

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

- `using Pkg`

- `using Interact`

```
url =  
"https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data,
```

- `url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv"`

```
"covid_data.csv"
```

- `download(url, "covid_data.csv")`

- `using CSV`

- `using DataFrames`

	Province/State	Country/Region	Lat	Long	1/2
1	missing	"Afghanistan"	33.9391	67.71	0
2	missing	"Albania"	41.1533	20.1683	0
3	missing	"Algeria"	28.0339	1.6596	0
4	missing	"Andorra"	42.5063	1.5218	0
5	missing	"Angola"	-11.2027	17.8739	0
6	missing	"Antigua and Barbuda"	17.0608	-61.7964	0
7	missing	"Argentina"	-38.4161	-63.6167	0
8	missing	"Armenia"	40.0691	45.0382	0
9	"Australian Capital Territory"	"Australia"	-35.4735	149.012	0
10	"New South Wales"	"Australia"	-33.8688	151.209	0
	more				
280	missing	"Zimbabwe"	-19.0154	29.1549	0

```
• CSV.read("covid_data.csv", DataFrame)
```

```
• data = CSV.read("covid_data.csv", DataFrame);
```

	Province/State	Country/Region	Lat	Long	1/2
1	missing	"Afghanistan"	33.9391	67.71	0
2	missing	"Albania"	41.1533	20.1683	0
3	missing	"Algeria"	28.0339	1.6596	0
4	missing	"Andorra"	42.5063	1.5218	0
5	missing	"Angola"	-11.2027	17.8739	0
6	missing	"Antigua and Barbuda"	17.0608	-61.7964	0
7	missing	"Argentina"	-38.4161	-63.6167	0
8	missing	"Armenia"	40.0691	45.0382	0
9	"Australian Capital Territory"	"Australia"	-35.4735	149.012	0
10	"New South Wales"	"Australia"	-33.8688	151.209	0
more					
280	missing	"Zimbabwe"	-19.0154	29.1549	0

- `data`

```
countries =
```

```
["Afghanistan", "Albania", "Algeria", "Andorra", "Angola", "Antigua and Barbuda", "A
```

- `countries = collect(data[:,2])`

```
unique_countries =
```

```
["Afghanistan", "Albania", "Algeria", "Andorra", "Angola", "Antigua and Barbuda", "A
```

- `unique_countries = unique(countries)`

- `using WebIO`

```
(::Base.Fix2{typeof(startswith), String}) (generic function with 1 method)
```

- `startswith("I")`

- `I_countries = [startswith(country, "I") for country in countries];`

1	missing	"Iceland"	64.9631	-19.0208	0	0	0	0
2	missing	"India"	20.5937	78.9629	0	0	0	0
3	missing	"Indonesia"	-0.7893	113.921	0	0	0	0
4	missing	"Iran"	32.4279	53.688	0	0	0	0
5	missing	"Iraq"	33.2232	43.6793	0	0	0	0
6	missing	"Ireland"	53.1424	-7.6921	0	0	0	0
7	missing	"Israel"	31.0461	34.8516	0	0	0	0
8	missing	"Italy"	41.8719	12.5674	0	0	0	0

- `data[I_countries, :]`

```
BitVector: [false, false, false, false, false, false, false, false, false, false, fa
```

- `countries` `==` "India"

IND\_row = 148

```
IND_row = findfirst(countries .=="India")
```

```
IND_data_row =
```

DataFrameRow (719 columns)

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20
	String63	String63	Float64?	Float64?	Int64	Int64	Int64	Int64
148	missing	India	20.5937	78.9629	0	0	0	0

```
IND_data_row = data[IND_row, :]
```

IND\_data =

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, more ,34799691, 3480888
```

```
IND_data = collect(IND_data_row[5:end])
```

- using Plots

- `plot(IND_data);`

```
date_strings =
```

```
["1/22/20", "1/23/20", "1/24/20", "1/25/20", "1/26/20", "1/27/20", "1/28/20", "1/29/
```

```
• date_strings = String.(names(data))[5:end]
```

```
• using Dates
```

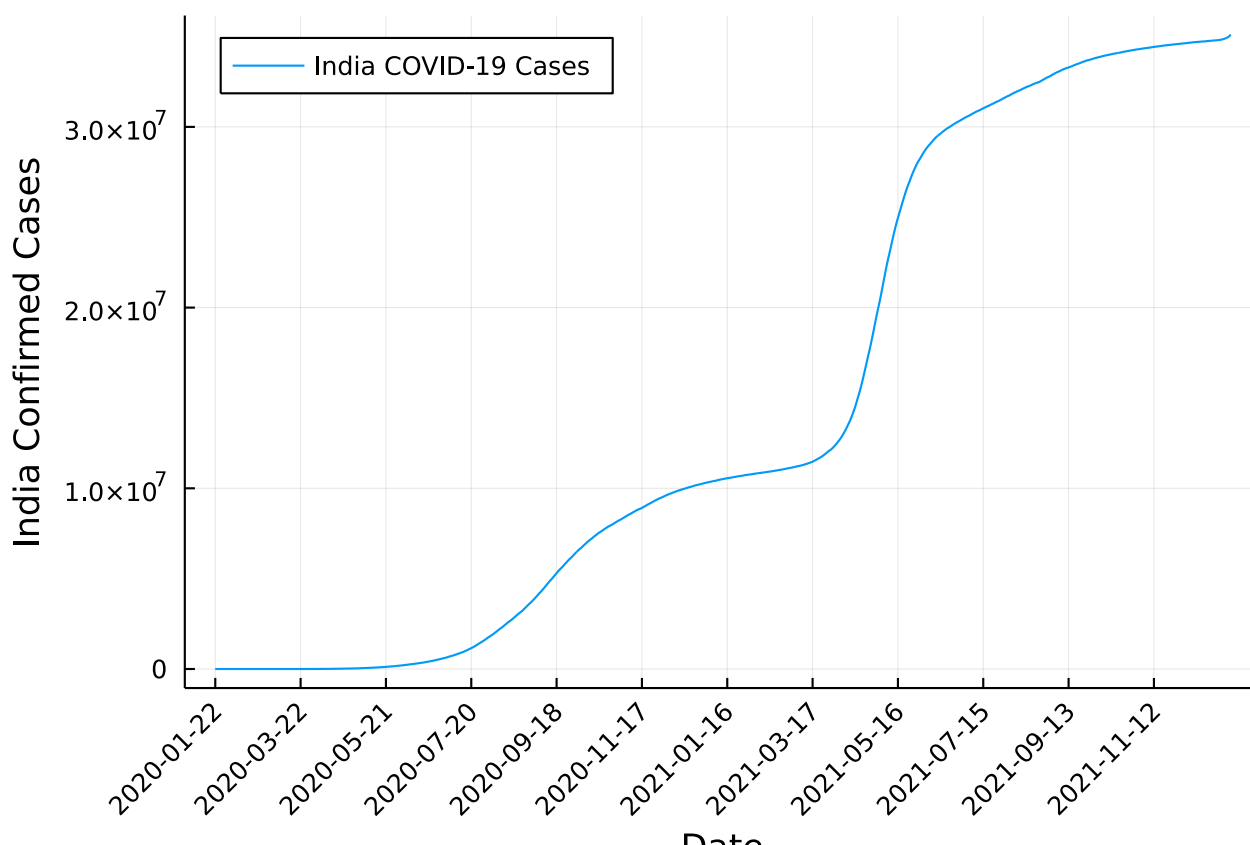
```
format = dateformat"m/d/Y"
```

