

For your code, the only random generator you can use is uniform (0,1) .

1. Normal distribution (50%)

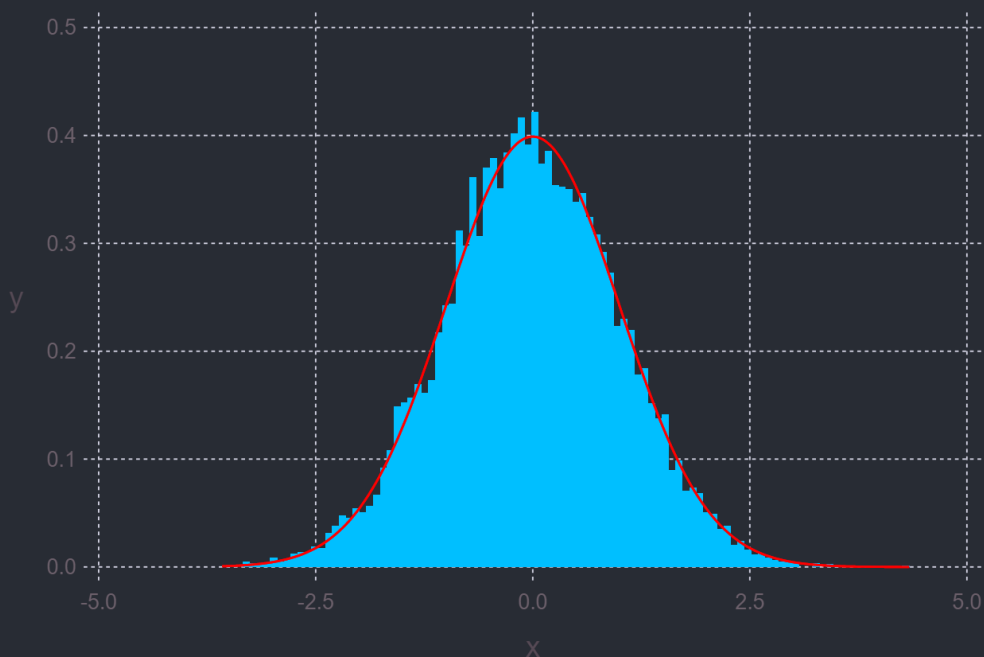
Write a program to generate normal distribution using **Box-Muller transformation**. Sample 10^4 data and draw the histogram to make sure you actually get a normal distribution.

```
function Box_Muller(n::Int)
    k = Int64(ceil(n/2))
    function f(r, θ)
        return [sqrt(r)*cos(θ), sqrt(r)*sin(θ)]
    end
    output = vcat(f.([-2log(u) for u in rand(k)], [u*2π for u in rand(k)])...)
    if isodd(n)
        return output[1:length(output)-1]
    else
        return output
    end
end

x = Box_Muller(10000)  # Vector{Float64} with 10000 elements
```

```
using Gadfly, DataFrames, Distributions

plot(DataFrame(x = x),
    layer(x -> pdf(Normal(0, 1), x), min(x...), max(x...), Geom.line, Theme(default_color="red")),
    layer(x = :x, Geom.histogram(bincount = 100, density = true))
)
```



2. Poisson Process (50%)

Write a program that uses the thinning algorithm to generate the first 10 time units of a nonhomogeneous Poisson process with intensity function

$$\lambda(t) = 3 + \frac{4}{t+1}$$

```
using Optim ☒
function NHPP(FUN::Function, start::Real, endT::Real)
    t, i = start, 0
    S = []
    λ = -minimum(optimize(x -> -FUN(x), start, endT))
    while true
        t += - 1/λ * log(rand(1)[1])
        if t > endT
            return S
        end
        if rand(1)[1] < FUN(t)/λ
            i += 1
            append!(S, t)
        end
    end
end ☒ > NHPP

print(NHPP(x -> 3 + 4/(x+1), 0, 10)) ☒
```

```
Any[0.1707611554721911, 0.20779393940298674, 0.24993118918546253, 0.3506181638946378, 0.37505376
336107615, 0.6549812336485789, 0.8974797125428274, 1.000953264584754, 1.024509025857148, 1.11000
73580255376, 1.1719654549104799, 1.3251202235992388, 1.442725134833686, 1.5768254985823862, 1.79
34017025974784, 2.3709866930244243, 2.395174159442776, 2.419861919535599, 2.750882213467985, 2.8
755826794865915, 3.310692195256405, 3.4324917952099128, 3.438292516719573, 3.597864652366532, 3.
788188324948758, 3.9052632927905937, 4.31504471009269, 5.097018009422954, 5.3283379576272765, 5.
50976275146895, 5.6779556652690335, 6.794438309383981, 6.8327460804177695, 6.9956159922764, 7.06
2115866015141, 7.962245095790614, 8.127974039128413, 8.199031898340309, 8.733073001533938, 9.024
128493276807, 9.280422953038771, 9.518108179953538]
```