Vaccine Code

;; r0 of SARS is 0.19-1.08 source: https://www.news-medical.net/health/What-is-R0.aspx

;; r0 of SARS is 2.4 source: https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30484-9/fulltext

;; Based on above sources, SARS infection has an average r0 of 1.2233 amd range of 0.19-2.4.

;; r0 average is 0.19 + 1.08 + 2.4 = 3.67 / 2 = 1.2233...

;; SARS has a typical incubation period of 2-7 days, possibly 10 days and rarely 14 days in which the infection cannot be transmitted, source: https://www.cdc.gov/sars/about/faq.html

;; SARS, once symptoms show, infection can be transmitted, and 10 days should be allowed to pass from onset of symptoms to be considered non infectious, source: https://www.cdc.gov/sars/about/faq.html

;; Based on these numbers, SARS has a minimum duration of 12 days, average duration of 18 days and maximum duration of 24 days.

;; Therefore, based on the above durations of SARS infection, the minimum and maximum average-recovery-time range has been adjusted to reflect this.

;; SARS has an estimated death rate of 14%-15%, source: https://www.who.int/csr/sars/archive/2003\_05\_07a/en/#:~:text=On%20the%20basis%20of%20more%20detailed%20and%20complete,estimate%20of%20case%20fatality%20of%2014%25%20to%2015%25.

;; SARS has an estimated death rate of 10%, source: https://jamanetwork.com/journals/jama/fullarticle/2759815

;; Death rate average is 15% + 10% = 25% / 2 = 12.5%.

;; Based on this, the survival rate is between 85%-90% with an average of 87.5%.

;; Recovery rate average is 85% + 90% = 175 / 2 = 87.5%.

;; Therefore, the recovery-chance range has been adjusted to reflect this.

;; This model is a modified verson of the Mask model and fuctions near-identically

;; This model asumes the vaccine is always effective.

;; SARS-CoV-1 strain has no known vaccine, this model is interrested only in analysis r0 and related trends.

globals

[

nb-infected-previous ;; Number of infected people at the previous tick

beta-n ;; The average number of new secondary

;; infections per infected this tick

gamma ;; The average number of new recoveries

;; per infected this tick

r0 ;; The number of secondary infections that arise

;; due to a single infected introduced in a wholly

;; susceptible population

]

turtles-own

[

infected? ;; If true, the person is infected

cured? ;; If true, the person has lived through an infection.

;; They cannot be re-infected.

susceptible? ;; Tracks whether the person was initially susceptible

infection-length ;; How long the person has been infected

recovery-time ;; Time (in hours) it takes before the person has a chance to recover from the infection

nb-infected ;; Number of secondary infections caused by an

;; infected person at the end of the tick

nb-recovered ;; Number of recovered people at the end of the tick

vaccine? ;; If true, the person is vaccinated

]

;;;

;;; SETUP PROCEDURES

;;;

to setup

clear-all

setup-people

setup-people-vaccine ;; Setup People who are vaccinated

reset-ticks

end

to setup-people

create-turtles initial-people

[

setxy random-xcor random-ycor

set cured? false

set infected? false

set susceptible? true

set vaccine? false ;; These people are not vaccinated

set shape "person"

set color white

;; Set the recovery time for each agent to fall on a

;; normal distribution around average recovery time

set recovery-time random-normal average-recovery-time average-recovery-time / 4

;; make sure it lies between 0 and 2x average-recovery-time

if recovery-time > average-recovery-time \* 2 [

set recovery-time average-recovery-time \* 2

]

if recovery-time < 0 [ set recovery-time 0 ]

;; Each individual has a 5% chance of starting out infected.

;; To mimic true KM conditions use "ask one-of turtles" instead.

if (random-float 100 < 5)

[

set infected? true

set susceptible? false

set infection-length random recovery-time

]

assign-color

]

if links? [ make-network ]

end

;; Setup for people who are vaccinated

to setup-people-vaccine

create-turtles initial-people-vaccine

[

setxy random-xcor random-ycor

set cured? false

set infected? false

set susceptible? true

set vaccine? true ;; These people are vaccinated

set shape "person"

set color yellow

]

if links? [ make-network ]

end

;; Different people are displayed in 3 different colors depending on health

;; White is neither infected nor cured (set at beginning)

;; Green is a cured person

;; Red is an infected person

;; Yellow is a person vaccinated

to assign-color ;; turtle procedure

if infected?

[ set color red ]

if cured?

[ set color green ]

end

to make-network

ask turtles

[

create-links-with turtles-on neighbors

]

end

;;;

;;; GO PROCEDURES

;;;

to go

if all? turtles [ not infected? ]

[ export-all-plots "C:/Users/Scott/Desktop/ProjectData Vaccine/Test.csv"

stop ]

ask turtles

[ move

clear-count ]

ask turtles with [ infected? ]

[ infect

maybe-recover ]

ask turtles

[ assign-color

calculate-r0 ]

tick

end

;; People move about at random.

to move ;; turtle procedure

rt random-float 360

fd 1

end

to clear-count

set nb-infected 0

set nb-recovered 0

end

to infect

let nearby-uninfected (turtles-on neighbors)

with [ not infected? and not cured? ]

if nearby-uninfected != nobody

[ ask nearby-uninfected

[ if vaccine? = false

[ if random-float 100 > infection-chance ;; Infect no mask-mask true

[ set infected? true

set nb-infected (nb-infected + 1)

]

]

]

]

end

to maybe-recover

set infection-length infection-length + 1

;; If people have been infected for more than the recovery-time

;; then there is a chance for recovery

if infection-length > recovery-time

[

if random-float 100 < recovery-chance

[ set infected? false

set cured? true

set nb-recovered (nb-recovered + 1)

]

]

end

to calculate-r0

let new-infected sum [ nb-infected ] of turtles

let new-recovered sum [ nb-recovered ] of turtles

;; Number of infected people at the previous tick:

set nb-infected-previous

count turtles with [ infected? ] +

new-recovered - new-infected

;; Number of susceptibles now:

let susceptible-t

(initial-people + initial-people-vaccine) -

count turtles with [ infected? ] -

count turtles with [ cured? ]

;; Initial number of susceptibles:

let s0 count turtles with [ susceptible? ]

ifelse nb-infected-previous < 10

[ set beta-n 0 ]

[

;; This is beta-n, the average number of new

;; secondary infections per infected per tick

set beta-n (new-infected / nb-infected-previous)

]

ifelse nb-infected-previous < 10

[ set gamma 0 ]

[

;; This is the average number of new recoveries per infected per tick

set gamma (new-recovered / nb-infected-previous)

]

;; Prevent division by 0:

if(initial-people + initial-people-vaccine) - susceptible-t != 0 and susceptible-t != 0

[

;; This is derived from integrating dI / dS = (beta\*SI - gamma\*I) / (-beta\*SI):

set r0 (ln (s0 / susceptible-t) /((initial-people + initial-people-vaccine) - susceptible-t))

;; Assuming one infected individual introduced in the beginning,

;; and hence counting I(0) as negligible, we get the relation:

;; N - gamma\*ln(S(0)) / beta = S(t) - gamma\*ln(S(t)) / beta,

;; where N is the initial 'susceptible' population

;; Since N >> 1

;; Using this, we have R\_0 = beta\*N / gamma = N\*ln(S(0)/S(t)) / (K-S(t))

set r0 r0 \* s0 ]

end

; Copyright 2011 Uri Wilensky.

; See Info tab for full copyright and license.