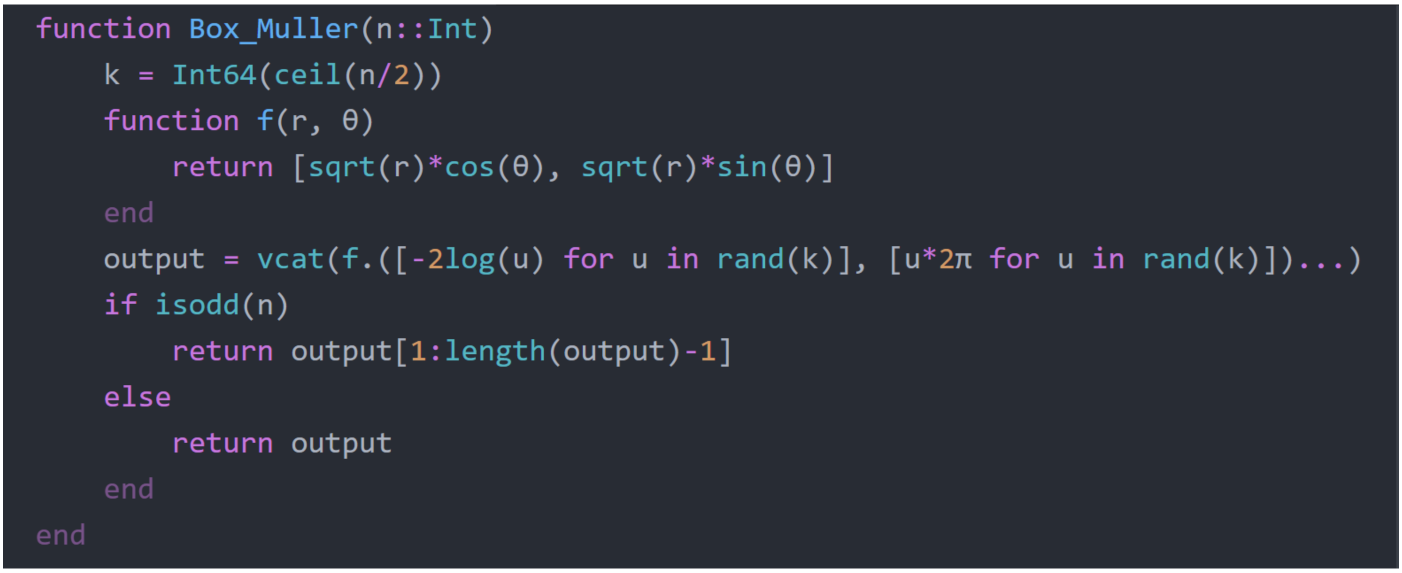
1. Normal distribution (50%)

Write a program to simulate a three-dimensional normal distribution with mean zero and

