Now, here's a very serious virus spreading simulation here:

https://www.youtube.com/watch?v=gxAaO2rsdIs

Please write a program to simulate the "base" scenario on 2:45, and the scenario with "double infection radius" on 3:25.

- You DON'T need to do the animation. You only need to get the final graph on the upper-left part.
- There are a lot of parameters that you can decide by your own. The point is to show the effect when you double the infection radius.
- For this particular homework, you can use whatever package you get, like rnorm() or so.

函數中所使用的參數為

- population (人口/平方公里)
- radius (感染半徑)
- prob (感染機率)
- step (每日步行數)
- span (步長)
- remove (平均康復或死亡的時間)

模擬的假設情境為，在一平方公里的封閉區域內，所有人口依照均勻的速度每天 24 小時不休息地走了訂定的每日步行數。每一時刻(每步停留時間)，若有人待在病人的感染半徑內，則有一定機率會遭到感染，假設染病的病人沒有被隔離且正常活動，而受到感染者在一段時間後才會康復或死亡，亦即不再有感染風險。

參數的預設值為，參考新竹市人口資料平均一平方公里有 4309 人，台人平均每日步行數為 5000 次，平均步長為 0.7 公尺，在模擬中指的是每$24 * 60 * 60/5000 = 17.28$ 秒走一步，在這 17.28 秒內若距離病人 1.5 公尺內將有 1%受到感染的機率，而染病的平均康復或死亡時間為 6 天

```
using Plots

function ep(;population::Int=4309, radius::Float64=1.5/1000, prob::Float64=0.01,
            step::Int=5000, span::Float64=0.7/1000, remove::Int=step*6)
    t = 0
    pS, pI, pR = [1-1/population], [1/population], [0.0]
    x = rand(population)
    y = rand(population)
    angle = rand(population)*2π
    condition = repeat([1], population)
    condition[rand(1:population)] = 2
    rm_time = repeat([Inf], population)
    rm_time[condition.==2] .= rand((remove-step):(remove+step))[1]
    while pI[length(pI)] != 0
        t += 1
        # 更新康復或死亡的人口狀態
        if any(rm_time .== t)
            condition[(1:population)[rm_time .== t]] .= 3
        end
```

pS, pI, pR 記錄各時間點 S,I,R 的比例

x,y 是每個人在時間 t 的位置

condition 紀錄染病狀態(S = 1, I = 2, R = 3)

while 迴圈一直跑直到病毒根除

康復或死亡所需天數的隨機變數X，在預設參數下$5000 * X$ 服從$Discrete\ \ Uniform(25000,35000)$

```
        for i in (1:population)[condition .!= 3]
            angle[i] += rand(1)[1]*π/2 - π/4
            while !(0 < x[i]+span*cos(angle[i]) < 1 || 0 < y[i]+span*sin(angle[i]) < 1)
                angle[i] = rand(1)[1]*2π
            end
            x[i] = x[i] + span*cos(angle[i])
            y[i] = y[i] + span*sin(angle[i])
        end
        # 染病
        for i in (1:population)[condition .== 2]
            for j in (1:population)[condition .== 1]
                if (x[i]-x[j])^2 + (y[i]-y[j])^2 < radius^2
                    if rand(1)[1] < prob
                        condition[j] = 2
                        rm_time[j] = t + rand((remove-step):(remove+step))[1]
                    end
                end
            end
        end
```

