Homework 1

Milena Belianovich
Wednesday, January 31

Part 1: Generate your own data and visualize it

1. Create a box plot for visualization of both arrays. (for code refer to the GitHub link in the footnote)

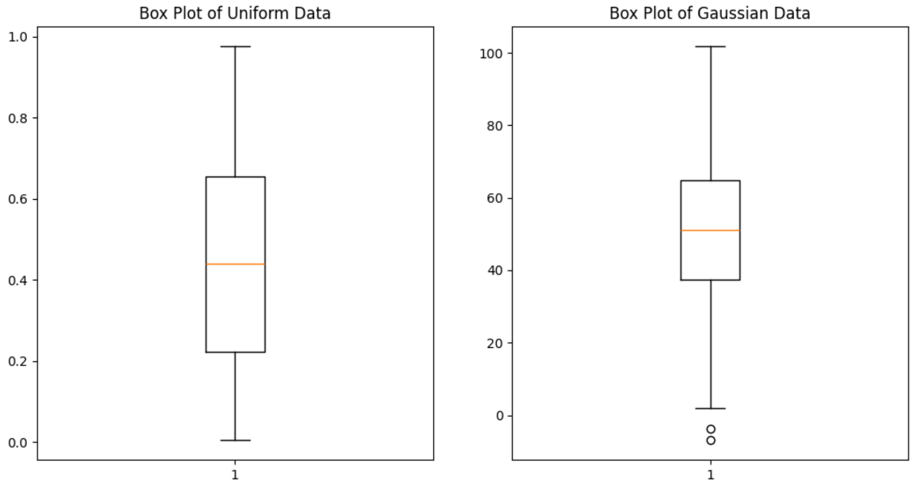

Figure 1: Boxplot visualizations for uniform and Gaussian distributions

2. Partition the data into 20 bins and create a histogram of both arrays using the 20 bins with a bar chart. You may not use a histogram function from a plotting library, however, you may use a bar chart function.

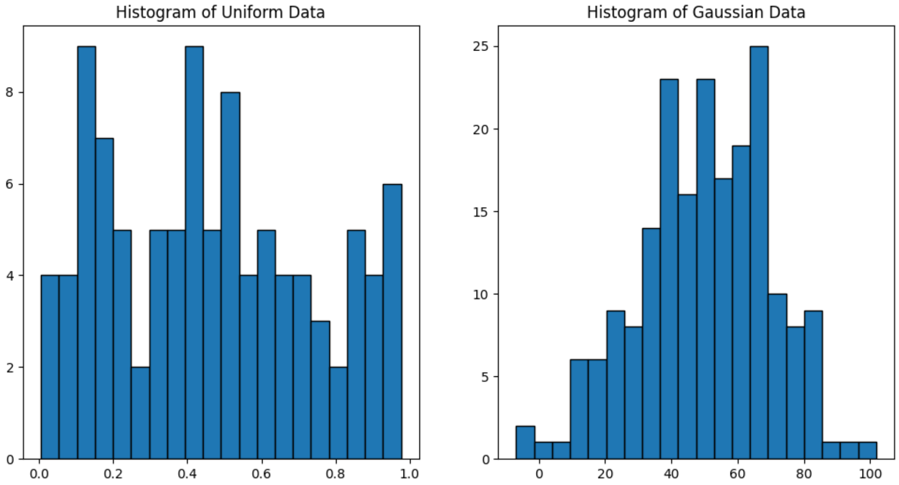

Figure 2: Histogram visualizations for uniform and Gaussian distributions

code: https://github.com/milenabel/CS6635-VisforScD

3. Write the arrays into a binary file. Read it back into an array. Visualize the arrays that were read in by plotting the cumulative distribution function as a line graph. (for code refer to the GitHub link in the footnote)

