Computer aided simulations and performance evaluation

Lab 1 - Estimated average of a stochastic process

Gabriele Cuni 277957

1.0 Introduction

The software is a simulator which estimates the average of a stochastic process. The process output is called X and it is uniformly distributed between 0 and 10, therefore its population mean is 5.

1.1 Input parameters

The simulator accepts input parameters which may be passed via command line or they can be omitted in order to use the default parameters by me.
The parameters are:

- **Seed**: A seed to initialize the numpy random generator [1][2]. The default value is 42.
- **noSamples**: The number of samples extracted from the uniform distribution each runs. The default value is 50
- **confidenceLevel**: The value of 1-α which is used to compute the confidence interval of the estimated average. The default value is 0.95
- **stoppingCondition**: the relative error expressed as a percentage which must be reached to stop the simulations. The default value is 1%

1.2 Formulas

During the simulation the software extracts N_i numbers from the uniform distribution, which are called d_j.

Then the sample mean is computed $x_i = \frac{1}{N_i} \cdot \sum_{j=1}^{N_i} d_j$ for each runs.

When all the runs are computed the average estimate is done with the formula $\hat{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$.

The standard deviation is computed by mean of the equation $s = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - \hat{x})^2}$.

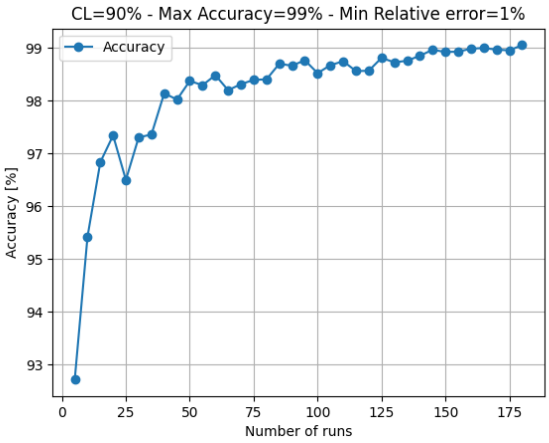
The confidence interval can be now evaluated with $\hat{x} \pm t_{n-1,\frac{\alpha}{2}} \cdot \frac{s}{\sqrt{n}}$.

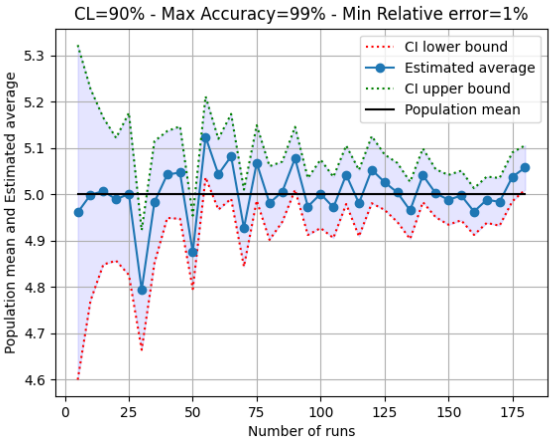
The relative error is $\frac{\Delta}{\hat{x}}$ and the accuracy is $1 - \frac{\Delta}{\hat{x}}$.