```
• format = Dates.DateFormat("m/d/Y")
```

```
• dates = parse.(Date, date_strings, format) + Year(2000);
```

```
• plot(dates, IND_data, xticks=dates[1:60:end], xrotation=45, leg=:topleft,  
label="India COVID-19 Cases");
```

```
• xlabel!("Date");
```



```
• ylabel!("India Confirmed Cases")
```

Simple Model of infection

```
• md"  
• Simple Model of infection  
• "
```

**UndefVarError: @cI\_n\_cmd not defined**

```

1. top-level scope @ :0
2. #macroexpand#51 @ expr.jl:115 [inlined]
3. macroexpand @ expr.jl:114 [inlined]
4. try_macroexpand(::Module, ::Base.UUID, ::Expr) @ PlutoRunner.jl:253
5. var"#run_expression#25"(::Bool, ::typeof(Main.PlutoRunner.run_expression),
  ::Module, ::Expr, ::Base.UUID, ::Nothing, ::Nothing) @ PlutoRunner.jl:482
6. top-level scope @ none:1

```

```
• ``I_{n+1} = I_n + (cI_n)``
```

```
I0 = 1
```

```
• I0 = 1
```

```
I_0 = 1
```

```
• I_0 = 1
```

```
I_0 = 1
```

```
• I_0 = 1
```

```
c = 0.01
```

```
• c = 0.01 # average no. of people that each individual infects on each day
```

```
• λ = 1 + c ;
```

```
I_1 = 1.01
```

```
• I_1 = λ * I_0
```

```
I_2 = 1.0201
```

```
• I_2 = λ * I_1
```

```
I_3 = 1.030301
```

```
• I_3 = λ * I_2
```

```
I =
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
• I = zeros(I)
```

```
1
```

```
• I[1] = I_0
```

```
T = 20
```

```
• T=20
```

```

• for n in 1:T-1
•     I[n+1] = λ * I[n]
•     @show n, I[n]
• end

```

```
[1.0, 1.01, 1.0201, 1.0303, 1.0406, 1.05101, 1.06152, 1.07214, 1.08286, 1.09369, 1.10476]
```

```
• I
```

```
• plot(I, m=:o, label="I[n]", legend=:topleft);
```

```
p = 0.02
```

```
• p = 0.02
```

```
α = 0.01
```

```
• α = 0.01
```

```
N = 1000
```

```
• N = 1000
```

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

```

• function run_infection(T=20)
•
•     I = zeros(T)
•     I[1] = I_0
•
•     for n in 1:T-1
•         I[n+1] = λ * I[n]
•     end
•
•     return I
• end

```

```
# 2 methods for generic function run_infection:
```

- run\_infection() in Main.workspace#134 at [/Users/virajvaidya/Downloads/MITMATH2.jl#===#afef1536-fd7e-47cc-8b6b-a9d4e716f882:1](#)
- run\_infection(T) in Main.workspace#134 at [/Users/virajvaidya/Downloads/MITMATH2.jl#===#afef1536-fd7e-47cc-8b6b-a9d4e716f882:1](#)

```
• methods(run_infection)
```

```
[1.0, 1.01, 1.0201, 1.0303, 1.0406, 1.05101, 1.06152, 1.07214, 1.08286, 1.09369]
```

```
• run_infection(10)
```

**Multiple definitions for `I_result`**

```
• I_result = run_infection(10);
```

**UndefVarError: `I_result` not defined**

1. top-level scope @ **Local: 1**

```
• I_result
```

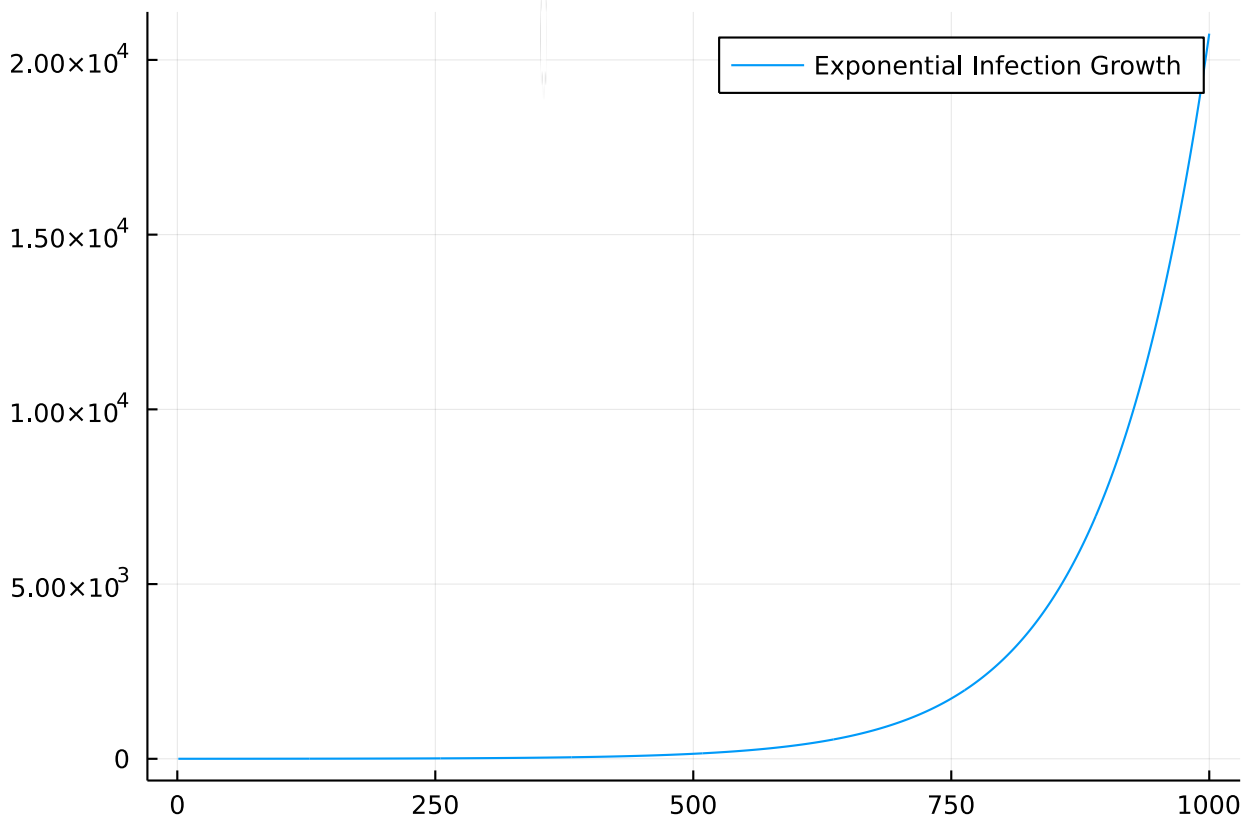
**UndefVarError: `I_result` not defined**

1. top-level scope @ **Local: 1**

