Assignment-1

Data Analysis and Interpretation

Nischal 23b1024

Pushpendra Uikey 23B1023

Nithin 23B0993

Professor: **Sunita Sarawagi**

August 25, 2024

1 Let's Gamble

Understanding the Problem

- **Friend A** has $n+1$ dice.
- **Friend B** has n dice.
- Probability of getting prime number on each throw is $\frac{1}{2}$
- The goal is to find the probability that Friend A will have more wins than Friend B after they both roll all their dice.

Key Idea

- The problem is symmetric in nature, meaning the probability that Friend A wins more often than Friend B is the same as the probability that Friend B wins more often than Friend A.
- Let's denote the probability that Friend A and Friend B have the same number of wins on n throws by K.
- Now if A got prime number on $n+1$th throw A will win.
- Hence the probability that A wins is $\frac{K}{2}$ in getting same wins for n throws case.
- And the probability that Friend A wins more than Friend B or vice versa on n throws is $\frac{1-K}{2}$.

$$P(X > Y) + P(X = Y) + P(X < Y) = 1$$

Summing the Probabilities

By the given points:

$$P(X > Y) = \frac{K}{2} + \frac{1-K}{2}$$

Final Result

Hence the probability of A winning more than B is symmetric, and the result is:

$$P(X > Y) = \frac{1}{2}$$

Conclusion

- The probability that Friend A will have more wins than Friend B is $\frac{1}{2}$, or 50%.

2 Trading Game Analysis

Reasoning

Let P_A be the probability of Team A winning against you, and P_B be the probability of Team B winning against you. Since Team B is better at trading, we assume:

$$P_B > P_A$$

In both sequences, if I win both of the first two trades, then the probability of winning is:

$$\text{Probability of first win} \times \text{Probability of second win}$$

$$\text{For A-B-A}: \quad (1 - P_A)(1 - P_B)$$

$$\text{For B-A-B}: \quad (1 - P_B)(1 - P_A)$$

This shows that if I win by winning the first two trades, I have an equal chance in both sequences. Next, let's consider if I lose the first trade and then win the next two consecutive trades:

$$\text{Probability of first lose} \times \text{Probability of second win} \times \text{Probability of third win}$$

$$\text{For A-B-A}: \quad P_A \times (1 - P_B) \times (1 - P_A)$$

$$\text{For B-A-B}: \quad P_B \times (1 - P_A) \times (1 - P_B)$$

Since the last two terms are the same in both sequences, the first term determines the winning probability. Given that $P_B > P_A$, the sequence B-A-B has a higher probability of winning if the first trade is lost.

Conclusion

Based on the analysis, sequence B-A-B provides a higher probability of winning in scenarios where the first trade is lost. Thus, the preferred choice is:

Choose B-A-B

3 Random Variables

3.1 Problem Statement

Let Q_1, Q_2 be non-negative random variables. Let $P(Q_1 < q_1) \geq 1 - p_1$ and $P(Q_2 < q_2) \geq 1 - p_2$, where q_1, q_2 are non-negative. Then show that $P(Q_1 Q_2 < q_1 q_2) \geq 1 - (p_1 + p_2)$.

Proof:

Let $A = \{Q_1 < q_1\}$ and $B = \{Q_2 < q_2\}$. We know that:

$$P(A) \geq 1 - p_1 \quad \text{and} \quad P(B) \geq 1 - p_2.$$

We seek to bound $P(A \cap B)$, which corresponds to $P(Q_1 < q_1 \text{ and } Q_2 < q_2)$, implying $P(Q_1 Q_2 < q_1 q_2)$.

By the union bound (Boole's inequality), we have:

$$P(A^c \cup B^c) \leq P(A^c) + P(B^c),$$

where A^c and B^c are the complements of A and B, respectively.

Taking complements:

$$P(A^c \cup B^c) = 1 - P(A \cap B).$$

Thus:

$$1 - P(A \cap B) \leq P(A^c) + P(B^c).$$

Substituting $P(A^c) = 1 - P(A)$ and $P(B^c) = 1 - P(B)$, we get:

$$1 - P(A \cap B) \leq (1 - P(A)) + (1 - P(B)).$$

$$1 - P(A \cap B) \leq p_1 + p_2.$$

So:

$$P(A \cap B) \geq 1 - (p_1 + p_2).$$

Since $P(A \cap B)$ represents $P(Q_1 Q_2 < q_1 q_2)$, we have shown that:

$$P(Q_1 Q_2 < q_1 q_2) \geq 1 - (p_1 + p_2).$$

3.2 Problem Statement

Given n distinct values $\{x_i\}_{i=1}^n$ with mean μ and standard deviation σ, prove that for all i, we have $|x_i - \mu| \leq \sigma\sqrt{n-1}$. How does this inequality compare with Chebyshev's inequality as n increases?

