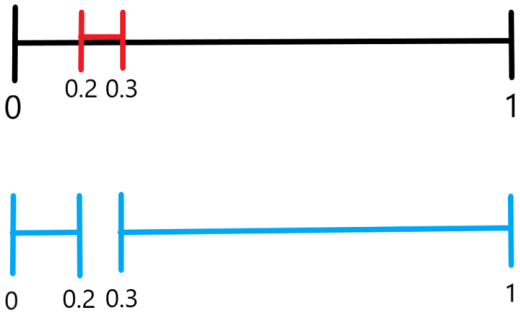
7. Give an efficient method for generating nine uniform points on (0, 1) conditional on the event than no two of them are within 0.1 of each other.

```
import Statistics.mean
import Random.shuffle!

function setd(a::Vector, b::Vector)
    i = 1
    while i <= length(a)
        if a[i][1] < b[1] < a[i][2]
            push!(a, [a[i][1], b[1]])
            push!(a, [b[1], a[i][2]])
        elseif a[i][1] < b[2] < a[i][2]
            push!(a, [a[i][1], b[2]])
            push!(a, [ b[2], a[i][2]])
        end
        i += 1
    end
    filter!(x -> !(b[1] <= x[1] < x[2] <= b[2]), a)
    filter!(x -> !(x[1] < b[1] < x[2]), a)
    filter!(x -> !(x[1] < b[2] < x[2]), a)
    return a
end
```

setd 函數用以計算區間的差集，例如 setd([[0, 1]], [0.2, 0.3])會等於[[0, 0.2], [0.3, 1]]

setd([[0, 1]], [0.2, 0.3])



= [[0, 0.2], [0.3, 1]]

```
function iunif(x::Vector)
    d = [a[2]-a[1] for a in x]
    v = rand(1)[1] * sum(d)
    i = findfirst(v .<= cumsum(d))
    return x[i][2] - cumsum(d)[i] + v
end
```

iunif 函數計算多個互斥實數區間之聯集上的 Uniform 隨機值

```
function p1(nsim = 1, k = 2e2, d = 0.1)
    output = Array{Array{Float64}}(undef, nsim)
    for i in 1:nsim
        y = shuffle!(collect(range(0, 1, length = 9)))
        for j in 1:k
            s = [[0., 1]]
            for r in setdiff(1:9, (j%9 + 1))
                s = setd(s, [y[r]-d, y[r]+d])
            end
            y[Int(j%9 + 1)] = iunif(s)
        end
        output[i] = y
    end
    return output
end
```

