CS215 - Assignment 1

Ramachandran S - 23b1052
Keyaan KR - 23b0977
Harith S - 23b1085

August 24, 2024

Contents

1 Let's Gamble

After you roll a dice, Probability(getting a prime) = 1/2 = Probability(not getting a prime)
So, after every roll, a player either wins or loses with equal probability. Since there are 2 equi-probable outcomes, the total number of possible outcomes is 2^{2n+1}.

Let's assume A had won k times. Then for A to have more wins than B, B should have won $< k$ times.
Number of ways in which A can win k times is

$$\binom{n+1}{k}$$

Number of ways in which B can win <k times is

$$\sum_{j=0}^{k-1}\binom{n}{j}$$

Total probability of A winning more number of times than B is

$$P_W = \frac{\sum_{k=1}^{n+1}\binom{n+1}{k}\left(\sum_{j=0}^{k-1}\binom{n}{j}\right)}{2^{2n+1}}$$

We know

$$\binom{n}{r} = \binom{n}{n-r}$$

Using it, we can choose $\binom{n+1}{r}$ and $\binom{n+1}{(n+1)-r}$ terms from the summation.

$$\begin{aligned}
&\binom{n+1}{r}\left(\sum_{k=0}^{r-1}\binom{n}{k}\right) + \binom{n+1}{n+1-r}\left(\sum_{k=0}^{n-r}\binom{n}{k}\right)\\
&= \binom{n+1}{r}\left(\sum_{k=0}^{r-1}\binom{n}{k}\right) + \binom{n+1}{r}\left(\sum_{k=r}^{n}\binom{n}{k}\right)\\
&= \binom{n+1}{r}\left(\sum_{k=0}^{n}\binom{n}{k}\right)\\
&= \binom{n+1}{r}\cdot 2^n
\end{aligned}$$

Case 1: If n is even, we will have

$$\begin{aligned}
&\binom{n+1}{0}\cdot 2^n, \binom{n+1}{1}\cdot 2^n, \ldots, \binom{n+1}{\frac{n}{2}}\cdot 2^n\\
&\sum_{k=0}^{\frac{n+1}{2}}\binom{n+1}{k} = \frac{1}{2}\left(\sum_{k=0}^{n+1}\binom{n+1}{k}\right)\\
&= \frac{1}{2}\cdot 2^{n+1}\\
&= 2^n\\
&P_w = (2^n\cdot 2^n)\cdot\frac{1}{2^{2n+1}} = \frac{1}{2}
\end{aligned}$$

Case 2: If n is odd, we will have

$$\binom{n+1}{0}\cdot 2^n, \binom{n+1}{1}\cdot 2^n, \ldots, \binom{n+1}{\frac{n-1}{2}}\cdot 2^n$$

Here,

$\binom{n+1}{\frac{n+1}{2}}\left(\sum_{k=0}^{\frac{n-1}{2}}\binom{n}{k}\right)$ won't have a complementary term. But $\sum_{k=0}^{\frac{n-1}{2}}\binom{n}{k}$ is self complementary and

$$\sum_{k=0}^{\frac{n-1}{2}}\binom{n}{k} = \frac{1}{2}\cdot 2^n$$

So, we will have:

$$\begin{aligned}\binom{n+1}{0}+\binom{n+1}{1}+\cdots+\frac{1}{2}\cdot\binom{n+1}{\frac{n+1}{2}} &= \frac{1}{2}\left(\sum_{k=0}^{n+1}\binom{n+1}{k}\right)\\ &= \frac{1}{2}\cdot 2^{n+1}\\ &= 2^n\end{aligned}$$

$$P_w = (2^n\cdot 2^n)\cdot\frac{1}{2^{2n+1}} = \frac{1}{2}$$

Irrespective of n being even or odd, Probability of A having more wins than B is 1/2. Hence, the final answer is **1/2**.

2 Two Trading Teams

Let P_A be the probability of winning against A and P_B be the probability of winning against B.
Clearly, from the question,

$$P_A > P_B$$

Since you are playing thrice and you must win 2 times in a row, you must win against player number 2. To win, you must win against **atleast one** from player number 1 and player number 3.

