Your Data:** Imagine a room full of people. Clustering algorithms act like invisible organizers, grouping similar data points together based on predefined criteria. We'll explore popular clustering techniques like k-means, which partitions data into a specific number of clusters, and hierarchical clustering, which creates a hierarchy of nested clusters.

Discover how clustering algorithms can be used to create visualizations like scatter plots with color-coded clusters, revealing hidden relationships within your data.

- **Dimensionality Reduction: Taming High-Dimensional Data:** The world is full of complex data with many variables. Dimensionality reduction algorithms come to the rescue, transforming high-dimensional data into a lower-dimensional space while preserving the most important information. We'll explore techniques like Principal Component Analysis (PCA), which identifies the most significant underlying dimensions in your data, and t-distributed Stochastic Neighbor Embedding (t-SNE), which excels at preserving local similarities between data points. See how dimensionality reduction allows you to visualize complex datasets in a way that the human brain can comprehend.

Charting the Future: Visualization Meets Machine Learning

The boundaries between visualization and machine learning are rapidly blurring. This chapter explores the exciting intersection of these fields, showcasing the potential for revolutionary advancements in data visualization:

- **Deep Learning-Driven Visualization: A New Frontier:** Deep learning, a subfield of machine learning inspired by the structure and function of the human brain, is making waves in visualization. We'll discuss how convolutional neural networks can be used to automatically generate visualizations based on specific data characteristics, and how generative adversarial networks (GANs) hold promise for creating new and innovative visualization formats. Dive into this emerging frontier and explore the possibilities for deep learning to automate and personalize the visualization creation process.
- **AI-Assisted Feature Extraction: Letting Machines Do the Heavy Lifting:** Feature extraction, the process of identifying the most relevant characteristics from your data, is crucial for effective visualization. Machine learning algorithms are increasingly playing a role in this process. We'll examine how automated feature engineering can discover hidden patterns within your data and how feature selection algorithms can help you identify the most informative features to include in your visualizations. Learn how AI can streamline your workflow and empower you to create more impactful visualizations.
- **Human-Machine Collaboration: A Powerful Partnership:** Imagine a future where humans and machines work together to create the most insightful visualizations possible. We'll discuss the potential for human-machine collaboration in visualization, where humans provide domain expertise and guide

the visualization process, while machines handle complex computations and automate repetitive tasks. Explore the exciting possibilities for this collaborative approach and how it can revolutionize the field of data visualization.

-

Summary

Chapter 1

The journey begins with understanding the foundations of visualization. This chapter dives deep into the human mind, exploring how we perceive and process visual information. You'll learn about color theory and its influence on emotional response and attention, how visual attention guides where our eyes focus within a visualization, and how spatial reasoning allows us to understand the relationships between elements. Additionally, the chapter explores cognitive theories that inform strong visual design, including Gestalt principles for clear organization, cognitive load management to minimize mental strain, and information processing principles to tailor visualizations for effective communication.

Chapter 2

Chapter 2 bridges the gap between theory and practice. It guides you through the visualization pipeline, the step-by-step process of transforming ideas into interactive visuals. Here, you'll gain hands-on knowledge about data preparation, a crucial step that ensures the quality of your visualizations. This includes techniques for cleaning and transforming data to reveal hidden patterns and trends.

Once your data is ready, the chapter dives into the technical aspects of bringing your visualizations to life. You'll explore data encoding, the art of translating data attributes into visual properties like color or size. Visualization grammars, a powerful framework for creating visualizations, are also introduced. The chapter delves into rendering engines, the software that translates your visualization specifications into pixels on the screen.

No interactive visualization is complete without engaging user interaction. This section equips you with the principles of interaction design, allowing viewers to explore and manipulate your visualizations. You'll learn how to capture user input through mouse clicks, hover events, and touch gestures, and how to respond to this input dynamically to provide deeper insights into the data. Animation techniques are also explored for their ability to enhance user experience and draw attention to specific elements.

