DT049A-HT22 Lab Report

Student: Kaan Tekin Öztekin

Date of submission: 24.11.2025

The full Python source codes and figures used in this report can be found in my GitHub repository at the following link: github.com/kaanoztekin99/stochastic-processes.

1 – Basic Probability

Task a

Let $X_1, X_2, \ldots, X_n$ be IID random variables drawn from the Uniform$(0, 1)$ distribution. The true cumulative distribution function (CDF) of a Uniform$(0, 1)$ random variable is

$$F(x) = x, \qquad 0 < x < 1.$$

To estimate the CDF, we generate n samples, sort them in ascending order, and assign

$$\hat{F}_n(x_{(i)}) = \frac{i}{n},$$

where $x_{(i)}$ denotes the i-th order statistic. We repeat this procedure for sample sizes $n = 10, 100, 1000$, following Algorithm 1.

Plot Interpretation

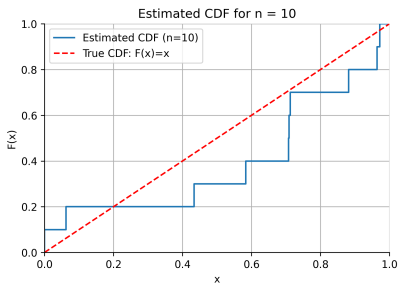

Figure 1: Estimated CDF for $n = 10$

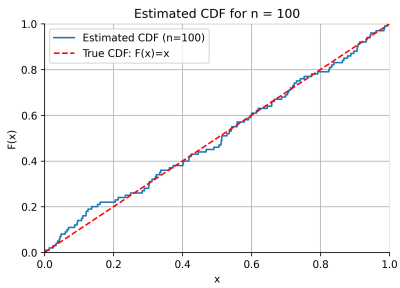

Figure 2: Estimated CDF for $n = 100$

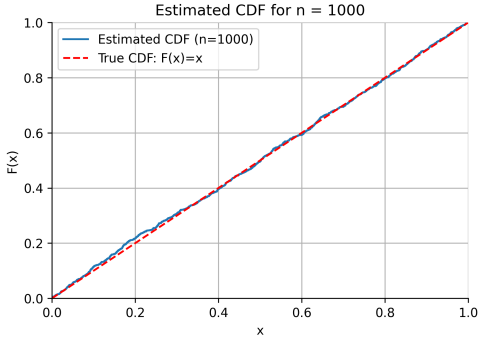

Figure 3: Estimated CDF for $n = 1000$

Case $n = 10$: The estimated CDF is highly irregular and shows significant deviations from the true CDF. Due to the small sample size, the steps are wide and the approximation has high variance.
Case $n = 100$: The estimated CDF is noticeably smoother. Although small fluctuations remain, the plot aligns much closer with the theoretical line $F(x) = x$.

Case $n = 1000$: The estimated CDF almost perfectly overlaps with the true CDF. Differences become visually negligible, and the approximation is highly accurate.

Behavior as n Becomes Large

As the number of samples increases, the estimated CDF converges uniformly to the true CDF:

$$\sup_x \left| \hat{F}_n(x) - F(x) \right| \xrightarrow{a.s.} 0,$$

Thus, for large n, the empirical estimate provides an excellent approximation of the true distribution, and the graphical difference becomes unnoticeable.

Task b

A fair coin is tossed 10 times. We want to compute the probability of obtaining at least one run of exactly five heads in a row, while excluding all runs of length six or more.

Let a run of five heads begin at position i ($1 \leq i \leq 6$). A run of length 5 occupies positions $i, i+1, i+2, i+3, i+4$. To ensure that the run is *exactly* five heads long, the toss immediately before and after the block (if they exist) must both be tails.

Because the sequence has length 10, the block can start at exactly the following six positions:

$$i = 1, 2, 3, 4, 5, 6.$$

These events are pairwise disjoint: it is impossible for a valid 5-head run to overlap with another without creating a forbidden 6-head run. Thus,

$$P(\text{at least one exact 5-head run}) = \sum_{i=1}^{6} P(E_i),$$

where E_i is the event that an exact run of five heads starts at position i.

1. Run at the beginning ($i = 1$). Pattern:

$$\underbrace{HHHHH}_{1-5}\ T\ ****$$

The first six positions are fixed. The last four positions are free, giving $2^4 = 16$ valid sequences:

$$P(E_1) = \frac{16}{2^{10}} = \frac{16}{1024}.$$

2. Run at the end ($i = 6$).

$$****\ T\ \underbrace{HHHHH}_{6-10}$$

Again four free positions:

$$P(E_6) = \frac{16}{1024}.$$

3. Run in the middle ($i = 2, 3, 4, 5$). Example for $i = 3$:

$$* \, T \, \underbrace{H H H H H}_{3-7} \, T \, * *$$

Each middle position has exactly three free bits, so:

$$P(E_i) = \frac{8}{1024}, \qquad i = 2, 3, 4, 5.$$

Four such positions:

$$P_{\text{middle}} = 4 \cdot \frac{8}{1024} = \frac{32}{1024}.$$

Total Probability

$$P_{\text{total}} = \frac{16}{1024} + \frac{32}{1024} + \frac{16}{1024} = \frac{64}{1024} = \frac{1}{16} = 0.0625.$$

Thus, the probability of obtaining a run of exactly five heads in ten tosses (with no runs of six or more) is:

$$P = \frac{1}{16} = 0.0625,$$

and the program's output is shown in Figure. 4.