Probability(winning against atleast one) = 1 - Probability(losing against both)

And we know winning/losing with the player number 2 has nothing to do with the performance against the other players. Therefore, the two events are independent. In that case,

Probability(winning) = Probability(winning with player 2)*Probability(winning against atleast one from player 1 and player 3)

Let me denote the probability of winning as P_W

- **Case-1: A-B-A**:

$$\begin{aligned}P_W &= P_B\cdot(1-(1-P_A)\cdot(1-P_A))\\ P_W &= P_B\cdot P_A\cdot(2-P_A)\end{aligned}$$

- **Case-2: B-A-B**:

$$\begin{aligned}P_W &= P_A\cdot(1-(1-P_B)\cdot(1-P_B))\\ P_W &= P_A\cdot P_B\cdot(2-P_B)\end{aligned}$$

We know, $P_A > P_B$. Therefore, $(2-P_A) < (2-P_B)$.
P_W from Case-1 is less than P_W from Case-2.

Therefore, the option that that maximizes the chance of winning and thus the option to go for is **B-A-B**.

3 Random Variables

3.1 Part-3.1

$Q_1Q_2 < q_1q_2$ if

- Case-A: $Q_1 < q_1$ and $Q_2 < q_2$.
- Case-B: $Q_1 > q_1$ and $Q_2 < q_2$ but Q_2 is chosen such that even if Q_1 is greater than q_1, the overall product still remains to be less than q_1q_2.
- Case-C: $Q_2 > q_2$ and $Q_1 < q_1$ but Q_1 is chosen such that even if Q_2 is greater than q_2, the overall product still remains to be less than q_1q_2.

There could be other cases as well. So,

$$P(Q_1Q_2 > q_1q_2) \geq P(A \cup B \cup C)$$

It can be observed that A, B and C are mutually exclusive events and therefore,

$$P(A \cup B \cup C) = P(A) + P(B) + P(C)$$

. Since $P(B), P(C) \geq 0$, we can say

$$P(A \cup B \cup C) \geq P(A)$$

Let D be event $Q_1 < q_1$ and E be event $Q_2 < q_2$. D and E are independent as choosing of Q_1 and Q_2 don't affect each other. Given in question, $P(D) = 1 - p_1$ and $P(E) = 1 - p_2$

$$\begin{aligned} P(A) &= P(D \cap E) \\ P(A) &= P(D) \cdot P(E) \\ P(A) &= (1 - p_1)(1 - p_2) \\ P(A) &\geq 1 - p_1 - p_2 \end{aligned}$$

Earlier, we proved

$$P(A \cup B \cup C) \geq P(A)$$

Combining both inequalities, we get

$$\begin{aligned} P(A \cup B \cup C) &\geq 1 - p_1 - p_2 \\ P(Q_1Q_2 > q_1q_2) &\geq 1 - (p_1 + p_2) \end{aligned}$$

3.2 Part-3.2

We know that,

$$\sigma^2 = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \mu)^2$$

$$(n-1)\sigma^2 = |x_i - \mu|^2 + \sum_{k=1, k \neq i}^{n} |x_k - \mu|^2 \quad (1)$$

Since all the terms in the summation are squares, we can say the summation is ≥ 0.
From (1),

$$(n-1)\sigma^2 - |x_i - \mu|^2 = \sum_{k=1, k \neq i}^{n} |x_k - \mu|^2 \geq 0$$

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

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

Comparing with Chebyshev's inequality:

According to Chebyshev, $|S_k| \leq \frac{n}{n-1}$ for $k = \sqrt{n-1}$. This means $|S_k| = 0$ or 1 for $|x_i - \mu| \geq \sigma\sqrt{n-1}$. According to the inequality in question, $\forall i$ such that x_i is in the dataset, we have $|x_i - \mu| \leq \sigma\sqrt{n-1}$.

Both the inequalities go hand-in-hand but the inequality in this question gives a strict number of elements that satisfy $|x_i - \mu| \leq \sigma\sqrt{n-1}$. On the other hand, from Chebyshev's inequality, we get an upper bound of number of elements that satisfy $|x_i - \mu| \geq \sigma\sqrt{n-1}$. Both don't contradict each other and can be used together to prove any other theorem if required. As N increases, the bound becomes tighter and all the elements lie inside the range making $|S_k| = 0$.

