Laboratory 2 - Computer Simulation. Monte Carlo Methods.

1 Estimating areas and volumes

RStudio. Don't forget to set the working directory: **Session** → **Set Working Directory** → **Choose Directory**.

Solved exercise. Area of unit circle equals π. Cover a circle with a 2 by 2 square and estimate number π based on 10000, 50000, and 100000 random numbers. Compare the results with the exact value $\pi = 3.14159265358...$

The unit circle is included in $[-1, 1] \times [-1, 1]$. The following function estimates π based on N random numbers.

```
disc_area = function(N) {
  N_C = 0;
  for(i in 1:N) {
    x = runif(1, -1, 1);
    y = runif(1, -1, 1);
    if(x*x + y*y <= 1)
      N_C = N_C + 1;
    }
  return(4*N_C/N);
}
```

Suppose we estimate a value α_{actual} by Monte Carlo method and we get α_{MC}; we can measure the error we make (by using α_{MC} instead of α_{actual}) in at least two ways:

- The **absolute error**: $\epsilon_{abs} = |\alpha_{MC} - \alpha_{actual}|$.
- The **relative error**: $\epsilon_{rel} = \dfrac{|\alpha_{MC} - \alpha_{actual}|}{|\alpha_{actual}|}$. This value can be written as a percent and we get the **percent error**: $\epsilon_{per} = \epsilon_{rel} \cdot 100\%$.

Exercises to work.

1.1 Estimate the volume of the unit sphere (which is known to be $4\pi/3$) using different sizes for the random numbers samples and compute the corresponding (absolute and percent) errors.

1.2 Estimate the area between the parabola of equation $y = -2x^2 + 5x - 2$ and the Ox axis (abscissa) - with 10000 uniform points. Find the exact area by integration and compute the relative error.

Hint: the parabola intersects the Ox axis in $(1/2, 0)$ and $(2, 0)$, and has its vertex in $(5/4, 9/8)$. A rectangular domain in the real plane that contains the area can be $[0, 2] \times [0, 2]$.

2 Monte Carlo integration

Solved exercise. Estimate the following integral using 20000 and 50000 random points (find 30 such approximations for each size and compute the average and the corresponding standard deviation)

$$\int_0^{10} e^{-u^2/2}\,du.$$

The following function gives an estimate for a sample of size N

```
MC_integration = function(N) {
   sum = 0;
   for(i in 1:N) {
      u = runif(1, 0, 10);
      sum = sum + exp(-u*u/2;
   }
   return(10*sum/N);
}
```

We can average $k = 30$ such approximations and compute the standard deviation also using the following function

```
MC_integr_average= function(k, N) {
   estimates = vector();
   for(i in 1:k)
      estimates[i] = MC_integration(N);
   print(mean(estimates));
   print(sd(estimates));
}
```

By running this function we get

```
> MC_integr_average(30, 20000
[1] 1.249768
[1] 0.02327472
> MC_integr_average(30, 50000
[1] 1.253072
[1] 0.01373724
```

Solved exercise. Estimate the following integral using 20000 and 50000 random points (find 30 such approximations for each size and compute the average and the corresponding standard deviation), by using the improved MC integration method, namely with the exponential distribution ($\lambda = 1$)

$$\int_0^{+\infty} e^{-u^2}\,du.$$

(The exact value of the above integral is $\sqrt{\pi}/2 \approx= 0.8862269$.)

First, the following function gives an estimate for a sample of size N

```
MC_improved_integration = function(N) {
   sum = 0;
   for(i in 1:N) {
      u = rexp(1, 1);
      sum = sum + exp(-u*u)/exp(-u);
   }
   return(sum/N);
}
```

We can average $k = 30$ such approximations and compute the standard deviation also using the following function

```
MC_imprvd_integr_average= function(k, N) {
estimates = 0;
   for(i in 1:k)
      estimates[i] = MC_improved_integration(N);
   print(mean(estimates));
   print(sd(estimates));
}
```

By running this function we get

```
> MC_imprvd_integr_average(30, 20000)
[1] 0.8858024
[1] 0.002743676
> MC_imprvd_integr_average(30, 50000)
[1] 0.8861285
[1] 0.00213069
```

Exercises to work.

2.1 ((a) or (b) and (c) or (d))Estimate the value of the following integrals (comparing the result with the exact value) and compute the absolute and the relative error:

$$(a) \int_0^{\pi} \sin^2 x \, dx = \frac{\pi}{2}; (b) \int_1^4 e^x \, dx = 51.87987;$$

$$(c) \int_0^1 \frac{dx}{\sqrt{1-x^2}} = \frac{\pi}{2}; (d) \int_1^{+\infty} \frac{dx}{4x^2-1} = \ln 3/4.$$

2.2 Estimate the value of the following integral by using the improved MC integration with the exponential distribution ($\lambda = 3$, $N = 50000$)

$$\int_0^{+\infty} e^{-2u^2} \, du = \sqrt{\pi/8}.$$

Compare the result with the exact value, and compute the absolute and the relative error. Then find 30 such approximations and compute the average and the corresponding standard deviation. (See the second solved exercise.)

