Project 1
Computational Numerical Statistics
DM, FCT-UNL

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Deadline: 15/11/21

The writing of the project report, either in LaTeX or R Markdown, should detail all the theoretical aspects of the methods and provide all the necessary graphical displays necessary to the understanding of the particular cases.

Random variable generation

In the resolution of the following problems **do not** use any of the `R` random variable generation built-in functions, nor any `R` function referring to the densities of random variables.

1. Let $X \sim TruncatedPareto(\alpha, L, H)$, which has probability density function

$$f(x) = \frac{\alpha L^{\alpha} x^{-\alpha-1}}{1-(\frac{L}{H})^{\alpha}}, \qquad \alpha, L > 0, \qquad H > L \qquad x \in [L, H].$$

Fix your `R` random seed to 123 in the simulations.

   (a) Analytically derive the cumulative distribution function (c.d.f.) F of X as well as its inverse F^{-1}.

   (b) Describe in detail (theory + algorithm) the **inverse-transform** (IT) method for generating samples from continuous distributions (as is our case). Implement this method in `R`. Call your routine `sim.IT()` and let it receive as input a generic sample size m as well as generic α, L and H parameters.

   (c) Use routine `sim.IT()` to generate a sample of size $m = 10000$ of

$$X \sim TruncatedPareto(0.5, 2, 4).$$

   Report the first 10 simulated values. Explicitly derive and simplify the expression of the p.d.f.. Plot the sample histogram with the true p.d.f. superimposed.

   (d) Describe in detail (theory + algorithm) the **acceptance-rejection** (AR) method for generating samples from continuous distributions (as is our case). Identify the candidate density function for the particular case of the $X \sim TruncatedPareto(0.5, 2, 4)$ and compute by hand the constants of the AR method (namely, M and the acceptance probability α). Use the `R` function `optimize()` or other to confirm the result you obtained for M.

(e) Implement the AR method in `R`. Call your routine `sim.AR()` and let it receive as input a generic sample size m as well as generic α, L and H parameters. Besides returning the simulated values of the target distribution, the `sim.AR()` routine should also return the simulated values that were rejected.

(f) Use routine `sim.AR()` to generate a sample of size $m = 10000$ of

$$X \sim TruncatedPareto(0.5, 2, 4).$$

Report the first 10 simulated values of the $TruncatedPareto(0.5, 2, 4)$ and the rejection rate. Plot the sample histogram with the true p.d.f. superimposed.

Graphically display, the candidate and p.d.f. functions against the hits and misses of the AR method (as done in class – week 2) when used to simulate just $m = 15$ sample values.

(g) Compare the computational times of routines `sim.IT()` and `sim.AR()` for generating a sample of size $m = 50000$ from the truncated Pareto distribution (you can use the `R` routine `proc.time()` as done in class – week 2).

2. Some random variables can be generated from the exponential distribution (**exponential-based** method), which we know how to obtain from the $U(0, 1)$ distribution. Such is the case of the random variable $X \sim Gamma(\alpha, \theta)$:

$$\text{If } Y_i \underset{iid}{\sim} Exp(1) \text{ then } X = \theta \sum_{i=1}^{\alpha} Y_i \sim Gamma(\alpha, \theta),\ \alpha = 1, 2, \ldots$$

Fix your `R` random seed to 654 in the simulations.

a) Implement the **exponential-based** method in `R` for generating a sample from $X \sim Gamma(\alpha, \theta)$, which has probability density function (p.d.f.)

$$f(x) = \frac{\theta^{-\alpha}}{\Gamma(\alpha)} e^{-\frac{x}{\theta}} x^{\alpha-1}, \qquad \alpha, \theta > 0, \qquad x \geq 0,$$

where Γ is the gamma function.

Call your routine `sim.gam()` and let it receive as input a generic sample size m and the Gamma distribution parameters α (note that here $\alpha \in \mathbb{N}$) and θ. Provide both algorithm and **R** code.

b) Use routine `sim.gam()` to generate a sample of size $m = 10000$ from $X \sim Gamma(2, 1)$. Plot the histogram with the pdf superimposed.

Monte Carlo methods: integration

In the resolution of the problems in this section you may already use the `R` random variable generation built-in functions as well as any `R` function referring to the densities of random variables.

Fix your `R` random seed to 987 in the simulations.

Let

$$\mathcal{I} = \int_0^1 \frac{e^{-x}}{1+x^2} \mathrm{d}x.$$

a) Use the `R` function `integrate()` to compute the value of $\mathcal{I}$.

b) Describe and implement in `R` the Monte Carlo method for estimating $\mathcal{I}$ (use size $m = 10000$). Report an estimate of the variance of the Monte Carlo estimator $\hat{\mathcal{I}}_{MC}$ of $\mathcal{I}$.

c) Describe and implement in `R` the Monte Carlo methods of based on **antithetic variables**, **control variables** and **importance sampling** for estimating $\mathcal{I}$ (size $m = 10000$). Report an estimate of the variance of all the Monte Carlo estimators $\hat{\mathcal{I}}_{ant}$, $\hat{\mathcal{I}}_{cont}$ and $\hat{\mathcal{I}}_{is}$ of $\mathcal{I}$.

d) What's the percentage of variance reduction that is achieved when using those MC estimators instead of $\hat{\mathcal{I}}_{MC}$?

Monte Carlo methods: confidence intervals

In the resolution of the problems in this section you may already use the `R` random variable generation built-in functions as well as any `R` function referring to the densities of random variables.

Fix your `R` random seed to 569 in all simulations.

4. Assume $X \sim N(\mu, \sigma^2)$ with μ **unknown** and σ^2 **known**. Let $X_1, \ldots, X_n$ be a random sample of population X. One has that the **random** interval given by

$$\left[\bar{X} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}, \bar{X} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}\right],$$

is a $(1-\alpha) \times 100\%$ confidence interval (CI) for μ regardless of sample size n.

In this exercise, one wishes to assess how a certain type of data contamination affects the coverage probability of a 95% CI for μ, given a random sample of a population $X \sim N(\mu, 1)$ of size $n = 20$, via a Monte Carlo simulation study (with $m = 10000$ simulations). For that purpose, consider that 90% of those n observations are drawn from a $X \sim N(0, 1)$ distribution and that the remaining 10% are drawn from the contaminated normal distribution $X \sim N(k, 1)$ with $k = 1, 5, 9$. The bad data points generated in this way are called *shift* outliers because the bad data points are shifted from $\mu = 0$ in k standard deviations.

Present a thorough discussion of your results. How harmful can this type of contamination be in terms of the level of confidence of a CI? Would your results still hold for other levels of confidence other than 95%?