Proof:

1. The mean μ is defined as:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

2. The variance σ^2 is:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2.$$

So:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}.$$

Consider the inequality:

$$\sum_{i=1}^{n}(x_i - \mu)^2 \geq (x_j - \mu)^2 + (n-1) \times 0$$

for some j. Therefore:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \geq \frac{1}{n}(x_j - \mu)^2.$$

Multiplying both sides by n:

$$n\sigma^2 \geq (x_j - \mu)^2.$$

Taking square roots:

$$\sqrt{n}\sigma \geq |x_j - \mu|.$$

This inequality can be sharpened to:

$$|x_j - \mu| \leq \sigma\sqrt{n-1}.$$

Comparison with Chebyshev's Inequality:

Chebyshev's inequality states that:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}.$$

As n increases, Chebyshev's inequality provides a probabilistic bound on deviations, while the inequality $|x_i - \mu| \leq \sigma\sqrt{n-1}$ gives a deterministic bound. For large n, Chebyshev's bound becomes less tight compared to the deterministic bound in the given inequality.

4 Staff Assistant

You have n candidates to interview and want to hire the best one. After interviewing the k-th candidate, you must either offer them the job or lose the chance to hire them forever. A strategy is proposed: reject the first m candidates to get an idea of the field's strength, then hire the first candidate who is better than all previous candidates you've interviewed.

Part (a): Probability of Hiring the Best Candidate

Let's denote E as the event of hiring the best candidate, and E_i as the event that the i-th candidate is the best and is hired.

To calculate $\Pr(E_i)$:

- The i-th candidate is the best with probability $\frac{1}{n}$.
- For the i-th candidate to be hired, they must be better than all previous candidates, and the best among the first $i-1$ candidates must be in the first m positions so they are rejected.

Given this, the probability of hiring the i-th candidate (who is the best) is:

$$\Pr(E_i) = \frac{m}{i-1} \times \frac{1}{n}$$

Summing this probability over all i from $m+1$ to n:

$$\Pr(E) = \frac{m}{n} \sum_{j=m}^{n-1} \frac{1}{j}$$

Part (b): Bounding the Sum

We need to bound the sum $\sum_{j=m+1}^{n} \frac{1}{j-1}$ to obtain bounds for $\Pr(E)$.

Using logarithmic approximations:

- **Upper Bound:** $\sum_{j=m+1}^{n} \frac{1}{j-1} \leq \ln(n-1) - \ln(m-1)$, giving:

$$\Pr(E) \leq \frac{m}{n}(\ln(n-1) - \ln(m-1))$$

- **Lower Bound:** $\sum_{j=m+1}^{n} \frac{1}{j-1} \geq \ln(n) - \ln(m)$, giving:

$$\Pr(E) \geq \frac{m}{n}(\ln(n) - \ln(m))$$

Part (c): Maximizing the Probability

We need to maximize $\frac{m}{n}(\ln(n) - \ln(m))$ with respect to m.

- Define the function $f(m) = \frac{m}{n}(\ln(n) - \ln(m))$.
- Differentiate $f(m)$ with respect to m and set it to zero to find the critical point:

$$f'(m) = \frac{1}{n}(\ln(n) - \ln(m) - 1) = 0$$

  This gives $\frac{n}{m} = e$, or $m = \frac{n}{e}$.

- Substituting $m = \frac{n}{e}$ back into the expression gives $\Pr(E) \geq \frac{1}{e}$.

Thus, the probability of hiring the best candidate is maximized when $m = \frac{n}{e}$, ensuring that $\Pr(E) \geq \frac{1}{e}$.

5 Free Trade

Problem Statement

In a queue of traders, each assigned an ID from 1 to 200, the goal is to choose a position to maximize the chances of being the first trader whose ID matches the ID of any previous trader.

Approach

Probability Formula: To maximize the chances, we calculate the probability $P(x)$ that the trader at position x will be the first to have an ID that matches one of the IDs of traders who have already placed their trades.(Assume all $x-1$ members infront of him are having unique ids)

The probability $P(x)$ is given by:

$$P(x) = \frac{(x-1) \cdot \frac{200!}{(200-x+1)!}}{200^x}$$

Ratio Calculation: To determine the best position, we need to compare the probabilities of consecutive positions. We compute the ratio $\frac{P(x+1)}{P(x)}$ to find out where the probability increases or decreases.

