

# Topic 2 - Univariate Simulation

## Assignment 1: Rejection Sampling

Isin Altinkaya

09.11.22

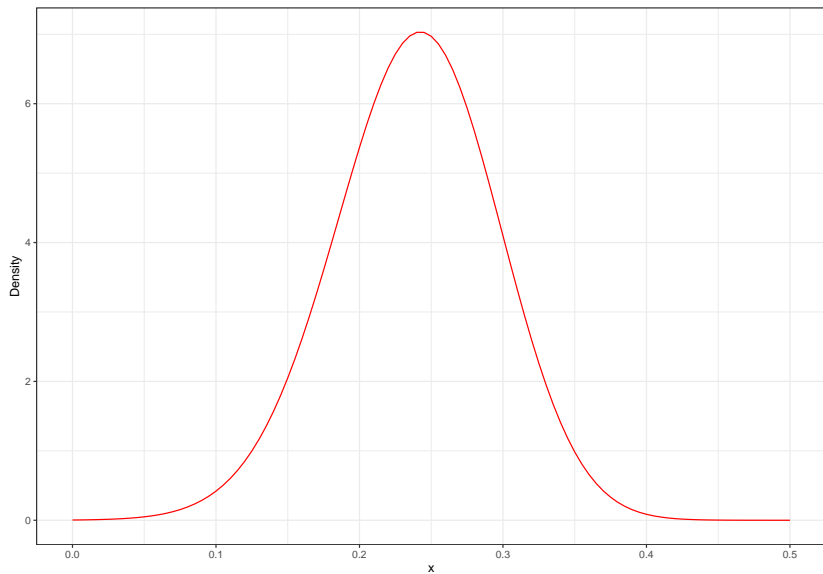
# Objective

- ▶ We want  $X \sim f(x)$  (target distribution)
- ▶ We know how to  $X \sim g(x)$  (proposal distribution, i.e. Gaussian envelope)
- ▶ Assume  $\frac{f(x)}{g(x)} \leq c, y \geq 0$ .

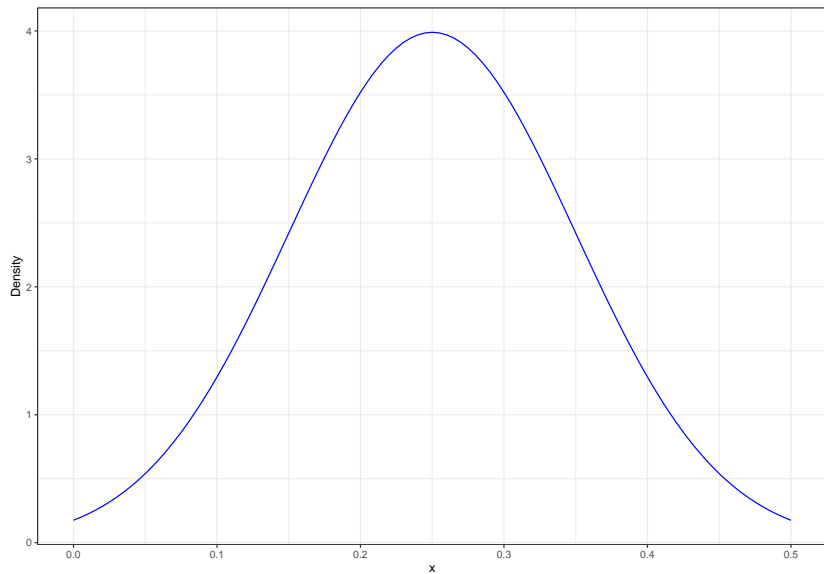
## Algorithm: Rejection sampling

1. Sample  $Y \sim g(y)$  (Proposal distribution)
2. Sample  $U \sim U[0, 1]$
3. If  $U \leq \frac{f(Y) \cdot \alpha}{g(Y)}$  return  $Y$ ; otherwise GOTO 1.

