Alyssa Decker

Geizi Llanes

Super Computing

March 29, 2017

Super Computing - Would You Survive?

In a small, hypothetical community the outbreak of chickenpox is tested along with the death rate. This project is a model of Chickenpox and its spread in a small hypothetical community. Chickenpox varies from person to person, meaning there is no constant, leaving most of the data represented by variables. The number of people within the community can range from 10 to 300 people, for our purpose, the population will stay at 300. This model shows the death, immunity, number of people and level of sickness. The reason for choosing chicken pox for my project was I did not know much about it, and I was interested in knowing the symptoms and how it would effect a community. To learn more about a virus that I knew nothing about, would help spread awareness of the different aspects of chickenpox. The aspects that I hope to bring awareness to are, they can be fatal and easily be spread. Recognizing chickenpox is one of the first steps in getting treatment to decrease spreading and uncomfortable symptoms.

Chicken pox presents itself through a blister like rash, which is itchy, irritating, and can cause a high fever. Before the vaccination became available many with chicken pox were hospitalized or died. With the assistance of vaccinations, there has been a steady decline in the number of individuals infected with chicken pox. In order to anticipate how chickenpox could infect a population there were Four different scenarios, with five test runs apiece, ran on a program created to show the spread of this virus. The program used was created with NetLogo, a coding interface. I believe that with a higher infectious rate and a lower chance of recovery that the fatality rate will increase.

After the program was created, two specific percentages, that represented the chance of recovery and infectiousness (dependent variables), were chosen based on my discretion; there is no conclusive data found to give a specific percentage to use in such a simulation. Changing the percentages for the chance of recovery and infectiousness. Each time I ran the scenario, was completed four different times and were run five times each. The hypothetical community consisted of 300 people, which was my independent variable. When the model is started, you begin with 3.3% of the population already infected with the virus. I used Netlogo, on my computer, to complete my experiment.

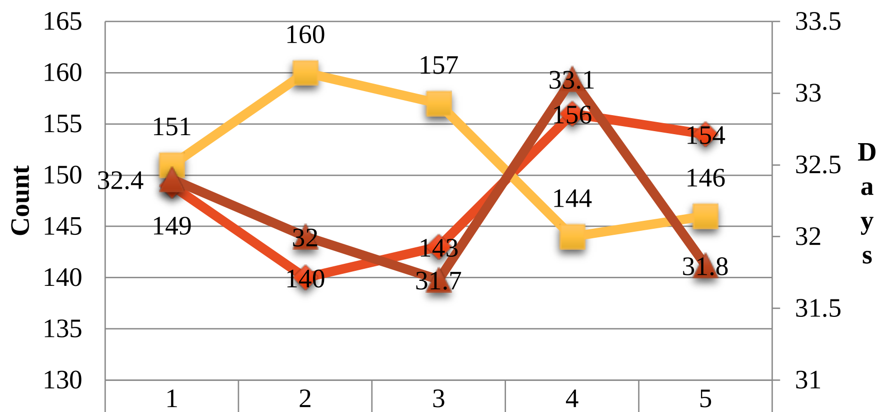
At first, chickenpox is disguised as a cold, but soon it covers it victims in a red, itchy rash of blisters, known as "pox". Pox usually showing up first on the face and chest, the blisters then spread over the rest of the body. In severe cases, blisters even show up in the mouth and ears. The virus can be in your system up to two weeks without any symptoms and you are still contagious even before you are aware you have chickenpox. The time you are most contagious is usually the first few days after the blister like rash appears. People help spread chickenpox by scratching their blisters, sneezing, or coughing, which spreads infected fluid to others. When inducted into the body, chickenpox goes straight into the nervous system which can cause shingles in the future. Chickenpox focuses on kids, attacking mostly those younger than 15. Adults who do get infected, though are more likely than kids to have serious complications. Once someone is infected, chickenpox cannot be stopped, but its symptoms can be treated.

The Varicella-Zoster Virus vaccine(*CDC)* may not prevent all chickenpox, but it is very effective at preventing severe cases. The chickenpox vaccine became available in the United

States in 1995. Between the years 2005- 2012 the chickenpox outbreaks have gone down (CDC). The second dose was implemented in 2006. Each year, more than 3.5 million cases of chickenpox, 9,000 hospitalizations, and 100 deaths are prevented yearly in the U.S.A (CDC). Hospitalizations also declined 93% in 2012 (CDC). For most people, getting chickenpox once provides immunity for life. However, for a few people, they can get chickenpox more than once, although this is not common.