```
• plot(I_result, m=:o)
```

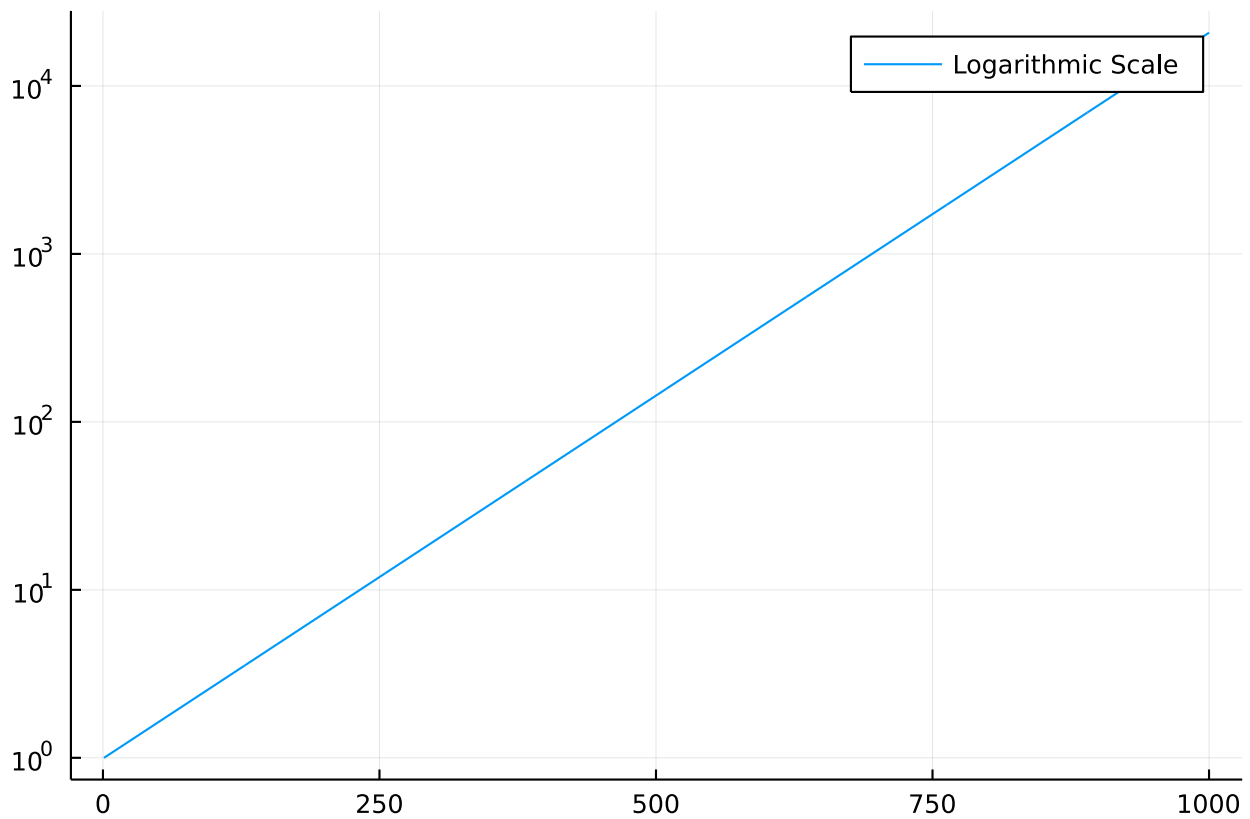
```
end_T = 1000
```

```
• end_T = 1000
```



```
• plot(run_infection(1000), label="Exponential Infection Growth")
```





```
• plot(run_infection(1000), yscale=:log10, label="Logarithmic Scale")
```

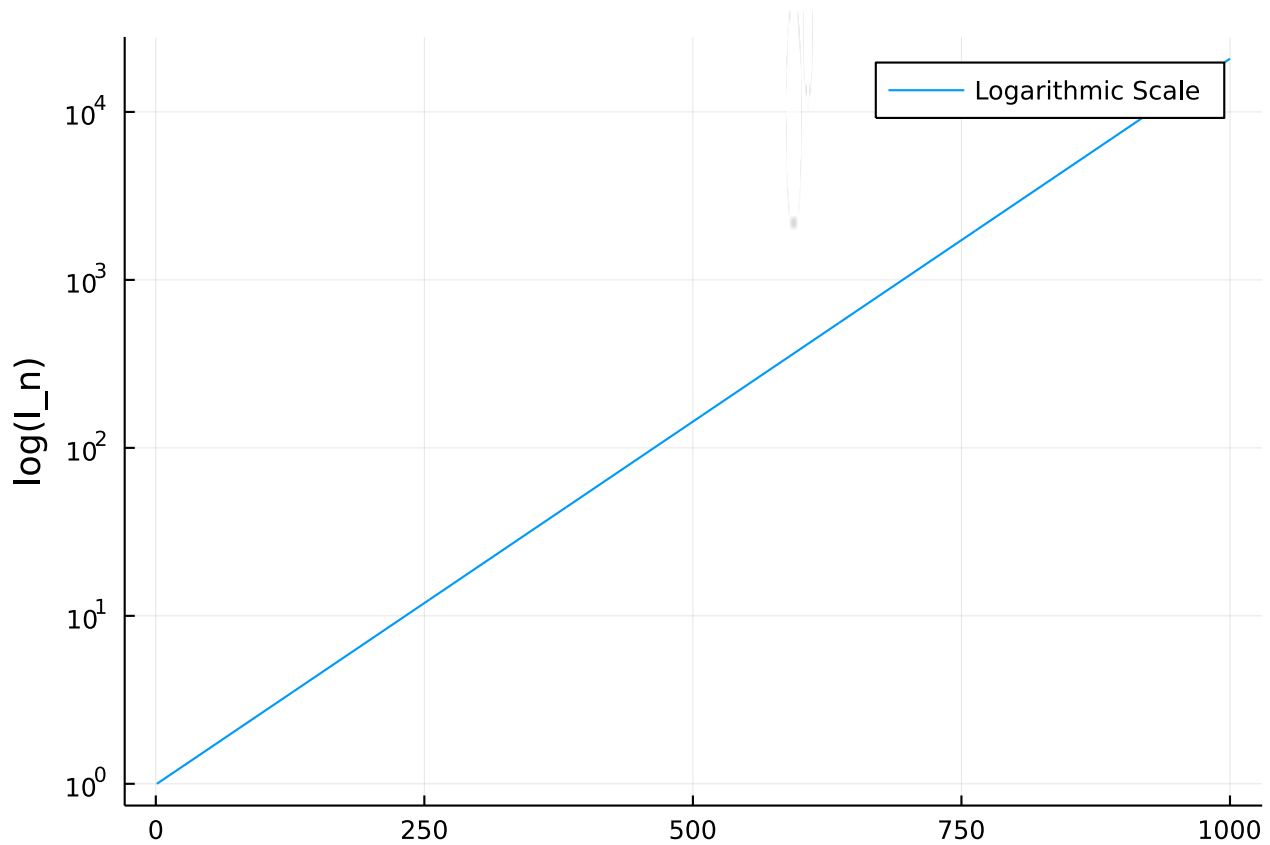
Multiple definitions for `I_result`