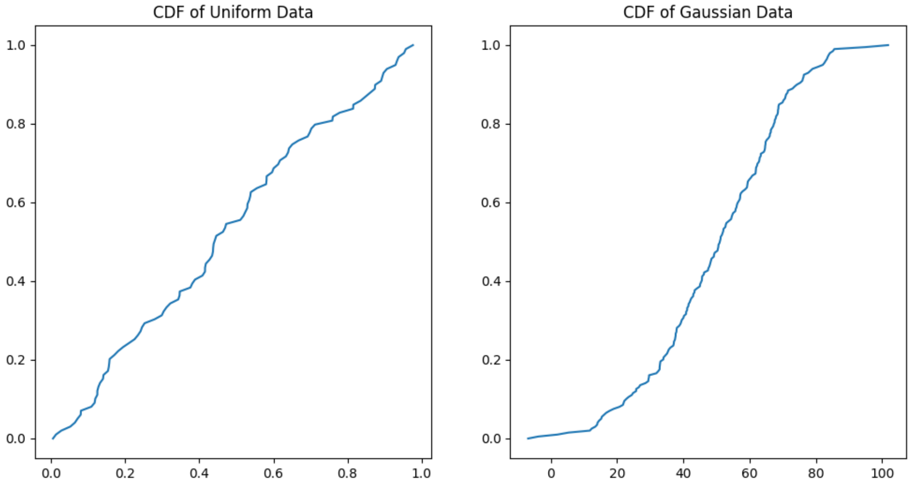

Figure 3: CDF visualizations for uniform and Gaussian distributions

4. Create 2D arrays using uniform random sampling and gaussian random sampling with 5,000 points on [0,1] x [0,1]. Plot the arrays with a scatter plot and compare. Note: You can use Python's built in random.sample package. (for code refer to the GitHub link in the footnote)

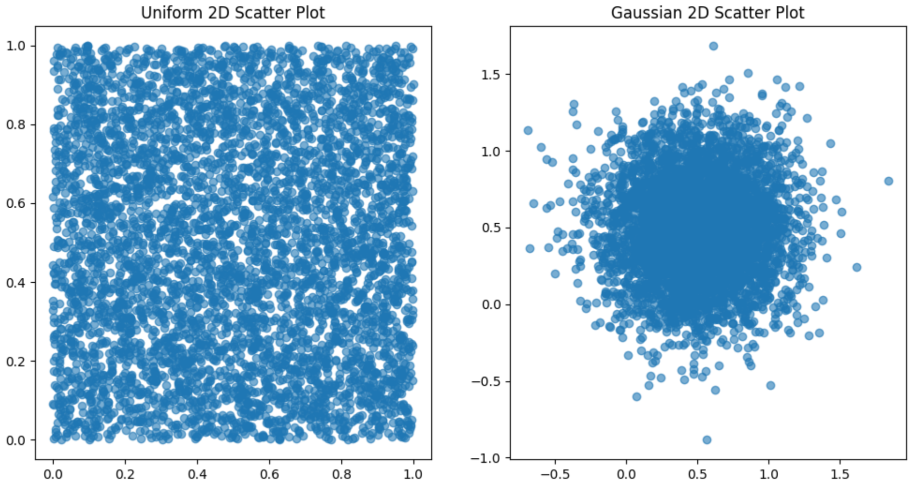

Figure 4: Scatter plot visualizations for uniform random sampling and gaussian random sampling

code: https://github.com/milenabel/CS6635-VisforScD

4b) For both sampling arrays, generate 100 bins along both dimensions(think of counting the number of points in each grid cell). The output will be a 2D array of size 100x100. Show these arrays as images. (for code refer to the GitHub link in the footnote)

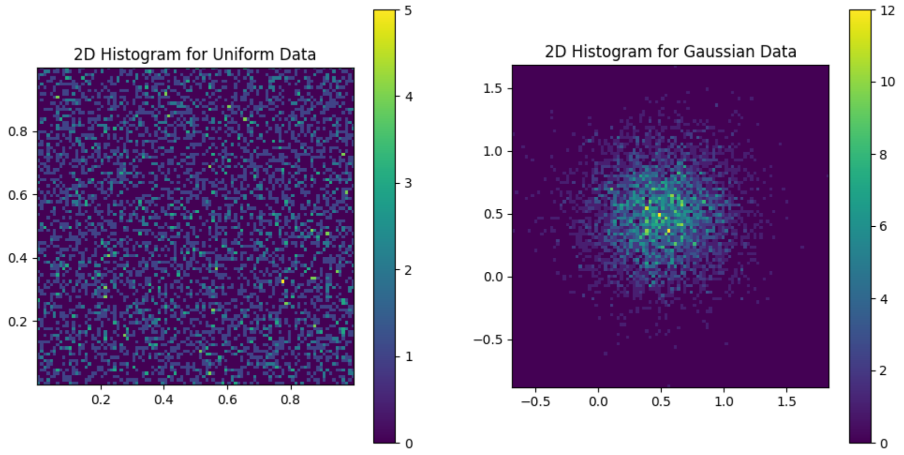

Figure 5: Histogram visualizations for uniform random sampling and gaussian random sampling

