CS215 - Assignment 1

Ujwal L Shankar (23b1050)
Madhav R Babu (23b1060)
Marumamula Venkata Pranay (23b1073)

Contents

1 Let's Gamble

Let P(win upon 1 throw) = $\frac{1}{2}$, since {2, 3, 5} are the only prime number that can be obtained on throwing a fair die.

Consider two players A and B who each throw a die n times. At the end of n throws, three cases are possible:

1. No: wins by A > No: Wins by B
2. No: wins by A < No: Wins by B
3. No: wins by A = No: Wins by B

By symmetry, $P(A > B) = P(A < B) = P'$.
The total probability must sum to 1:

$$P' + P(A = B) + P' = 1$$

From this equation, we get:

$$2P' + P(A = B) = 1$$

$$P' = \frac{1 - P(A = B)}{2}$$

Now consider the case of the $(n + 1)^{th}$ throws:

- If $A > B$, the next throw for A can be either prime or not prime.
- If $A = B$, the next throw for A must be prime.
- If $A < B$, A cannot end up with more wins than B.

Therefore, the probability that A ends up with more wins than B after $n+1$ throws is: P' + P(A = B) $\cdot \frac{1}{2}$
Substituting P' into the equation, we get:

P(A having more wins than B) = $\frac{1-P(A=B)}{2} + \frac{P(A=B)}{2}$
P(A having more wins than B) =$\frac{1}{2}$

2 Two Trading Teams

Let P(winning against A) = a where $0 \leq a \leq 1$
Let P(winning against B) = b where $0 \leq b \leq 1$
Since B is better than A, we have $b < a$.

Consider two strategies:
1. Strategy 1: A-B-A
The probability of winning is given by:

$$P(\text{winning}) = ab + (1 - a) \cdot ab$$

$$P(\text{winning}) = ab\,[1 + (1 - a)] = ab\,[2 - a]$$

2. Strategy 2: *B-A-B*
The probability of winning is given by:

$$P(\text{winning}) = ab + (1 - b) \cdot ab$$

$$P(\text{winning}) = ab\,[1 + (1 - b)] = ab\,[2 - b]$$

Since $b < a$, we have $2 - b > 2 - a$. Therefore:

$$ab\,[2 - b] > ab\,[2 - a]$$

Thus, the second strategy *B-A-B* is better since $P(winning)$ is greater.

3 Random Variables

3.1

Given:

$$P(Q_1 < q_1) \geq 1 - p_1$$

$$P(Q_2 < q_2) \geq 1 - p_2$$

$Q_1 < q_1$ and $Q_2 < q_2$ implies that $Q_1Q_2 < q_1q_2$ but not vice-versa. Hence,

$$P(Q_1Q_2 < q_1q_2) \geq P(Q_1 < q_1) \cap P(Q_2 < q_2)$$

$$P(Q_1Q_2 < q_1q_2) \geq P(Q_1 < q_1) + P(Q_2 < q_2) - (P(Q_1 < q_1) \cup P(Q_2 < q_2))$$

Since $0 \leq (P(Q_1 < q_1) \cup P(Q_2 < q_2)) \leq 1$,

$$P(Q_1Q_2 < q_1q_2) \geq P(Q_1 < q_1) + P(Q_2 < q_2) - 1$$

On substituting,

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

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

Hence Proved.

3.2

For n distinct values $\{x_\}_{i=1}^{n}$, we know that:

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

Without Loss of Generality, for any single element x_i for $i \in \{1, 2, \ldots n\}$,

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

as all the terms in the summation are positive.