```
• I_result = run_infection(1000)
```

UndefVarError: `I_result` not defined

1. top-level scope @ **Local: 1**

```
• plot(log10.(I_result))
```



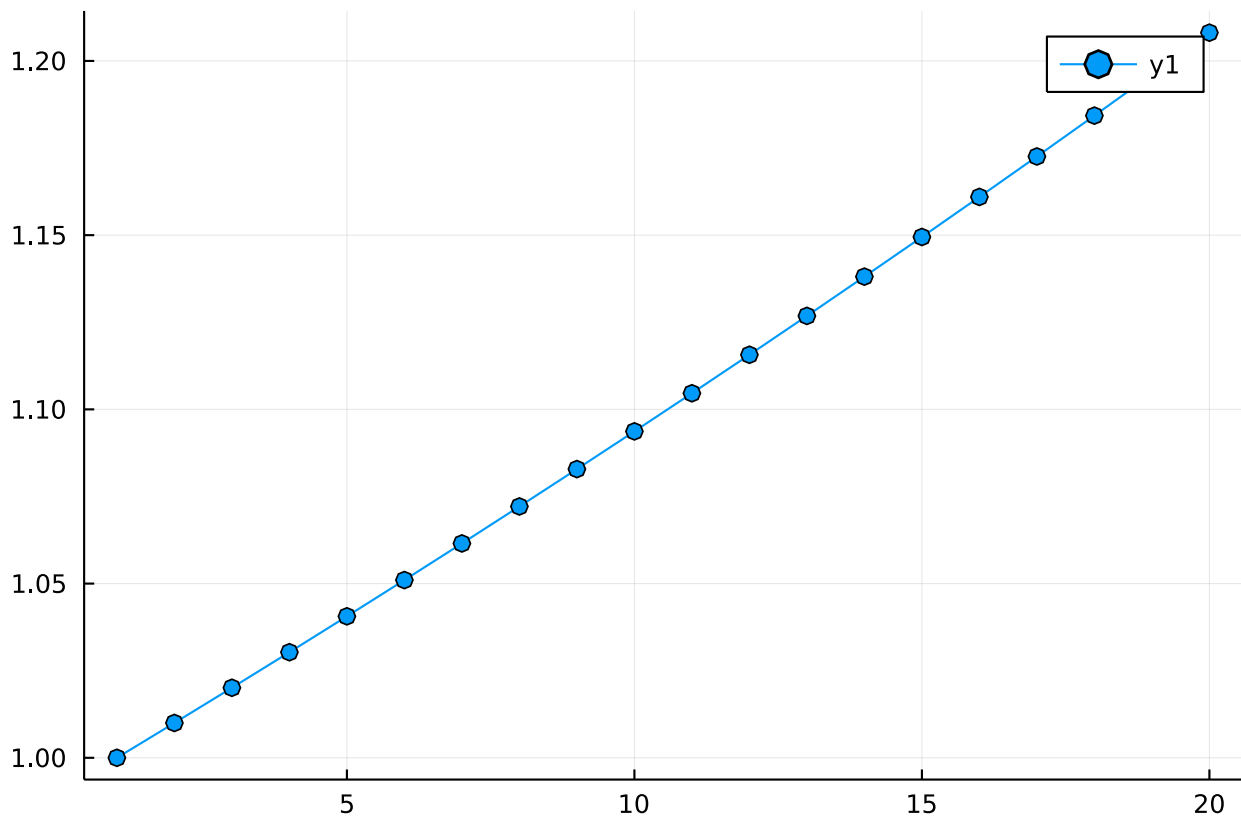
```
• ylabel!("log(I_n)")
```

$\beta$  (generic function with 1 method)

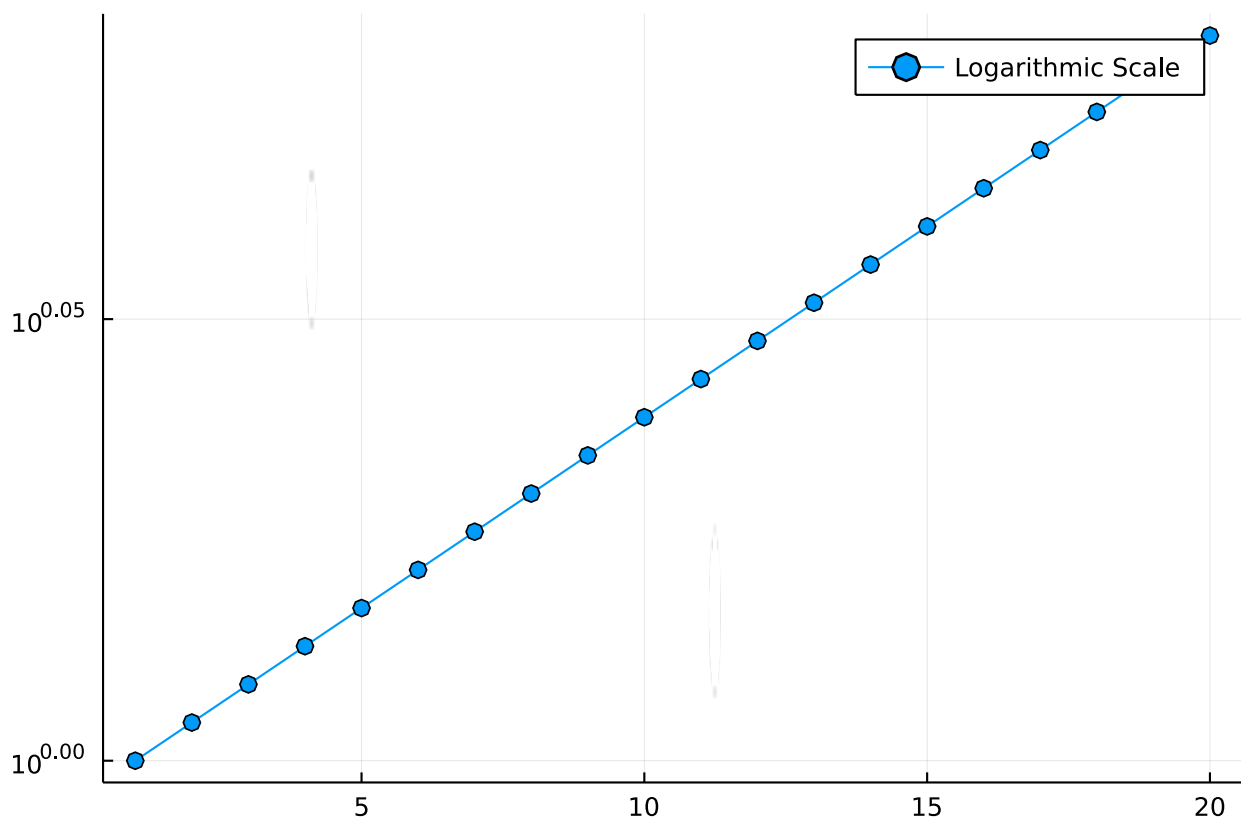
```
•  $\beta(I, S) = p * \alpha * S$ 
```

$I_{20} =$