The ratio is given by:

$$\frac{P(x+1)}{P(x)} = \frac{x \cdot (200-x+1)}{200 \cdot (x-1)}$$

Compute the Ratio for Specific Positions:

- **For $x = 14$:**

$$\frac{P(15)}{P(14)} = \frac{15 \cdot (200-14+1)}{200 \cdot (14-1)}$$

$$\frac{P(15)}{P(14)} = \frac{15 \cdot 187}{200 \cdot 13}$$

$$\frac{P(15)}{P(14)} = \frac{2805}{2600}$$

$$\frac{P(15)}{P(14)} = 1.078846$$

  Since $\frac{P(15)}{P(14)} > 1$, the probability of being the first trader with a matching ID increases at position 15 compared to position 14.

- **For $x = 15$:**

$$\frac{P(16)}{P(15)} = \frac{16 \cdot (200-15+1)}{200 \cdot (15-1)}$$

$$\frac{P(16)}{P(15)} = \frac{16 \cdot 186}{200 \cdot 14}$$

$$\frac{P(16)}{P(15)} = \frac{2976}{2800}$$

$$\frac{P(16)}{P(15)} = 1.062857$$

$$\frac{P(16)}{P(15)} = 0.960$$

Since $\frac{P(16)}{P(15)} < 1$, the probability of being the first trader with a matching ID decreases at position 16 compared to position 15.

Conclusion

From the calculations, the probability $P(x)$ increases as you move from position 14 to 15, and decreases when moving to position 16. This indicates that the probability of being the first to have a matching ID is maximized at position $x = 15$.

Thus, **to maximize your chances of receiving the free trade, you should choose position 15** in the queue.

6 Update Functions

Problem Statement

Suppose that you have computed the mean, median and standard deviation of a set of n numbers stored in array A where n is very large. Now, you decide to add another number to A. Write a python function to update the previously computed mean, another python function to update the previously computed median, and yet another python function to update the previously computed standard deviation. Note that you are not allowed to simply recompute the mean, median or standard deviation by looping through all the data. You may need to derive formulae for this. Include the formulae and their derivation in your report. Note that your python functions should be of the following form:

$$\text{function newMean} = \text{UpdateMean}(\text{OldMean}, \text{NewDataValue}, n, A),$$
$$\text{function newMedian} = \text{UpdateMedian}(\text{OldMedian}, \text{NewDataValue}, n, A),$$
$$\text{function newStd} = \text{UpdateStd}(\text{OldMean}, \text{OldStd}, \text{NewMean}, \text{NewDataValue}, n, A).$$

Also explain, how would you update the histogram of A, if you received a new value to be added to A? (Only explain, no need to write code.) Please specify clearly if you are making any assumptions.

Proof:

Update Mean

To update the mean, use the following formula:

$$\text{new\_mean} = \frac{\text{old\_mean} \times n + \text{new\_data\_value}}{n+1} \tag{1}$$

where n is the number of data points before adding the new value.

Update Median

To update the median with a new value, consider the following cases:

- **Even number of data points**:
  - If the new value is less than or equal to the smaller middle value, the new median is the larger middle value.
  - If the new value is greater than or equal to the larger middle value, the new median is the smaller middle value.
  - If the new value is between the two middle values, the new median is the new value itself.
- **Odd number of data points**:
  - If the new value is less than or equal to the middle value, the new median remains the same middle value.
  - If the new value is greater than the middle value, the new median is the next value in the sorted dataset.

Update Standard Deviation

To update the standard deviation, use the following formula:

$$\text{new\_std} = \sqrt{\frac{(n-1) \times \text{old\_std}^2 + (\text{new\_data\_value} - \text{old\_mean})^2}{n}} \tag{2}$$

where old_std is the standard deviation before adding the new value, old_mean is the mean before adding the new value, and n is the number of data points after adding the new value.

Update Histogram

To update the histogram:

- Add the new value to the dataset.
- Re-bin the data into the existing histogram bins:
  - Increment the count for the bin that contains the new value.

– If the new value falls outside the existing bins, create a new bin or adjust the bin edges as needed.