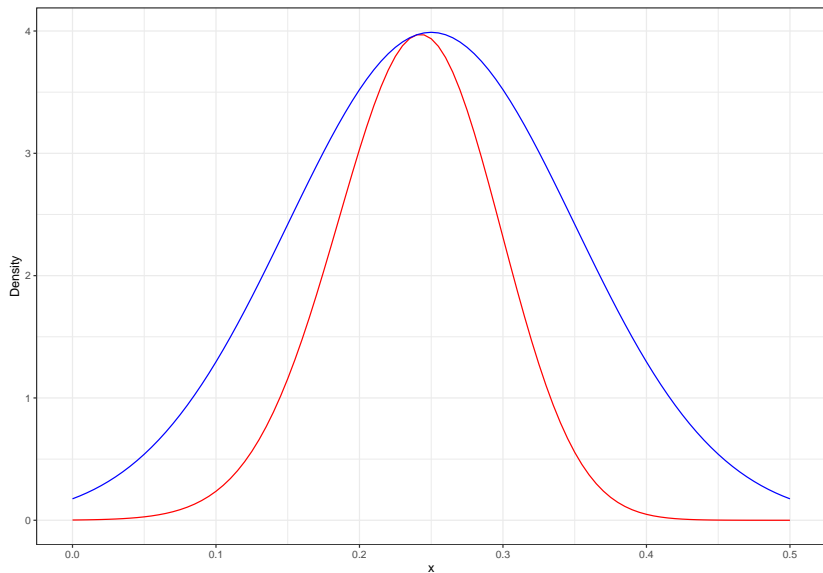
# Target



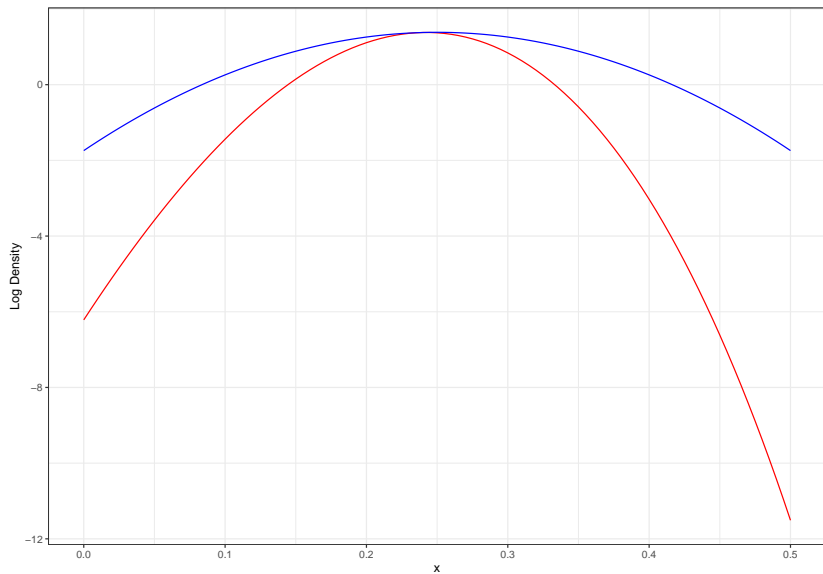
# Proposal



## Target proposal fit with optimized $\alpha$ parameter

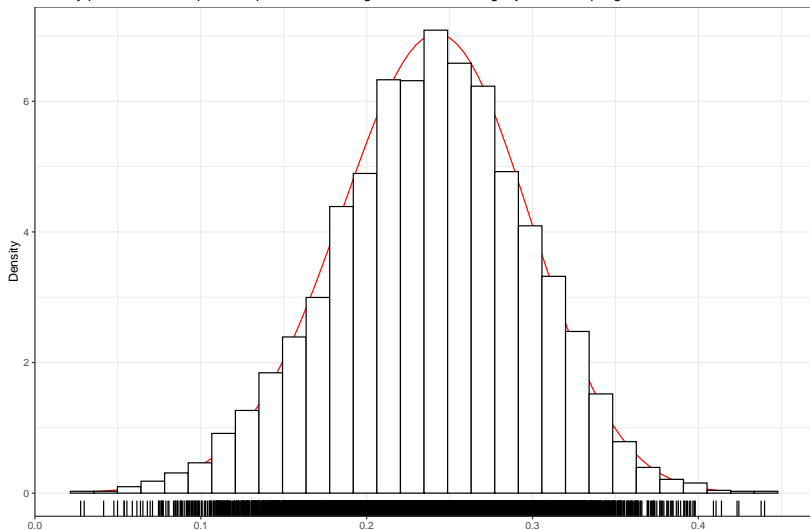


## Log scale



# Testing the accuracy of the rejection sampling implementation

Density plot of 5000 samples sampled from the target distribution using rejection sampling method





