

CS215 Assignment 1 Solutions

Satyam Sinoliya, 23B0958
Vaibhav Singh, 23B1068
Shaik Awez Mehtab, 23B1080

August 2024

Question 1. There are two friends playing a dice-roll game. Friend A has $(n + 1)$ fair dice and Friend B has n fair dice (a fair die has equal probability of every face). On every roll, a win is achieved if we get a prime number on the top. What is the probability that A will have more wins than B if both roll all of their dice?

[5 marks]

Solution 1. So A or B wins when they get either 2, 3 or 5, whose probability is $\frac{1}{2}$. So the probability that A wins i number of times when all of the $(n + 1)$ dice are rolled is

$$p(A_i) = \binom{n+1}{i} \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{n+1-i} \quad (1)$$

$$= \binom{n+1}{i} \left(\frac{1}{2}\right)^{n+1} \quad (2)$$

Similarly, probability that B gets i wins when all of the n are rolled is

$$p(B_i) = \binom{n}{i} \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{n-i} \quad (3)$$

$$= \binom{n}{i} \left(\frac{1}{2}\right)^n \quad (4)$$

Now the probability that A has more wins than B is

$$p(i > j) = \sum_{i=0}^{n+1} \sum_{\substack{j=0 \\ i>j}}^n p(A_i)p(B_j) \quad (5)$$

Since number of wins of A is independent from that of B . By writing each possible term $\binom{n+1}{i}$ and $\binom{n}{j}$ in a grid manner. We observe that our required terms cover exactly half of the grid. Thus equation (5) simplifies to

$$p(i > j) = \frac{\sum_{i=0}^{n+1} \binom{n+1}{i} \left(\frac{1}{2}\right)^{n+1} \cdot \sum_{j=0}^n \binom{n}{j} \left(\frac{1}{2}\right)^n}{2} = \frac{1}{2} \quad (6)$$

Thus, the probability of A winning more times than B is $\frac{1}{2}$.

Question 2. You are playing a trading game against two teams A and B (will happen in reality soon). The game is played in the form of a three-set series with A and B alternately. Also, Team B is better at trading than Team A . To encourage your trading career, the exchange (an organization responsible for managing the trades) gives you two options A - B - A (which means you play a game with Team A , then Team B and at last Team A again) or B - A - B . You will win if you win two sets in a row. Which of the two options should you choose? Justify your choice with proper calculations.

[5 marks]

Solution 2. Let p_A and p_B be the probability of winning the set against A and B respectively. Since, Team B is better than team A ,

$$p_A > p_B$$

For the first option, i.e. A - B - A , to win, we either have to win first and second set and lose the third set or win the second and third set and lose the first set or we can win all of them. Probability of this happening is, (let it be P_1)

$$= p_A \times p_B \times (1 - p_A) + (1 - p_A) \times p_B \times p_A + p_A \times p_B \times p_A$$

$$= p_A p_B - p_A^2 p_B + p_A p_B - p_A^2 p_B + p_A^2 p_B$$

$$= 2p_A p_B - p_A^2 p_B$$

Similar for second option, which is B - A - B , the probability for winning will be (let it be P_2)

$$= p_B \times p_A \times (1 - p_B) + (1 - p_B) \times p_A \times p_B + p_B \times p_A \times p_B$$

$$\begin{aligned}
&= p_{APB} - p_A p_B^2 + p_{APB} - p_A p_B^2 + p_A p_B^2 \\
&= 2p_{APB} - p_A p_B^2
\end{aligned}$$

We know that,

$$p_A > p_B$$

As, both p_A and p_B is positive, we can multiple them to both the sides without changing the inequality sign,

$$\begin{aligned}
p_A \times p_{APB} &> p_B \times p_{APB} \\
p_A^2 p_B &> p_A p_B^2
\end{aligned}$$

Multiplying -1 on both the sides,

$$-p_A^2 p_B < -p_A p_B^2$$

Adding $2p_{APB}$ on both the sides,

$$2p_{APB} - p_A^2 p_B < 2p_{APB} - p_A p_B^2$$

Which is basically,

$$P_1 < P_2$$

So, playing in the **second** order will give us better chances to win the game.

