Happy holiday! Remember to take care of yourself and your loved ones!

```
v =  [1.0]
```

```
v = [1.0]
```

```
[1.0, 7.0]
```

```
push!(v, 7.0)
```

```
run_infection (generic function with 2 methods)
```

```
function run_infection(I_0, λ, T=20)  # T=20 is default value

    Is = [I_0]
    I = I_0   # current value of I

    for n in 1:T-1
        I_next = λ * I

        push!(Is, I_next)

        I = I_next
    end

    return Is
end
```

```
[1.0, 1.1, 1.21, 1.331, 1.4641, 1.61051, 1.77156, 1.94872, 2.14359, 2.35795, 2.59374,
```

```
run_infection(1.0, 1.1)
```

```
bernoulli (generic function with 1 method)
```

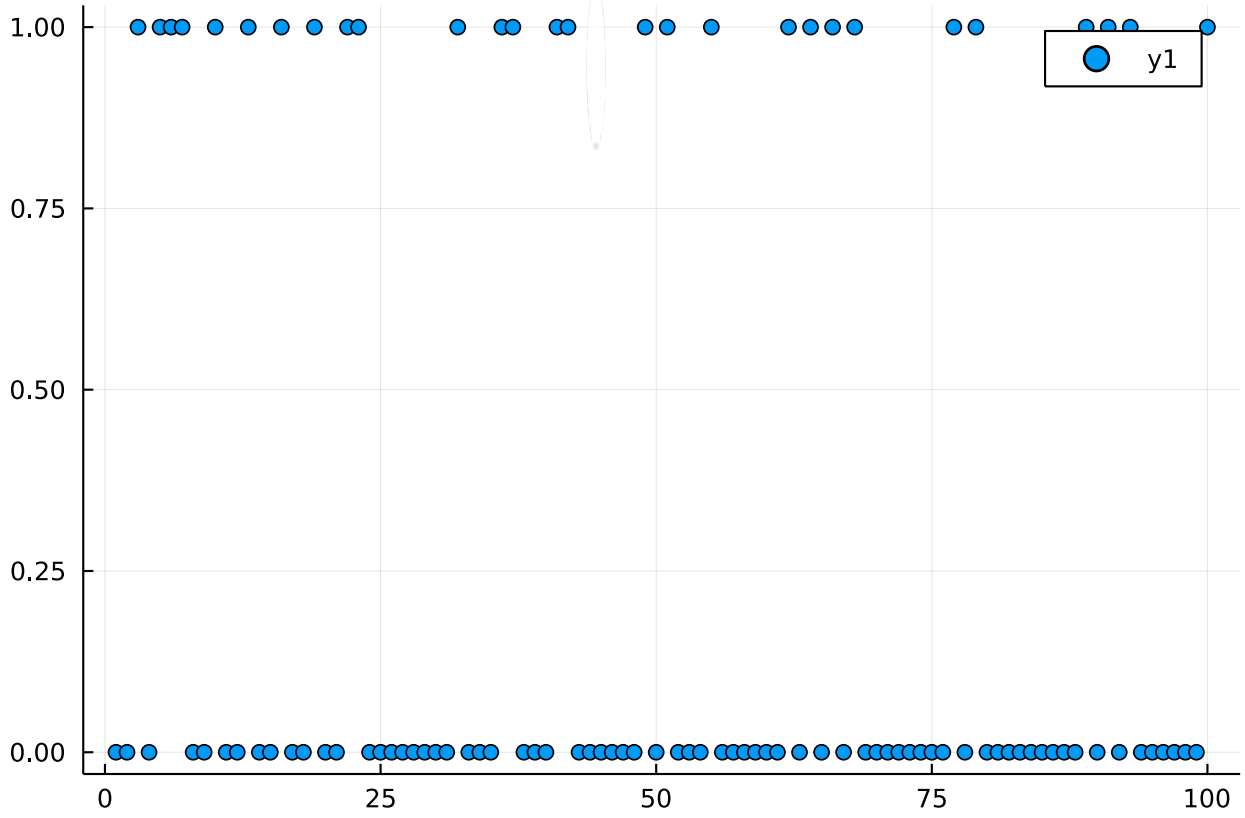
```
function bernoulli(p)

    r = rand()

    if r < p
        return true
    else
        return false
    end
end
```

```
p = 0.25
```

```
p = 0.25
```

```
trials = [bernoulli(p) for i in 1:100];
```

```
using Plots
```