4 Staff Assistant

4.1 Part a

The number of people sitting for interviews is n. First, we'll figure out the probability of the best candidate being the i^{th} person being interviewed. Here, as the question mentions, E is the event of selecting the best candidate and E_i is the probability of i^{th} candidate is the best and we hire them.
We proceed by using the going back to the basics of probability: taking the cases we need in the numerator by dividing the total number of cases in the denominator. In the denominator, we have the total number of permutations, which is $n!$. We have $i-1$ people before the i^{th} person and $n-i$ people after them.
In the $i-1$ people before i, m are going to be rejected. Now, the only way that i can be selected is if the best person among $i-1$ is was one among the rejected, as otherwise, they would be selected before the best person. So, we get the probability $P(E_i)$ as:

$$P(E_i) = \frac{\binom{n-1}{i-1} \cdot \binom{m}{1} \cdot (i-2)! \cdot (n-i)!}{n!}$$

We first divide the remaining people into two groups $i-1$ and $n-i$ representing the people before and after the i^{th} person. That is the first term in the numerator. Then, we find the second best person and put them in a spot in one of the first m slots. We permute the remaining people as their order doesn't matter.

$$P(E) = \frac{(n-1)! \cdot m \cdot (i-2)! \cdot \cancel{(n-i)!}}{n! \cdot (i-1)! \cdot \cancel{(n-i)!}} = \frac{m}{n}\left(\frac{1}{i-1}\right) \tag{1}$$

Now summing this over all E_i's, we get $P(E)$. And since we want $i > m$ as we need the best person to be selected, i starts from m + 1. Thus:

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

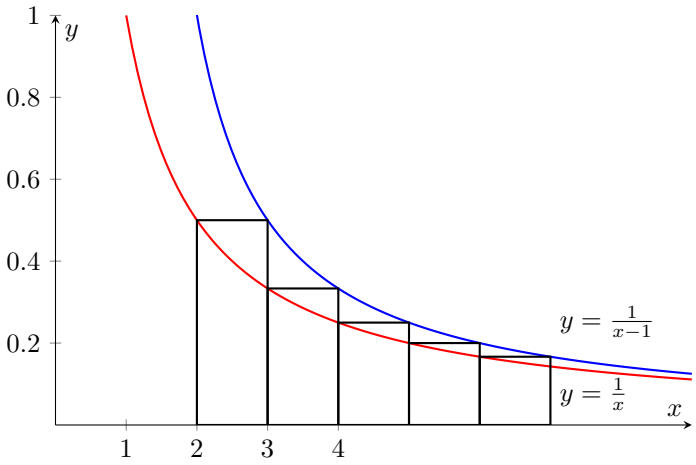
4.2 Part b

We basically have to bound the following expression denoted by S:

$$S = \sum_{i=m+1}^{n}\frac{1}{i-1} = \sum_{i=m}^{n-1}\frac{1}{i}$$

We can do that with integrals. We have:

$$\int_m^n \frac{1}{x}\,dx \leq \sum_{i=m}^{n-1}\frac{1}{i} = \sum_{i=m+1}^{n}\frac{1}{i-1} \leq \int_m^n \frac{1}{x-1}\,dx$$



This is using the properties of integrals, but instead of considering the lower sums and upper sums, we take the sum itself(which we want to approximate) and consider two integrals above and below it. The sums satisfy the inequality as both are the lower bounds of both integrals. On integrating we get:

$$\ln(n) - \ln(m) \leq S \leq \ln(n-1) - \ln(m-1)$$

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

(Multiplying by $\frac{m}{n}$)

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

(Since $P(E) = \frac{m}{n}S$)

4.3 Part c

Since we want to maximize the lower bound and figure out m in terms of n, we put $\frac{m}{n}$ as x. Thus we get:

$$\frac{m}{n}(\ln(n) - \ln(m)) = x \ln(1/x) = -x \ln x$$

Differentiating and equating to zero we get:

$$-(\ln x + x \cdot 1/x) = -(\ln x + 1) = 0$$

This implies $x = 1/e$ and it is the maximum because the sign of the derivative changes from positive to negative. Therefore

$$\frac{m}{n} = \frac{1}{e} \implies m = \frac{n}{e}$$

5 Free Trade

We have to find the probability of being the first repetition and check where the probability hits a maximum. Let i denote the position of the first repetition and X_i denote the event that the first repetition happens at i. This would mean all the numbers before i should've been distinct. The probability (say P_1) of this happening(similar to birthday paradox) is

$$P_1 = \frac{\binom{200}{i-1} \cdot (i-1)!}{(200)^{i-1}}$$

We choose(to get the cases we desire to find the probability of) i - 1 numbers without repetition as the numbers in the first (i - 1) positions. After this, if we get any number chosen so far in the i^{th} position, we get the prize. The probability (say P_2 of that happening is

$$P_2 = \frac{i-1}{200}$$

Since these two events are independent(each getting a number is similar to rolling a die), we have the final probability as:

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

In order to maximize this, we check the inequality

$$\frac{P(X_{i+1})}{P(X_i)} > 1$$

Whenever the fraction goes less than 1, we have the function to be decreasing. Let the fraction be denoted by M. Let (i - 1) be j.

$$M = \frac{i/200}{(i-1)/(200-i+1)} = \frac{i \cdot (200-i+1)}{200 \cdot (i-1)} = \frac{(j+1) \cdot (200-j)}{200 \cdot j} > 1$$

We want to check whenever this condition is satisfied. Simplifying we get:

$$200 \cdot (j+1) - j \cdot (j+1) > 200 \cdot j$$

$$200 - j^2 - j > 0 \implies j^2 + j - 200 < 0$$

The quadratic after a point grows unbounded. Therefore is only a certain region where this condition is true. Solving the quadratic we get:

$$-14.65 < j < 13.65$$

The integer j where this quadratic is satisfied is at j = 13. Since we chose the expression as

$$\frac{P(X_{i+1})}{P(X_i)} > 1$$

So, we have

$$\frac{P(X_{15})}{P(X_{14})} > 1$$

And after that the value of value of probability starts dropping. Thus, the position where the maximum is attained is at 15. Hence, we should go for position 15.

6 Update Functions

The functions implemented in Question 6 are as follows:

- **UpdateMean:** The basic idea behind it's derivation is that mean is sum of all values divided by the total number of elements. Since we know the number of elements and we can get the sum of the elements just by multiplying the previous mean with the number of elements and we don't have to sum all the elements again. The Formula derived goes as follows:

$$\text{NewMean} = (\text{OldMean} \cdot n + \text{NewDataValue}) \,/\, (n+1)$$

- **UpdateMedian:** Clearly, the definition of median differs when the number of elements in the array is even compared to when it is odd. We proceed by assuming the array is sorted. We storing the sorted array in a different array named "temp" and using temp for future purposes. The positioning of the NewDataValue using a few if-else statements is then checked and it is decided if the NewDataValue will take part in the calculating NewMedian or not.
  Assumption: The median when the number of elements is even is taken as the average of middle two elements.
  if-else statements can be found below used in deciding median.

```
def UpdateMedian(OldMedian, NewDataValue, n, A):
    temp = np.array(sorted(A)) # If we don't assume A to be sorted
    if ( n % 2 == 0 ):
        if ( NewDataValue > temp[n//2] ):
            return temp[n//2]
        elif ( NewDataValue < temp[(n//2)-1] ):
            return temp[(n//2)-1]
        else:
            return NewDataValue
    else:
        if ( NewDataValue > temp[(n+1)//2] ):
            return (OldMedian+temp[(n+1)//2])/2
        elif ( NewDataValue < temp[(n-3)//2] ):
            return (OldMedian+temp[(n-3)//2])/2
        else:
            return (OldMedian+NewDataValue)/2

```

Listing 1: NewMedian

- **UpdateStd:** The defintion of standard deviation is the sum of squares about the deviation. Hence, we have:

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

  On expanding, we get a formula which is more suitable for our case. It goes as follows:

$$\sigma^2 = \frac{\left(\sum_{i=1}^{n} x_i^2\right) - n \cdot \text{mean}^2}{n-1}$$

  To get the new Standard Deviation, we just have to add the square of NewDataValue and the term related to mean can be changed as we have access to both OldMean and the NewMean values and we have access to the number of elements in the array.

$$\sigma_{\text{new}} = \sqrt{\frac{\sigma_{\text{old}}^2 \cdot (n-1) + \text{NewDataValue}^2 + n \cdot \text{OldMean}^2 - (n) \cdot \text{NewMean}^2}{n}}$$

  The above expression will give the value of New Standard Deviation without having to compute it again.