```
[1.0, 1.01, 1.0201, 1.0303, 1.0406, 1.05101, 1.06152, 1.07214, 1.08286, 1.09369, 1.10469, 1.11596, 1.12788, 1.14045, 1.15367, 1.16754, 1.18206, 1.19724, 1.21308, 1.22958, 1.24674, 1.26456, 1.28304, 1.30219, 1.32201, 1.34251, 1.36369, 1.38555, 1.40809, 1.43132, 1.45524, 1.47985, 1.50515, 1.53115, 1.55785, 1.58525, 1.61335, 1.64215, 1.67165, 1.70185, 1.73275, 1.76435, 1.79665, 1.82965, 1.86335, 1.89775, 1.93285, 1.96865, 2.00515, 2.04235, 2.08025, 2.11885, 2.15815, 2.19815, 2.23885, 2.28025, 2.32235, 2.36515, 2.40865, 2.45285, 2.49775, 2.54335, 2.58965, 2.63665, 2.68435, 2.73275, 2.78185, 2.83165, 2.88215, 2.93335, 2.98525, 3.03785, 3.09115, 3.14515, 3.19985, 3.25525, 3.31135, 3.36815, 3.42565, 3.48385, 3.54275, 3.60235, 3.66265, 3.72365, 3.78535, 3.84775, 3.91085, 3.97465, 4.03915, 4.10435, 4.17025, 4.23685, 4.30415, 4.37215, 4.44085, 4.51025, 4.58035, 4.65115, 4.72265, 4.79485, 4.86775, 4.94135, 5.01565, 5.09065, 5.16635, 5.24275, 5.31985, 5.39765, 5.47615, 5.55535, 5.63525, 5.71585, 5.79715, 5.87915, 5.96185, 6.04525, 6.12935, 6.21415, 6.29965, 6.38585, 6.47275, 6.56035, 6.64865, 6.73765, 6.82735, 6.91775, 7.00885, 7.10065, 7.19315, 7.28635, 7.38025, 7.47485, 7.57015, 7.66615, 7.76285, 7.86025, 7.95835, 8.05715, 8.15665, 8.25685, 8.35775, 8.45935, 8.56165, 8.66465, 8.76835, 8.87275, 8.97785, 9.08365, 9.19015, 9.29735, 9.40525, 9.51385, 9.62315, 9.73315, 9.84385, 9.95525, 10.06735, 10.18015, 10.29365, 10.40785, 10.52275, 10.63835, 10.75465, 10.87165, 10.98935, 11.10775, 11.22685, 11.34665, 11.46715, 11.58835, 11.71025, 11.83285, 11.95615, 12.08015, 12.20485, 12.33025, 12.45635, 12.58315, 12.71065, 12.83885, 12.96775, 13.09735, 13.22765, 13.35865, 13.49035, 13.62275, 13.75585, 13.88965, 14.02415, 14.15935, 14.29525, 14.43185, 14.56915, 14.70715, 14.84585, 14.98525, 15.12535, 15.26615, 15.40765, 15.54985, 15.69275, 15.83635, 15.98065, 16.12565, 16.27135, 16.41775, 16.56485, 16.71265, 16.86115, 17.01035, 17.16025, 17.31085, 17.46215, 17.61415, 17.76685, 17.92025, 18.07435, 18.22915, 18.38465, 18.54085, 18.69775, 18.85535, 19.01365, 19.17265, 19.33235, 19.49275, 19.65385, 19.81565, 19.97815, 20.14135, 20.30525, 20.46985, 20.63515, 20.80115, 20.96785, 21.13525, 21.30335, 21.47215, 21.64165, 21.81185, 21.98275, 22.15435, 22.32665, 22.49965, 22.67335, 22.84775, 23.02285, 23.19865, 23.37515, 23.55235, 23.73025, 23.90885, 24.08815, 24.26815, 24.44885, 24.63025, 24.81235, 25.09515, 25.37865, 25.66285, 25.94775, 26.23335, 26.51965, 26.80665, 27.09435, 27.38275, 27.67185, 27.96165, 28.25215, 28.54335, 28.83525, 29.12785, 29.42115, 29.71515, 30.00985, 30.30525, 30.60135, 30.89815, 31.19565, 31.49385, 31.79275, 32.09235, 32.39265, 32.69365, 32.99535, 33.29775, 33.60085, 33.90465, 34.20915, 34.51435, 34.82025, 35.12685, 35.43415, 35.74215, 36.05085, 36.36025, 36.67035, 36.98115, 37.29265, 37.60485, 37.91775, 38.23135, 38.54565, 38.86065, 39.17635, 39.49275, 39.80985, 40.12765, 40.44615, 40.76535, 41.08525, 41.40585, 41.72715, 42.04915, 42.37185, 42.69525, 43.01935, 43.34415, 43.66965, 43.99585, 44.32275, 44.65035, 44.97865, 45.30765, 45.63735, 45.96775, 46.29885, 46.63065, 46.96315, 47.29635, 47.63025, 47.96485, 48.30015, 48.63615, 48.97285, 49.31025, 49.64835, 49.98715, 50.32665, 50.66685, 51.00775, 51.34935, 51.69165, 52.03465, 52.37835, 52.72275, 53.06785, 53.41365, 53.76015, 54.10735, 54.45525, 54.80385, 55.15315, 55.50315, 55.85385, 56.20525, 56.55735, 56.91015, 57.26365, 57.61785, 57.97275, 58.32835, 58.68465, 59.04165, 59.39935, 59.75775, 60.11685, 60.47665, 60.83715, 61.19835, 61.56025, 61.92285, 62.28615, 62.65015, 63.01485, 63.38025, 63.74635, 64.11315, 64.48065, 64.84885, 65.21775, 65.58735, 65.95765, 66.32865, 66.69935, 67.07075, 67.44285, 67.81465, 68.18715, 68.56035, 68.93425, 69.30885, 69.68415, 70.06015, 70.43685, 70.81425, 71.19235, 71.57115, 71.95065, 72.33085, 72.71175, 73.09335, 73.47565, 73.85865, 74.24235, 74.62675, 75.01185, 75.39765, 75.78415, 76.17135, 76.55925, 