The chapter concludes by helping you navigate the vast landscape of visualization tools and technologies. You'll gain insights into the strengths and weaknesses of popular programming languages like Python, R, and JavaScript used for visualization. Additionally, you'll explore a range of visualization libraries like D3.js, Matplotlib, and Seaborn, packed with pre-built components for efficient visualization creation. Finally, the chapter discusses frameworks and platforms like Tableau and Power BI that offer user-friendly interfaces and functionalities for data exploration and visualization.

Chapter 3

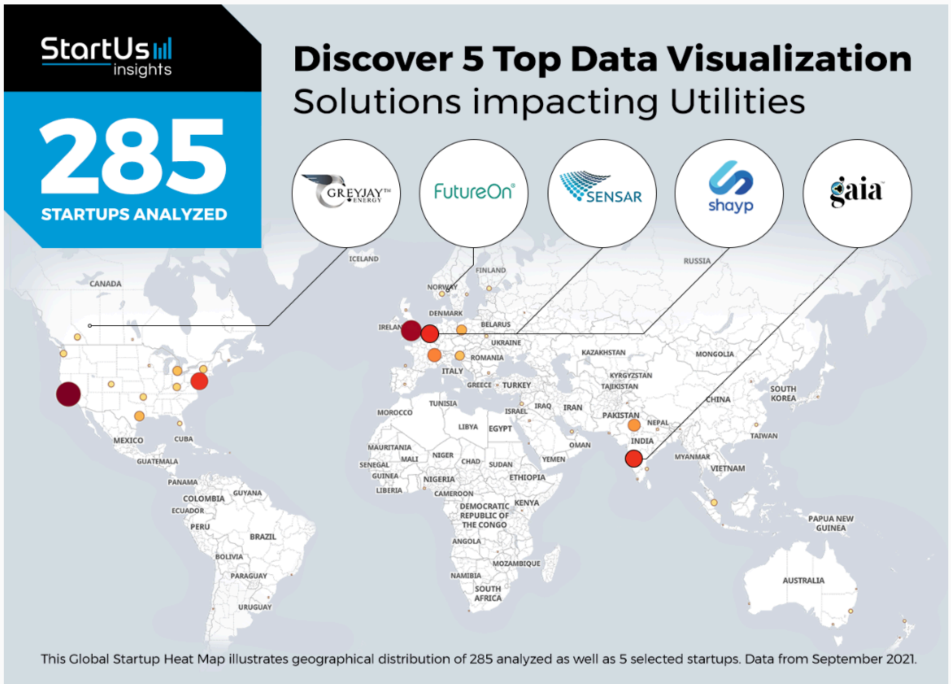
Effective data visualization goes beyond aesthetics; it leverages algorithms to manipulate data for clear insights. Chapter 3 unveils the hidden world of algorithms that power visualizations. You'll delve into essential algorithms like data aggregation (summarizing data points to reveal patterns), data filtering (focusing on specific subsets of data), and clustering (grouping similar data points together). Dimensionality reduction algorithms are also explored, which are crucial for transforming complex, high-dimensional data into a format comprehensible by the human brain.

The chapter also explores the exciting frontier where visualization meets machine learning. You'll learn about deep learning-driven visualization, where algorithms can automatically generate visualizations based on data characteristics. AI-assisted feature extraction is another area of exploration, where machine learning algorithms help identify the most relevant features from your data for visualization. The chapter concludes with a thought-provoking discussion on human-machine collaboration in visualization, where humans provide domain expertise while machines handle complex computations, paving the way for a future of revolutionary advancements in this field.

Exercises

Exercise 1: Business Analytics

Create a comprehensive data visualization report that delves into market trends, customer behaviors, and operational efficiencies within a fictional business scenario. Utilize a variety of visual elements such as charts, graphs, and maps to present key insights that can drive strategic decision-making processes. Consider incorporating interactive features to allow stakeholders to explore the data dynamically and extract valuable information. Focus on simplifying complex data sets to make them easily understandable at a glance, showcasing the power of data visualization in enhancing business analytics.