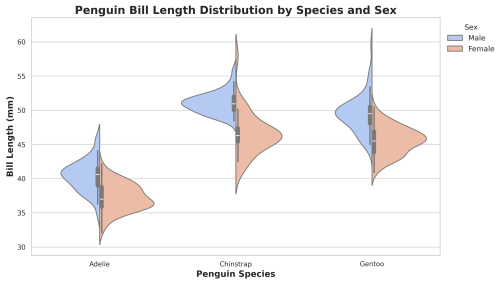
Updating the Histogram: Histograms plot the number of entries present in every bin. When a new value is added, the frequency or number of entries of the bin that contains the new data value is increased by 1.

For running the program: You will find a python file named `A1-Q6.py` in the zip file. To run any testcase, open the python file and edit the first 2 lines of the code (Read the comments in the python file for better understanding of what each variable represent). After editing the python file, save the file and use `python3 A1-Q6.py`.

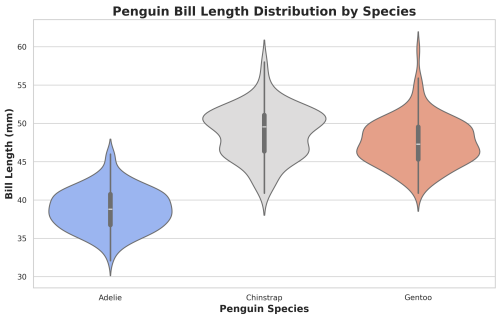
7 Plots

In this question, we studied about 4 major plots and with the learning, we have described the major uses of each of the plots and 1 example plot that was generated on sample data.

- **Violin Plot:** A Violin plot uses density curves to depict distribution of numeric data for one or more groups (You can plot values for more than one category in the same plot). A Violin plot is made by building density curves on center lines. Along with the density curves, the Violin plot also has a box plot along the center line to provide additional information like quartiles, medians.



(a) Violin Plot 1



(b) Violin Plot 2

- **Pareto Chart:** A Pareto Chart is a bar graph along with a line graph that represents the cumulative frequency in percentage. One important point to note is that the bar graph has values in descending order. The line demonstrates the aggregated impact of all the categories/factors and their outcomes in terms of frequency or quantity. A good example to demonstrate the use of the line graph is x-axis containing factors affecting people to reach late to office and y-axis depicts number of people affected by it. Here, the cumulative frequency would convey how many people are affected by a combination of these factors.

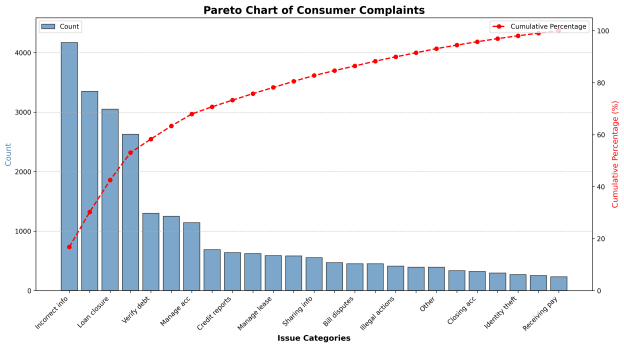


Figure 2: Pareto Chart

- **Coxcomb Chart:** These are modified pie charts. The angle assigned to every factor is same (unlike pie charts where the percentage is decided based on the angle). The area of the sector determines the measure of the values of every category. Coxcomb can be used to measure the values of a particular category for different data sets in the same coxcomb chart. Example: A Coxcomb chart was used to represent the amount of deaths due to Covid in every month for the last 5 Years. In the above example, the sectors represent the months of a year and in every sector, you can use different colours to represent different years and choose areas accordingly.

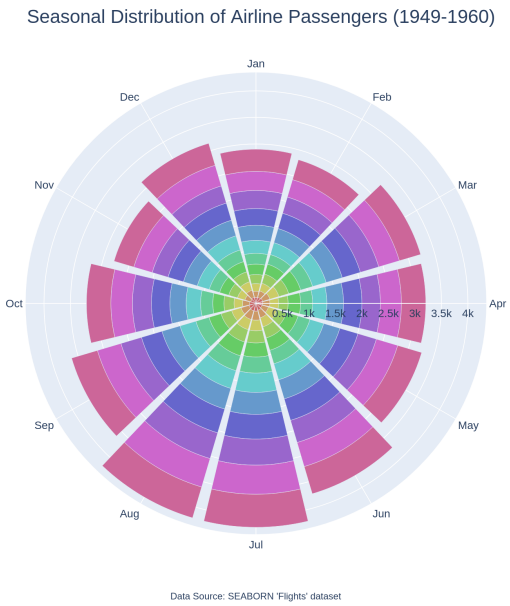


Figure 3: Coxcomb Chart

- **Waterfall Plot:** A Waterfall Plot is a three-dimensional graph that displays the frequencies of an asset over time. These are particularly useful when you want to study the trends and changes in the factors you are measuring.