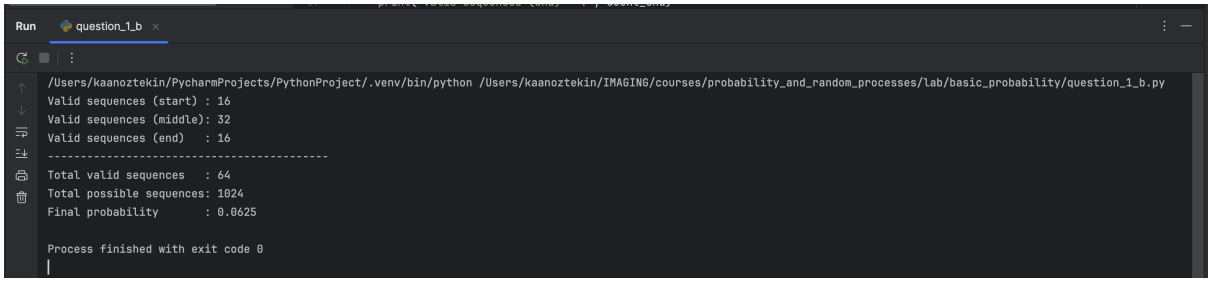

Figure 4: Python program output confirming that valid sequences total to 64 out of 2^{10} possible sequences, yielding a probability of 1/16.

2 – Poisson Processes

Let X be an exponential random variable with parameter λ, modelling the inter-arrival times of a Poisson arrival process. The probability density function (PDF) of X is

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

In this problem, we plot $f_X(x)$ for three different rate parameters:

$$\lambda = 0.3, \quad \lambda = 1, \quad \lambda = 3,$$

and observe how the shape of the exponential distribution changes.

PDF Plots for Different Values of λ

For each value of λ, the exponential density is computed using

$$f_X(x) = \lambda e^{-\lambda x}.$$

The Python script used to generate the figure evaluates the PDF over the range $x \in [0, 10]$ and plots all three curves on the same axes. The resulting figure is shown in Fig. 5.

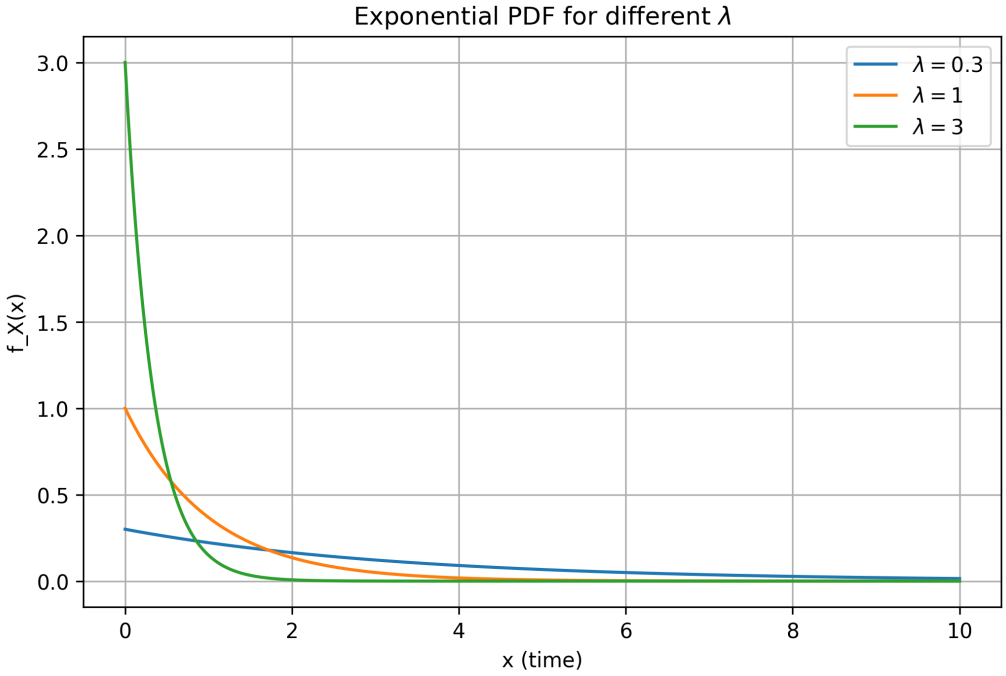

Figure 5: Exponential PDFs for $\lambda = 0.3, 1, 3$. Larger λ values make the distribution more concentrated near $x = 0$, corresponding to shorter waiting times between arrivals.