76.94785, 77.33715, 77.72715, 78.11785, 78.50925, 78.90135, 79.29415, 79.68765, 80.08185, 80.47675, 80.87235, 81.26865, 81.66565, 82.06335, 82.46175, 82.86085, 83.26065, 83.66115, 84.06235, 84.46425, 84.86685, 85.27015, 85.67415, 86.07885, 86.48425, 86.89035, 87.29715, 87.70465, 88.11285, 88.52175, 88.93135, 89.34165, 89.75265, 90.16435, 90.57675, 90.98985, 91.40365, 91.81815, 92.23335, 92.64925, 93.06585, 93.48315, 93.90115, 94.31985, 94.73925, 95.15935, 95.58015, 95.99165, 96.40385, 96.81675, 97.23035, 97.64465, 98.05965, 98.47535, 98.89175, 99.30885, 99.72665, 100.14515, 100.56435, 100.98425, 101.40485, 101.82615, 102.24815, 102.67085, 103.09425, 103.51835, 103.94315, 104.36865, 104.79485, 105.22175, 105.64935, 106.07765, 106.50665, 106.93635, 107.36675, 107.79785, 108.22965, 108.66215, 109.09535, 109.52925, 109.96385, 110.39915, 110.83515, 111.27185, 111.70925, 112.14735, 112.58615, 113.02565, 113.46585, 113.90675, 114.34835, 114.79065, 115.23365, 115.67735, 116.12175, 116.56685, 117.01265, 117.45915, 117.90635, 118.35425, 118.80285, 119.25215, 119.70215, 120.15285, 120.60425, 121.05635, 121.50915, 121.96265, 122.41685, 122.87175, 123.32735, 123.78365, 124.24065, 124.69835, 125.15675, 125.61585, 126.07565, 126.53615, 126.99735, 127.45925, 127.92185, 128.38515, 128.84915, 129.31385, 129.77925, 130.24535, 130.71215, 131.17965, 131.64785, 132.11675, 132.58635, 133.05665, 133.52765, 133.99935, 134.47175, 134.94485, 135.41865, 135.89315, 136.36835, 136.84425, 137.32085, 137.79815, 138.27615, 138.75485, 139.23425, 139.71435, 140.19515, 140.67665, 141.15885, 141.64175, 142.12535, 142.60965, 143.09465, 143.58035, 144.06675, 144.55385, 145.04165, 145.53015, 146.01935, 146.50925, 146.99985, 147.49115, 147.98315, 148.47585, 148.96925, 149.46335, 149.95815, 150.45365, 150.95085, 151.44875, 151.94735, 152.44665, 152.94665, 153.44735, 153.94875, 154.45085, 154.95365, 155.45715, 155.96135, 156.46625, 156.97185, 157.47815, 157.98515, 158.49285, 158.99125, 159.49035, 159.99015, 160.49065, 160.99185, 161.49375, 161.99635, 162.49965, 162.99365, 163.48835, 163.98375, 164.47985, 164.97665, 165.47415, 165.97235, 166.47125, 166.97085, 167.47115, 167.97215, 168.47385, 168.97625, 169.47935, 169.98315, 170.48765, 170.99285, 171.49875, 171.99535, 172.49265, 172.99065, 173.48935, 173.98875, 174.48885, 174.98965, 175.49115, 175.99335, 176.49625, 176.99985, 177.50415, 177.99915, 178.49485, 178.99125, 179.48835, 179.98615, 180.48415, 180.98235, 181.48075, 181.97935, 182.47815, 182.97715, 183.47635, 183.97575, 184.47535, 184.97515, 185.47515, 185.97535, 186.47575, 186.97635, 187.47715, 187.97815, 188.47935, 188.98075, 189.48235, 189.98415, 190.48615, 190.98835, 191.49075, 191.99335, 192.49615, 192.99915, 193.50235, 193.99575, 194.48935, 194.98315, 195.47715, 195.97135, 196.46575, 196.96035, 197.45515, 197.95015, 198.44535, 198.94075, 199.43635, 199.93215, 200.42815, 200.92435, 201.42075, 201.91735, 202.41415, 202.91115, 203.40835, 203.90575, 204.40335, 204.90115, 205.39915, 205.89735, 206.39575, 206.89435, 207.39315, 207.89215, 208.39135, 208.89075, 209.39035, 209.89015, 210.39015, 210.89035, 211.39075, 211.89135, 212.39215, 212.89315, 213.39435, 213.89575, 214.39735, 214.89915, 215.40115, 215.90335, 216.40575, 216.90835, 217.41115, 217.91415, 218.41735, 218.92075, 219.42435, 219.92815, 220.43215, 220.93635, 221.44075, 221.94535, 222.45015, 222.95515, 223.46035, 223.96575, 224.47135, 224.97715, 225.48315, 225.98935, 226.49575, 226.99235, 227.48915, 227.98615, 228.48335, 228.98075, 229.47835, 229.97615, 230.47415, 230.97235, 231.47075, 231.96935, 232.46815, 232.96715, 233.46635, 233.96575, 234.46535, 234.96515, 235.46515, 235.96535, 236.46575, 236.96635, 237.46715, 237.96815, 238.46935, 238.97075, 239.47235, 239.97415, 240.47615, 240.97835, 241.48075, 241.98335, 242.48615, 242.98915, 243.49235, 243.99575, 244.49935, 244.99315, 245.49715, 245.99135, 246.49575, 246.99035, 247.49515, 247.99015, 248.49535, 248.99075, 