為了不使人走起來像個醉漢，所以每個人走的下一步的方向會是上一步行走方向再加一個

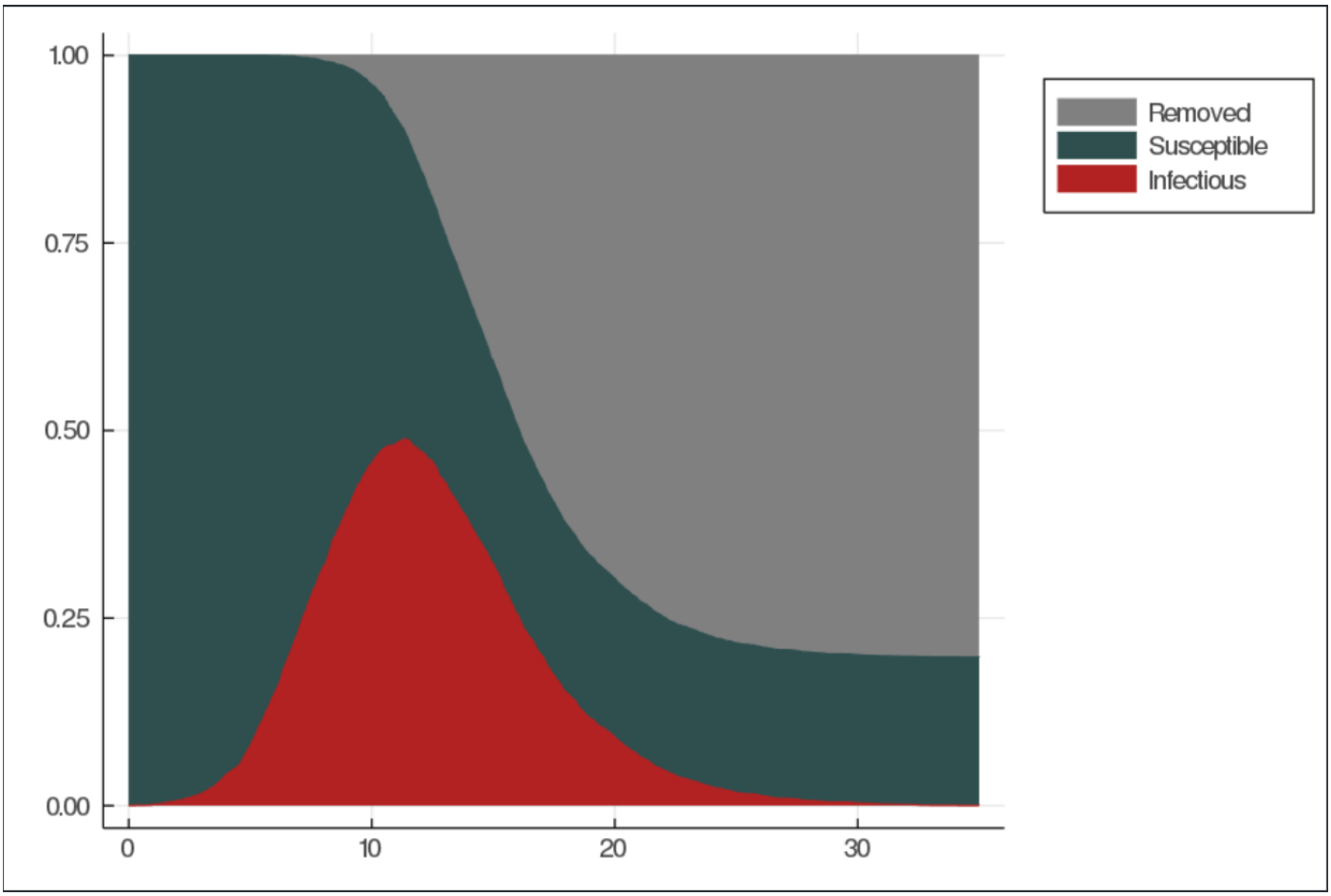
$Uniform\left(-\frac{\pi}{4},\frac{\pi}{4}\right)$的隨機值，除非會撞到邊界時才會選擇其他方向。

```
        append!(pS, sum(condition.==1)/population)
        append!(pI, sum(condition.==2)/population)
        append!(pR, sum(condition.==3)/population)
    end
    gr()
    Plots.GRBackend()
    plot((0:t)/step, [repeat([1], t+1) 1 .- collect(pR) collect(pI)], fill = 0,
        color = permutedims(["Gray", "DarkSlateGray", "Firebrick"]), legend = :outertopright,
        label = ["Removed" "Susceptible" "Infectious"])
end  > ep
```

紀錄尚未被感染、已染病、康復或死亡的比例。
當 while 迴圈跑完即開始作圖。

```
ep(radius = 1.5/1000)  Plot{Plots.GRBackend() n=3}
```

感染半徑為 1.5/1000 公里的模擬，其他參數維持預設。



```
ep(radius = 3.0/1000)    Plot{Plots.GRBackend() n=3}
```

感染半徑為 3/1000 公里的模擬，其他參數維持預設。