CS215 Data Analysis and Interpretation

Assignment 1

Aditi Singh (23b1053)

Abhilasha Sharma Suman (23b1011)

Navya Garg (23b0982)

Indian Institute of Technology, Bombay

August 2024

Contents

1 Let's Gamble

I claim that the probability of A winning the game is $\frac{1}{2}$ irrespective of the value of n. Here is how it goes

- First let us start with What is the probability that a given die gives a win. For a given dice there are 3 favourable outcomes which are $2, 3, 5$ among the total of 6 outcomes . Therefore

$$P(win) = \frac{3}{6} = \frac{1}{2}$$

  Similarly

$$P(lose) = \frac{3}{6} = \frac{1}{2}$$

  Therefore we observe that probabilities of a die giving a win and that of it giving a loss are equal.

- Now I claim that the scenario where A has more wins than B is symmetrical to the scenario where A has more losses than B. This symmetry arises from the fact that each die roll is independent, and the probability of rolling a prime number (a win) is the same as the probability of not rolling a prime number (a loss). Since the dice are fair and there is no bias towards rolling a prime number, the outcomes of the rolls are equally likely. I can write it as

$$P(A_{wins} > B_{wins}) = P(A_{loses} > B_{loses})$$

- Now since A has one more die (n+1) than B (n), A can either have more wins than B or more losses than B but not both. This can be proved as follows . In any case one of the below 3 must be true.
  - Case 1:- A has more wins than B .
  - Case 2:- A has same number of wins as B. In this case A has more loses because he has one dice more than B.
  - Case 3:- A has less wins than B . In this case also A definitely has more wins than B.

  These 3 are the only possible conditions and are mutually exhaustive. So I can write that

$$P(A_{loses} > B_{loses}) = P(A_{wins} = B_{wins}) + P(A_{wins} < B_{wins})$$

  and

$$P(A_{wins} = B_{wins}) + P(A_{wins} < B_{wins}) + P(A_{wins} > B_{wins}) = 1$$

- Using the above two observations we conclude that

$$P(A_{win} > B_{win}) = P(A_{wins} = B_{wins}) + P(A_{wins} < B_{wins})$$

  So,

$$2 * P(A_{wins} > B_{wins}) = 1$$

  Therefore,

$$P(A_{win} > B_{win}) = \frac{1}{2}$$

Therefore I claim that A having more wins than B denoted by $P(A > B)$ is

$$P(A_{win} > B_{win}) = \frac{1}{2}$$

regardless of the value of n.

2 Two Trading Teams

Let p be the possibility of my team winning against team A and that against team B be q. Since B is a better team than A, $p > q$. To win the whole game we need to win two consecutive games. Now in both scenarios,

1. **B-A-B Sequence**
   There are two ways in which my team can win: either we win the first two games (Say sitB1) (ie we defeat B and then defeat A), or we lose the first one and then win the next two (Say sitB2) (ie be defeated by B once and then win against A and then against B).

$$\text{P (sitB1)= P(win against B) } * \text{ P(win against A)}$$
$$\text{P(sitB1)=}qp$$
$$\text{P (sitB2)= P(lose against B) } * \text{ P(win against A) } * \text{ P(win against B)}$$
$$\text{P(sitB2)= } (1-q)\ p\ q\ = \ pq - pq^2$$
$$\text{P(winning in BAB)= P(sitB1)+ P(sitB2) } = pq + pq - pq^2$$
$$\text{P(winning in BAB)= } 2pq - pq^2$$

2. **A-B-A Sequence**
   There are two ways in which my team can win: either we win the first two games (Say sitA1) (ie we defeat A and then defeat B), or we lose the first one and then win the next two (Say sitA2) (ie be defeated by A once and then win against B and then against A).

$$\text{P (sitA1)= P(win against A) } * \text{ P(win against B)}$$
$$\text{P(sitA1)=}pq$$
$$\text{P (sitA2)= P(lose against B) } * \text{ P(win against A) } * \text{ P(win against B)}$$
$$\text{P(sitA2)= } (1-p)\ q\ p\ = \ pq - p^2q$$
$$\text{P(winning in ABA)= P(sitA1)+ P(sitA2) } = pq + pq - p^2q$$
$$\text{P(winning in ABA)= } 2pq - p^2q$$

It is obvious that $(2pq - pq^2) > (2pq - p^2q)$ since $p > q$. Since P(win in BAB) > P(win in ABA), choosing BAB sequence is the better choice.