Question 3. This question has two parts:

- **3.1** 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 $P(Q_1 Q_2 < q_1 q_2) \geq 1 - (p_1 + p_2)$ [3 marks]
- **3.2** 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? (give an informal answer)

[3+2 marks]

Solution 3. 3.1

Define two events, E_1 and E_2 :

1. $E_1 = \{Q_1 < q_1\}$
2. $E_2 = \{Q_2 < q_2\}$

So,

$$\begin{aligned}
P(E_1) &\geq 1 - p_1 \\
P(E_2) &\geq 1 - p_2
\end{aligned}$$

We need to prove that,

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

Let's define another event E_3 , where $E_3 = \{Q_1 Q_2 < q_1 q_2\}$

If we consider the complement of E_3 , which is

$$\begin{aligned}
\{Q_1 Q_2 < q_1 q_2\}^c &= \{Q_1 Q_2 \geq q_1 q_2\} \\
E_3^c &= \{Q_1 Q_2 \geq q_1 q_2\}
\end{aligned}$$

If $Q_1 Q_2 \geq q_1 q_2$, and Q_1, Q_2 are non-negative integers, it is very clear that, atleast one of the following has to be true:

$$\begin{aligned}
&Q_1 \geq q_1 \text{ or } Q_2 \geq q_2 \\
\implies E_3^c &\subseteq \{Q_1 \geq q_1\} \cup \{Q_2 \geq q_2\}
\end{aligned}$$

$$\begin{aligned}\implies P(E_3^c) &\leq P(\{Q_1 \geq q_1\} \cup \{Q_2 \geq q_2\}) \\ &\leq P(\{Q_1 \geq q_1\}) + P(\{Q_2 \geq q_2\})\end{aligned}$$

It is clear that:

$$\{Q_1 \geq q_1\} = E_1^c \text{ and } \{Q_2 \geq q_2\} = E_2^c$$

So,

$$\begin{aligned}1 - P(E_3) &\leq P(E_1^c) + P(E_2^c) \\ &\leq 1 - P(E_1) + 1 - P(E_2) \\ &\leq 1 - (1 - p_1) + 1 - (1 - p_2) \\ &\leq p_1 + p_2 \\ \implies P(E_3) &\geq 1 - (p_1 + p_2)\end{aligned}$$

Solution 3.2 We know that,

$$\begin{aligned}\frac{\sum_{i=0}^n (x_i - \mu)^2}{n - 1} &= \sigma^2 \\ \implies \sum_{i=0}^n (x_i - \mu)^2 &= \sigma^2(n - 1)\end{aligned}$$

For any i ,

$$(x_i - \mu)^2 \geq 0$$

So, for each i ,

$$(x_i - \mu)^2 \leq \sigma^2 \times (n - 1)$$

Again, as both $(x_i - \mu)^2$ and $\sigma^2 \times (n - 1)$ are greater than or equal to zero, we can take square root on both sides.

$$\sqrt[2]{(x_i - \mu)^2} \leq \sqrt[2]{\sigma^2(n - 1)} \implies |x_i - \mu| \leq \sigma\sqrt{n - 1}$$

Chebyshev's inequality states that for any distribution, the proportion of values that lie more than k standard deviations away from the mean is bounded by:

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

As n increases, the deterministic bound $|x_i - \mu| \leq \sigma\sqrt{n - 1}$ can potentially become less tight as $\sqrt{n - 1}$, but it remains a stronger bound for small datasets compared to Chebyshev's inequality. As Chebyshev's inequality, on the other hand, remains a probabilistic statement that applies to a portion of the distribution and becomes less restrictive for larger values of k .

Question 4. You need a new staff assistant, and you have n people to interview. You want to hire the best candidate for the position. When you interview a candidate, you can give them a score, with the highest score being the best and no ties being possible.

You interview the candidates one by one. Because of your company's hiring practices, after you interview the k th candidate, you either offer the candidate the job before the next interview or you forever lose the chance to hire that candidate. We suppose the candidates are interviewed in a random order, chosen uniformly at random from all $n!$ possible orderings.

We consider the following strategy. First, interview m candidates but reject them all: these candidates give you an idea of how strong the field is. After the m th candidate, hire the first candidate you interview who is better than all of the previous candidates you have interviewed.