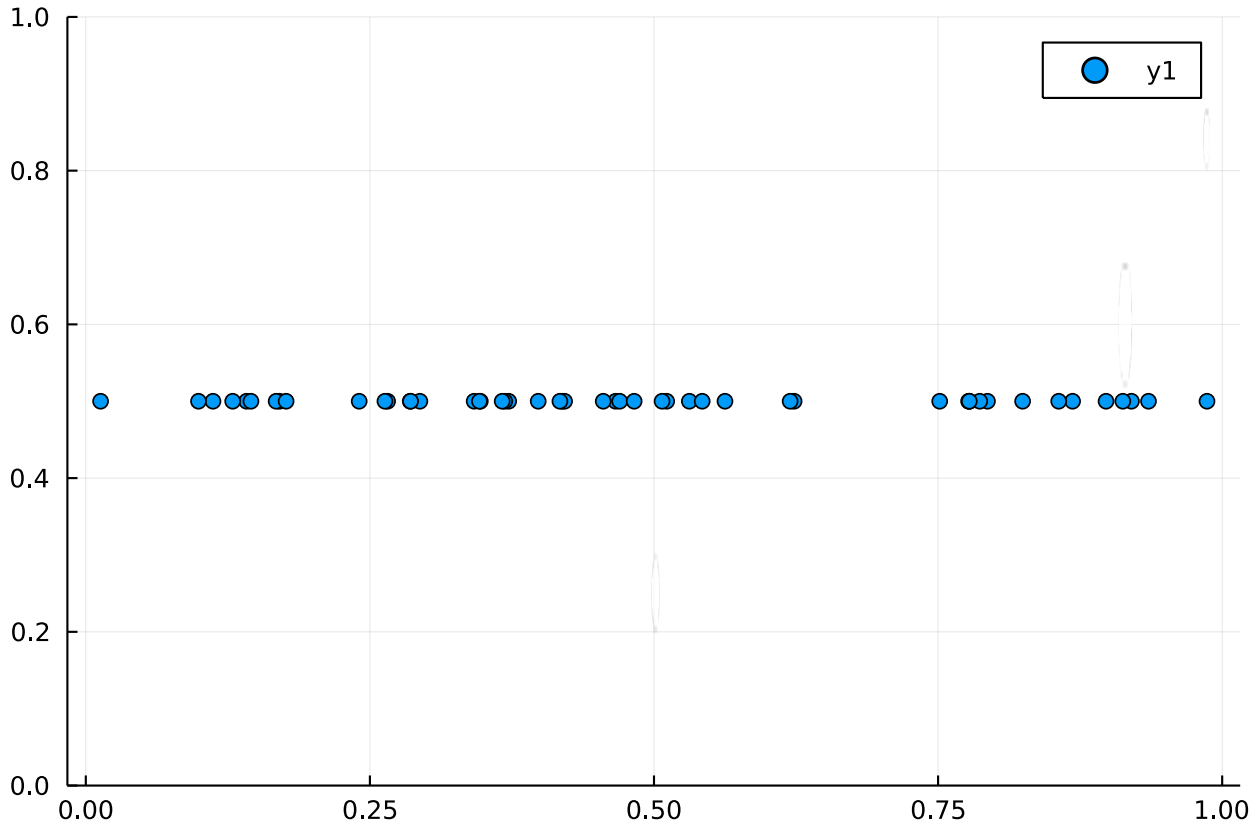



```
scatter(trials)
```

```
1.0
```

```
one(0.5)
```

```
r = rand(50);
```




```
scatter(r, 0.5 .* one.(r), ylim=(0, 1))
```

```
using Interact
```

```
num_points = 100
```

```
num_points = 100
```

```
random =
 [0.842289, 0.268025, 0.766607, 0.416435, 0.128462, 0.395565, 0.0299531, 0.71491, 0.5
```

```
random = rand(num_points)
```

```
using WebIO
```

```
@manipulate for n in 1:num_points
    scatter(random[1:n], 0.5 .* one.(random[1:n]), ylim=(0, 1), xlim=(0, 1))
end
```

```
bernoulli_experiment (generic function with 2 methods)
```

```
function bernoulli_experiment(p, N=100)
    trials = [bernoulli(p) for i in 1:N];
    return count(trials)
end
```

```
100
```

```
count(trials .== false) + count(trials)
```

```
72
```

```
count(.!(trials))
```

```
20
```

```
bernoulli_experiment(0.25)
```

```
25
```

```
bernoulli_experiment(0.25)
```

```
N = 20
```

```