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

Therefore,

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

Comparison with Chebyshev's inequality
Chebyshev's Inequality:

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

For $k = \sqrt{n-1}$,

$$P(|x_i - \mu| \geq \sigma\sqrt{n-1}) \leq \frac{1}{n-1}$$

$$\implies P(|x_i - \mu| < \sigma\sqrt{n-1}) \geq 1 - \frac{1}{n-1} = \frac{n-2}{n-1}$$

Thus, as the value of n(and k) increases, the probability of an element x_i lying in the the range $(\mu - \sigma\sqrt{n-1}, \mu + \sigma\sqrt{n-1})$ approaches to 1 as,

$$\lim_{n\to\infty}(\frac{n-2}{n-1}) = 1$$

While Chebyshev's inequality gives the fraction of elements outside a particular range from the mean, this result gives us the outer bound for most element values as n increases.

4 Staff Assistant

4.1 Part A

Let E_i denote the event that the i^{th} assistant is the best and we hire him.
There are ${}^{n-1}C_{i-1}$ ways of picking i-1 candidates apart from the best during the first i-1 tries. $m \cdot (i-2)! \cdot (n-i)!$ ways of putting the best one in one of the first m slots and arranging the remaining candidates. If the best candidate appears in the first m, we reject him. Hence:

$$P(E_i) = \begin{cases} \frac{{}^{n-1}C_{i-1} \cdot m \cdot (i-2)! \cdot (n-i)!}{n!} & \text{if } m < i \leq n \\ 0 \qquad \text{if } 1 \leq i \leq m \end{cases}$$

$$\frac{{}^{n-1}C_{i-1} \cdot m \cdot (i-2)! \cdot (n-i)!}{n!} = \frac{m}{n} \cdot \frac{1}{i-1}$$

$$P(E) = \Sigma_{j=1}^{n} P(E_j) = \Sigma_{j=1}^{m} P(E_j) + \Sigma_{j=m+1}^{n} P(E_j)$$

$$P(E) = \Sigma_{j=m+1}^{n} \frac{m}{n} \cdot \frac{1}{j-1} = \frac{m}{n} \cdot \Sigma_{j=m+1}^{n} \frac{1}{j-1}$$

4.2 Part B

Consider the function $f(x) = \frac{1}{x-1}$.

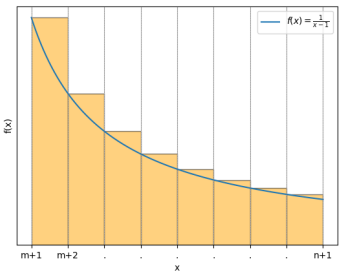

Figure 1:

In this figure[1] each bin has an area of $\frac{1}{j-1}$

$$\Sigma_{j=m+1}^{n} \frac{1}{j-1} \geq \int_{m+1}^{n+1} \frac{1}{x-1} dx = \ln n - \ln m$$

Now, consider the function $f(x) = \frac{1}{x}$.

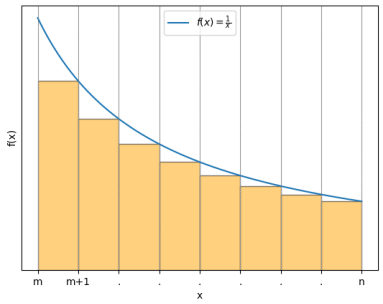

Figure 2:

In this figure [2], each bin has an area of $\frac{1}{j}$.

$$\Sigma_{j=m}^{n-1} \frac{1}{j} = \Sigma_{j=m+1}^{n} \frac{1}{j-1} \leq \int_{m}^{n} \frac{1}{x-1} dx = \ln(n-1) - \ln(m-1)$$

Hence:

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

[1] Code for generating this plot is present in graph_1.py inside Q4 directory
[2] Code for generating this plot is present in graph_2.py inside Q4 directory