Observations

- When $\lambda = 0.3$, the PDF decreases slowly and has a long tail. The expected waiting time is large: $\mathbb{E}[X] = 1/0.3 \approx 3.33$.
- When $\lambda = 1$, the PDF decays at a moderate rate, with expected waiting time $\mathbb{E}[X] = 1$.
- When $\lambda = 3$, the PDF begins at a much higher value ($f_X(0) = 3$) and decays very rapidly. The expected waiting time is short: $\mathbb{E}[X] = 1/3 \approx 0.33$.

Overall,

as λ increases, the inter-arrival times become shorter.

This matches the interpretation of λ as the *rate* of a Poisson process.

- λ is the average number of arrivals per unit time.

- The exponential variable X represents the waiting time between consecutive arrivals.
- Larger λ means arrivals occur more frequently, hence the waiting time X tends to be smaller.

Thus, the exponential PDF becomes more steeply decreasing as λ grows.

3 – Markov Processes

In this problem, we simulate the queue length of a taxi stand modeled as a Markov process. At each time step, one customer is serviced and leaves the queue, while the number of new arrivals follows a Poisson distribution with rate $\lambda = 0.85$. The queue length is updated for $N = 10{,}000$ time units, and the resulting distribution is visualized using a histogram.

The simulation shows the long-term behavior of the queue length. Since the arrival rate (0.85) is lower than the service rate (1 customer per unit time), the queue remains stable and typically stays within small values.

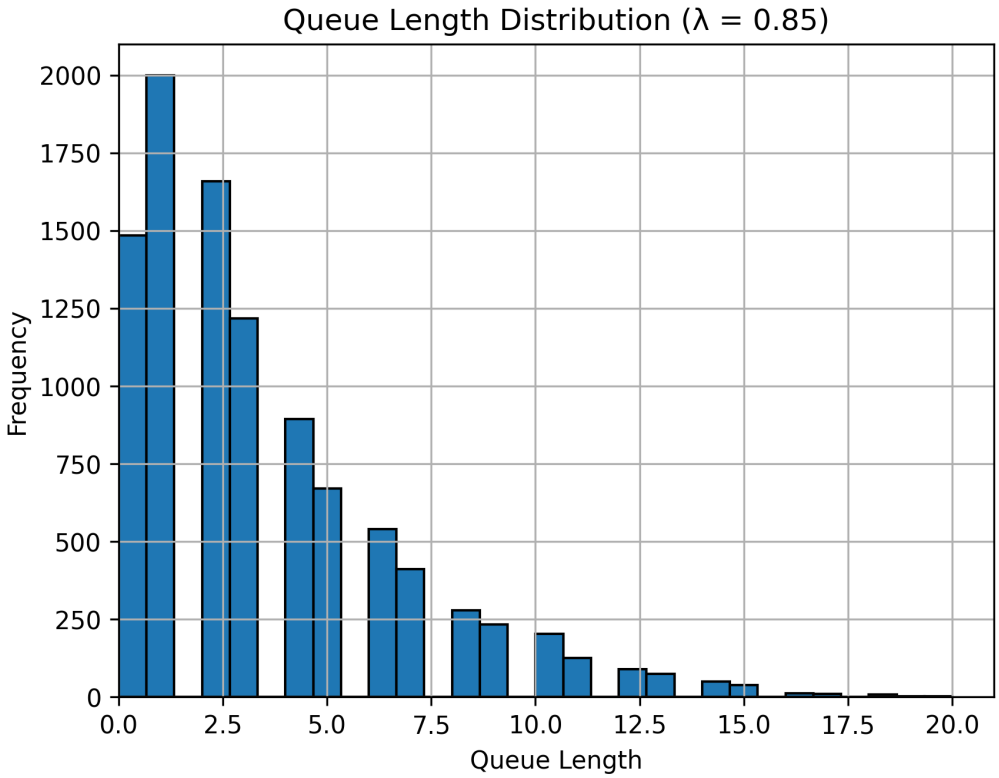

Figure 6: The probability decreases exponentially as the queue length increases, which is expected for a stable M/M/1-type Markov queue with $\lambda = 0.85$ and $\mu = 1$

The figure 6 shows that the queue length is most frequently between 0 and 4 customers, with the highest probability around 2–3 customers. . Since $\lambda < \mu$, the system remains stable, and long queues (greater than 10 customers) occur very rarely. This simulation confirms that the taxi stand can handle the customer arrival rate efficiently without excessive waiting times.