4c) Now plot both uniform and gaussian sampled arrays as contour plots with 10 levels. Hint: Use tricontourf in matplotlib as it is unstructured data. (for code refer to the GitHub link in the footnote)

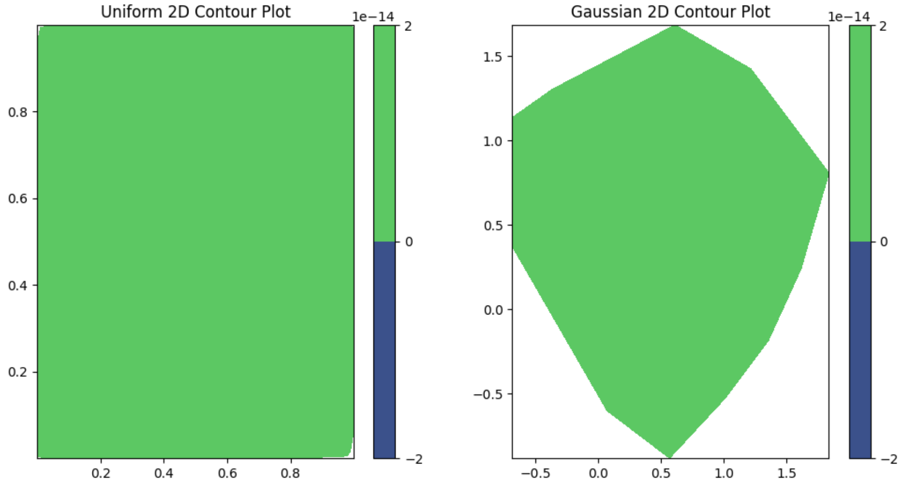

Figure 6: CDF visualizations for uniform and Gaussian distributions

code: https://github.com/milenabel/CS6635-VisforScD

1 Part 2: Interesting data sets for visualization

1. Download the NOAA Land Ocean Temperature Anomalies Data Set: https://my.eng.utah.edu/ cs6635/NOAA-Temperatures.csv. Create a bar plot of the data. Include a label called "Year" along the x-axis and a label called Degrees F +/- From Average along the y-axis. Color each bar with either red/blue based on whether there is a positive/negative change in temperature. Describe trends in the data. (for code refer to the GitHub link in the footnote)

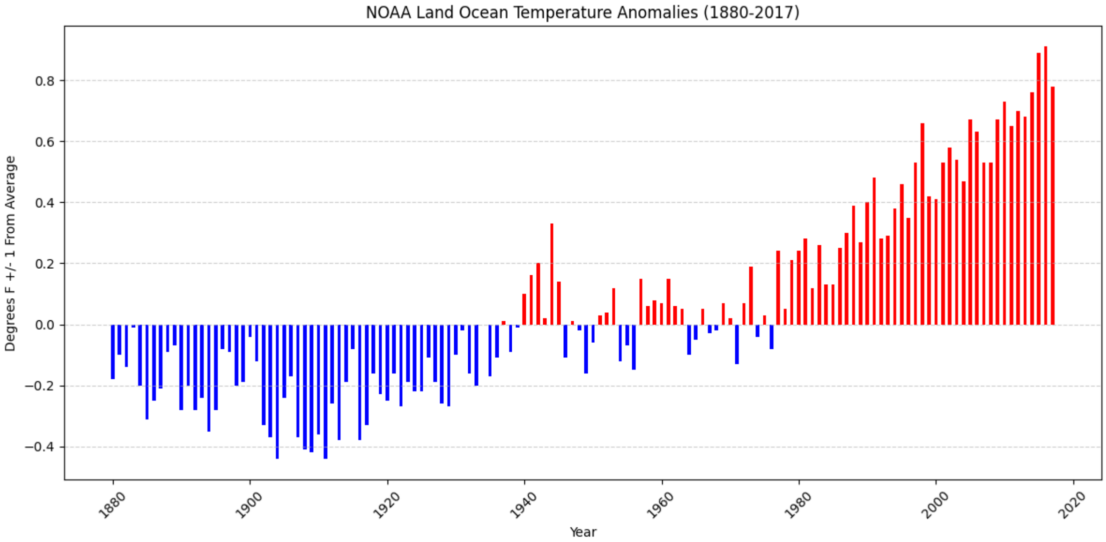

Figure 7: NOAA Land Ocean Temperature Anomalies (1880-2017)

