

# Lab9

## Importance sampling

Piotr Ginalski

2025-04-14

As a first example, we are going to simulate random walk with different probability distribution of step.

```
func <- rnorm
start_point <- 10

up_barrier <- 20
down_barrier <- 0

Z <- start_point
while(Z >= down_barrier && Z <= up_barrier){
  Z <- Z - func(1)
}
```

Now we are going to pack it in function.

```
simulation <- function(start_point, up_barrier, down_barrier, func){
  Z <- start_point
  while(Z >= down_barrier && Z <= up_barrier){
    Z <- Z - func(1)
  }
  return(ifelse(Z < down_barrier, 1, 0))
}
```

Let's pack it in function for calculating probabilities for different distributions.

```
calculate_probability <- function(start_point, up_barrier, down_barrier, func, n_sim){
  outputs <- numeric(n_sim)
  for (i in 1:n_sim){
    outputs[i] <- simulation(10, 20, 0, func)
  }

  paste("Estimator:", mean(outputs), "+-", 2*sd(outputs) / sqrt(n_sim))
}
```

After definition of our function, let's calculate some examples.

```
gen_norm <- function(x) rnorm(1, 0, 1)
calculate_probability(10, 20, 0, gen_norm, 10000)
```

```
## [1] "Estimator: 0.4979 +- 0.0100004118327038"
```

```
gen_norm <- function(x) rnorm(1, -0.1, 1)
calculate_probability(10, 20, 0, gen_norm, 10000)
```

```
## [1] "Estimator: 0.1079 +- 0.00620539174336873"
```

```
gen_norm <- function(x) rnorm(1, -0.2, 1)
calculate_probability(10, 20, 0, gen_norm, 10000)
```

```
## [1] "Estimator: 0.016 +- 0.00250962742712585"
```

```
gen_norm <- function(x) rnorm(1, -0.3, 1)
calculate_probability(10, 20, 0, gen_norm, 10000)
```

```
## [1] "Estimator: 0.0021 +- 0.000915598073383293"
```

```
gen_norm <- function(x) rnorm(1, -0.4, 1)
calculate_probability(10, 20, 0, gen_norm, 10000)
```

```
## [1] "Estimator: 1e-04 +- 2e-04"
```

```
gen_norm <- function(x) rnorm(1, 0.5, 1)
calculate_probability(10, 20, 0, gen_norm, 10000)
```

```
## [1] "Estimator: 1 +- 0"
```

Now we are going to focus on importance sampling. Lets' clear an environment.

```
rm(list = ls())
```

```
weighted_simulation <- function(start_point, up_barrier, down_barrier, func, mean){
  Z <- start_point
  while(Z >= down_barrier && Z <= up_barrier){
    Z <- Z - func(1)
  }
  return(ifelse(Z < down_barrier, 1, 0) * exp(2*mean*Z) * exp(-2*mean*start_point))
}
```

```
calculate_probability_weighted <- function(start_point, up_barrier, down_barrier, func, mean, n_sim){
  outputs <- numeric(n_sim)
  for (i in 1:n_sim){
    outputs[i] <- weighted_simulation(10, 20, 0, func, mean)
  }
  paste("Estimator:", mean(outputs), "+-", 2*sd(outputs) / sqrt(n_sim))
}
```

Let's use our function!

```

mean <- 0
sd <- 1

gen_norm <- function(x) rnorm(1, mean, sd)
calculate_probability_weighted(10, 20, 0, gen_norm, mean, 10000)

```

```
## [1] "Estimator: 0.504 +- 0.0100001800163815"
```

```

mean <- 0.1
sd <- 1

gen_norm <- function(x) rnorm(1, mean, sd)
calculate_probability_weighted(10, 20, 0, gen_norm, mean, 10000)

```

```
## [1] "Estimator: 0.107138513935028 +- 0.00078259424664142"
```

```

mean <- 0.2
sd <- 1

gen_norm <- function(x) rnorm(1, mean, sd)
calculate_probability_weighted(10, 20, 0, gen_norm, mean, 10000)

```

```
## [1] "Estimator: 0.014341302504523 +- 6.51695920195722e-05"
```

```

mean <- 0.3
sd <- 1

gen_norm <- function(x) rnorm(1, mean, sd)
calculate_probability_weighted(10, 20, 0, gen_norm, mean, 10000)

```

```
## [1] "Estimator: 0.00174155618185163 +- 9.7999348716984e-06"
```

```

mean <- 0.5
sd <- 1

gen_norm <- function(x) rnorm(1, mean, sd)
calculate_probability_weighted(10, 20, 0, gen_norm, mean, 10000)

```

```
## [1] "Estimator: 2.56779282230561e-05 +- 2.28207739694599e-07"
```