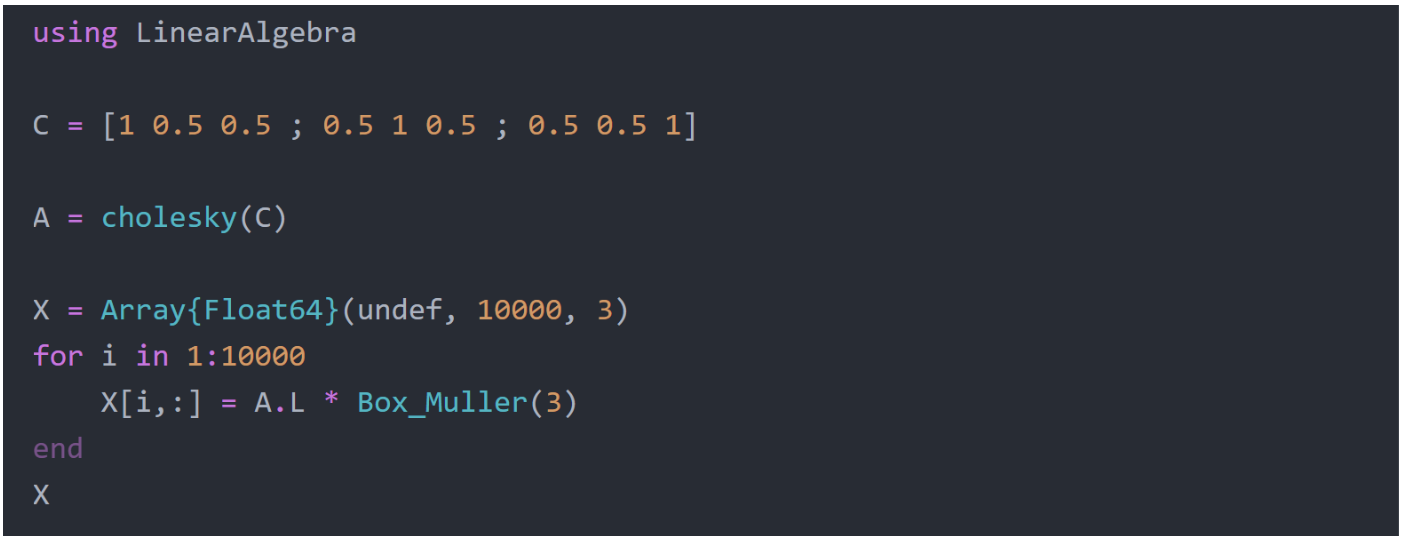
- Var(X_i) = 1
- Corr(X_i, X_j) = 0.5

Simulate 10^4 points. Compute the mean, variance and correlation of your sample.



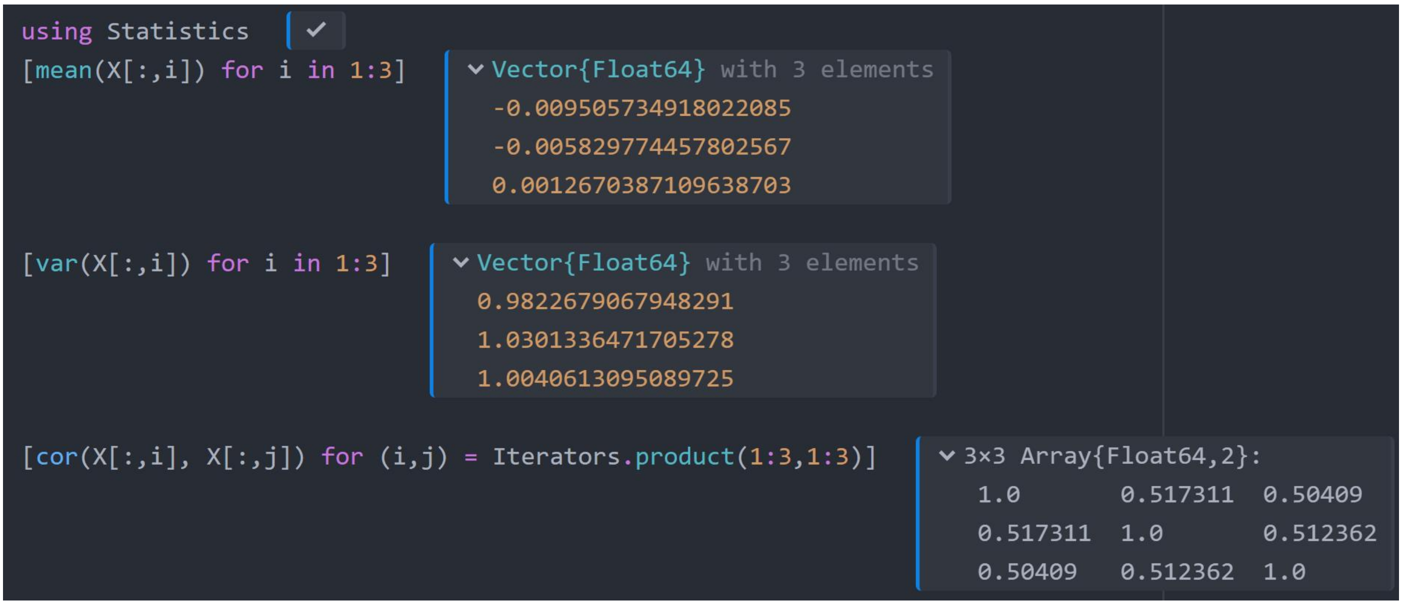
```
function Box_Muller(n::Int)
    k = Int64(ceil(n/2))
    function f(r, θ)
        return [sqrt(r)*cos(θ), sqrt(r)*sin(θ)]
    end
    output = vcat(f.([-2log(u) for u in rand(k)], [u*2π for u in rand(k)])...)
    if isodd(n)
        return output[1:length(output)-1]
    else
        return output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

借用上次作業 Box_Muller 的程式來生成 Normal(0,1)的隨機值。



```
using LinearAlgebra

C = [1 0.5 0.5 ; 0.5 1 0.5 ; 0.5 0.5 1]

A = cholesky(C)