Trends in the data:

Early Period (1880 - 1930): The bars are predominantly blue, suggesting that during this period, the temperatures were frequently below the long-term average. This indicates a cooler climate phase relative to the later years.

Mid 20th Century (1930 - 1975): There is a mix of red and blue bars, but blue bars still dominate. The frequency and intensity of the positive anomalies start to increase, hinting at a shift towards warmer temperatures.

Late 20th Century to Early 21st Century (1975 - 2017): A clear and consistent trend of red bars is observed, indicating that temperatures have been above the average almost every year. Moreover, the magnitude of these positive anomalies appears to increase over time, especially notable from the 1980s onwards.

Recent Decades: The red bars become taller in the most recent decades, which signifies that not only are the temperatures above average, but the degree to which they are above average is also increasing. This is particularly pronounced in the 2000s and 2010s, indicating a significant warming trend.

In summary, the data illustrates a long-term warming trend, with temperatures increasingly above average in recent decades, consistent with the global warming phenomenon observed by climate scientists. The increasing height of the red bars in recent years is particularly concerning as it indicates an accelerating rate of temperature rise.

code: https://github.com/milenabel/CS6635-VisforScD

2. Download the dataset https://my.eng.utah.edu/ cs6635/Breakfast-Cereals.xls and generate a radar/star chart with 8 nutritional statistics for 3 cereals. (for code refer to the GitHub link in the footnote)