N = 20   # num of trials
```

```
num_expts = 100
```

```
num_expts = 100
```

```
results =
  [5, 5, 3, 5, 7, 4, 9, 4, 4, 7, 5, 4, 5, 3, 6, 4, 6, 6, 7, 7,    more ,5, 3, 5, 4, 3, 10,
```

```
results = [bernoulli_experiment(p, N) for i in 1:num_expts]
```

```
counts =   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
counts = zeros(Int, maximum(results) + 1)
```

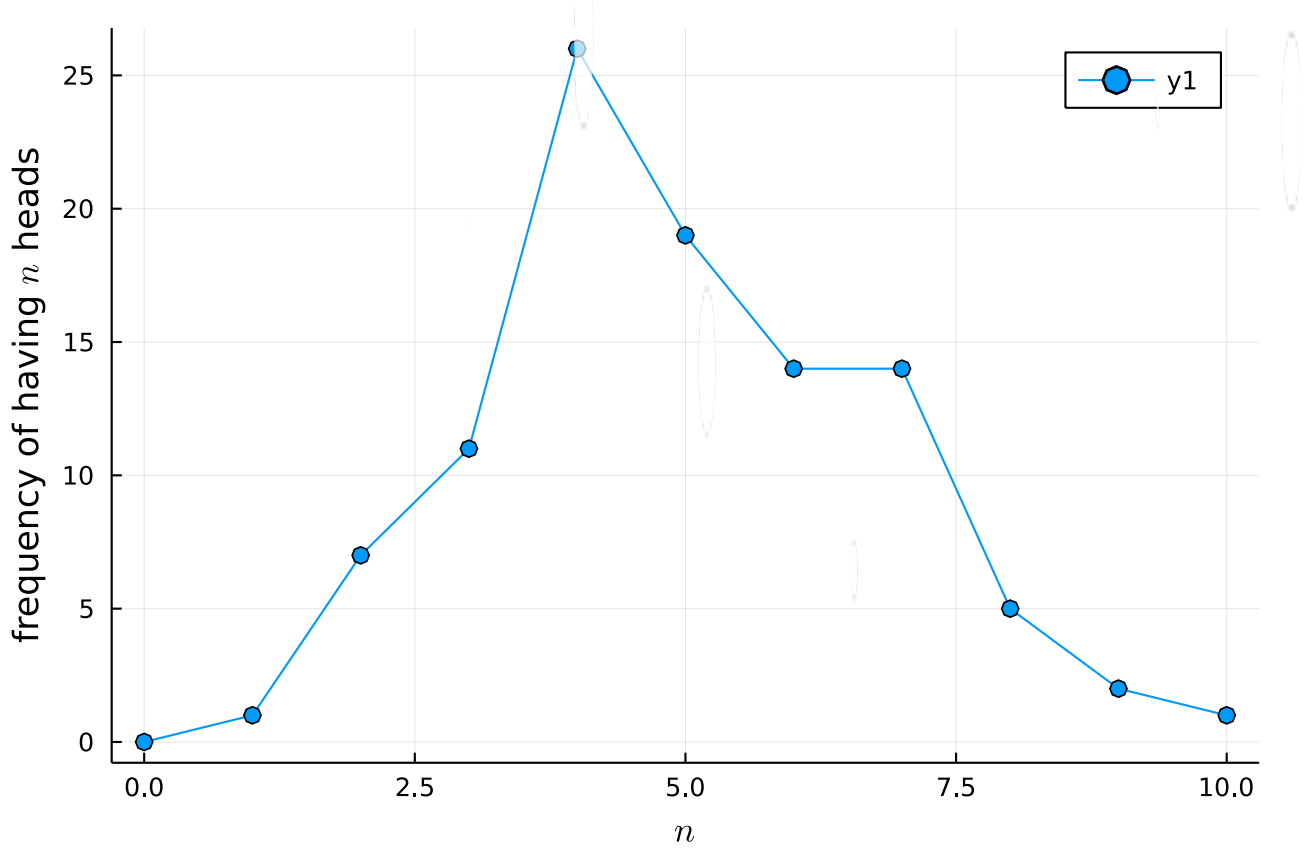
```
for score in results[1:10]   # for i in 1:length(results)
    @show score
end
```

```
for score in results
    counts[score + 1] += 1   # increment by 1
end
```

```
using LaTeXStrings
```

```
plot(0:maximum(results), counts, m=:o);
```

```
ylabel!(L"frequency of having $n$ heads");
```




```
xlabel!(L"n")
```

count_them (generic function with 1 method)

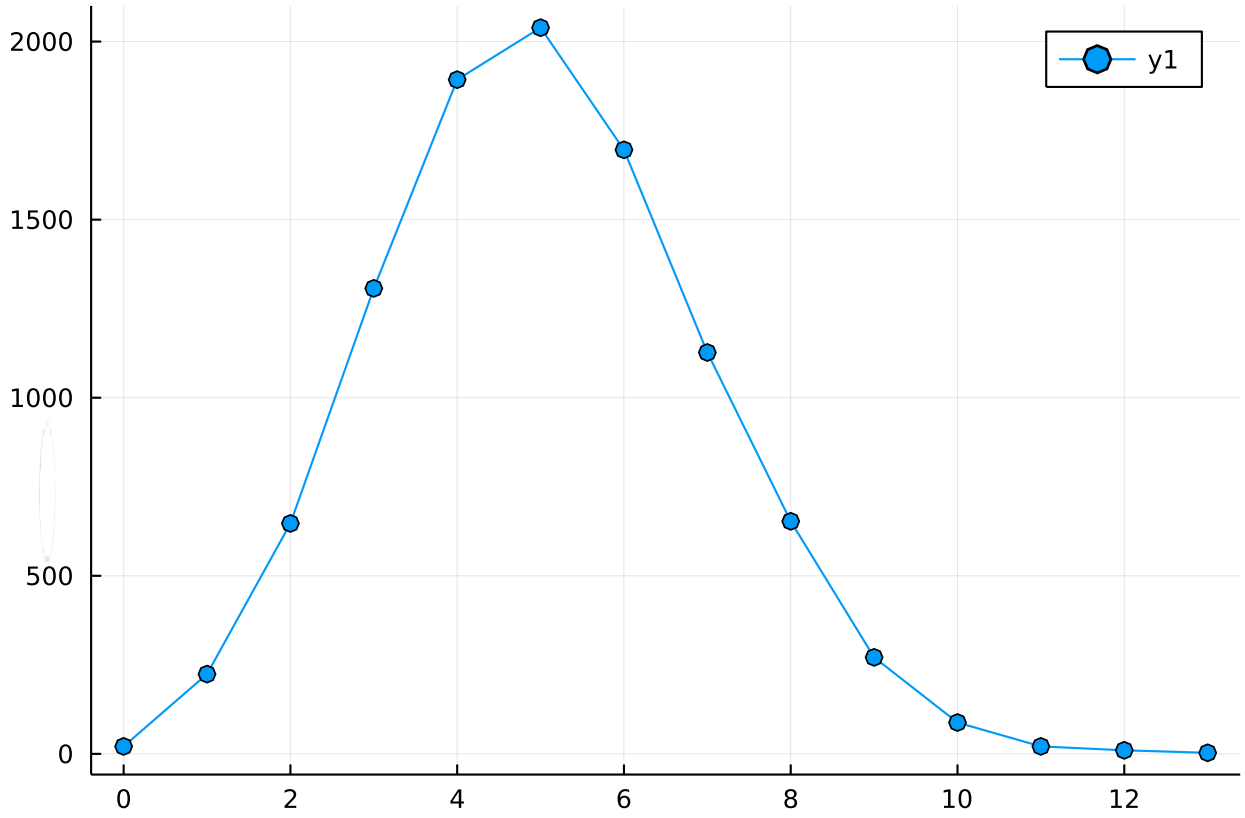
```
function count_them(results)

    counts = zeros(Int, maximum(results) + 1)

    for score in results