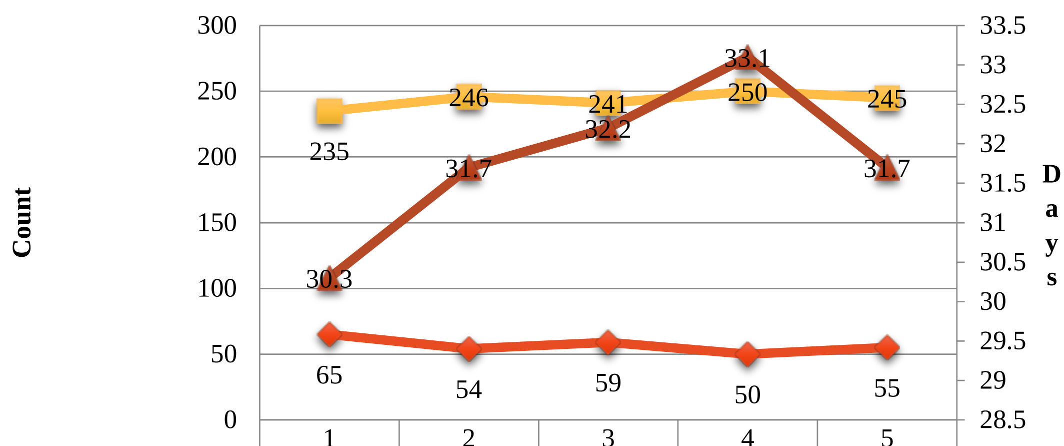
	Alive	Dead	Days to Immune	Chance-Recovery	Infectiousness	Number of People
	236	64	32.8	80%	60%	300
	230	70	32.1	80%	60%	300
	251	49	31.8	80%	60%	300
	246	54	32.1	80%	60%	300
	243	57	31.6	80%	60%	300
mean	187.4	52.6	26.48			
	273	27	32.8	90%	40%	300
	270	30	33.3	90%	40%	300
	269	31	35.8	90%	40%	300
	284	16	34.7	90%	40%	300
	278	22	32.9	90%	40%	300
mean	217.4	22.6	26.22			
	65	235	30.3	50%	20%	300
	54	246	31.7	50%	20%	300
	59	241	32.2	50%	20%	300
	50	250	33.1	50%	20%	300
	55	245	31.7	50%	20%	300
mean	54	186	23.96			
	149	151	32.4	50%	50%	300
	140	160	32	50%	50%	300
	143	157	31.7	50%	50%	300
	156	144	33.1	50%	50%	300
	154	146	31.8	50%	50%	300
mean	118.2	121.8	26.04			

Recovery:50% Infectiousness:50%

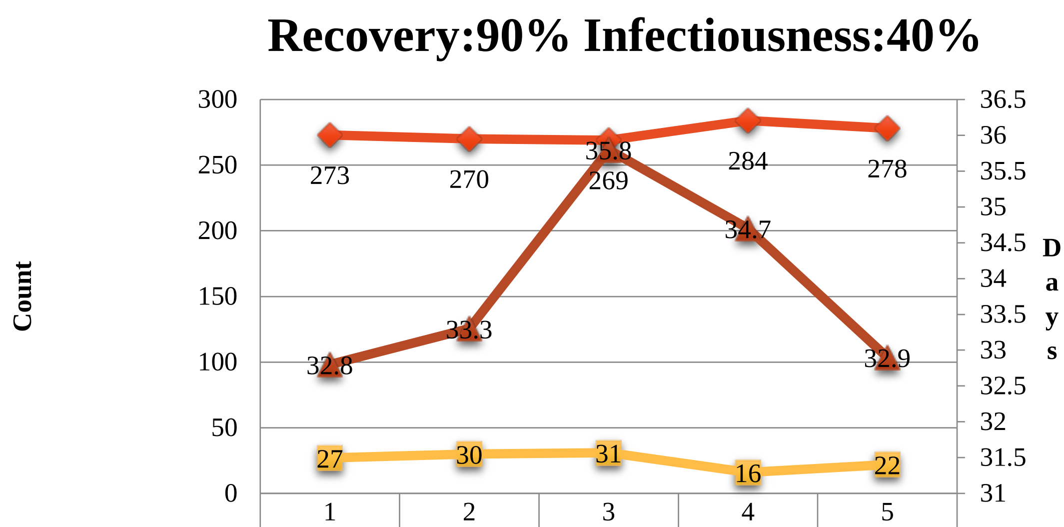


	1	2	3	4	5
Alive	149	140	143	156	154
dead	151	160	157	144	146
days to immune	32.4	32	31.7	33.1	31.8