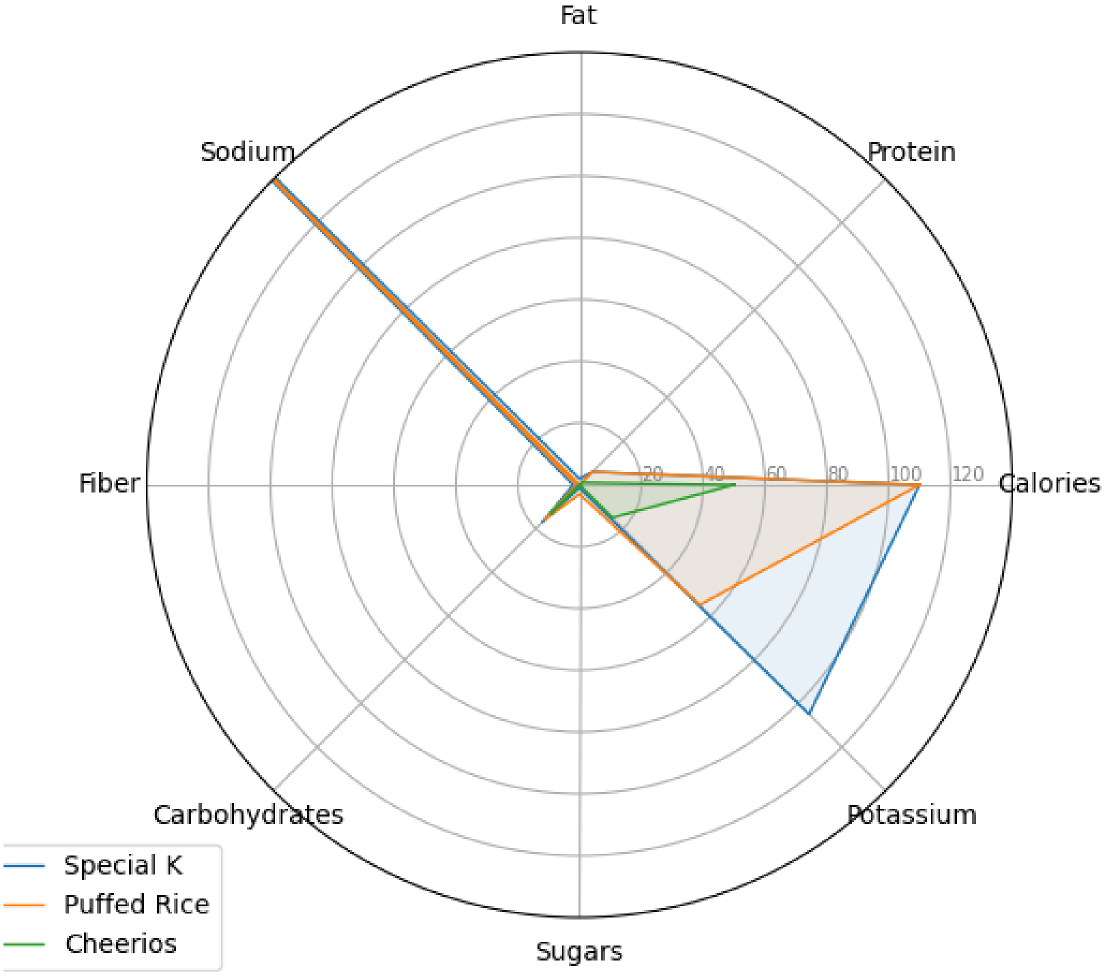

Figure 8: Radar chart of 3 different cereal brands

In this chart I chose to show a closer view of the data (which cut of some of the sodium values) for a more vivid comparison of the variables.

code: https://github.com/milenabel/CS6635-VisforScD

3. Five Thirty Eight maintains a sever with many interesting datasets: https://github.com/fivethirtyeight/data. Choose two different data sets to visualize. Visualize each data set using what you think is the most appropriate plot for the given data. Choose between Parallel Coordinates and Scatter Plot for each data set and use each plotting type only once. It is also helpful to color lines or points according to certain features in the data set to get more insight. Describe the trends you can find in the data by experimenting with these plots. (for code refer to the GitHub link in the footnote)

Picked datasets:

1. USbirths2000-2014SSA.csv contains U.S. births data for the years 2000 to 2014, as provided by the Social Security Administration

2. mmsa-icu-beds.csv combines data from the Centers for Disease Control and Prevention's Behavioral Risk Factor Surveillance System (BRFSS), a collection of health-related surveys conducted each year of more than 400,000 Americans, and the Kaiser Family Foundation to show the number of people who are at high risk of becoming seriously ill from COVID-19 per ICU bed in each metropolitan area, micropolitan area or metropolitan division for which we have data.

For the US Births data, a scatter plot is appropriate to visualize trends over time, such as the number of births by day of the week or month. We can color the points based on the day of the week to see if there are any weekly trends.

For the COVID-19 High-Risk Geography data, a parallel coordinates plot is more suitable. This type of plot can effectively show multi-dimensional data and allow us to observe how different areas compare across various metrics like high risk per ICU bed, number of hospitals, etc. We can color the lines based on total percent at risk