1. Let E be the event that we hire the best assistant, and let E_i be the event that i th candidate is the best

and we hire him. Determine $Pr(E_i)$, and show that

$$Pr(E) = \frac{m}{n} \sum_{j=m+1}^n \frac{1}{j-1} \quad (7)$$

[4 marks]

2. Bound $\sum_{j=m+1}^n \frac{1}{j-1}$ to obtain:

$$\frac{m}{n}(\ln n - \ln m) \leq Pr(E) \leq \frac{m}{n}(\ln(n-1) - \ln(m-1)) \quad (8)$$

[3 marks]

3. Show that $\frac{m}{n}(\ln(n) - \ln(m))$ is maximized when $m = \frac{n}{e}$, and explain why this means $Pr(E) \geq \frac{1}{e}$ for this choice of m . [3 marks]

Solution 4.

Question 5. Imagine an infinitely long line of traders waiting outside a brokerage firm to place their trades. Each trader is assigned an ID number from 1 to 200 (both inclusive, obviously these IDs are not unique). The firm's director announces a special offer: the first trader in the queue whose ID number matches the ID of any trader who has already placed a trade will receive a free trade (i.e., a trade without any margins). You have the option to choose your position in this queue. However, you don't know the ID numbers of the traders ahead of you or behind you. Your goal is to maximize your chances of being the first trader whose ID matches someone who has already placed a trade. Given this situation, what position in the queue should you choose to maximize your chances of receiving the free trade? [6 marks]

Solution 5. We have infinite number of trader where each trader is assigned an ID number from 1 to 200 waiting in a queue. We need to find the position which maximises the chance of getting a free trade, where free trade is awarded to a person whose ID number matches with someone ahead in the queue, that is, person who has traded. Since there are 200 unique IDs, the last position one can win is 201 (by pigeon hole principle). 1st person can never win as it has no one ahead of him. Therefore for i^{th} person $i \in \{2, 3, 4, \dots, 201\}$ to win, the probability of winning a free trade is given by

$$P(i) = \frac{200}{200} \times \frac{199}{200} \times \dots \times \frac{202-i}{200} \times \frac{i-1}{200} \quad (9)$$

$$= \prod_{j=2}^i \left(\frac{202-j}{200} \right) \times \frac{i-1}{200} \quad (10)$$

For maximising this probability I have used python and I have attached a file for the code. The position which maximises the probability of winning a free trade is 15 with a probability of 0.0439.