StartUs insights

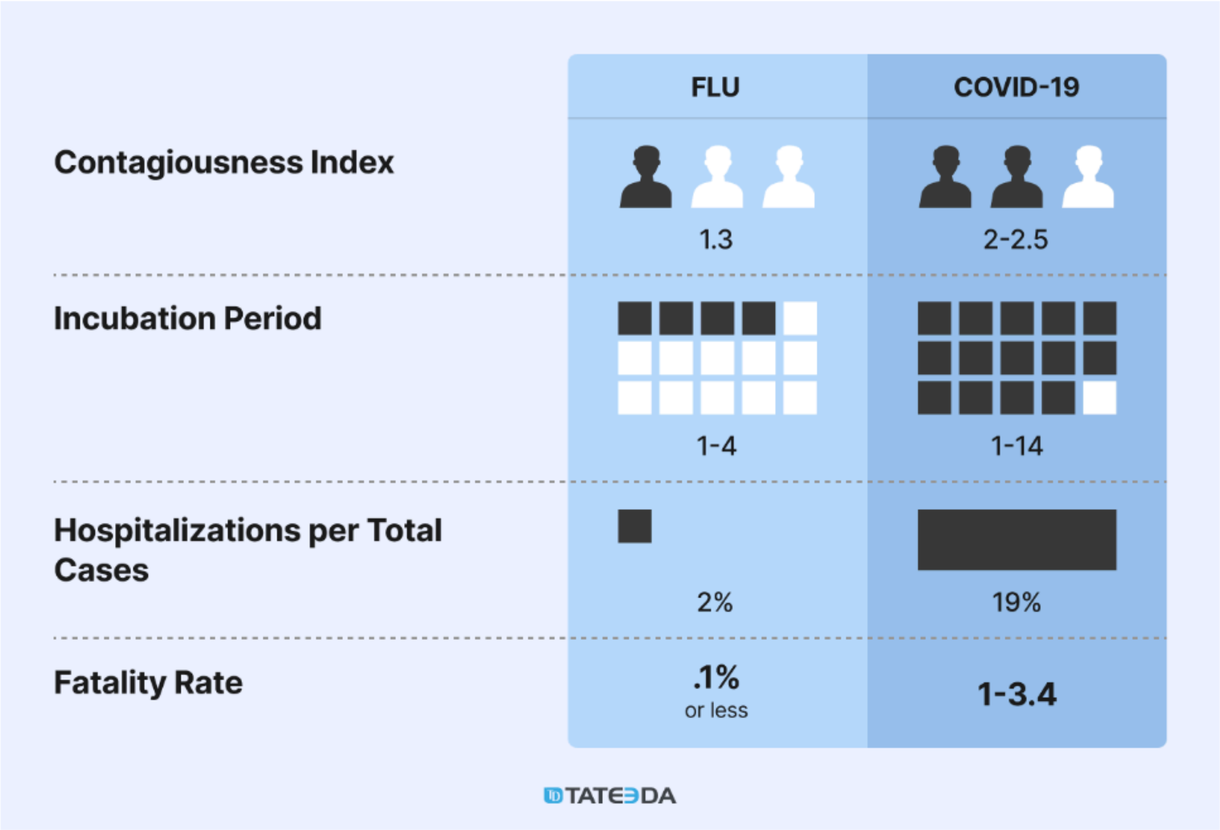
285 STARTUPS ANALYZED

Discover 5 Top Data Visualization Solutions impacting Utilities

This Global Startup Heat Map illustrates geographical distribution of 285 analyzed as well as 5 selected startups. Data from September 2021.