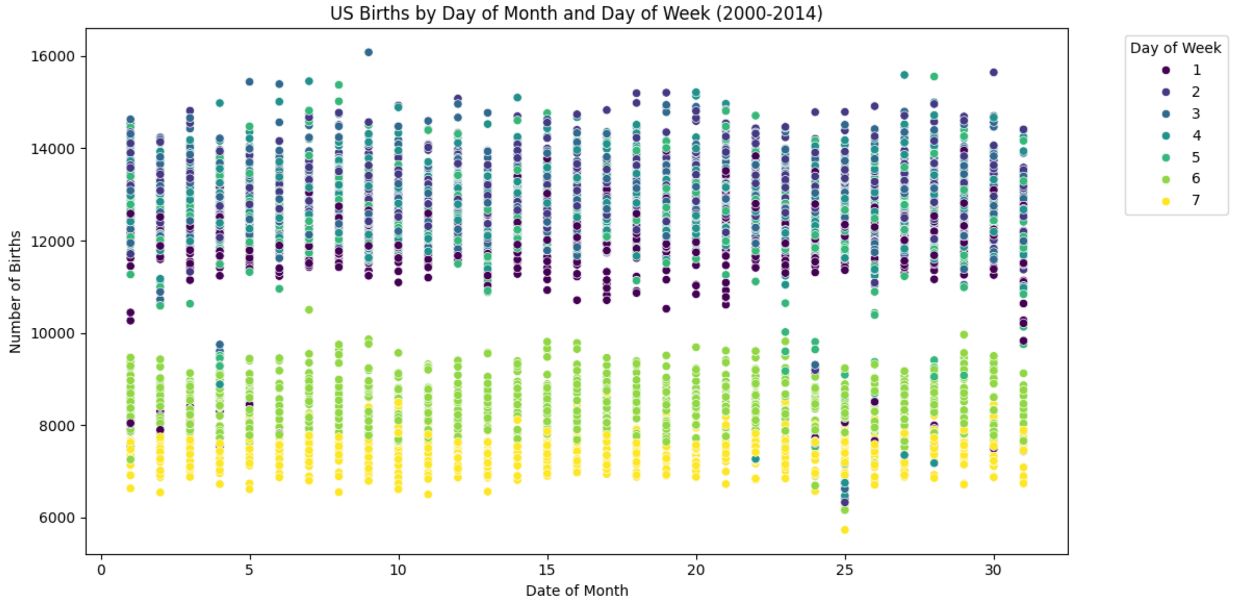

Figure 9: US Births by Day of Month and Day of Week (2000-2014)

The visualizations provide interesting insights into each dataset:

US Births Scatter Plot (2000-2014):

This scatter plot shows the number of births by the date of the month. Different colors represent different days of the week (1 being Monday and 7 being Sunday). Trends such as variations in the number of births on different days of the week can be observed. For example, there may be fewer births on weekends compared to weekdays.

code: https://github.com/milenabel/CS6635-VisforScD

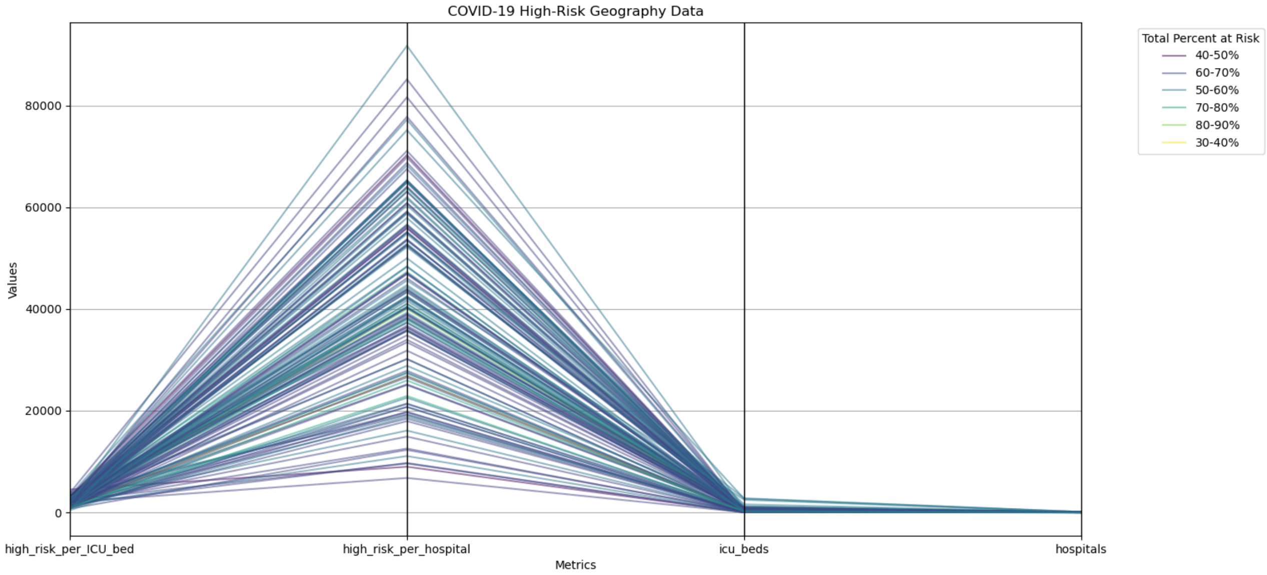

Figure 10: COVID-19 High-Risk Geography Data

COVID-19 High-Risk Geography Parallel Coordinates Plot:

This plot compares different metrics (high risk per ICU bed, high risk per hospital, number of ICU beds, number of hospitals) across various areas. Lines represent different areas, colored by the total percent of individuals at high risk. Trends such as areas with a higher total percent at risk might have a different distribution of healthcare resources (like ICU beds and hospitals) compared to areas with a lower risk percent.