249.49635, 249.99215, 250.49815, 250.99435, 251.49075, 251.98735, 252.48415, 252.98115, 253.47835, 253.97575, 254.47335, 254.97115, 255.46915, 255.96735, 256.46575, 256.96435, 257.46315, 257.96215, 258.46135, 258.96075, 259.46035, 259.96015, 260.46015, 260.96035, 261.46075, 261.96135, 262.46215, 262.96315, 263.46435, 263.96575, 264.46735, 264.96915, 265.47115, 265.97335, 266.47575, 266.97835, 267.48115, 267.98415, 268.48735, 268.99075, 269.49435, 269.99815, 270.50215, 270.99635, 271.49075, 271.98535, 272.48015, 272.97515, 273.47035, 273.96575, 274.46135, 274.95715, 275.45315, 275.94935, 276.44575, 276.94235, 277.43915, 277.93615, 278.43335, 278.93075, 279.42835, 279.92615, 280.42415, 280.92235, 281.42075, 281.91935, 282.41815, 282.91715, 283.41635, 283.91575, 284.41535, 284.91515, 285.41515, 285.91535, 286.41575, 286.91635, 287.41715, 287.91815, 288.41935, 288.92075, 289.42235, 289.92415, 290.42615, 290.92835, 291.43075, 291.93335, 292.43615, 292.93915, 293.44235, 293.94575, 294.44935, 294.95315, 295.45715, 295.96135, 296.46575, 296.97035, 297.47515, 297.98015, 298.48535, 298.99075, 299.49635, 299.99215, 300.49815, 300.99435, 301.49075, 301.98735, 302.48415, 302.98115, 303.47835, 303.97575, 304.47335, 304.97115, 305.46915, 305.96735, 306.46575, 306.96435, 307.46315, 307.96215, 308.46135, 308.96075, 309.46035, 309.96015, 310.46015, 310.96035, 311.46075, 311.96135, 312.46215, 312.96315, 313.46435, 313.96575, 314.46735, 314.96915, 315.47115, 315.97335, 316.47575, 316.97835, 317.48115, 317.98415, 318.48735, 318.99075, 319.49435, 319.99815, 320.50215, 320.99635, 321.49075, 321.98535, 322.48015, 322.97515, 323.47035, 323.96575, 324.46135, 324.95715, 325.45315, 325.94935, 326.44575, 326.94235, 327.43915, 327.93615, 328.43335, 328.93075, 329.42835, 329.92615, 330.42415, 330.92235, 331.42075, 331.91935, 332.41815, 332.91715, 333.41635, 333.91575, 334.41535, 334.91515, 335.41515, 335.91535, 336.41575, 336.91635, 337.41715, 337.91815, 338.41935, 338.92075, 339.42235, 339.92415, 340.42615, 340.92835, 341.43075, 341.93335, 342.43615, 342.93915, 343.44235, 343.94575, 344.44935, 344.95315, 345.45715, 345.96135, 346.46575, 346.97035, 347.47515, 347.98015, 348.48535, 348.99075, 349.49635, 349.99215, 350.49815, 350.99435, 351.49075, 351.98735, 352.48415, 352.98115, 353.47835, 353.97575, 354.47335, 354.97115, 355.46915, 355.96735, 356.46575, 356.96435, 357.46315, 357.96215, 358.46135, 358.96075, 359.46035, 359.96015, 360.46015, 360.96035, 361.46075, 361.96135, 362.46215, 362.96315, 363.46435, 363.96575, 364.46735, 364.96915, 365.47115, 365.97335, 366.47575, 366.97835, 367.48115, 367.98415, 368.48735, 368.99075, 369.49435, 369.99815, 370.50215, 370.99635, 371.49075, 371.98535, 372.48015, 372.97515, 373.47035, 373.96575, 374.46135, 374.95715, 375.45315, 375.94935, 376.44575, 376.94235, 377.43915, 377.93615, 378.43335, 378.93075, 379.42835, 379.92615, 380.42415, 380.92235, 381.42075, 381.91935, 382.41815, 382.91715, 383.41635, 383.91575, 384.41535, 384.91515, 385.41515, 385.91535, 386.41575, 386.91635, 387.41715, 387.91815, 388.41935, 388.92075, 389.42235, 389.92415, 390.42615, 390.92835, 391.43075, 391.93335, 392.43615, 392.93915, 393.44235, 393.94575, 394.44935, 394.95315, 395.45715, 395.96135, 396.46575, 396.97035, 397.47515, 397.98015, 398.48535, 398.99075, 399.49635, 399.99215, 400.49815, 400.99435, 401.49075, 401.98735, 402.48415, 402.98115, 403.47835, 403.97575, 404.47335, 404.97115, 405.46915, 405.96735, 406.46575, 406.96435, 407.46315, 407.96215, 408.46135, 408.96075, 409.46035, 409.96015, 410.46015, 410.96035, 411.46075, 411.96135, 412.46215, 412.96315, 413.46435, 413.96575, 414.46735, 414.96915, 415.47115, 415.97335, 416.47575, 416.97835, 417.48115, 417.98415, 418.48735, 418.99075, 419.49435, 419.99815, 420.50215, 420.99635, 421.49075, 421.98535, 422.48015, 422.97515, 423.47035, 423.96575, 424.46135, 424.95715, 
```



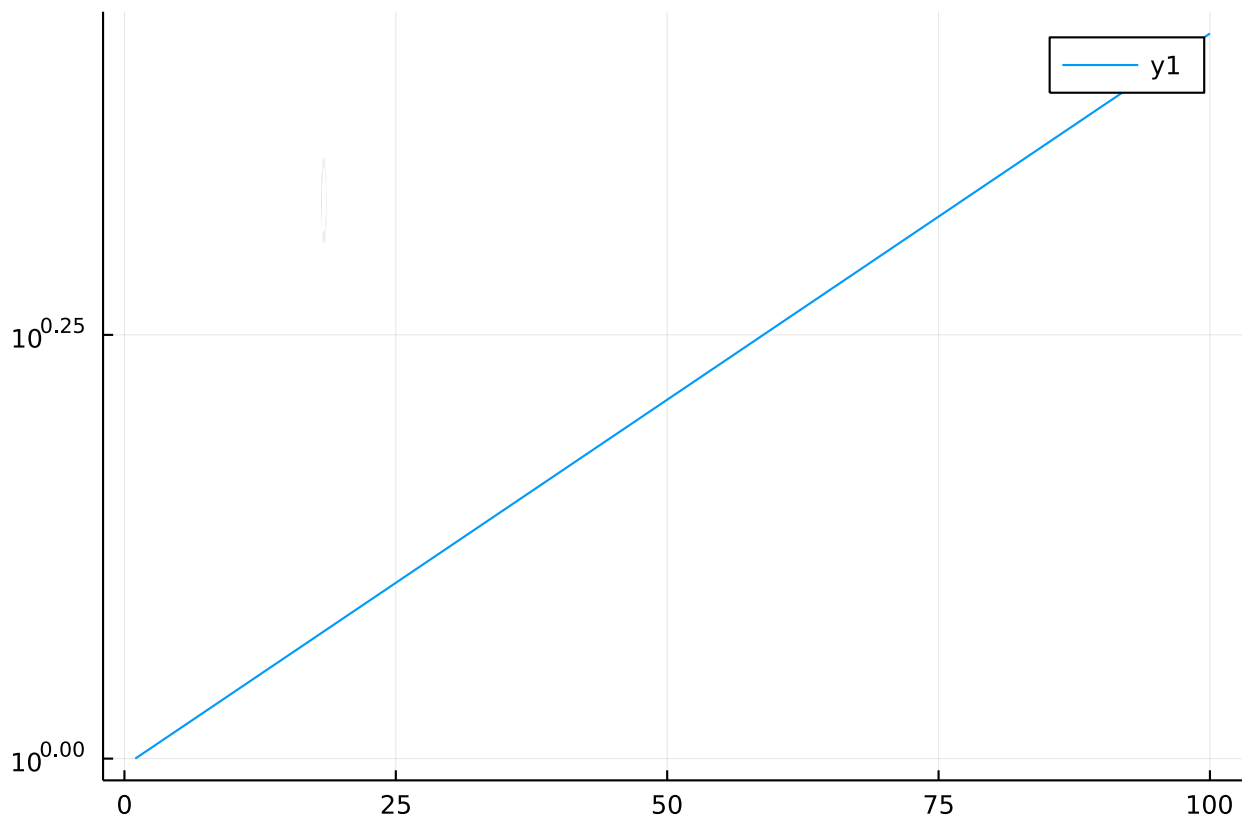
```
plot(I_20, m=:o)
```