Question 6. 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:

```
function newMean = UpdateMean(OldMean, NewDataValue, n, A),
function newMedian = UpdateMedian(OldMedian, NewDataValue, n, A),
function newStd = UpdateStd(OldMean, OldStd, NewMean, 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. [10 marks]

Solution 6.

Question 7. Read about the following plots:

1. Violin Plot
2. Pareto Chart
3. Coxcomb Chart
4. Waterfall Plot

Describe the uses of these plots. Take some sample data and generate one example plot for each of them.

[8 marks]

Solution 7. Here are the descriptions and usages of the given plots along with their examples:

1. **Violin plot:** It's a hybrid of box plot and kernel density plot. A box plot represents data in a linear fashion. It's made of a straight line from lowest value to highest value along with a box from first to third quartiles, marking all the quartiles of the dataset. Here's an example in figure 1:



Figure 1: Box plot

A kernel density plot represents the density/frequency of data points. It's similar to a histogram, but smooth. In a violin graph, this is kept vertical, with two mirror images of it reflected along y-axis. This is useful in the sense that we can look into both central tendencies of the data (like mean, median etc.) but also how the data is distributed. Both at once. We can visualise the following data which I've taking from (*) containing the average number of hours a person studies given the number of courses taken

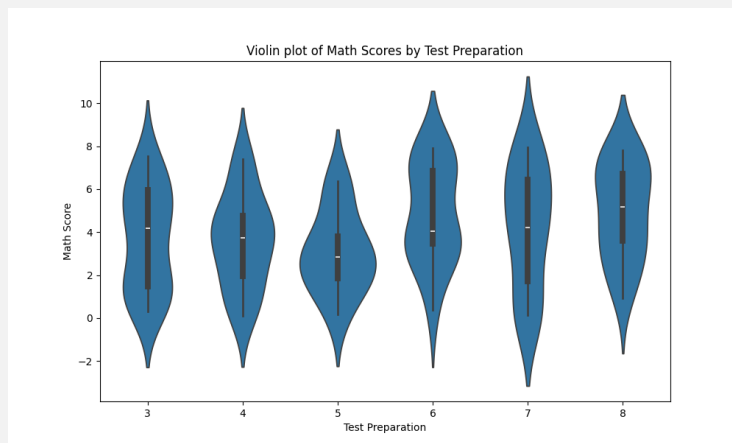


Figure 2: Violin Plot

As mentioned before, a violin plot can show both statistical summary along with distribution, which a normal plot can't. Here in figure 2, the gray line represents the box plot component of it. And the plot you get by rotating it by 90° is the distribution plot.

2. **Pareto Chart:** A pareto chart consists of both a bars and a line graph in the same plot. The bar graph represents the individual data like frequency, cost, height, weight of individual values. These values are arranged in decreasing order of the height of corresponding graphs. The line