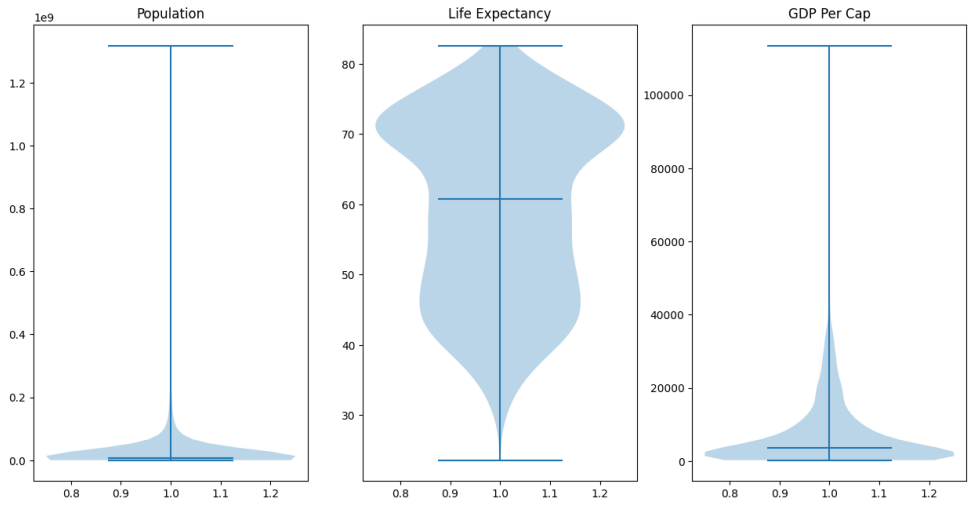
Alternatively, consider using an online histogram update algorithm, such as the "Streaming Histogram" algorithm, which efficiently updates the histogram without re-processing the entire dataset.

7 Plots

In statistics, a plot is a graphical technique for representing a data set. It is used to show the relationship between two or more variables.

Violin Plot

A violin plot is a statistical graph that combines aspects of a box plot and a kernel density plot to provide a more comprehensive view of data distribution. It displays data points across different categories along with their probability density, showing the full distribution of the data. The "violin" shape is created by plotting a mirrored density plot on either side of the box plot, which represents the interquartile range and median of the data. This visualization allows for quick identification of the data's central tendency, spread, and overall distribution, including any potential skewness or multimodality.



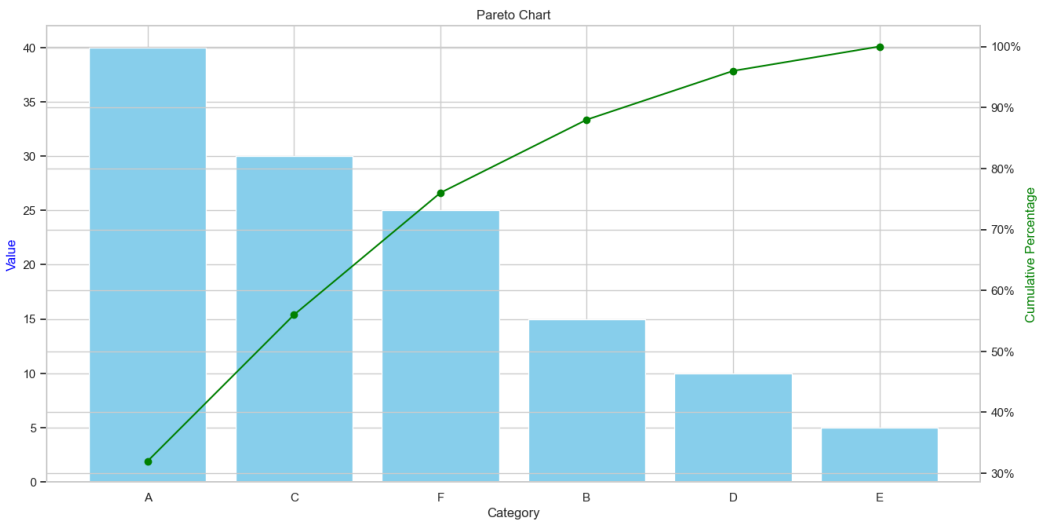
Uses of Violin Plot

- **Comparing Data Distributions:** Violin plots are ideal for comparing the distribution of data across multiple categories or groups. They show variations within each group, including the spread, central tendency, and density, making it easier to see differences and similarities.
- **Visualizing Skewness and Kurtosis:** Unlike traditional box plots, violin plots can reveal the skewness (asymmetry) and kurtosis (tailedness) of the data distribution. The shape of the violin plot indicates whether the data is skewed to the left or right and how concentrated or dispersed the data is.