> p1

模擬 $0 < X_1, \dots, X_9 < 1$，在沒有任何兩點距離小於 0.1 的區域服從 9 維的 Uinform

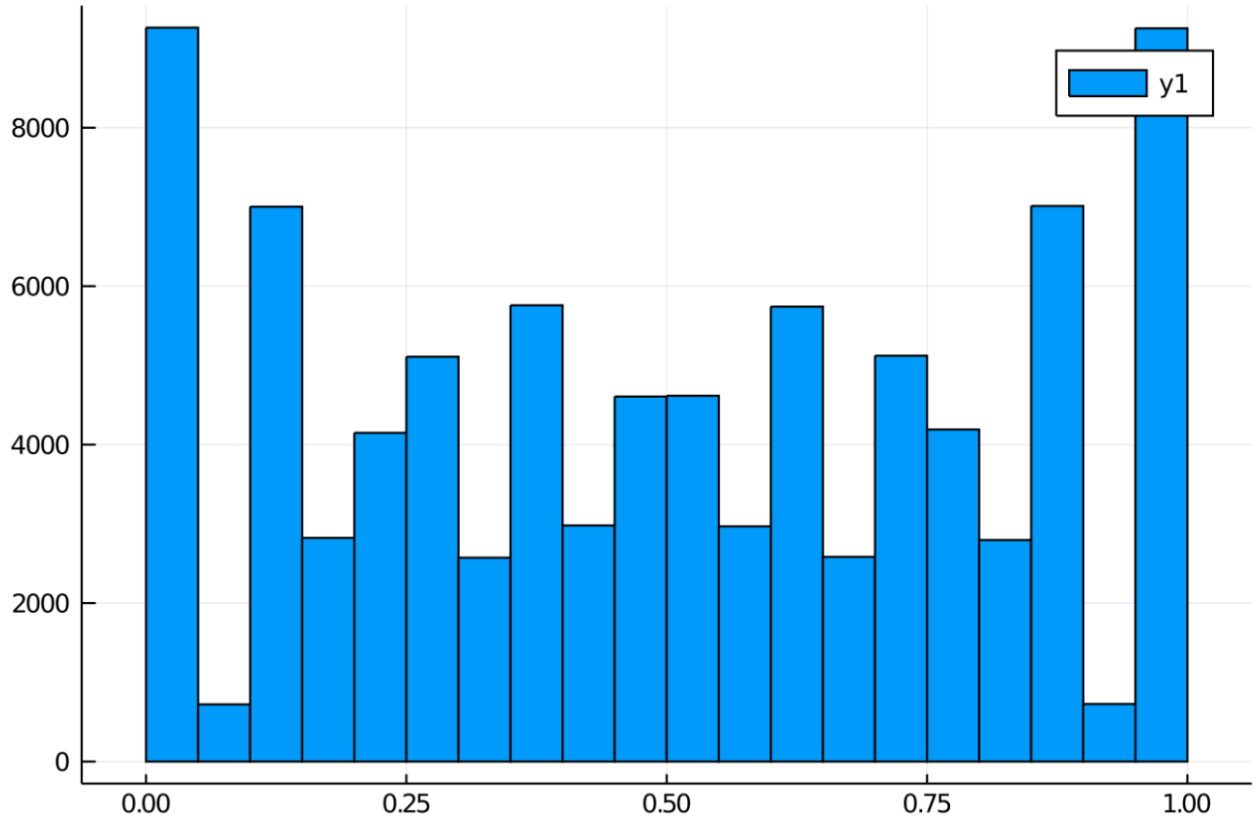
$X_1|X_2, \dots, X_9$服從在$[0,1]\setminus\bigcup_{i=2}^{9}[X_i - d, X_i + d]$上的 Uniform，如此類推，作為 Gibbs sampler 用Hastings − Metropolis algorithm 做 MCMC

```
p1()
```

```
Vector{Array{Float64,N} where N} with 1 element
  Vector{Float64} with 9 elements
    0.45371638089710153
    0.5906120904075525
    0.9814813022389108
    0.21401551048758338
    0.10322052161709068
    0.0009922415126070677
    0.8501529218391621
    0.32647471619959967
    0.7000887465041845
```

抽樣一次的結果。

```
using Plots ✔
histogram(vcat(p1(10000)...))    Plot{Plots.GRBackend() n=1}
```



抽 10000 組放在一起畫直方圖。

9. Let $X_i, i = 1, 2, 3$, be independent exponentials with mean 1. Run a simulation study to estimate

(a) $E[X_1 + 2X_2 + 3X_3 | X_1 + 2X_2 + 3X_3 > 15]$.
(b) $E[X_1 + 2X_2 + 3X_3 | X_1 + 2X_2 + 3X_3 < 1]$.

(a)

```
function p2a(nsim = 1, k = 2e2)
    output = Array{Array{Float64}}(undef, nsim)
    for i in 1:nsim
        x = [1.5, 2.5, 3.5]
        for j in 1:k
            if (j%3 + 1) == 1
                x[1] = -log(rand(1)[1]) + max(0, 15 - 2x[2] - 3x[3])
            elseif (j%3 + 1) == 2
                x[2] = -log(rand(1)[1]) + max(0, (15 - x[1] - 3x[3])/2)
            else
                x[3] = -log(rand(1)[1]) + max(0, (15 - x[1] - 2x[2])/3)
            end
        end
        output[i] = x
    end
    return output
end    > p2a

mean([x[1]+2x[2]+3x[3] for x in p2a(100000)])    18.091725186994065
```

平均值為 18.09

(b)

```
function p2b(nsim = 1, k = 2e2)
    output = Array{Array{Float64}}(undef, nsim)
    for i in 1:nsim
        x = [0.05, 0.1, 0.2]
        for j in 1:k
            if (j%3 + 1) == 1
                x[1] = -log(1-rand(1)[1]*(1-exp(-(1-2x[2]-3x[3]))))
            elseif (j%3 + 1) == 2
                x[2] = -log(1-rand(1)[1]*(1-exp(-((1-x[1]-3x[3])/2))))
            else
                x[3] = -log(1-rand(1)[1]*(1-exp(-((1-x[1]-2x[2])/3))))
            end
        end
        output[i] = x
    end
    return output
end    > p2b

mean([x[1]+2x[2]+3x[3] for x in p2b(100000)])    0.7267908764915858
```

平均值為 0.73