```
plot(I_20, m=:o, yscale=:log10, label="Logarithmic Scale")
```

```
2.44863, 2.47312, 2.49785, 2.52283, 2.54806, 2.57354, 2.59927, 2.62527, 2.65152, 2.67
```

```
I_100 = run_infection(100)
```



```
plot(I_100, yscale=:log10)
```

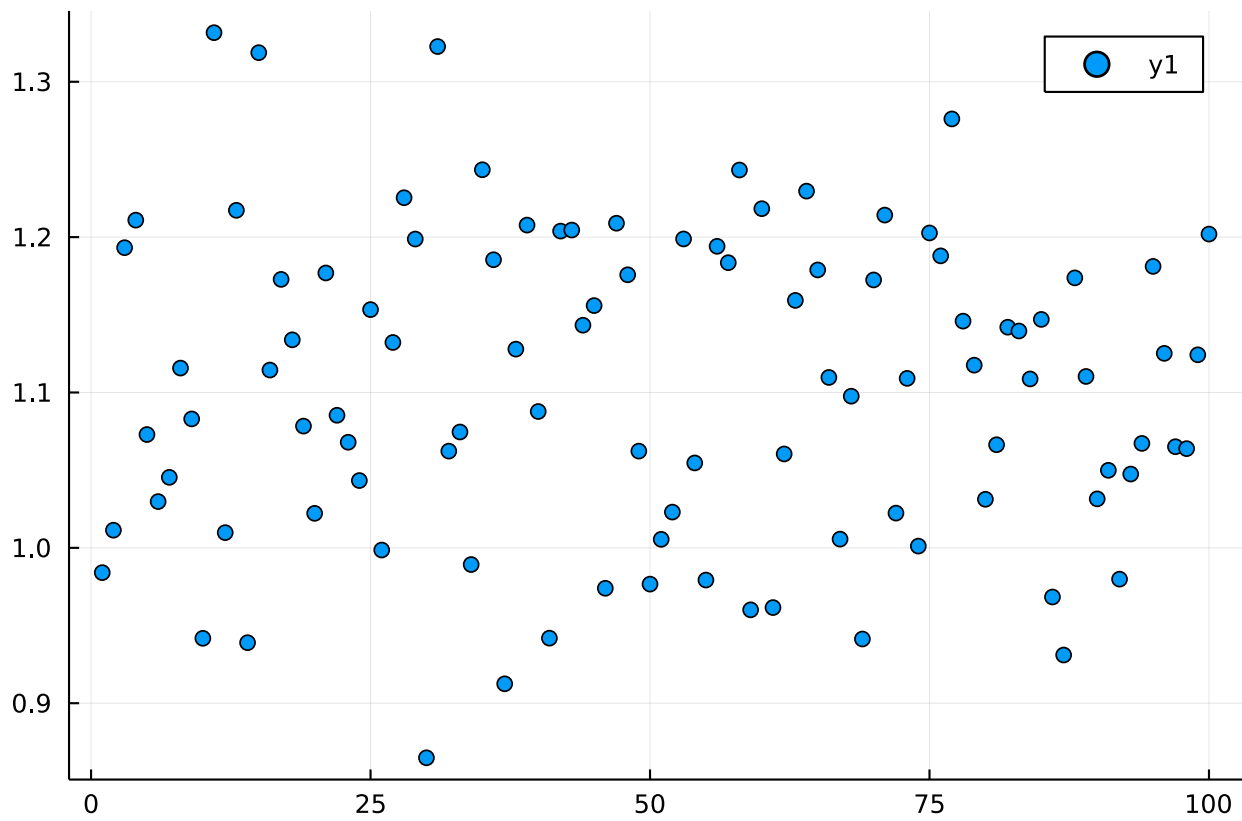
```
c_average = 1.1
```

```
c_average = 1.1
```

```
cs =
```

```
[0.984051, 1.01137, 1.1932, 1.21093, 1.07294, 1.02978, 1.04538, 1.1157, 1.08298, 0.9
```

```
cs = [c_average + 0.1*randn() for i in 1:100]
```



```
scatter(cs)
```