4.3 Part C

$$\frac{m}{n} \cdot (\ln n - \ln m) = \frac{-m}{n} \cdot \ln \frac{m}{n}$$

Let $x = \frac{m}{n}$,

$$g(x) = -x \ln x$$

$$g^{'}(x) = -1 - \ln x$$

$$g^{''}(x) = \frac{-1}{x} < 0 \quad \forall x > 0$$

$g^{'}(x_0) = 0 \implies -1 - \ln x_0 = 0 \implies x_0 = \frac{1}{e}$
g(x) is maximised at $x_o = \frac{1}{e} = \frac{m}{n}$
Hence $m = \frac{n}{e}$

$$\frac{m}{n} \cdot (\ln n - \ln m) = \frac{1}{e} \quad \text{when m= } \frac{n}{e}$$

$$\implies P(E) \geq \frac{1}{e}$$

5 Free Trade

Let E_j be the event that the first repetition of ID number happens with the j^{th} person standing in the queue.

Now,

$$P(E_j) = \frac{^{200}P_{j-1} \cdot (j-1)}{(200)^j}$$

On simplifying,

$$P(E_j) = \frac{(200)! \cdot (j-1)}{(201-j)! \cdot (200)^j}$$

If standing at the j^{th} position maximizes the chances of winning a free trade, then:

1. $P(E_j) > P(E_{j-1})$:

$$\frac{(200)! \cdot (j-1)}{(201-j)! \cdot (200)^j} > \frac{(200)! \cdot (j-2)}{(202-j)! \cdot (200)^{j-1}}$$

$$(202-j) \cdot (j-1) > 200 \cdot (j-2)$$

$$j^2 - 3j - 198 < 0$$

Ignoring the negative case,

$$j < 15.65 \tag{1}$$

2. $P(E_j) > P(E_{j+1})$

$$\frac{(200)! \cdot (j-1)}{(201-j)! \cdot (200)^j} > \frac{(200)! \cdot j}{(200-j)! \cdot (200)^{j+1}}$$

On simplifying,

$$(200) \cdot (j-1) > (201-j) \cdot j$$

$$j^2 - j - 200 > 0$$

Ignoring the negative case,

$$j > 14.65 \tag{2}$$

From (1) and (2),

$$j = 15$$

Therefore, standing at the 15^{th} position in the queue maximises the chances of winning a free trade.

We have written a Python code to compute the same, which can be found in the submission as Q5.py

6 Update Functions

We have added the code to update the mean, median, and standard deviation in O(1) time inside the file Q6.py.

6.1 Updating Mean

$$S = \Sigma_{i=1}^{i=n} x_i = n \cdot OldMean$$

$$NewMean = \frac{S + NewDataValue}{n+1}$$

6.2 Updating Median

We divided the problem into two cases when n=even and n=odd. We check where the new data should go if sorted and then we update the median appropriately.

6.3 Updating Standard Deviation

$$S = \Sigma_{i=1}^{i=n} x_i^2 = (n-1) \cdot OldStd + n \cdot OldMean$$

$$S\_New = \Sigma_{i=1}^{i=n+1} x_i^2 = S + NewDataValue^2$$

$$NewStd = \sqrt{\frac{S\_New - NewMean^2}{N}}$$

6.4 Updating Histogram

To update the histogram, loop through the frequency table and check where the new data value belongs. If we can find an appropriate place for it update the frequency table and adjust the value corresponding to that row of the frequency table in the histogram. If we are unable to find an appropriate place, either create a new row in the frequency table or create a new frequency table that can accommodate the new value.

7 Plots

7.1 Violin Plot

Violin plots use density curves [3] The width of each curve is proportional to the frequency of data points in each region. They are useful when comparing the date distributions between multiple groups. Violin plots can be constructed both horizontally and vertically depending on the need, extending the plot on its vertical axis is easier than horizontal.