3 Random Variables

3.1 Part I

Consider an event A corresponding to $Q_1 < q_1$ and an event B corresponding to $Q_2 < q_2$

Since we know that if A and B are two events, then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$
$$\text{i.e. } P(A \cap B) = P(A) + P(B) - P(A \cup B)$$
$$\text{i.e. } P(A \cap B) \geqslant P(A) + P(B) - 1 \qquad \because P(A) \leqslant 1$$

We can say that

$$P(A \cap B) = P(Q_1 < q_1, Q_2 < q_2) \geqslant 1 - p_1 - p_2$$

Now consider $P(Q_1Q_2 < q_1q_2)$ as:

$$P(Q_1Q_2 < q_1q_2) = P\left(Q_1 \geqslant q_1, Q_2 < \frac{q_1q_2}{Q_1}\right) + P(Q_1 < q_1, Q_2 < q_2) + P\left(Q_1 < \frac{q_1q_2}{Q_2}, Q_2 < q_2\right)$$

We split our given probability into three parts, each representing a mutually exclusive case. Now since $P(A) \geqslant 0$ for any event A, we can say that:

$$P(Q_1Q_2 < q_1q_2) \geqslant P(Q_1Q_2 < q_1q_2)$$

But since $P(Q_1 < q_1, Q_2 < q_2) \geqslant 1 - (p_1 + p_2)$, we have

$$P(Q_1Q_2 < q_1q_2) \geqslant 1 - (p_1 + p_2)$$

3.2 Part II

For any $i(1 \leqslant i \leqslant n)$, LHS is

$$|x_i - \mu| = \sqrt{(x_i - \mu)^2} \leqslant \sqrt{\sum_{i=1}^{n}(x_i - \mu)^2}$$

and expanding the expression for σ, the RHS becomes

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

Thus, $\forall i(0 \leqslant i \leqslant n)$,

$$|x_i - \mu| \leqslant \sigma\sqrt{n-1}$$

How does this inequality compare with Chebyshev's inequality as n increases?
Consider the set

$$S = \{x : |x - \mu| \leqslant \sigma\sqrt{n-1}\}$$

Note that it is the complement of the set,

$$S' = \{x : |x - \mu| > k\sigma\}$$

$k = \sqrt{n-1}$, where my sample space is the set $\{x_1, x_2, \ldots, x_n\}$ Thus $|S| + |S'| = n$
Now using Chebyshev's inequality on S', we have

$$\frac{|S'|}{n} \leqslant \frac{1}{n-1}$$

which means that

$$\frac{|S|}{n} \geqslant 1 - \frac{|S'|}{n}$$

i.e.,

$$\frac{|S|}{n} \geqslant \frac{n-2}{n-1}$$

We can see that as n increases, the Chebyshev's inequality is increasingly easily able to explain the given inequality $|x_i - \mu| \leqslant \sigma\sqrt{n-1}$. To be more explicit, we can see that as $n \to \infty$, $|S'| \to 0$ (from Chebyshev's inequality for S'), i.e., the number of elements in the set not satisfying the given inequality goes to 0, thus asserting that the inequality is true.

4 Staff Assistant

4.1 Part (a)

Let B_i be the event that the i^{th} person is the best candidate. It is clear that

$$B_i = \frac{1}{n} \qquad \forall i \in \{1, 2, \ldots, n\}$$

It means $P(E_i|B_i)$ is the event that the best person is hired given that we know that i^{th} person is the best.
Now suppose that i^{th} person is the best candidate, then we can say that

$$P(E_i|B_i) = 1 - P(i^{th} \text{ person does not get hired}|B_i)$$

which is the same as saying either of $(m+1)^{th}, (m+2)^{th}, \ldots$ or $(i-1)^{th}$ person gets hired.
Suppose the scores of the people on positions $1, 2, ..., m, ..., i-1$ belong to the set $\{s_1, s_2, ...s_m, ..., s_{i-1}\}$, then consider cases:

Case 1 When the person at position $i-1$ gets selected
In this case, the person needs to the the one with score s_{i-1}. The probability of this happening, say, p_1 is:

$$p_1 = \frac{\text{number of permutations of remaining } (i-2) \text{ people in } \{1, 2, ..., i-2\} \text{ places}}{\text{total number of permutations of } (i-1) \text{ people in } (i-1) \text{ places}}$$

i.e.,

$$p_1 = \frac{(i-2)!}{(i-1)!}$$

Case 2 When the person at position $i-2$ gets selected
In this case the person at $i-2$ position can be the one with score s_{i-2} or s_{i-1}.