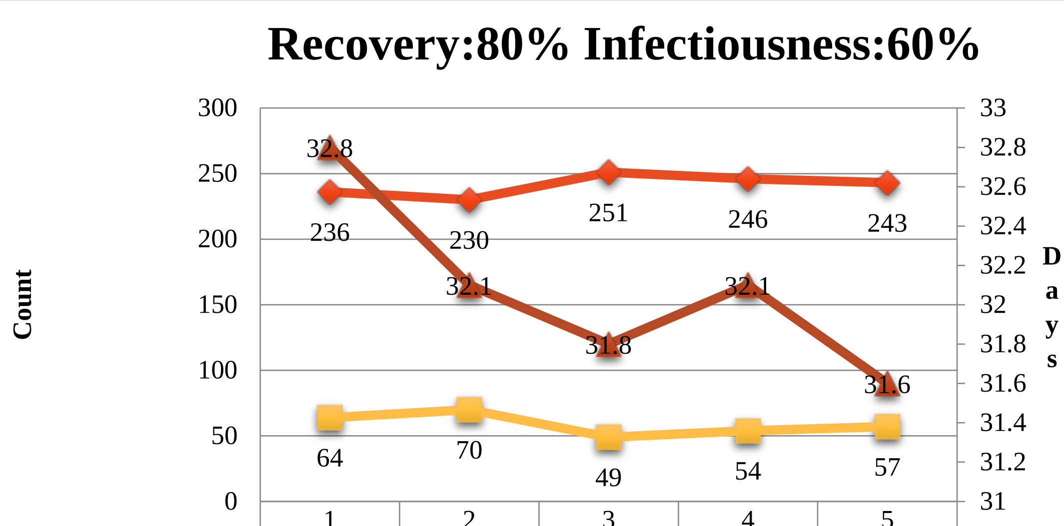
Recovery:50% Infectiousness:20%



	1	2	3	4	5
Alive	65	54	59	50	55
dead	235	246	241	250	245
days to immune	30.3	31.7	32.2	33.1	31.7



	1	2	3	4	5
Alive	273	270	269	284	278
dead	27	30	31	16	22
days to immune	32.8	33.3	35.8	34.7	32.9



	1	2	3	4	5
Alive	236	230	251	246	243
dead	64	70	49	54	57
days to immune	32.8	32.1	31.8	32.1	31.6

I observed that the lowest death rate was 16 people with the dependent variables of 90% (chance of recovery) and 40% (infectiousness). The highest death rate was 250 people with the dependent variables of 50% (chance of recovery) and 20% (infectiousness). This data surprised me because the chance of recovery was 50% and the infectiousness percent was low, I expected to see a lower death rate. When the chance of recovery rate was tested at 80% and 90% I observed that the death rate was lower. When the chance of recovery rate was tested at 50% the death rate was lower, even when the infectiousness rate was low. Based on my results my hypothesis was incorrect. With a higher rate of infectiousness and a lower chance of recovery the death rate decreased.