        counts[score + 1] += 1    # increment by 1
    end

    return counts
end
```

run_experiments (generic function with 2 methods)

```
function run_experiments(p, N, num_expts=1000)
    results = [bernoulli_experiment(p, N) for i in 1:num_expts]
end
```

```
data =   [21, 224, 647, 1307, 1893, 2039, 1696, 1127, 653, 271, 88, 21, 10, 3]
```

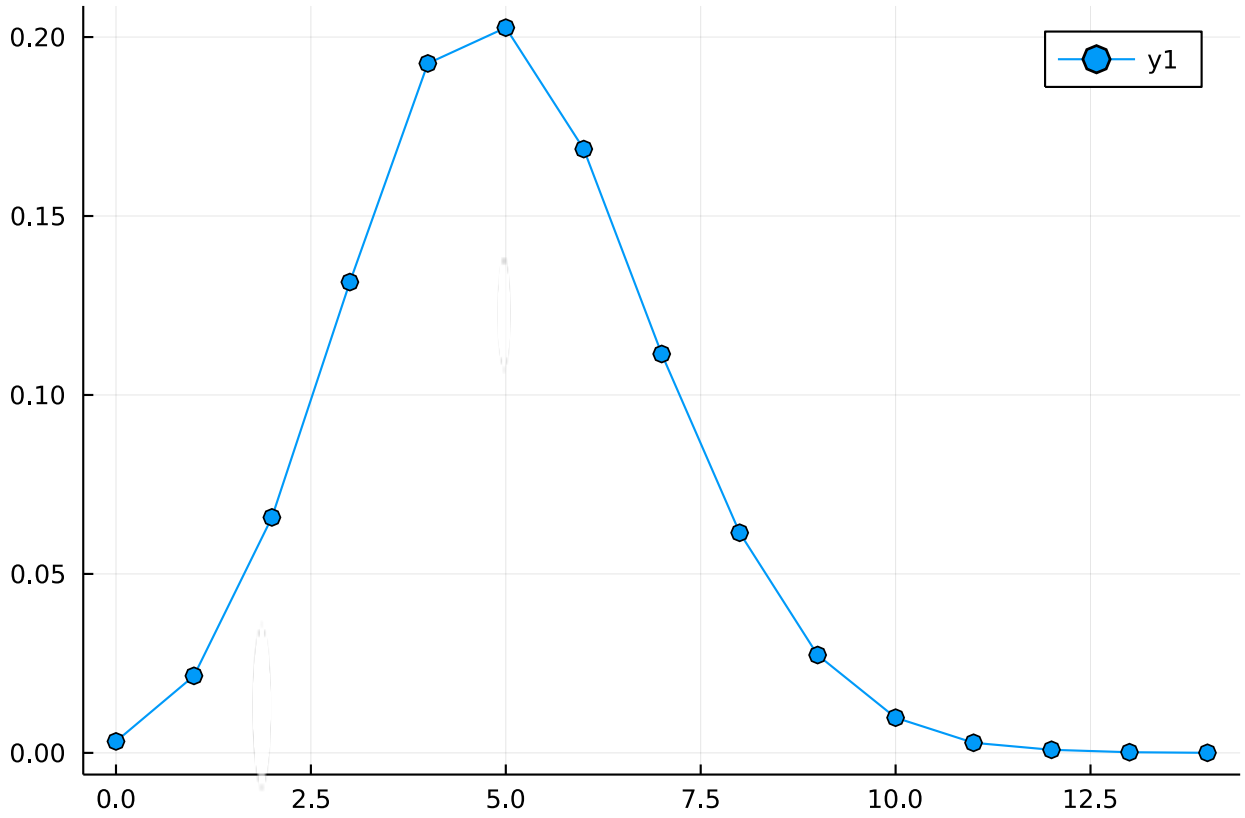
```
data = count_them(run_experiments(0.25, 20, 10000))
```




```
plot(0:length(data)-1, data, m=:o)
```

```
data1 =
  [320, 2152, 6579, 13154, 19263, 20262, 16870, 11145, 6150, 2734, 983, 282, 86, 18, 2]
```

```
data1 = count_them(run_experiments(0.25, 20, 10^5))
```




```
plot(0:length(data1)-1, data1 ./ sum(data1), m=:o)
```

```
@time data2 =
  [336, 2092, 6631, 13417, 18879, 20366, 16818, 11153, 6097, 2800, 1010, 311, 75, 13, 2
```

```
@time data2 = count_them(run_experiments(0.25, 20, 10^5))
```

```
@time data3 =
  [3131, 21074, 66870, 134375, 189776, 201909, 168704, 112457, 60738, 26971, 9955, 3045
```

```
@time data3 = count_them(run_experiments(0.25, 20, 10^6))
```

```
@time data4 =
  [31746, 211301, 668480, 1339050, 1896207, 2022789, 1687435, 1123660, 608381, 271961,
```

```
@time data4 = count_them(run_experiments(0.25, 20, 10^7))
```

```
data5 =  [0, 0, 8, 14, 21, 19, 23, 7, 6, 2]
```

```
data5 = count_them(run_experiments(0.25, 20, num_expts))
```

```
probs =  [0.0, 0.0, 0.08, 0.14, 0.21, 0.19, 0.23, 0.07, 0.06, 0.02]
```

```
probs = data5 ./ num_expts
```

```
1.0000000000000002
```

```
sum(probs)
```

```
probs1 =
  [0//1, 0//1, 2//25, 7//50, 21//100, 19//100, 23//100, 7//100, 3//50, 1//50]
```

```
probs1 = data5 .// num_expts
```

```
1//1
```

```
sum(probs1)
```

```
using Statistics
```

```
results1 = run_experiments(0.25, 20, 10^5);
```

```
4.99798
```

```
mean(results1)
```