- When s_{i-2} is present at $i-2$ position, then necessarily s_{i-1} needs to be at $i-1$ position. But this case has been counted already in the previous case. Thus we don't need to account for it.

So the person with score s_{i-1} is fixed at position $i-2$ and let the probability of this case be p_2, then

$$p_2 = \frac{\text{number of permutations of remaining } (i-2) \text{ people in } \{1, 2, ..., i-3, i-1\} \text{ places}}{\text{total number of permutations of } (i-1) \text{ people in } (i-1) \text{ places}}$$

i.e.,

$$p_2 = \frac{(i-2)!}{(i-1)!}$$

$\vdots$
$\vdots$
$\vdots$
and so on.
Similarly for all $i-m-1$ cases till when the person at position $m+1$ gets selected, we have

$$p_j = \frac{(i-2)!}{(i-1)!} \qquad \forall j \in \{m+1, m+2, ..., i-1\}$$

Summing these $p'_j s$, we get

$$P(i^{th} \text{ person does not get hired}|B_i) = (i-m-1).\frac{(i-2)!}{(i-1)!}$$

Thus,

$$\begin{aligned} P(E_i|B_i) &= 1-(i-m-1).\frac{(i-2)!}{(i-1)!} \\ &= \frac{m}{i-1} \end{aligned}$$

Thus,

$$\begin{aligned} P(E_i) &= P(B_i)P(E_i|B_i) \\ &= \frac{m}{n}\frac{1}{i-1} \end{aligned}$$

And since $P(E) = \sum_{i=m+1}^{n} P(E_i)$, we have

$$P(E) = \frac{m}{n}\sum_{i=m+1}^{n}\frac{1}{i-1}$$

4.2 Part (b)

We are given the function $\sum_{j=m+1}^{n}\frac{1}{j-1}$ which we need to bound. We will herein give the explanation of both how to obtain the lower bound and the upper bound

Lower Bound For calculating the lower bound for the function $\sum_{j=m+1}^{n}\frac{1}{j-1}$ I can write it as

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

followed by this I can draw the histogram for the above function and claim that the area under the histogram is always greater than a supporting function $\frac{1}{x-1}$ where the limits of the function go from $m+1$ to $n+1$. An illustration for the same is attached herein in figure 1 in which I chose m=5 and n=20 Followed by this I can claim that

$$\sum_{j=m+1}^{n}\frac{1}{j-1} \geqslant \int_{m+1}^{n+1}\frac{1}{x-1}dx$$

Now I will further calculate the integral term

$$\begin{aligned} &\int_{m+1}^{n+1}\frac{1}{x-1}dx \\ &= [\ln(x-1)]_{m+1}^{n+1} \\ &= ln(n) - ln(m) \end{aligned}$$

So followed by this I can infer that

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

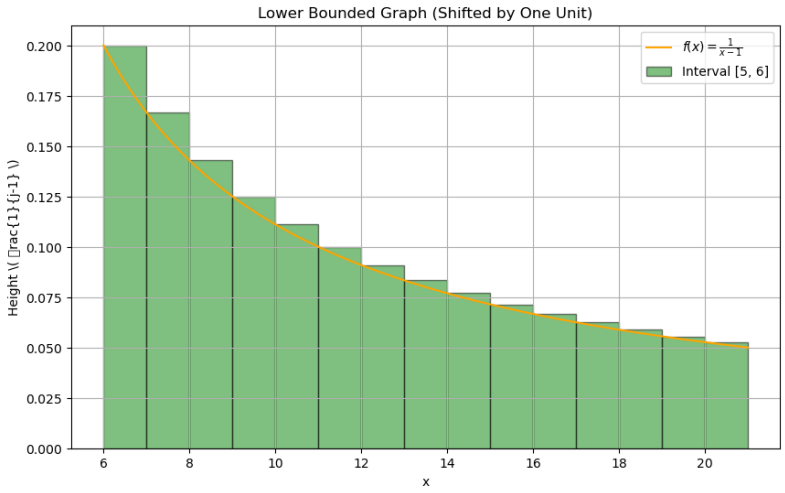

Figure 1: Lower Bound for the histogram

Upper Bound For calculating the upper bound for the function $\sum_{j=m+1}^{n} \frac{1}{j-1}$ I can write it as

$$\sum_{j=m+1}^{n} \frac{(j-1)-(j-2)}{j-1}$$

Followed by this I can draw the histogram for the above function and claim that the area under the histogram is always less than a supporting function $\frac{1}{x-1}$ where the limits of the function go from m to n . An illustration for the same is attached herein in figure 2 in which I chose m=5 and n=20 Followed by this I can claim that