Exercise 2: Healthcare Data Analysis

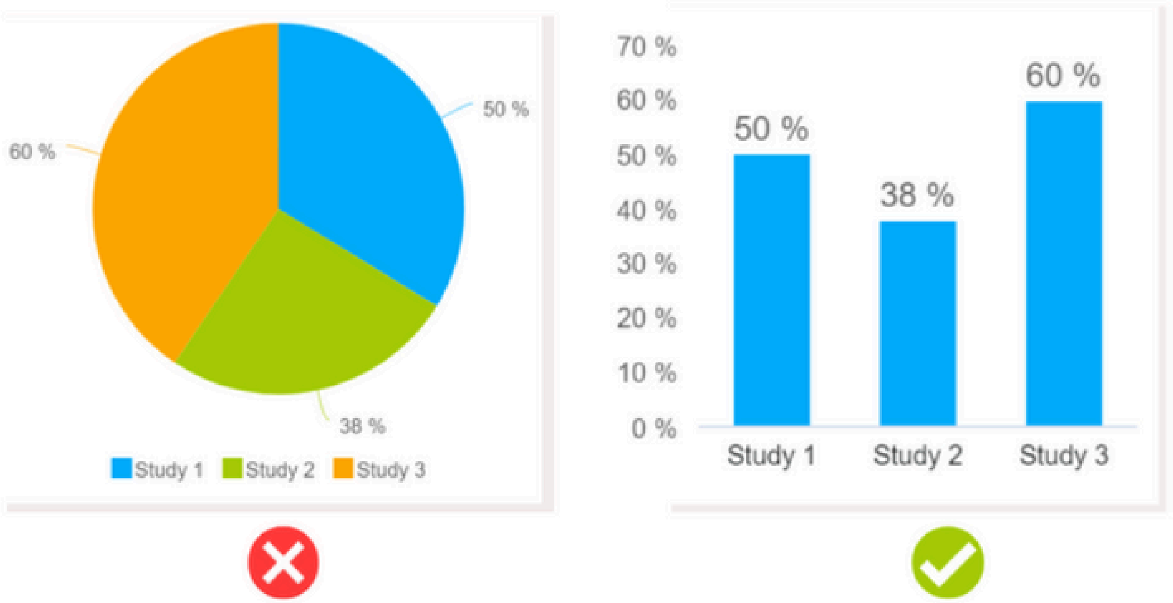
Develop an in-depth data visualization presentation that centers on tracking disease outbreaks, analyzing patient outcomes, and evaluating treatment efficacy in a healthcare setting. Utilize advanced visualization techniques to communicate intricate healthcare data effectively, enabling healthcare professionals to gain valuable insights and make informed decisions. Consider incorporating storytelling elements into your visualizations to highlight key messages and insights that may be overlooked in text-based data presentations. Emphasize the importance of data visualization in improving healthcare outcomes and driving evidence-based practices.



	FLU	COVID-19
Contagiousness Index	1.3	2-2.5
Incubation Period	1-4	1-14
Hospitalizations per Total Cases	2%	19%
Fatality Rate	.1% or less	1-3.4

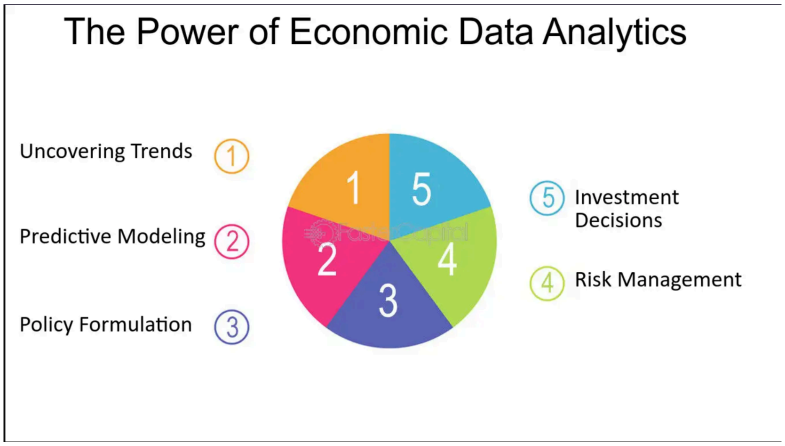
Exercise 3: Scientific Research Visualization

Design a series of interactive visualizations to explore research data, uncover new findings, and communicate results to a broader audience within the realm of scientific research. Utilize cutting-edge data visualization tools and techniques to present complex research data in a clear and engaging manner. Leverage principles of design and cognitive science to enhance comprehension, retention, and decision-making through your visualizations. Consider incorporating dynamic elements such as animations and interactive features to captivate viewers and facilitate a deeper understanding of the research findings.