# Speed profiling using profvis

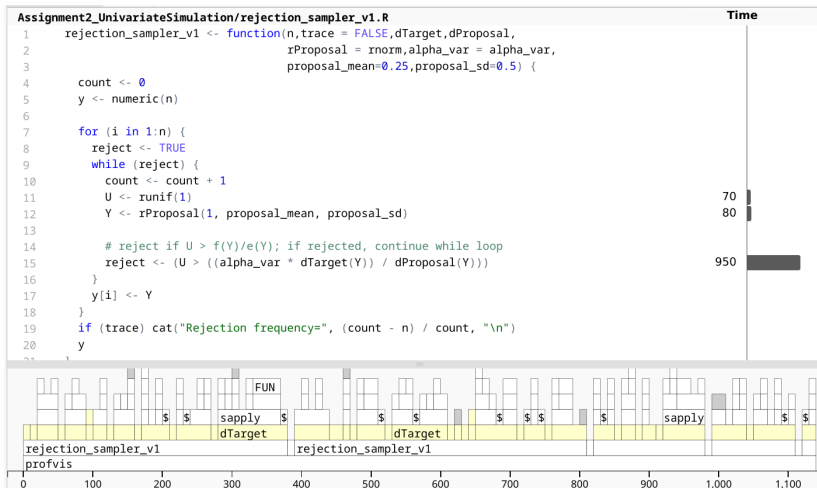
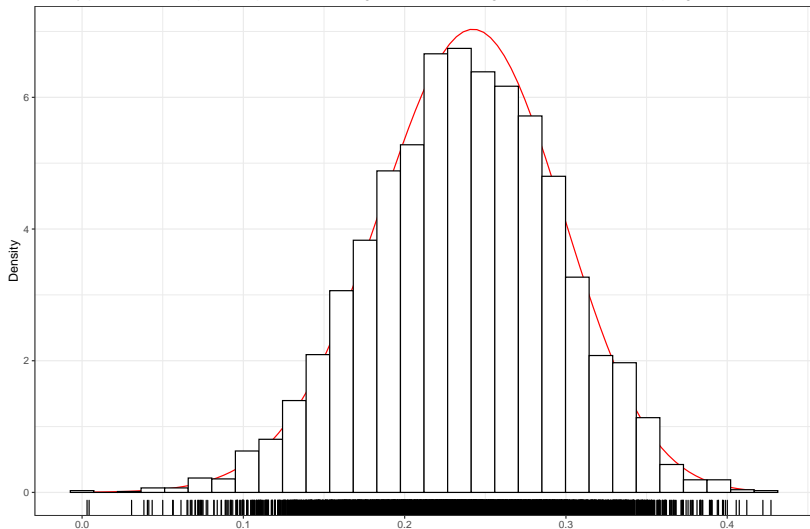


Figure 1: Speed profiling for `rejection_sampler_v1` using `profvis`

# Vectorized implementation of rejection sampling

Density plot of 5000 samples sampled from the target distribution using vectorized rejection sampling method



# Speed profiling using profvis

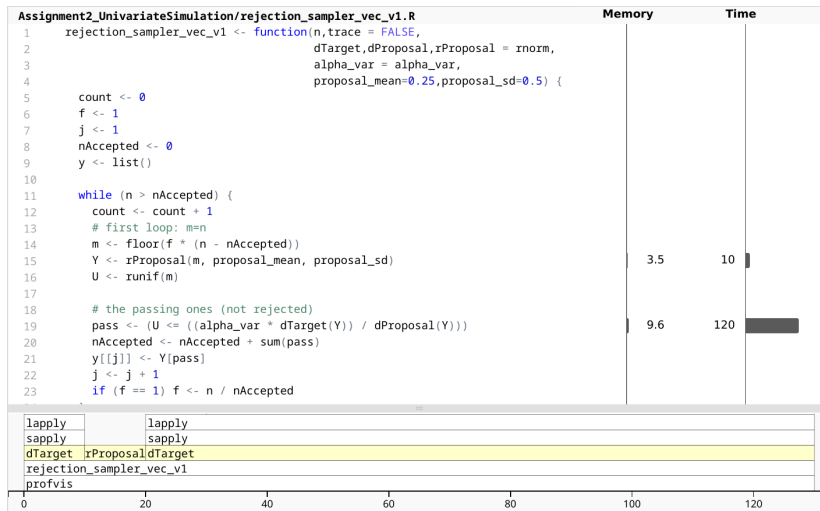
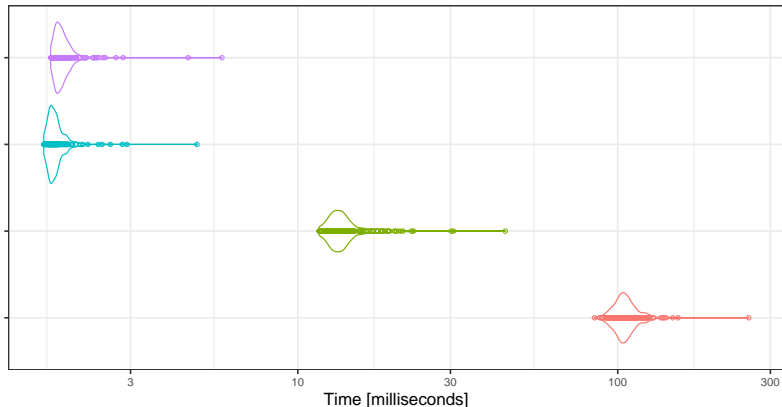