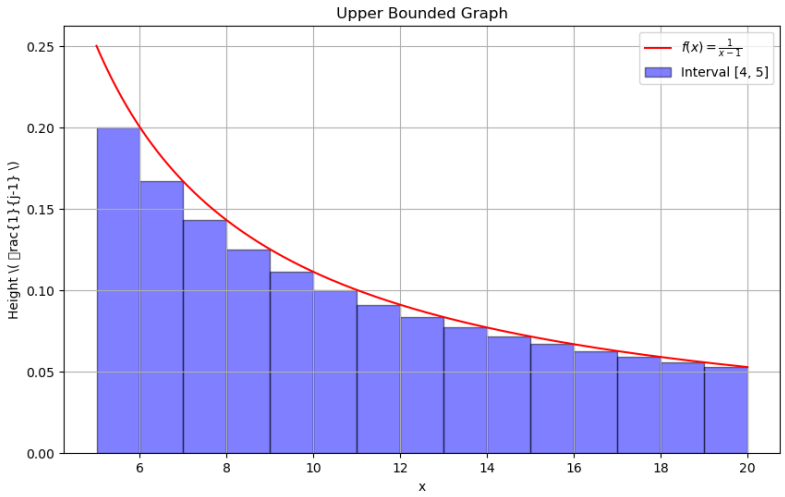

Figure 2: Upper bound for the histogram

$$\sum_{j=m+1}^{n} \frac{1}{j-1} \leqslant \int_{m}^{n} \frac{1}{x-1} dx$$

Now I will further calculate the integral term

$$\int_m^n \frac{1}{x-1}dx$$
$$= [\ln(x-1)]_m^n$$
$$= ln(n-1) - ln(m-1)$$

So followed by this I can infer that

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

So using both the above inferences I say that

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

4.3 Part (c)

To maximise $\frac{m}{n}(ln(n) - ln(m))$ wrt m, we will take it's derivative wrt m and then set that to zero.

$$f(m) = \tfrac{m}{n}(ln(n) - ln(m))$$
$$\tfrac{df}{dm} = \tfrac{1}{n}(ln(n) - ln(m)) \; - \; \tfrac{m}{n}\,\tfrac{1}{m}$$
$$\tfrac{df}{dm} = \tfrac{ln(n)-ln(m)-1}{n}$$
$$ln(n) - ln(m) - 1 = 0$$
$$ln(n) = ln(m) + 1$$
$$ln(n) = ln(me)$$
$$me = n \implies m = \tfrac{n}{e}$$

So the value $\frac{m}{n}(ln(n)ln(m))$ is maximized when $m = \frac{n}{e}$. Now substituting this value of m in the function, we get

$$f(m = \tfrac{n}{e}) = \tfrac{\frac{n}{e}}{n}(ln(n)ln(\tfrac{n}{e}))$$
$$f(m = \tfrac{n}{e}) = \tfrac{\frac{n}{e}}{n}(ln(n)ln(n) + 1)$$
$$f(m = \tfrac{n}{e}) = \tfrac{1}{e}$$

Since it is known that $P(E) \geqslant f(m)$, we can also say $P(E) \geqslant f(m = \frac{n}{e}) = \frac{1}{e}$

5 Free Trade

In an infinitely long line of traders I need to choose a position such that all the traders before me in that line have unique ids and my id matches one of them. Lets say that I choose the k^{th} position. Note that from $k \leqslant 201$ otherwise according to the Pigeon Hole Principle, there must be two traders with the same id.