X = Array{Float64}(undef, 10000, 3)
for i in 1:10000
    X[i,:] = A.L * Box_Muller(3)
end
X
```

生出 10^4 個 variance 為 1, 兩兩 correlation 為 0.5 的三維常態。

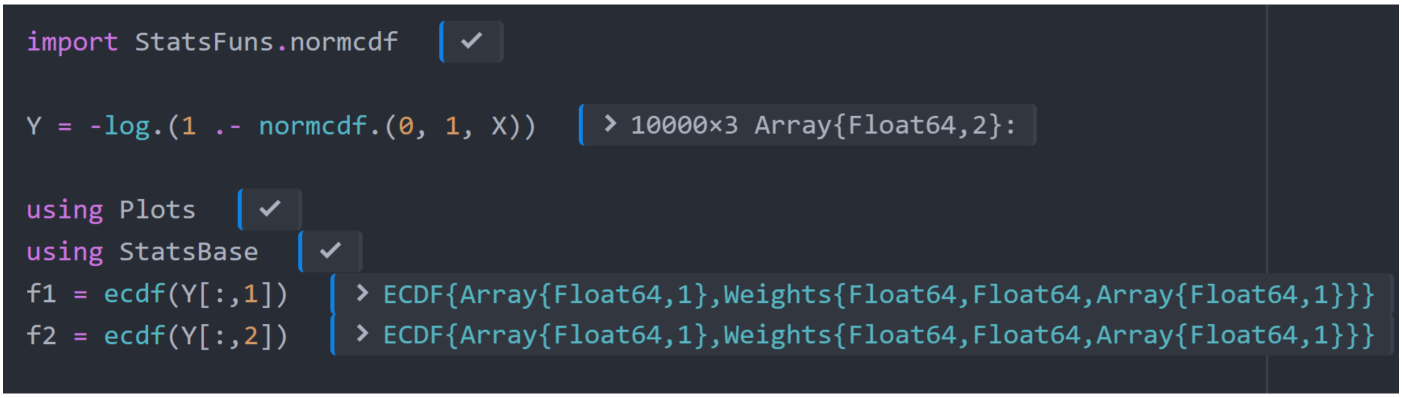


```
using Statistics                                      ✓
[mean(X[:,i]) for i in 1:3]                           ⌄ Vector{Float64} with 3 elements
                                                        -0.009505734918022085
                                                        -0.005829774457802567
                                                        0.0012670387109638703

[var(X[:,i]) for i in 1:3]                            ⌄ Vector{Float64} with 3 elements
                                                        0.9822679067948291
                                                        1.0301336471705278
                                                        1.0040613095089725