Figure 2: Speed profiling for rejection\_sampler\_vec\_v1 using profvis

# Benchmarking using microbenchmark

## Benchmarking results with 200 evaluations

- rejection\_sampler\_v1(n = 1000, dT = dTarget\_v1, dP = dProposal\_v2, alpha\_var = oa, pMean = 0.25, pSD = 0.2)
- rejection\_sampler\_vec\_v1(n = 1000, dT = dTarget\_v1, dP = dProposal\_v2, alpha\_var = oa, pMean = 0.25, pSD = 0.2)
- rejection\_sampler\_cpp\_v1(n = 1000, xis = pois\_data\$x, zis = pois\_data\$z, alpha\_var = oa, pMean = 0.25, pSD = 0.2)
- rejection\_sampler\_cpp\_v2(n = 1000, xis = pois\_data\$x, zis = pois\_data\$z, alpha\_var = oa, pMean = 0.25, pSD = 0.2)



## Algorithm: Adaptive rejection sampling

Given  $s_k$  squeezing function,  $e_k$  envelope function, grid  $T_k$ ,

1. Sample  $Y \sim g(y)$  (Proposal)
2. Sample  $U \sim U[0, 1]$
3. [Squeezing test] If  $U \leq \frac{s_k(Y)}{e_k(Y)}$  return  $Y$ ; otherwise GOTO 4
4. [Rejection test] If  $U \leq \frac{f(Y)}{e_k(Y)}$ ; accept  $Y$  and GOTO 5; otherwise GOTO 2
5. Insert  $Y$  into  $T_k$  to obtain  $T_{k+1}$ ; update  $s_k$  and  $e_k$

## Algorithm: Adaptive rejection sampling

Sample from the probability distribution on  $[0, \infty)$  with density

$$f(y) \propto \prod_{i=1}^{100} \exp(yz_i x_i - \exp(yx_i)), y \geq 0.$$

Log likelihood function

$$l(y) = \sum_{i=1}^{100} yz_i x_i - \exp(yx_i)$$

Derivative

$$l'(y) = \sum_{i=1}^{100} z_i x_i - x_i \exp(yx_i)$$

## Reducing the amount of computations by precomputing

$$x_i z_i$$

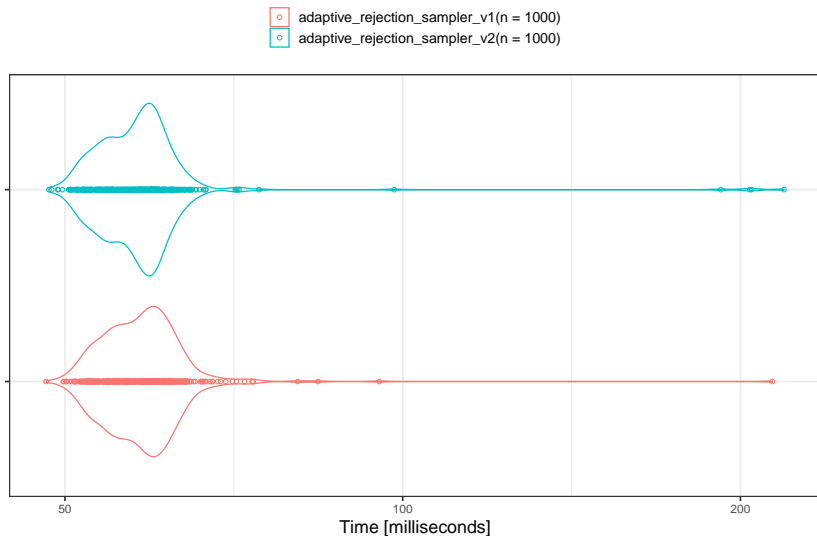
```
# adaptive_rejection_sampler_v1
lf <- function(x) sapply(x, function(x) sum(x *
  xis * zis - exp(x * xis)))
lf_deriv <- function(x) sapply(x, function(x) sum(xis *
  zis - xis * exp(x * xis)))

# adaptive_rejection_sampler_v2
xzis <- zis * xis
lf <- function(x) sapply(x, function(x) sum(x *
  xzis - exp(x * xis)))
lf_deriv <- function(x) sapply(x, function(x) sum(xzis -
  xis * exp(x * xis)))
```

# Reducing the amount of computations by precomputing

$x_i z_i$

Benchmarking results with 500 evaluations





# Comparing all

## Benchmarking results with 200 evaluations