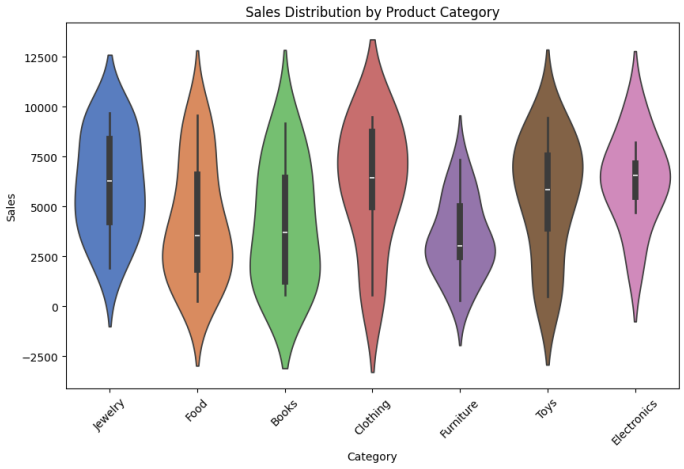

Figure 3: Violin Plot

7.2 Pareto Chart

Pareto chart consists of a bar chart and a line chart. The bar chart is sorted in descending order and the line represents the cumulative total of the individuals

[3] A density curve, or Kernel Density Estimate (KDE), which depicts data distribution by assigning a small area around each data point, shaped by a kernel function, which can vary in shape and width. The final curve is formed by stacking these areas, where regions with more data points have higher peaks to visualize data.

in percentage. This plot is inspired by the Pareto principle which states that 80% of the work is done by 20% of the people. Pareto chart allows the user to understand the role played by each cause in the net effect and make decisions accordingly.

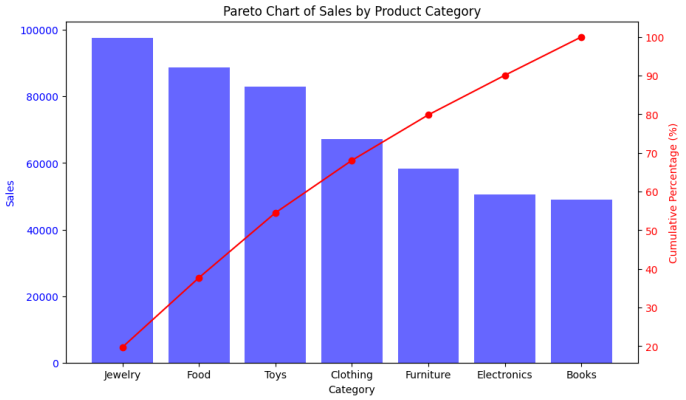

Figure 4: Pareto Chart

7.3 Coxcomb Chart

It is a modified pie chart, in which each slice have an equal angle but the are is proportional to the data value (the area is scaled by data thus resulting in a different radius for each slice). Each section of the coxcomb contains subsections with different color that overlap with each other.

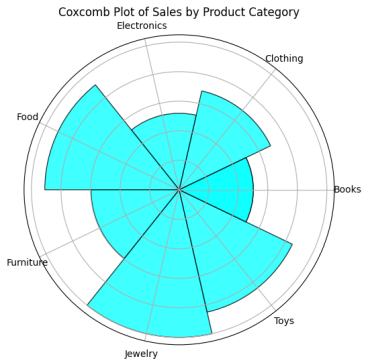

Figure 5: Coxcomb Chart

7.4 Waterfall Plot

A waterfall plot is a 2D or 3D visualization that shows how a phenomenon changes over time or another variable. Consider a series of overlapping "mountains", distributed horizontally and vertically on the screen, each representing a spectrum or data set. The closer a "mountain" is to you, the more it obscures the "mountains" behind it. This plot is used to see how spectrograms or cumulative spectral decay.

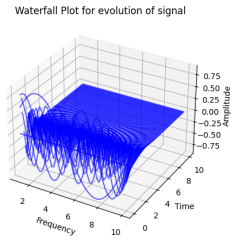

Figure 6: Waterfall Plot

8 Monalisa

The code for solving this question can be found inside Q8.ipynb. We have added the external libraries numpy and matplotlib for handling data and making plots respectively. The python notebook also contains the plots that are obtained on executing the code.