Exercise 4: Public Policy Analysis

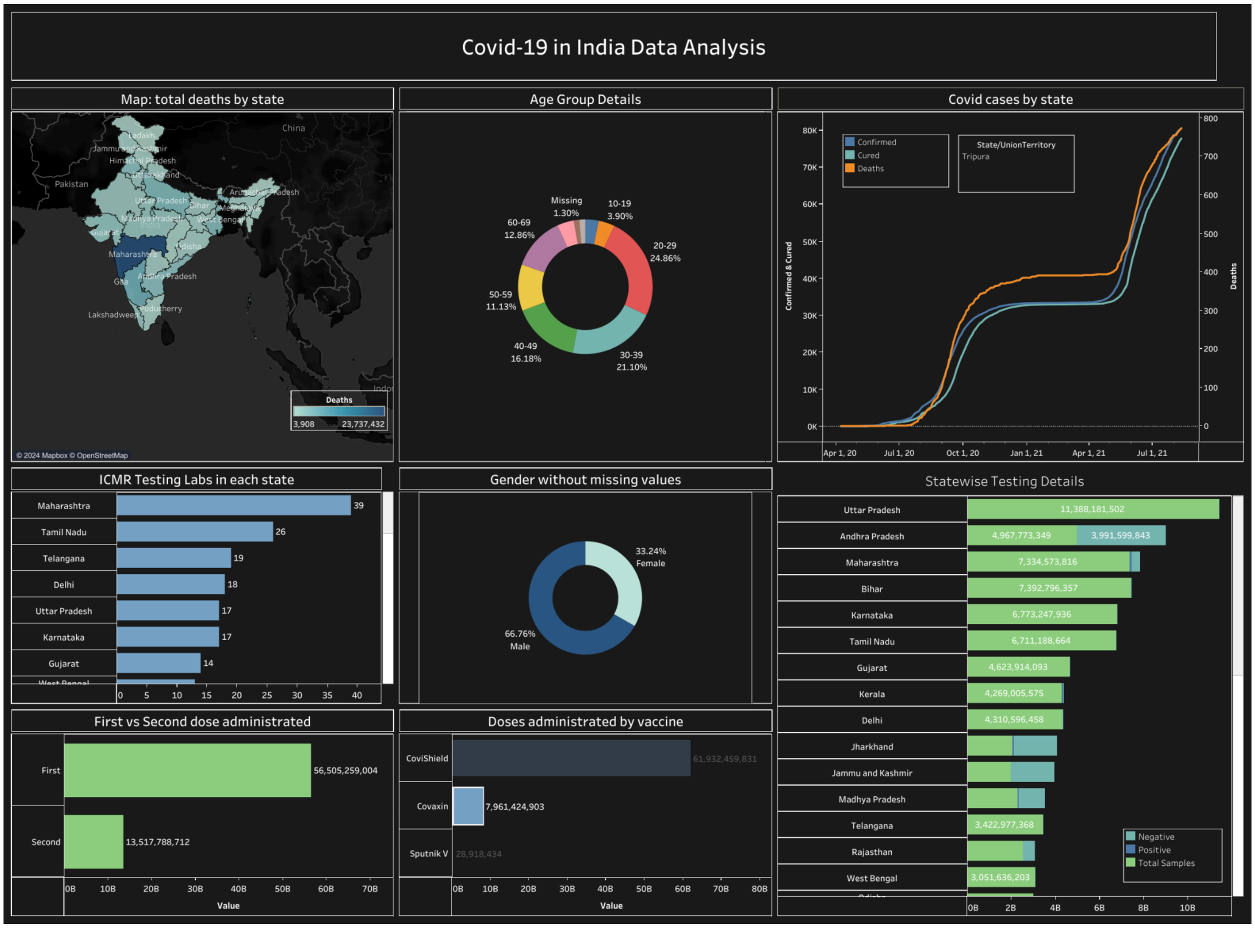
Create a set of visualizations to analyze and address societal issues, ranging from economic trends to environmental impacts, within a public policy context. Utilize data visualization techniques to present complex data sets in a visually appealing and actionable format for policymakers and stakeholders. Explore the use of choropleth maps, heatmaps, and network visualizations to highlight patterns and trends in public policy data. Emphasize the role of data visualization in facilitating data-driven decision-making and fostering greater transparency in policy discussions.