P (Free Trade) $\implies$ (One of the $k-1$ traders has the same id as the k^{th} trader) AND (Rest of the $k-2$ traders have unique ids)

$$\text{P(Free Trade)}=(\tbinom{k-1}{1}\ \tfrac{1}{200})(\tfrac{199}{200}\ \tfrac{198}{200}\cdots\tfrac{200-(k-2)}{200})$$

$$\text{P (Free Trade at } k) = \frac{k-1}{200} \ \frac{199 \times 198 \ldots \times (200-(k-2))}{(200)^{k-2}}$$

$$\text{P(Free Trade at } k) = \frac{k-1}{200} \ \frac{(200)!}{(200-(k-1))! \times (200)^{k-1}}$$

$$\text{P(Free Trade at } k) = \frac{(k-1) \times (200)!}{(200-(k-1))! \times (200)^{k}}$$

The best position would be k such that P(Free Trade at k) $\geqslant$ P(Free trade at $(k-1)$) and P(Free Trade at k) $\geqslant$ P(Free trade at $(k+1)$). That means that,

$$\frac{(k-1) \times (200)!}{(200-(k-1))! \times (200)^{k}} \geqslant \frac{(k) \times (200)!}{(200-(k))! \times (200)^{k+1}} \text{ and } \frac{(k-1) \times (200)!}{(200-(k-1))! \times (200)^{k}} \geqslant \frac{(k-2) \times (200)!}{(200-(k-2))! \times (200)^{k-1}}$$

$$\implies k^2 - k - 200 \geqslant 0 \text{ and } k^2 - 3k - 199 \leqslant 0$$

$$\implies k \epsilon [-\infty, -13.65] \cup [14.65, \infty] \text{ and } k \epsilon [-12.68, 15.68]$$

This leaves us with only one sensible value of k which is $k = 15$. So to maximize our chances of getting a free trade, we should choose to be at the 15^{th} position.

6 Update Functions

6.1 Updating the Mean, Median and Mode

Suppose we are given an array A containing n elements and sorted in an ascending order, i.e., $A = \{x_1, x_2, \ldots, x_n\}$ and say we add a new value x_{n+1} to the data.

- **Mean**
  Let the initial value of mean $= \mu$. This means that,

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

$$\text{or } \sum_{i=1}^{n} x_i = n\mu$$

  So after adding x_0, we have

$$\sum_{i=1}^{n+1} x_i = n\mu + x_{n+1}$$

  And since new mean, $\mu' = \frac{\sum_{i=1}^{n+1} x_i}{n+1}$,

$$\mu' = \frac{n\mu + x_{n+1}}{n+1}$$

```
def UpdateMean(OldMean, NewDataValue, n, A):
    newMean = OldMean*(n/(n+1)) + NewDataValue*(1 / (n+1))
    return newMean
```

Figure 3: Python implementation of `UpdateMean` function

- **Median**
  This is the python implementation of the `UpdatMedian` function

```python
def UpdateMedian(OldMedian, NewDataValue, n, A):
    #assuming the given array is in ascending order
    if n%2 == 0:
        if NewDataValue <= A[int(n/2)]:
            return A[int(n/2)]
        elif NewDataValue <=A[int(n/2) + 1]:
            return NewDataValue
        else:
            return A[int(n/2) + 1]
    else:
        if NewDataValue <= A[int((n-1)/2)]:
            return (A[int((n-1)/2)] + A[int((n+1)/2)])/2
        elif NewDataValue <= A[int((n+3)/2)]:
            return (A[int((n+1)/2)] + NewDataValue)/2
        else:
            return (A[int((n+1)/2)] + A[int((n+3)/2)])/2
```

Figure 4: Python implementation of `UpdateMedian` function

- **Standard Deviation**
  Since we know,

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

$$\text{or } \sigma = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 + n\mu^2 - 2\mu\sum_{i=1}^{n} x_i}{n-1}}$$

$$\text{or } \sigma = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 + n\mu^2 - 2n\mu^2}{n-1}}$$

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

where μ is the old mean value. This means

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

Thus,

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

Hence, the new Standard deviation, σ_1 can be written as:

$$\sigma_1 = \sqrt{\frac{(n-1)\sigma^2 + n\mu^2 + x_{n+1}^2}{n} - \frac{n+1}{n}\mu_1^2}$$

where $mu_1 = \frac{n\mu + x_{n+1}}{n+1}$ is the updated mean.

```python
def UpdateStd(OldMean, OldStd, NewMean, NewDataValue, n, A):
    newStd = ((OldStd**2)*((n-1)/(n)) + OldMean**2 + (NewDataValue**2)/n - ((NewMean**2)*(n+1)/n))**(1/2)
    return newStd
```