1 School name-San Juan college high school

2 Team number

3 The project's area of Science-Medical science

4 The computer language(s) Used-Scala

5 Team members grade(s) in school- 10th

6 email addresses-alyssadek@gmail.com

7 (team-00000.doc/team-00000.docx/team-00000.odt/team-00000.pdf,

the source code

```
turtles-own
  [ sick?                ;; if true, the turtle is infectious
    remaining-immunity   ;; how many weeks of immunity the turtle has left
    sick-time            ;; how long, in weeks, the turtle has been infectious
    age ]                ;; how many weeks old the turtle is

globals
  [ %infected            ;; what % of the population is infectious
    %immune              ;; what % of the population is immune
    lifespan             ;; the lifespan of a turtle
    chance-reproduce     ;; the probability of a turtle generating an offspring each tick
    carrying-capacity    ;; the number of turtles that can be in the world at one time
    immunity-duration
    Dead
    duration
   ]  ;; how many weeks immunity lasts

;; The setup is divided into four procedures
to setup
  clear-all
  setup-constants
  setup-turtles
  update-global-variables
```

```
  update-display
  reset-ticks

end


;; We create a variable number of turtles of which 10 are infectious,
;; and distribute them randomly
to setup-turtles
  create-turtles NumOfPeople
    [ setxy random-xcor random-ycor
      set age random lifespan
      set sick-time 168
      set remaining-immunity 0
      set size 1.5  ;; easier to see
      get-healthy ]
  ask n-of 10 turtles
    [ get-sick ]
end


to get-sick ;; turtle procedure
  set sick? true
  set remaining-immunity 0
end
```

```
to get-healthy ;; turtle procedure
  set sick? false
  set remaining-immunity 0
  set sick-time 0
end

to become-immune ;; turtle procedure
  set sick? false
  set sick-time 0
  set remaining-immunity immunity-duration
end

;; This sets up basic constants of the model.
to setup-constants

  ask patches [    set pcolor 5  ]
  set lifespan 613620 ; hour in 70 years    50 * 52      ;; 50 times 52 weeks = 50 years = 2600
weeks old
  set carrying-capacity 300
  set chance-reproduce 1
  set immunity-duration 613620
  set duration 672
```

```
end

to go

;ask patches set [pcolor red]
  ask turtles [
    get-older
    move
    if sick? [ recover-or-die ]
    ifelse sick? [ infect ] [ reproduce ]
  ]
  update-global-variables
  update-display
  tick
end

to update-global-variables

  if count turtles > 0
    [ set %infected (count turtles with [ sick? ] / count turtles) * 100
      set %immune (count turtles with [ immune? ] / count turtles) * 100
      set Dead (NumOfPeople - count turtles)
```

```
    ]
end


to update-display


  ask turtles
  [ if shape != turtle-shape [ set shape turtle-shape ]
    set color ifelse-value sick?  [ scale-color red sick-time  duration 0 ] [ ifelse-value immune?
[ sky ] [ yellow ] ]


  ]
  ;;ask patches [    set pcolor 5  ]
;;[ set color scale-color 5 age  50 0]
    ;[ if shape != turtle-shape [ set shape turtle-shape ]
     ; set color ifelse-value sick?  [ red ] [ ifelse-value immune? [ sky ] [ green ] ] ]
end


;;Turtle counting variables are advanced.
to get-older ;; turtle procedure
  ;; Turtles die of old age once their age exceeds the
  ;; lifespan (set at 50 years in this model).
  set age age + 1 ; one hour older
  if age > lifespan [ die ]
```

```
  if immune? [ set remaining-immunity remaining-immunity - 1 ]
  if sick? [ set sick-time sick-time + 1 ]
end

;; Turtles move about at random.
to move ;; turtle procedure
  rt random 100
  lt random 100
  fd 1
end

;; If a turtle is sick, it infects other turtles on the same patch.
;; Immune turtles don't get sick.
to infect ;; turtle procedure
  ask other turtles-here with [ not sick? and not immune? ]
    [ if random-float 100 < infectiousness
      [ get-sick ] ]
end

;; Once the turtle has been sick long enough, it
;; either recovers (and becomes immune) or it dies.
to recover-or-die ;; turtle procedure
```

```
  if sick-time > duration                        ;; If the turtle has survived past the virus' duration,
then
    [ ifelse random-float 100 < chance-recover   ;; either recover or die
      [ become-immune ]
      [ die ] ]
end


;; If there are less turtles than the carrying-capacity
;; then turtles can reproduce.
to reproduce
 ; if count turtles < carrying-capacity and random-float 100 < chance-reproduce
 ;   [ hatch 1
 ;     [ set age 1
 ;       lt 45 fd 1
 ;       get-healthy ] ]
end


to-report immune?
  report remaining-immunity > 0
end


to startup
```

```
  setup-constants ;; so that carrying-capacity can be used as upper bound of NumOfPeople
slider
end
```

Work cited

Healthline. www.healthline.com/health/chickenpox.

Medlineplus. medlineplus.gov/chickenpox.html.

Mayoclinic www.mayoclinic.org/diseases-conditions/chickenpox

CDC. www.cdc.gov/chickenpox/index.html