2 Part 3: Questions on The Value of Visualization Paper

1. Why is assessing value of visualizations important? What are the two measures for deciding the value of visualizations?

It's important to assess the value of visualizations to make informed decisions about their use and development. The two measures for deciding the value of visualizations are effectiveness and efficiency.

2.Briefly describe a mathematical model for the visualization block shown in Fig. 1.

The model represents the transformation of data (D) into an image (I(t)) based on a specification (S). This process is encapsulated by the visualization function V, leading to a time-varying image that increases the user's knowledge (K).

3. State four parameters that describe the costs associated with any visualization technique.

Cost Parameters of Visualization Techniques:

Initial development costs (Ci): Costs for developing and implementing the visualization method.

Initial costs per user (Cu): User's time spent on selecting, understanding, and tailoring the visualization method.

Initial costs per session (Cs): Costs for data conversion and setting up the initial visualization specification.

Perception and exploration costs (Ce): Time spent by the user to understand the visualization and explore the data set.

4. What are the pros and cons of interactivity of visualizations?

code: https://github.com/milenabel/CS6635-VisforScD

Pros and Cons of Interactivity in Visualizations:

Pros: Enhances understanding of data, supports exploration when the amount of data is too large for a single image, provides cues for identifying interesting features.

Cons: Can lead to subjectivity in visualizations, increase the cost of exploration (Ce), and make it hard to compare different visualizations due to high customization.

3 Part 4: 3D scalar volume data sets (Only for CS 6635)

MATLAB/Python also can be used for analysis and visualization of 3D volume data sets, such as brain MRI images. Download the brain MRI data set from https://my.eng.utah.edu/ cs6635/T2.nii.gz . The data format is .nii with 320 x 320 x 256 dimensions. Load data in MATLAB/Python. Extract one slice for each axis (three slices total) from the volume and save them as images. Use at least two colormaps to show the three image set and describe the difference this choice makes. (for code refer to the GitHub link in the footnote)

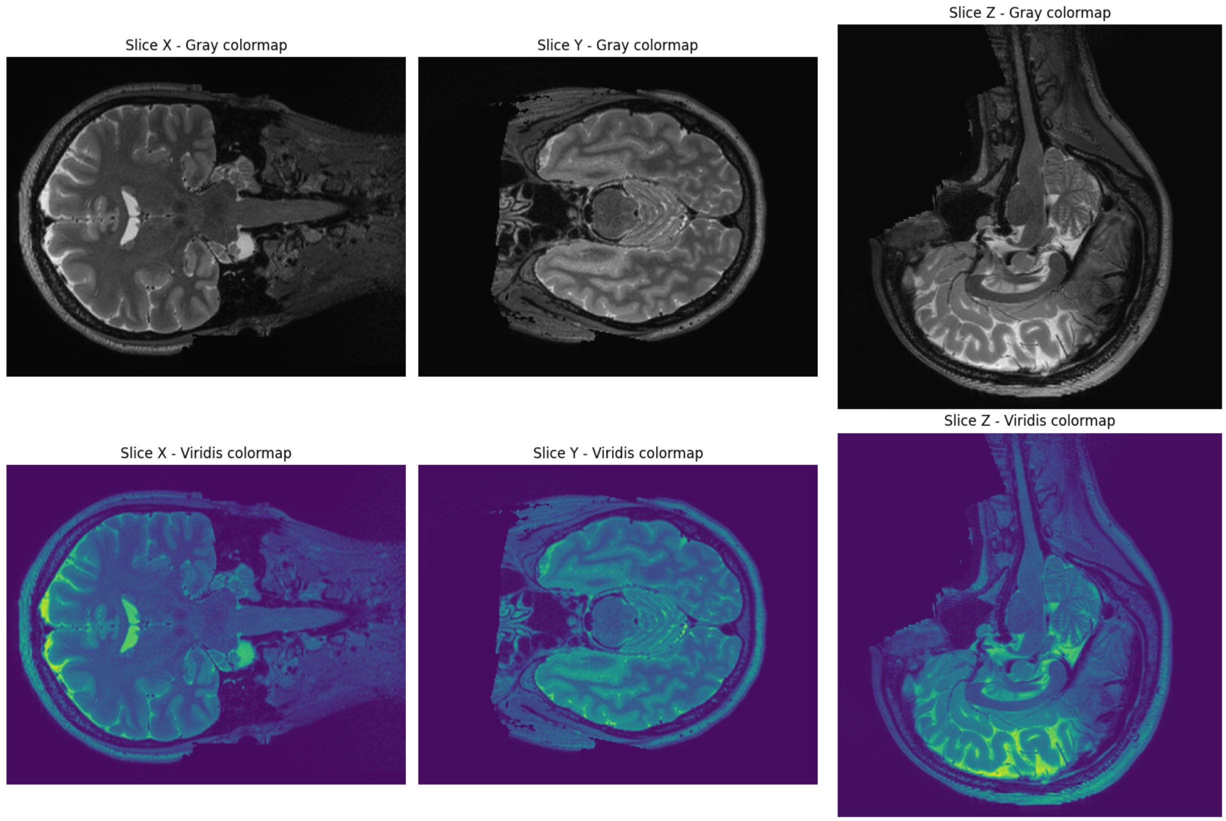
Slice X - Gray colormap
Slice Y - Gray colormap
Slice Z - Gray colormap
Slice X - Viridis colormap
Slice Y - Viridis colormap
Slice Z - Viridis colormap