Figure 5: Python implementation of `UpdateStd` function

6.2 Updating The Histogram of A

Assuming that we do not change the size of the bins on the addition of the new value, there can be two scenarios: one where the new value can be put in one of the existing bins and the second where it is an outlier for which separate bin needs to be created. If it is the first case, we can simply increase the frequency of the bin that the new value goes into by 1 and then plot the histogram again. In the second case, depending on how far off the new value is from the older distribution we may need to add multiple bins. Suppose we were studying the distribution of heights students from a class. In this distribution we had bins from 100cm-110cm, 110cm-120cm, 120cm-130cm, but now three new students of heights 115cm, 134cm and 152 cm may join the class. The latter two students' heights are now **outliers** from the old data distribution. For the first student (who can be sorted into the 110cm-120cm bin), we can simply add 1 to the frequency of his/her bin to update the histogram. For the second student we will have to create a new bin and set it's frequency to 1. For the third student we will have to add 3 new bins 130cm-140cm, 140cm-150cm and 150cm-160cm and set only the lattermost's frequency to 1 and keep the frequency of the other two 0 to update the histogram.

7 Plots

7.1 Violin Plot

Violin plot depict distribution of data using density curves. They are very useful when we wish to compare the distribution of multiple groups. To decide the boundary of the plot, violin plots use Kernel Density Estimate which estimates the probability distribution functions of random variables based on our observations.

The wider the violin plot is at specific value, the higher is the occurrence of that value in our dataset. To indicate the distribution of the data, there is a central white dot indicating the median of the data, and a box plot showing the first and the third quartiles' end.

They are used in model evaluation to compare predicted and actual values, identify bias and variance in the data. It can also be used to detect any outliers. They are also used to compare model performance with different hyper parameter setups when working with multiple machine learning models.

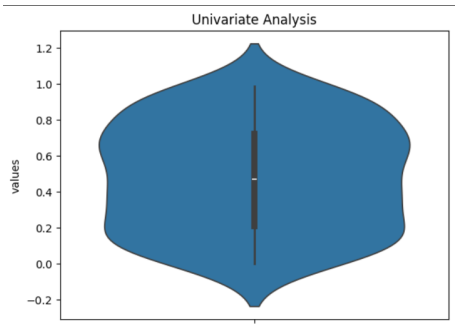

Figure 6: Example of a Violin Plot

7.2 Pareto Chart

The Pareto Chart is a bar graph containing a line graph. It represents individual values in **descending order** using bars. Meanwhile, a line represents the cumulative total of the individual values in percentage form. The chart was inspired by Italian economist Vilfredo Pareto and his 80/20 principle, which states that 80% of effects arise from 20% of causes.

The use of Pareto charts has a lot of advantages. due to their presentation of data in a descending order, our attention will be drawn to the more important or significant factors. Their minimal representation of data makes it easy to draw conclusions from them. Their visually engaging representation also simplifies the process of pattern recognition and relationship management.

We find a real life use of Pareto plots in healthcare where they are used to find patterns in patients administration and other data. For example, hospitals and pharmacies could use the data for the ailments with which patients come to them to maintain a sufficient stock of medicines and other supplies that might be needed. In retail industry, businesses could organise the investments in a way that they put more money into the channels which give them the maximum returns.

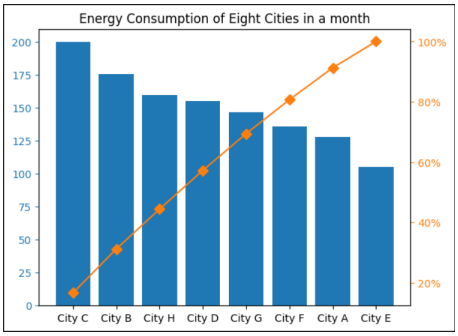

Figure 7: Example of a Pareto Chart

7.3 Coxcomb Plots

Interesting history: They were first used by Florence Nightingale to assess the number of soldiers who died every month and whether their death was caused by the wounds in the battlefield or diseases due to bad hygiene. The book which contained the charts was called coxcomb and the plots just inherited this name.

They are modified pie charts in which each slice represents a parameter or an interval. The main difference between the two is that pie charts encode values using the angle subtended by the subsection but in coxcomb plots, each subsection subtends the same angle at the centre but the values are encoded by the fraction of area of the subsection covered by the plotted values.