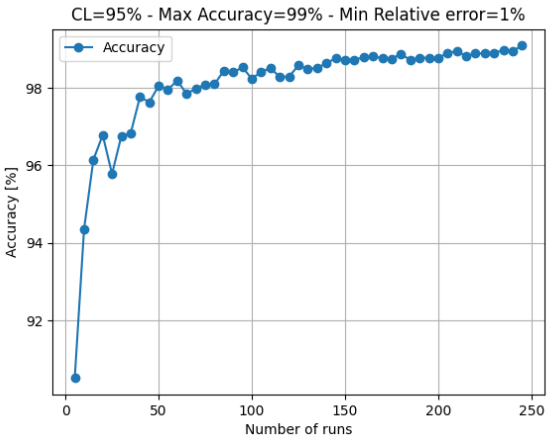
1.3 Python functions

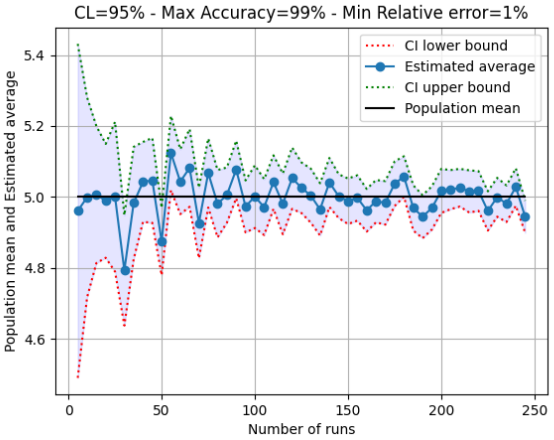
The function used to compute the mean is numpy.mean() [3]
The function used to compute the standard deviation is numpy.std(ddof=1) [4] which is the square root of the estimated variance and it is an unbiased estimator.
The function used to compute the threshold of the t-student is scipy.stats.t.ppf() [5] which calculates the inverse cumulative distribution function from the t-student distribution.
The other computations are simple arithmetic operations.

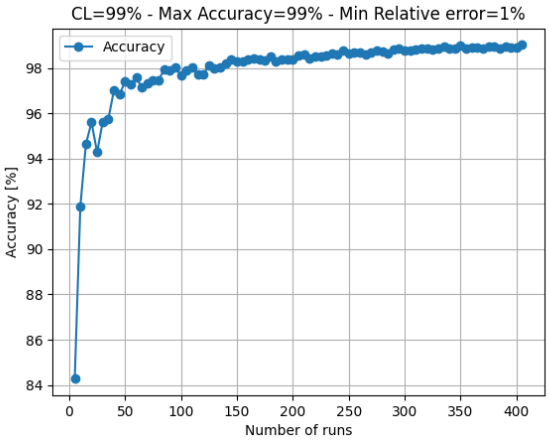
1.4 Charts

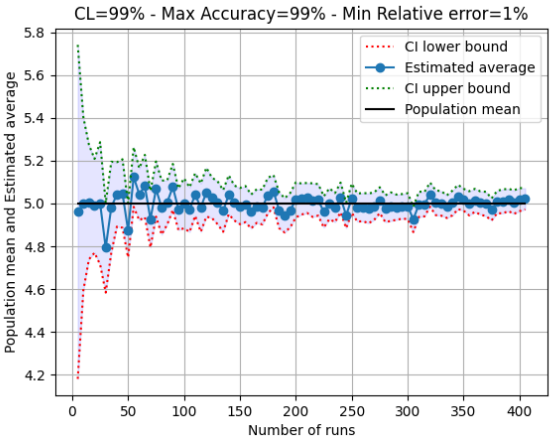

CL=90% - Max Accuracy=99% - Min Relative error=1%
Accuracy
Accuracy [%]
Number of runs


CL=90% - Max Accuracy=99% - Min Relative error=1%
CI lower bound
Estimated average
CI upper bound
Population mean
Population mean and Estimated average
Number of runs


CL=95% - Max Accuracy=99% - Min Relative error=1%
Accuracy
Accuracy [%]
Number of runs


CL=95% - Max Accuracy=99% - Min Relative error=1%
CI lower bound
Estimated average
CI upper bound
Population mean
Population mean and Estimated average
Number of runs


CL=99% - Max Accuracy=99% - Min Relative error=1%
Accuracy
Accuracy [%]
Number of runs


CL=99% - Max Accuracy=99% - Min Relative error=1%
CI lower bound
Estimated average
CI upper bound
Population mean
Population mean and Estimated average
Number of runs

1.5 Conclusions

The above section shows six charts, which all have in common the x axis, that represents the number of times 50 samples are extracted from the uniform distribution and their sample mean is computed. The first experiment has 5 runs, the second one has 10 runs, ecc, ecc.
It is shown that higher the confidence level, higher is the number of runs per experiment which are needed in order to get the desired accuracy.
It is also important to notice as the confidence level increases, the confidence interval width increases as well.
Finally, given a confidence level, increasing the number of runs makes the confidence interval shrinking.

1.6 References

[1] https://numpy.org/doc/stable/reference/random/generator.html#numpy.random.default_rng
[2] https://towardsdatascience.com/stop-using-numpy-random-seed-581a9972805f
[3] https://numpy.org/doc/stable/reference/generated/numpy.mean.html
[4] https://numpy.org/doc/stable/reference/generated/numpy.std.html
[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.t.html