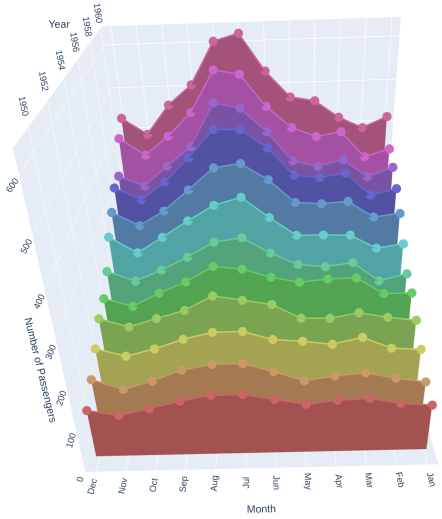


Figure 4: Waterfall Plot

8 Mona Lisa

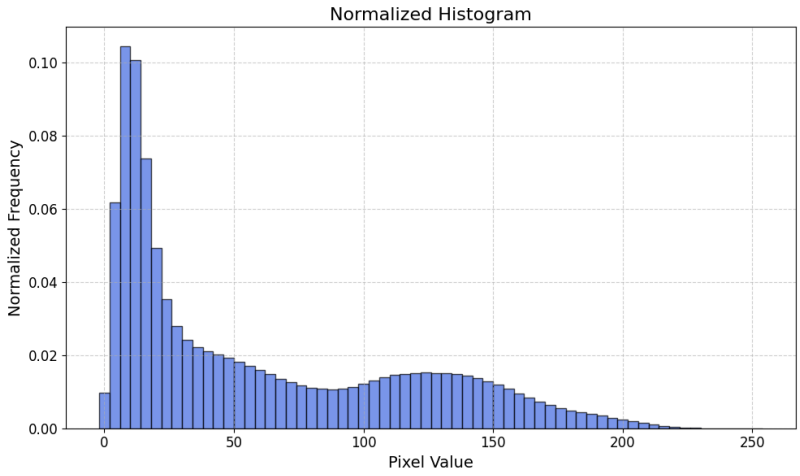
For running the program: python3 A1-Q8.py

The following libraries were used:

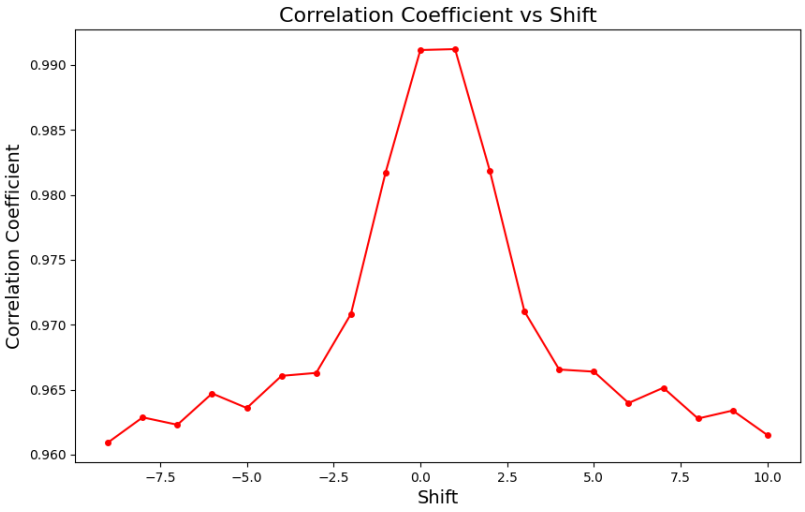
- ⋆ cv2
- ⋆ matplotlib.pyplot
- ⋆ numpy

Shifting the image is done after checking whether the result of j + t_x is within bounds. The formula behind the correlation coefficient is:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$



(a) Histogram



(b) Correlation

Both plots are as expected. The correlation plot has a maximum when no pixels are shifted and it slowly decreases as pixels are shifted as the pixels become zero so the correlation which was once linear slowly drops. The histogram shows the probability of each type of color.