The Power of Economic Data Analytics

Exercise 5: COVID-19 Impact Analysis

Given the ongoing COVID-19 pandemic, conduct a detailed analysis of data related to disease spread, vaccination rates, and healthcare outcomes. Develop interactive dashboards, choropleth maps, and time series visualizations to illustrate the impact of the pandemic on different regions and demographics. Utilize color, size, and motion to highlight trends and patterns in the data, emphasizing the importance of data visualization in understanding and responding to public health crises. Incorporate storytelling elements to convey the urgency and complexity of the COVID-19 situation, showcasing how data visualization can aid in effective crisis management and decision-making.


Covid-19 in India Data Analysis
Map: total deaths by state
China
Ladakh
Jammu and Kashmir
Himachal Pradesh
Uttarakhand
Pakistan
Arunachal Pradesh
Uttar Pradesh
Bihar
Meghalaya
Madhya Pradesh
West Bengal
India
Gujarat
Odisha
Maharashtra
Andhra Pradesh
Goa
Puducherry
Lakshadweep
Indo
Deaths
3,908
23,737,432
© 2024 Mapbox © OpenStreetMap
Age Group Details
Missing 1.30%
10-19 3.90%
60-69 12.86%
20-29 24.86%
50-59 11.13%
40-49 16.18%
30-39 21.10%
Covid cases by state
Confirmed
Cured
Deaths
State/UnionTerritory
Tripura
Confirmed & Cured
80K
70K
60K
50K
40K
30K
20K
10K
0K
800
700
600
500
400
300
200
100
0
Deaths
Apr 1, 20
Jul 1, 20
Oct 1, 20
Jan 1, 21
Apr 1, 21
Jul 1, 21
ICMR Testing Labs in each state
Maharashtra 39
Tamil Nadu 26
Telangana 19
Delhi 18
Uttar Pradesh 17
Karnataka 17
Gujarat 14
West Bengal
0
5
10
15
20
25
30
35
40
Gender without missing values
33.24% Female
66.76% Male
Statewise Testing Details
Uttar Pradesh 11,388,181,502
Andhra Pradesh 4,967,773,349 3,991,599,843
Maharashtra 7,334,573,816
Bihar 7,392,796,357
Karnataka 6,773,247,936
Tamil Nadu 6,711,188,664
Gujarat 4,623,914,093
Kerala 4,269,005,575
Delhi 4,310,596,458
Jharkhand
Jammu and Kashmir
Madhya Pradesh
Telangana 3,422,977,368
Rajasthan
West Bengal 3,051,636,203
Odisha
Negative
Positive
Total Samples
0B
2B
4B
6B
8B
10B
First vs Second dose administrated
First 56,505,259,004
Second 13,517,788,712
0B
10B
20B
30B
40B
50B
60B
70B
Value
Doses administrated by vaccine
CoviShield 61,932,459,831
Covaxin 7,961,424,903
Sputnik V 28,918,434
0B
10B
20B
30B
40B
50B
60B
70B
80B
Value

Citations

[1] Yousif, Z., & Rogers, H. (2020, March 19). *Data Visualization with Tableau*. ResearchGate. https://www.researchgate.net/publication/353047392_Data_Visualization_with_Tableau

[2] *"The Importance of Data Visualization." Harvard Business Review - Bing*. (n.d.). Bing. https://www.bing.com/search?q="The+Importance+of+Data+Visualization."+Harvard+Business+Review&cvid=835899ba1f3149e2b32dc28164d6110b&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBBzg3OWowajSoAgiwAgE&FORM=ANAB01&PC=U531

