

Lab 08 - Report - Lakshya Kohli - 210123077

In the presented model, having a lot of storms in the next month ($S \geq 6$) is a rare occurrence because of the comparatively low value of λ ($\lambda = 2.9$), hence grouping $S \geq 6$ into one strata is justified. Due to its rarity, the likelihood of such incidents is low. Combining them into a single strata makes the stratification procedure easier, expands the sample size within the strata, and boosts the calculation's overall efficiency. It also makes the analysis more useful and computationally efficient.

The screenshot shows a JupyterLab interface with a terminal window. The terminal displays the output of a Python script executed in a virtual environment. The script performs a Monte Carlo simulation for two different sample sizes, n=100 and n=10000, comparing the results of a Simple Monte Carlo method with a Stratification Method. The output includes the Probability, Confidence Interval, Variance, and Interval Length for each method and sample size.

```

/usr/local/bin/python3 /Users/kohlilakshya/Documents/Sem5/MA323/Lab08/q1.py
kohlilakshya@Lakshyas-MacBook-Air Lab08 % /usr/local/bin/python3 /Users/kohlilakshya/Documents/Sem5/MA323/Lab08/q1.py
For n = 100
Simple Monte Carlo
Probability = 0.74
Confidence Interval = [0.6400037085095548, 0.8399962914904452]
Variance = 0.1507070707070708
Interval Length = 0.1999925829808904

Stratification Method
Probability = 0.8169523203608925
Confidence Interval = [0.7878459710399872, 0.8460586696817978]
Variance = 0.1507070707070708
Interval Length = 0.058212698641810556
-----
For n = 10000
Simple Monte Carlo
Probability = 0.7703
Confidence Interval = [0.7594248384516114, 0.7811751615483886]
Variance = 0.1782531753175283
Interval Length = 0.02175032309677727

Stratification Method
Probability = 0.8011054299833138
Confidence Interval = [0.797837453367397, 0.8043734065992306]
Variance = 0.1782531753175283
Interval Length = 0.006535953231833647
-----
kohlilakshya@Lakshyas-MacBook-Air Lab08 %

```

The status bar at the bottom of the terminal window indicates the current line and column: 'Ln 29, Col 5'. It also shows the active environment: 'Python 3.12.0 64-bit'.

Observations:

- Stratification improved precision by grouping rare events, resulting in more stable probability estimates as we might be able to do better still by oversampling within the important strata and under-sampling those in which f is nearly constant.
- The stratification method typically led to narrower 99% confidence intervals, enhancing result reliability, by reducing the variance.
- Stratification optimised sample utilisation and computational efficiency when dealing with rare events.
- The analysis has practical applications in water resource management, aiding proactive planning based on probability estimates.

Answer 2.

The code effectively calculates the probability $\mu = 0.02636$ using the conditional Monte Carlo technique.

Observations:

- The problem involves generating random variables from a Dirichlet distribution with specific alpha parameters, which is essential for probability estimation.
- The code utilises the gamma distribution to generate random variables with shape parameters corresponding to the given alpha values, facilitating the calculation of the Dirichlet random variable.
- The code conditions on the value of Y_{19} (associated with α_{19}) to determine if it is the largest Y_j , simplifying the probability calculation.
- The approach is computationally efficient and avoids the need to calculate the Dirichlet density directly.

```
kohlilakshya@Lakshyas-MacBook-Air Lab08 % /usr/local/bin/python3 /Users/kohlilakshya/Documents/Sem5/MA323/Lab08/q2.py
Value of  $\mu = P(X_{19} = \max_i (X_i))$  using conditional monte carlo technique is 0.0257
Conditional Expectation (using scipy library) is: 0.026266
```

Answer 3.

Observations:

- The code successfully implements the covariate technique to estimate $\mu = E(f(X))$, considering specific log-normal random variables.
- It allows users to specify parameters (μ and σ^2) for each log-normal distribution, offering flexibility for different scenarios.
- A function to sample log-normal variables is defined, making it reusable and aligned with the chosen parameters.
- Samples are generated for each log-normal variable, with a substantial sample size ($n = 10,000$) for accurate estimation.
- The code features a dedicated function for estimating μ using the covariate technique, considering the maximum of the specific function $f(X)$.
- The estimated value of μ is printed to the console, providing a numerical result for the problem.

```
kohlilakshya@Lakshyas-MacBook-Air Lab08 % /usr/local/bin/python3 /Users/kohlilakshya/Documents/Sem5/MA323/Lab08/q3.py
mu_values : [1.1, 1.2, 0.9, 1.5, 1.3]
sigma2_values : [0.3, 0.2, 0.15, 0.1, 0.25]
Total number of iteration to estimate actual expectation is 100000
Estimated mean using covariate method comes out to be 3.650870727246618
```