graph represents the cumulative data like frequency, cost, etc. till that point. Since the values decreases, this line graph is concave.

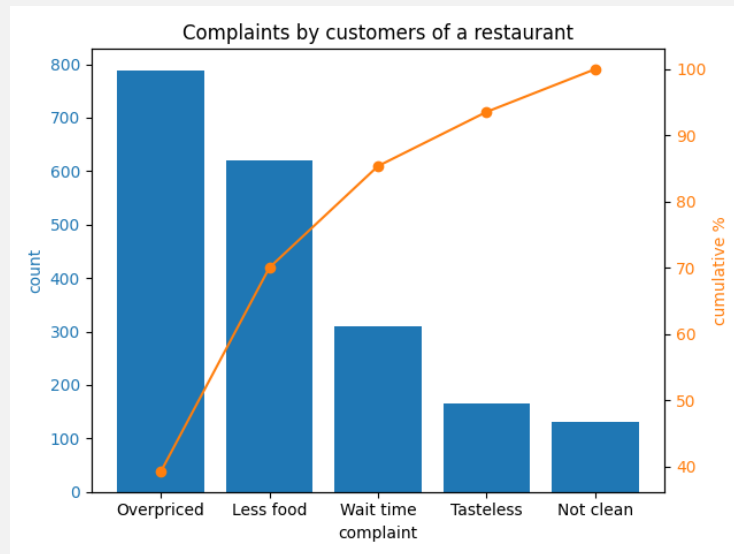


Figure 3: Pareto Plot

An example of a pareto plot is shown where I've collected data of complaints from customers about a given restaurant. It's useful in this case because it is able to show the proportion of the problem particular category is causing as well as the magnitude of the problem.

3. **Coxcomb Chart:** A coxcomb chart is similar to a pie chart. It's also known as a polar chart. But unlike a pie chart, here the area of each sector represents the proportion of the problem caused by a particular category of data unlike a pie chart, where the angle represents the proportion.

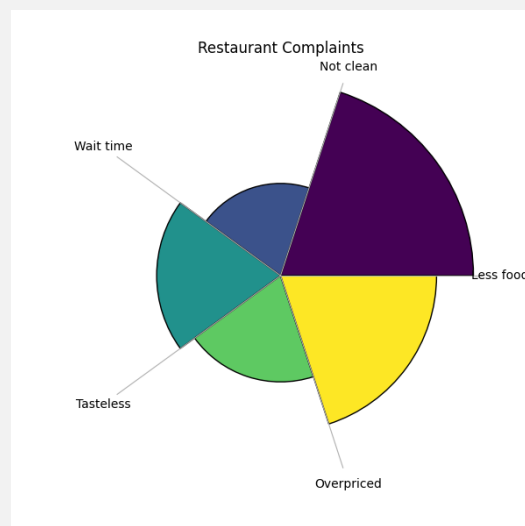


Figure 4: Coxcomb Plot

Here in figure 4 I've shown the same data shown before in figure 3 in a coxcomb chart. Like a pie chart, a coxcomb chart isn't useful when it comes to to large number of categories of data. It's useful when we have a small number of data which is categorical.

4. **Waterfall plot:**

Question 8. Download the image of Monalisa from here. Read the image using matplotlib (example). Write a piece of python code to shift the image along the X direction by tx pixels where tx is an integer ranging from -10 to $+10$ (so, in total you need to do this for 20 values). While doing so, assign a value of 0 to unoccupied pixels. For each shift, compute the correlation coefficient between the original image and its shifted version. Make a plot of correlation coefficients across the shift values. Also, generate a normalized histogram for the original image. You might need to refer to section 3.3 from this book. You are not allowed to use any inbuilt function for generating the histogram. If you are using any other libraries, then please mention about them in the pdf.

[8 marks]

Solution 8. The image of Mona Lisa was read using the matplotlib library. The image was then shifted horizontally by tx pixels for each value of tx in the range of -10 to $+10$. The shifting operation was implemented manually, ensuring that unoccupied pixels were assigned a value of 0.