[3]Fragapane, R. (2024, February 6). *The Importance of Data Visualization Tools for Business*. Maptive. https://www.maptive.com/importance-of-data-visualization-tools-for-business/

[4]*Friendly, Michael. "A Brief History of Data Visualization." Springer - Bing*. (n.d.). Bing. https://www.bing.com/search?q=Friendly%2C+Michael.+"A+Brief+History+of+Data+Visualization."+Springer&cvid=c46f1a21958b45cabd5c1c8445bbf3d3&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDEwNTVqMGo5qAIIsAIB&FORM=ANAB01&PC=U531

[5] *Art of Visualization*. (n.d.). https://www.artofvisualization.com/blog/history-and-evolution-of-data-visualization

[6] *Breaking data barriers with generative AI- and LLM-driven insights*. (2024, May 14). MIT Technology Review. https://www.technologyreview.com/2024/05/01/1092400/breaking-data-barriers-with-generative-ai-and-llm-driven-insights/

[7] Belova, K., & Belova, K. (2023, December 1). *Storytelling With Data Visualization [2024 Guide]*. PixelPlex. https://pixelplex.io/blog/storytelling-with-data-visualization/

[8] Booker, E. (2023, August 22). *How Data Visualization Experts See the Future*. InformationWeek. https://www.informationweek.com/machine-learning-ai/how-data-visualization-experts-see-the-future

[9] *Information visualization : perception for design : Ware, Colin : Free Download, Borrow, and Streaming : Internet Archive*. (2004). Internet Archive. https://archive.org/details/informationvisua0000ware

[10] Wang, W. Y. C., & Wang, Y. (2020, April 1). *Analytics in the era of big data: The digital transformations and value creation in industrial marketing*. Industrial Marketing Management. https://doi.org/10.1016/j.indmarman.2020.01.005

[11] Hürtgen, H., & Mohr, N. (2018, April 27). *Achieving business impact with data*. McKinsey & Company. https://www.mckinsey.com/capabilities/quantumblack/our-insights/achieving-business-impact-with-data

[12] *"Data Visualization in Healthcare." Health IT Analytics. - Bing*. (n.d.). Bing. https://www.bing.com/search?q=%22Data+Visualization+in+Healthcare.%22+Health+IT+Analytics.&cvid=e18b606fcc294e3eb0d4846a657d5dea&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBBzUyNGowajmoAgiwAgE&FORM=ANAB01&PC=U531

[13] Wong, B. (2012, December 1). *Visualizing biological data*. Nature Methods. https://doi.org/10.1038/nmeth.2258

[14] *Data Privacy*. (n.d.). World Bank. https://www.worldbank.org/en/programs/accountability/data-privacy

[15] *An Overview of Data Visualization*. (2019, November 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9012031

[16] *InfoGuides: Computer Graphics: ACM SIGGRAPH Course Notes*. (n.d.). https://infoguides.rit.edu/computer-graphics/coursenotes

[17] Küçükkarakurt, F. (2021, December 27). *INTRODUCTION TO GAME GRAPHICS*. DEV Community. https://dev.to/fkkarakurt/introduction-to-game-graphics-3l89

[18] *"Graphics Rendering Techniques for Film and Animation." Animation Magazine - Bing*. (n.d.). Bing. https://www.bing.com/search?q="Graphics+Rendering+Techniques+for+Film+and+Animation."+Animation+Magazine&cvid=0a7f7ea7c9744e7a8dd74d9cd219f213&gs_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRRg70gEHMzgwajBqOagCCLACAQ&FORM=ANAB01&PC=U531

[19] Keller, H. (2018, August 14). *How to Get the Most Out of Your Architectural Renderings*. Architectural Digest. https://www.architecturaldigest.com/story/architectural-renderings-tips-adam-divino-z-renderings

[20] *A deep dive into NVIDIA's vision for the future of GPU rendering*. (n.d.). https://garagefarm.net/blog/nvidias-vision-for-the-future-of-gpu-rendering-a-deep-dive#:~:text=The%20AI%20Revolution%3A%20NVIDIA's%20emphasis,haven't%20even%20imagined%20yet.