They are used obviously in health department data to maintain a database of the causes of death in the population to keep a tab on an epidemic or any other environmental or other factors. They are also used by law enforcement to recognize analyze patterns in crimes across locations or time to ensure more optimal allocation of resources to prevent any untoward incident.

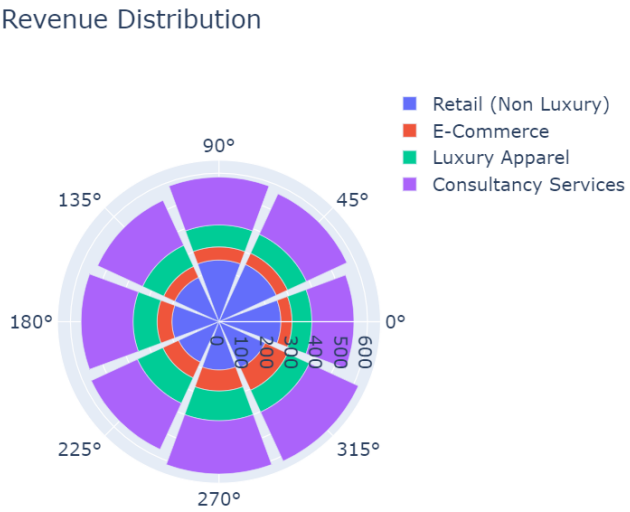

Figure 8: Example of a Coxcomb Plot

7.4 Waterfall Plots

Their name comes from the cascade or mountain like structures formed because of multiple curves being staggered across and vertically on the screen which causes the curves nearer to the viewer to mask those behind giving it the look of a waterfall.

They are used to visualise the variation of 2 dimensional idea with time. They are often used to depict spectograms and in acoustic analysis.

Waterfall Plots can also be used in Marketing Strategy analysis where we can compare the reach and impact of different campaigns and illustrate the customer acquisition costs through different channels.They are also used in clinical trials to track patient progress and efficacy of different treatments and analyse the efficacy of each treatment and the response to intervention showing how

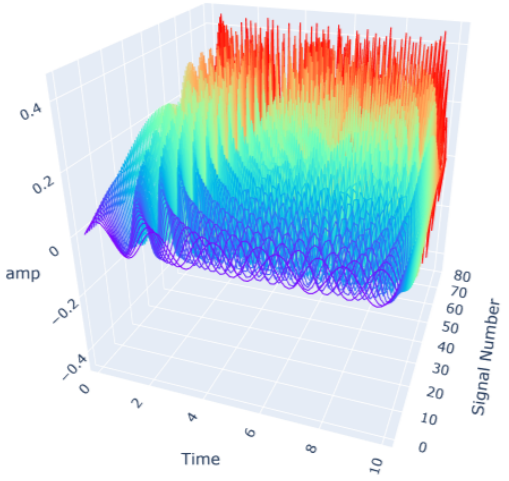

Figure 9: Example of a Waterfall Plot

8 Monalisa

The solution for this question is present in the file `q8.ipynb`. We will briefly explain about the question and it's workings.

- After downloading the image from the provided link we convert the image into gray-scale image and read the image using matplotlib.
- We then using a python code shift the image along the X direction by tx pixels where tx is an integer ranging from -10 to +10. For each shift, we had computed the correlation coefficient between the original image and its shifted version. Figure 10 is the plot for the same. The