- **Identifying Multimodal Distributions:** Violin plots can show whether a dataset has multiple modes (peaks), which box plots may not reveal. This feature is useful for identifying underlying patterns in the data that might suggest subgroups or different data-generating processes.
- **Exploratory Data Analysis:** During exploratory data analysis, violin plots provide a quick and detailed overview of data distribution, helping analysts understand data behavior and detect outliers, clusters, or trends that might need further investigation.
- **Analyzing Experimental Results:** In scientific research, violin plots are often used to analyze experimental results, especially when comparing the effects of different treatments or conditions. They allow researchers to visualize the distribution of outcomes and compare variability and central tendencies effectively.

Pareto Chart

A Pareto chart is a type of bar chart used for visualizing the distribution of data. Named after the Italian economist Vilfredo Pareto, this chart combines both bar graphs and line graphs to display the relative importance of various factors or causes in a dataset. The bars represent individual values or categories in descending order, while the line graph shows the cumulative total. By organizing data this way, the Pareto chart highlights the most significant factors that contribute to a problem or outcome, following the Pareto Principle—often known as the 80/20 rule—which states that roughly 80% of effects come from 20% of causes.



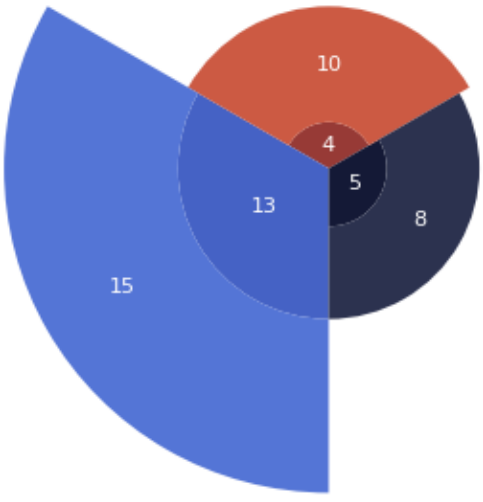
Uses of a Pareto Chart

- **Quality Control and Improvement:** Pareto charts are widely used to identify the most common defects or issues in manufacturing and service processes. By focusing on these critical areas, organizations can implement targeted quality control measures to reduce errors and enhance overall product or service quality.

- **Problem Solving and Root Cause Analysis:** These charts help identify the primary causes of problems by displaying data in order of significance. This allows teams to concentrate their efforts on addressing the most impactful issues, leading to more effective and efficient problem-solving.
- **Customer Complaint Analysis:** Businesses use Pareto charts to categorize and prioritize customer complaints. By addressing the most frequently reported issues, companies can improve customer satisfaction and loyalty.
- **Cost and Resource Management:** Pareto charts assist in identifying the primary drivers of costs or resource use within an organization. By targeting these high-impact areas, companies can implement cost-reduction strategies and optimize resource allocation.
- **Process Optimization:** Organizations use Pareto charts to find bottlenecks or inefficiencies in their processes. Focusing on the most significant factors causing delays or issues helps improve overall efficiency and productivity.

Coxcomb Chart

A Coxcomb chart, also known as a polar area diagram or rose diagram, is a type of circular statistical graphic that displays data in a visually striking way. Named after Florence Nightingale, who popularized its use in the 19th century, the Coxcomb chart represents data in sectors of a circle, with each sector corresponding to a category. The length of each sector from the center represents the magnitude of the data value, while the angle of each sector is fixed. The sectors are typically arranged in a circular fashion around a central point, resembling the petals of a flower. This design makes it easy to compare different categories and see patterns or trends in the data.