Figure 11: 3 slices of MRI data represented in 2 different colormaps

Gray Colormap:

- It provides a clear representation of the data, showing the intensity of the MRI signal without any color distraction.

- This colormap is excellent for identifying structures and abnormalities that are familiar to medical professionals.

- It allows for natural interpretation of the images, as brighter areas correspond to higher intensities and darker areas to lower intensities.

Viridis Colormap:

code: https://github.com/milenabel/CS6635-VisforScD

- This colormap is designed to be perceptually uniform, meaning that changes in data values result in perceptually consistent changes in color.

- It can enhance contrast and highlight subtle features that might not be as apparent in a grayscale image.

- However, the interpretation of colors is less intuitive than grayscale and can introduce artificial distinctions that do not correspond to actual differences in tissue properties.

4 Conclusion.

In this assignment, I have explored various data analysis and visualization techniques using Python and its powerful libraries. Here's a summary of the work I accomplished:

- Statistical Distribution Visualization: generated random datasets to represent uniform and Gaussian distributions. Through box plots, I visualized the spread and central tendencies of these datasets, which is crucial in understanding any inherent variability within the data.

- Histograms Without Library Functions: created histograms manually to understand the frequency distribution of the synthetic datasets. This approach provided me with a deeper insight into the fundamental methods of histogram creation, bypassing the convenience of high-level functions for educational purposes.

- Data Serialization: practiced writing to and reading from binary files using numpy's serialization capabilities. This skill is essential for data persistence, especially for large datasets or when preparing data for machine learning models.

- Cumulative Distribution Functions (CDFs): By plotting the CDFs, I observed how values in the datasets cumulatively distribute, which helps in assessing the probability distribution and the expected value over a range.

- 2D Data Scatter Plots and Histograms: examined the distribution of 2D random data using scatter plots and 2D histograms. These visualizations are instrumental in recognizing patterns, clusters, or anomalies within multidimensional datasets.

- Contour Plots: created contour plots to visualize potential densities and distributions in 2D space, which is particularly useful in fields like meteorology, geography, and medical imaging.

- NOAA Land Ocean Temperature Anomalies Visualization: created a bar plot to visualize the temperature changes over time over a century of temperature anomaly data, coloring bars in red and blue to indicate positive and negative anomalies, respectively. This visualization highlighted a clear trend towards increasing temperatures in recent years, revealing the concerning trend of climate change.

- Health Risk Analysis: Using the parallel coordinates plot, analyzed and visualized health risk data related to COVID-19. By categorizing the risk percentage into intervals, I made the data more interpretable and the visualization more accessible.

- MRI Data Visualization: loaded and visualized MRI scan data using different colormaps, which illustrated the impact of color mapping on the interpretation of medical imaging data.

Throughout the assignment, I've encountered and overcame challenges such as data preprocessing, handling various data formats, and choosing appropriate visualizations for different types of data. Each step and visualization provided valuable insights, contributing to my overall understanding of the data's story. The skills honed during this assignment are transferable to a wide array of data science applications, from exploratory data analysis to advanced machine learning tasks.