A custom function `shifting` was created to handle the shifting process:

- If $tx > 0$, pixels were shifted rightwards by tx units.
- If $tx < 0$, pixels were shifted leftwards by tx units.
- If $tx = 0$, the function returned the original image.

All the shifted images are displayed in the Jupyter notebook(q8.ipynb). Some of them are displayed below.



Figure 5: Image with $tx=10$



Figure 6: Image with $tx=-5$

For each shifted image, the correlation coefficient between the original and shifted image was calculated. This coefficient quantifies the linear relationship between the two images, with values ranging from -1 to 1 , where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. This done using a function `corrcoef` in numpy library. This function require 2 1-D arrays to determine the correlation coefficient. The flattening of arrays was done using `.flatten()`. Then correlation for each shift is plotted on a graph shown below.

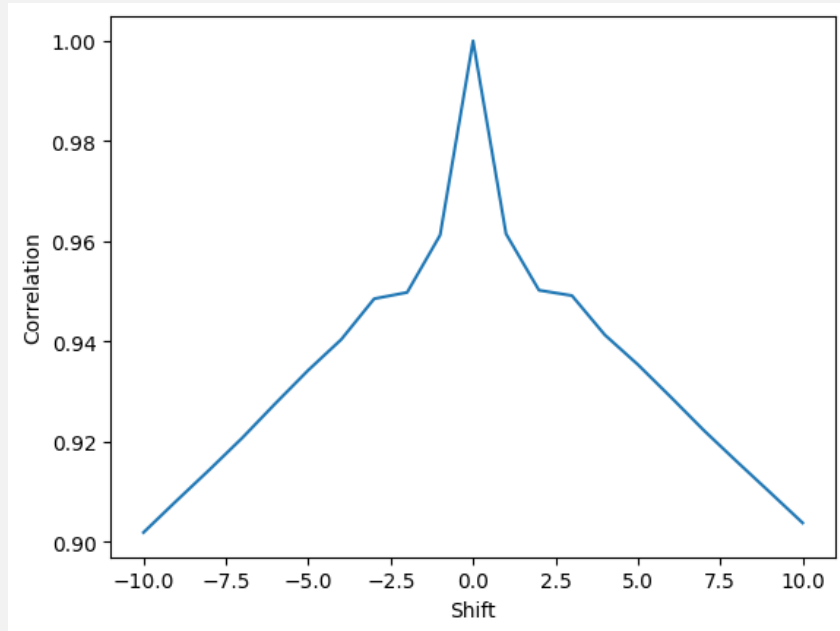


Figure 7: Correlation Coefficient vs Shift Values

As observed in Figure 7, the correlation decreases as the shift increases in either direction. This is expected as the more the image is shifted, the less it resembles the original, resulting in lower correlation values.

Following this, normalized histogram is plotted for each channel(R,G,B). This is done without use of any library, by creating a function `draw_hist` which takes the image of a single channel, transverses through each pixel taking the note of its value and finally each value is divided by the total number of pixels($\text{width} \times \text{height}$) to normalize the histogram.

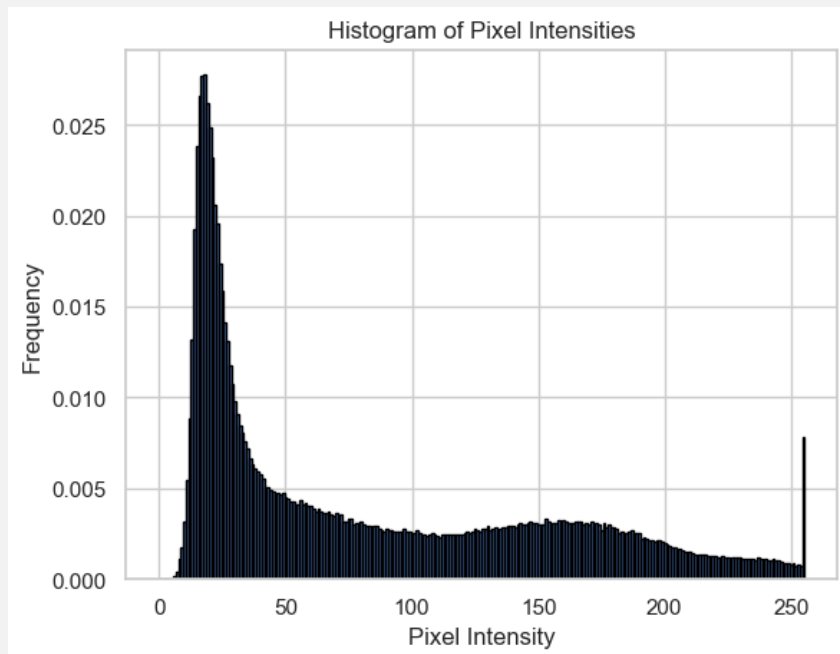


Figure 8: Normalized Histogram of red channel of the Original Image

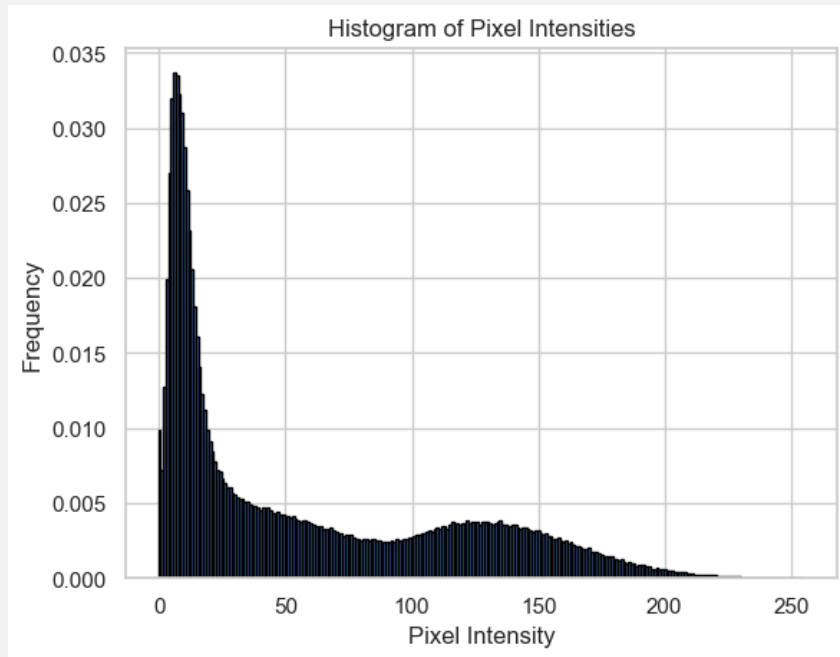


Figure 9: Normalized Histogram of green channel of the Original Image

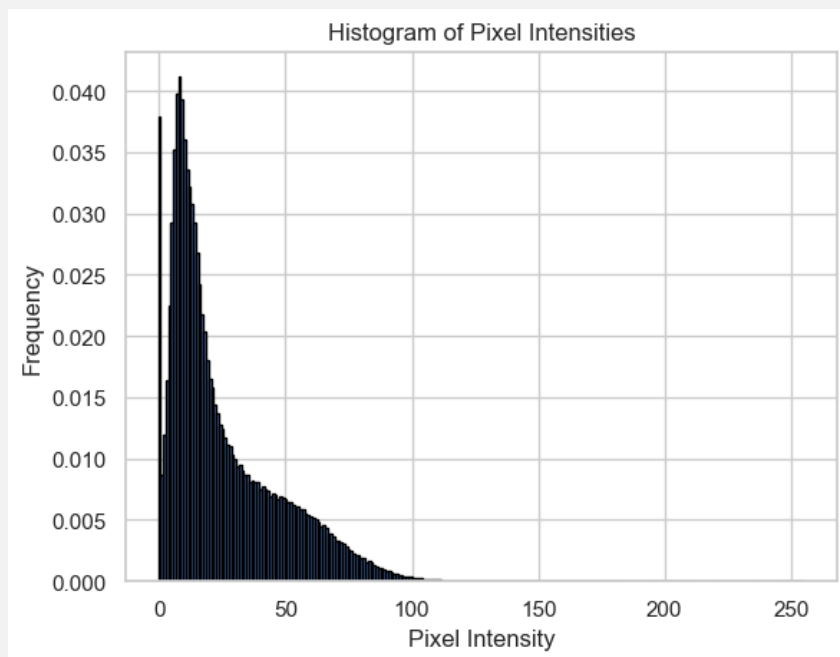


Figure 10: Normalized Histogram of blue channel of the Original Image