Uses of Coxcomb Chart

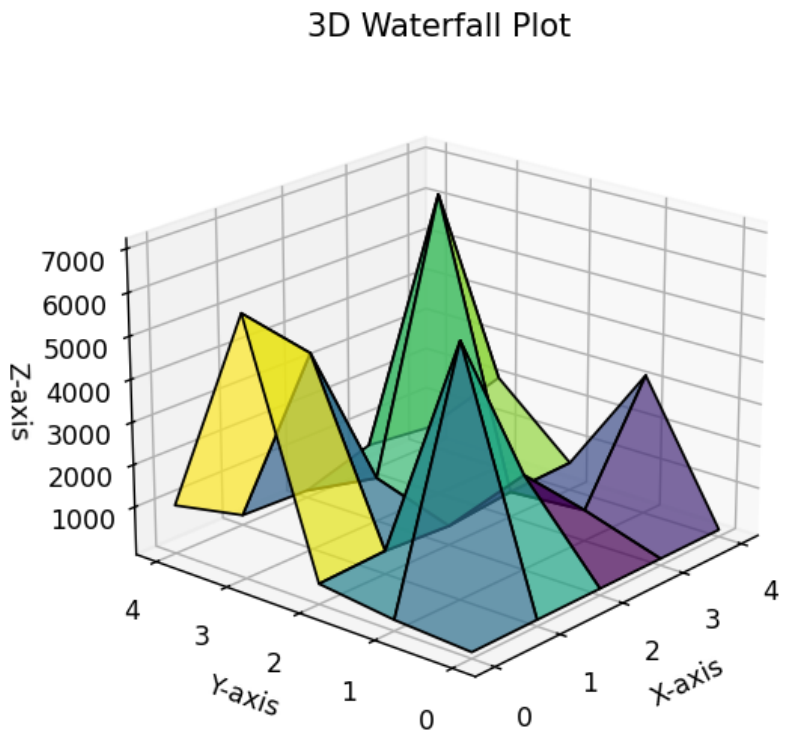
- **Visualizing Categorical Data:** Coxcomb charts are useful for displaying categorical data where each category's size is proportional to its value. They provide a clear visual comparison of how

different categories contribute to a whole, making them ideal for showing proportions and distributions.

- **Historical Data Analysis:** one of the most famous uses of the Coxcomb chart was by Florence Nightingale to illustrate causes of mortality in the British army during the Crimean War. Coxcomb charts are still used today for historical data analysis, especially when comparing different categories over time to highlight trends and changes.
- **Public Health and Epidemiology:** In public health, Coxcomb charts are used to represent data related to disease outbreaks, vaccination coverage, and health interventions. They help convey complex health data in a way that is easy to understand for policymakers and the public.
- **Comparing Temporal Data:** Coxcomb charts are effective for displaying changes in data over specific time periods. For example, they can be used to show monthly sales figures, annual budget allocations, or seasonal variations in data. The circular layout helps in recognizing patterns and fluctuations over time.

Waterfall Plot

A waterfall plot is a three-dimensional graphical representation used to illustrate how two-dimensional phenomena change over time or with another variable, such as frequency or rotational speed. Unlike the more commonly known waterfall charts, which are used for visualizing financial data as a series of floating columns, waterfall plots display multiple curves of data simultaneously. These curves are typically staggered both horizontally and vertically, creating a visual effect similar to a series of mountain ranges. Each "mountain" represents a different point in time or a different value of the variable being analyzed, with "nearer" curves partially masking those behind them. This staggered effect makes waterfall plots particularly useful for showing changes and trends in spectral data, such as audio frequencies, over time.



Uses of Waterfall Plot

- **Analyzing Spectrograms:**Waterfall plots are widely used to visualize spectrograms, which represent the frequency spectrum of a signal as it varies with time. By displaying these spectra simultaneously in a staggered, three-dimensional manner, waterfall plots allow for an easy comparison of changes in frequency content over time.
- **Cumulative Spectral Decay (CSD) Analysis:** In audio and acoustics, waterfall plots are used to analyze cumulative spectral decay, which shows how sound energy decays over time at different frequencies. This is useful for evaluating the performance of audio equipment, room acoustics, and understanding how sound behaves in various environments.
- **Signal Processing:** Waterfall plots are employed in signal processing to observe how signals change over time, particularly in applications involving vibrations, rotations, or any scenario where the signal evolves with another variable. This helps in identifying patterns, anomalies, or specific characteristics in the signal behavior.
- **Vibration and Rotational Analysis:** In mechanical engineering and diagnostics, waterfall plots are used to study vibrations and rotational dynamics. They can illustrate how the amplitude and frequency of vibrations change over time or with varying rotational speeds, helping to diagnose issues with machinery or equipment.

8 Monalisa

Libraries and Tools Used

The project utilized the following Python libraries:

- **Matplotlib**: Used for displaying images and plotting graphs such as the correlation coefficients and the normalized histogram.
- **NumPy**: Provided tools for efficient numerical computations, particularly for array manipulations, calculating the histogram, and computing the correlation coefficient.
- **Scikit-Image (`skimage`)**: Used for reading the Monalisa image and converting it to grayscale. It offers a wide range of image processing functionalities that simplified the development of this problem.