[cor(X[:,i], X[:,j]) for (i,j) = Iterators.product(1:3,1:3)]   ⌄ 3×3 Array{Float64,2}:
                                                                   1.0       0.517311  0.50409
                                                                   0.517311  1.0       0.512362
                                                                   0.50409   0.512362  1.0
```

求得模擬出的 mean, variance 還有 correlation。

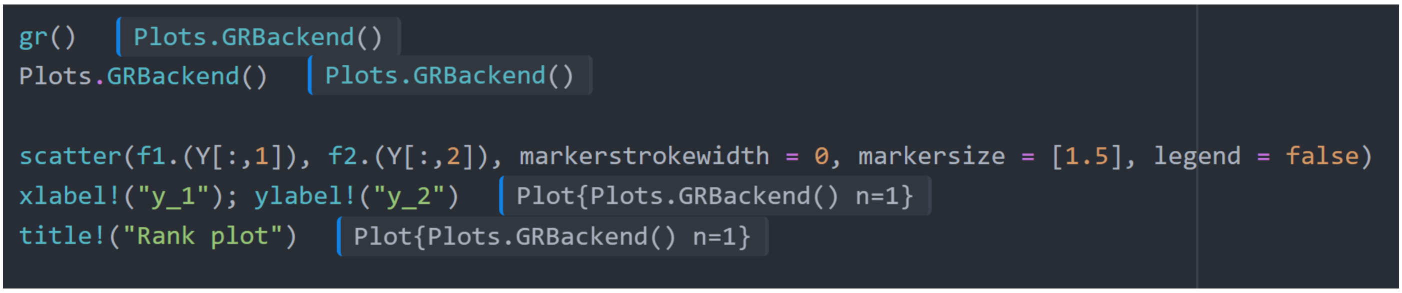
2. Copula (50%)

Now simulate a three dimensional (Y_1, Y_2, Y_3) so that Y_i is marginally exp(1), and have the copula as same as the X in part 1. Simulate 10^4 points. Plot the rank plot between Y_1 and Y_2 (as in page 6 in Week 4_2 file)

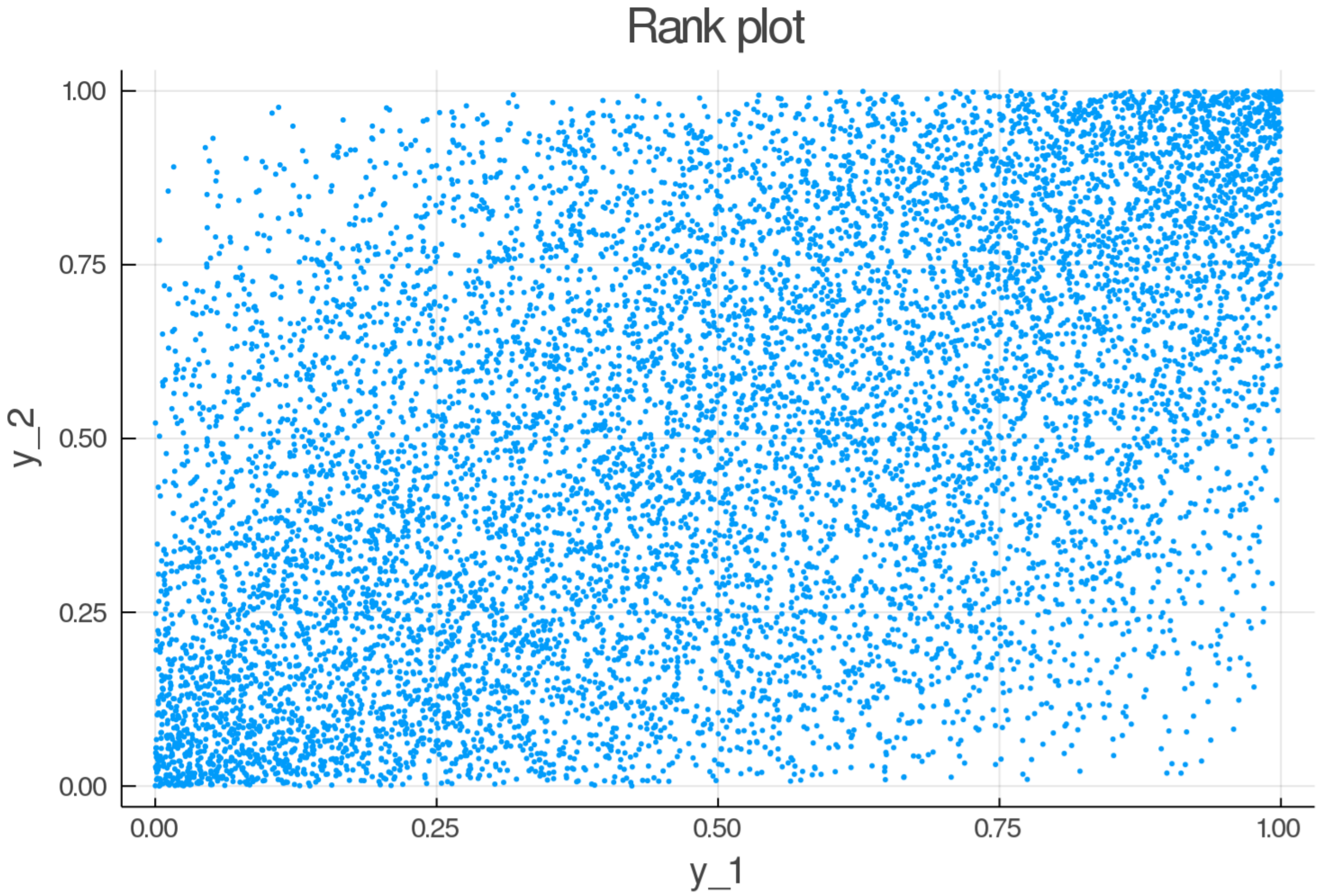


```
import StatsFuns.normcdf              ✓

Y = -log.(1 .- normcdf.(0, 1, X))     › 10000×3 Array{Float64,2}:

using Plots        ✓
using StatsBase    ✓
f1 = ecdf(Y[:,1])  › ECDF{Array{Float64,1},Weights{Float64,Float64,Array{Float64,1}}}
f2 = ecdf(Y[:,2])  › ECDF{Array{Float64,1},Weights{Float64,Float64,Array{Float64,1}}}
```

將各維度的值帶入 Standard Normal 的 cdf，再取 exp(1)的 cdf 反函數。
然後計算 Y_1, Y_2 各點的 empirical cdf。



```
gr()    Plots.GRBackend()
Plots.GRBackend()    Plots.GRBackend()

scatter(f1.(Y[:,1]), f2.(Y[:,2]), markerstrokewidth = 0, markersize = [1.5], legend = false)
xlabel!("y_1"); ylabel!("y_2")    Plot{Plots.GRBackend() n=1}
title!("Rank plot")    Plot{Plots.GRBackend() n=1}
```


Rank plot
y_2
y_1