3 Estimating expectations

Solved exercise. The stochastic model for the number of errors (bugs) found in new software release is decribed like follows. Every day, software testers find a random number of errors and correct them. The number of errors found on day i has Poisson(λ_i) distribution whose parameter is the lowest number of errors found during the previous two days:

$$\lambda_i = \min\{X_{i-2}, X_{i-1}\}.$$

What is the expected number of days needed to find all errors? (We suppose that during the first two days, the testers found 31 and 27 errors.) Use 10000 runs for the Monte Carlo estimate.

We generate the number of errors found on each day until this number equals 0. The following function gives the number of days until no more errors appear for a single run.

```
Nr_days = function() {
   nr_days = 1;
   last_errors = c(27, 31);
   nr_errors = 27;
   while(nr_errors > 0) {
      lambda = min(last_errors);
      nr_errors = rpois(1, lambda);
      last_errors = c(nr_errors, last_errors[1]) ;
      nr_days = nr_days + 1;
   }
   return(nr_days);
}
```

We run this function $N = 10000$ times and return their average

```
MC_nr_days = function(N) {
   s = 0;
   for(i in 1:N)
      s = s + Nr_days();
   return(s/N);
}
```

The result is 28.0686, thus, in 4 weeks all the errrors will be found.

Exercises to work.

3.1 Rework the solved exercise by considering that λ_i is the average number of errors in the previous three days (in the first three days were found 9, 15, and 13 errors).

3.2 The stochastic model for the number of fake-news from the social network PokPik can be described as follows: the competent authorities determine a number of accounts which generate the fake news and require from the network to remove them. The number of fake-news found in the ith day, denoted by X_i, is $Poisson(\min(X_{i-1}, X_{i-2}))$.

What is the expected number of days until the number of daily found fake-news becomes less than 10 (which is considered a safe threshhold)? (We suppose that in the first two day the number of found fake-news was 32 şi 25, respectively.) Use $N = 100000$ "runs" for the corresponding Monte Carlo estimator.

4 Estimating probabilities

Solved exercise. The stochastic model for the number of errors (bugs) found in new software release is decribed like follows. Every day, software testers find a random number of errors and correct them. The number of errors found found on day i has Poisson(λ_i) distribution whose parameter is the lowest number of errors found during the previous 3 days:

$$\lambda_i = \min\{X_{i-3}, X_{i-2}, X_{i-1}\}.$$

(a) Estimate the probability that some errors will remain undetected after 21 days using 500 runs. (We suppose that during the first three days, the testers found 28, 22, and 18 errors.)

(b) Estimate this probability, attaining the margin of error ± 0.01 with probability 0.95.

(a) We generate $N = 5000$ Monte Carlo runs; in each run we generate the number of errors found on each day until this number equals 0. The following function gives the number of days until no more errors appear for a single run.

```
Nr_days = function() {
   nr_days = 2;
   last_errors = c(18, 22, 28);
   nr_errors = 18;
   while(nr_errors > 0) {
      lambda = min(last_errors);
      nr_errors = rpois(1, lambda);
      last_errors = c(nr_errors, last_errors[1:2]) ;
      nr_days = nr_days + 1;
   }
   return(nr_days);
}
```

We run this function $N = 5000$ times and return the proportion of runs giving (strictly) more than 21 days.

```
MC_nr_days_21 = function(N) {
   s = 0;
   for(i in 1:N) {
      if(Nr_days() > 21) ;
         s = s + 1;
   }
   return(s/N);
}
```

The computed proportion, 0.246, is an estimate of the probability that some errors will remain undetected after 21 days.

(b) We will estimate the probability in two ways.

First, we can use the "guess", $p^* = 0.246$, and $N \geqslant p^*(1-p^*)\left(\frac{z_{\frac{\alpha}{2}}}{\epsilon}\right)^2$:

```
> alfa = 1 - 0.95
> z = qnorm(alfa/2)
> epsilon = 0.01
> p = 0.246
> N_min = p(1 - p)*(z/epsilon)^2
> N_min
[1] 7125.291
> MC_nr_days_21(N_min + 1)
[1] 0.2547264
```

We get $N \geqslant 7125.291$, and we can run MC_nr_days(N_min + 1)

The second method use the lower bound $N \geqslant \left(\frac{z_{\frac{\alpha}{2}}}{2\epsilon}\right)^2$:

```
> alfa = 1 - 0.95
> z = qnorm(alfa/2)
> epsilon = 0.01
> p = 0.246
> N_min = (1/4)(z/epsilon)^2
> N_min
[1] 9603.647
> MC_nr_days(N_min + 1)
[1] 0.2496968
```

We get $N \geqslant 9603.647$, and we run MC_nr_days(N_min + 1). Usually, with the second method the number of runs gets larger.

Exercises to work.

4.1 Estimate the probability $P(X > Y^2)$, where X and Y are independent Geometric random variables with parameters 0.3 and 0.5. Then estimate the same probability with an error not exceeding 0.005 with probability 0.95. What should be the number of runs?

4.2 The stochastic model for the number of fake-news from the social network PokPik can be described as follows: the competent authorities determine a number of accounts which generate the fake news and require from the network to remove them. The number of fake-news found in the ith day, denoted by X_i, is $Poisson(\min(X_{i-1}, X_{i-2}))$.

(We suppose that in the first two day the number of found fake-news was 32 şi 25, respectively.) Use $N = 100000$ "runs" for the corresponding Monte Carlo estimator.

(a) Estimate the probability that 15 days are enough for cleaning all fake-news.

(b) Estimate this last probability, attaining the margin of error ± 0.01 with probability 0.95.