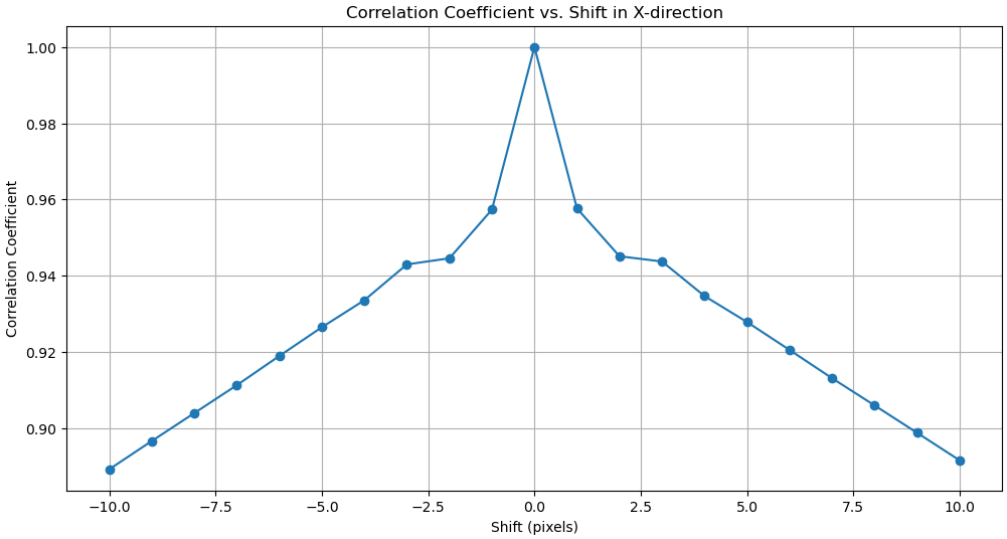

Figure 10: Graph for co-relation coefficient

Correlation coefficient was calculated using the shifting function provided in Figure 11

```python
def shift_image(image, tx):
    shifted_img = np.zeros_like(image)
    if tx > 0:
        shifted_img[:, tx:] = image[:, :-tx]
    elif tx < 0:
        shifted_img[:, :tx] = image[:, -tx:]
    else:
        shifted_img = image
    return shifted_img
```

Figure 11: Shifting function for images

- In the second part of the question, we focused on generating a normalized histogram for the original image. This process began with the creation of a NumPy array containing 256 elements, where each element represented a specific gray-scale value of the image, ranging from 0 (black) to 255 (white).

To simplify the analysis, we grouped these gray-scale values into 16 bins, each covering an interval of 16 gray-scale levels. For example, the first bin captured gray-scale values from 0 to 15, the second from 16 to 31, and so on, up to the final bin, which covered the range from 240 to 255.

Using the frequency data from these bins, we constructed a normalized histogram, which provided a visual representation of the distribution of gray-scale values in the image. The resulting histogram, illustrated in Figure 12, offers valuable insights into the tonal characteristics of the image, highlighting the concentration of pixels across different gray-scale ranges.

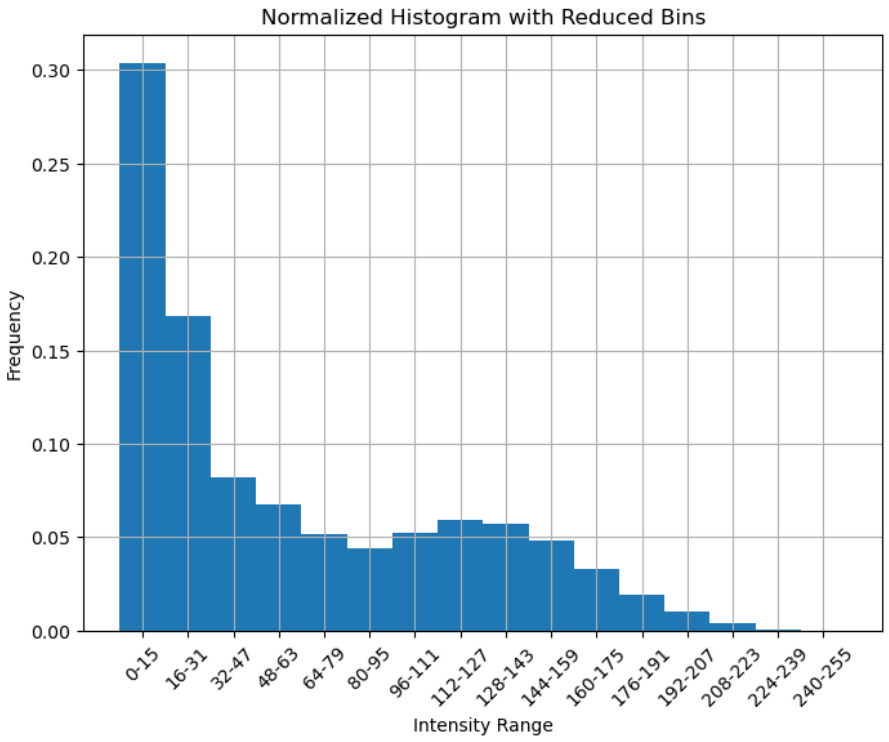

Figure 12: Histogram for gray-scaled Mona Lisa image