

# README

Breanna Richards, Tim Hedsbeth, Nancy Liu

12/15/2021

## PHP2560-Shiny-Final-Project

SARS is an infectious disease that spreads via droplets spread through contact with an infected individual (Zhou et al., 2004). This disease was identified in the early 2000s and due to its high infectivity was able to spread quickly and internationally (Zhou et al., 2004). Through drastic public health measures the spread of the disease was controlled, but peaked mathematical modler's interest (Zhou et al., 2004). In order to have an idea of how this disease was spreading the authors created a mathematical model that allows for them to predict the number of individuals that contracted SARS over a given amount of days (Zhou et al., 2004). In their model they have a multitude of parameters that they estimate based on their given data, to predict the number of individuals diagnosed with SARS on a given day based on the starting values and parameter values (Zhou et al., 2004). The authors utilize their own data to estimate the model parameters, but we find that this model could be beneficial for other similar infectious diseases. Thus we have created this app which allows for public health officials to input parameters that were estimated based on their own data to predict the number of sick, exposed or infected individuals after  $t$  days.

### App Set up

We have three tabs in our app: the first tab is a home page that explains what the app is doing, the second tab allows for the user to set the model parameters, and the third tab outputs the results of the simulation.

1. What is this app doing: This tab provides an overview of the motivations behind the development of this model and why we are creating this application. This tab includes a flowchart of individuals from different states. The user may click on an action button that will show a pop up explaining the parameters and details of the flowchart.
2. Parameters: This tab has a drop down that allows the user to select how they would like to run the simulation. The user may choose between three options corresponding to the time of pandemic: Start, Middle, and Random. If the user is at the start of the pandemic the user must input the number of people in the exposed and infective groups. If the user is at the middle of the pandemic, the user must input the number of people in the exposed, infective, quarantined, diagnosed, and recovered groups. If a user selects the random option, they will have same option to input the number of people in each group as those who selected the middle of the pandemic. The user may choose to use predetermined population values used by Zhou et al. 2004, with no values less than 1.

The second column uses a reactive input which allows the user to change various transmission rates via sliders. The user should have estimates for their transmission rate per day ( $\beta$ , **beta**), death rate ( $\delta$ , **delta**), recovery rate ( $\gamma$ , **gamma**), transfer rate from the exposed to infected group ( $\epsilon$ , **epsilon**), transfer rate from the exposed to quarantined group ( $\lambda$ , **lambda**), transfer rate from the infective to the diagnosed group ( $\theta$ , **theta**), and the infectivity fraction ( $k$ , **k**) (Zhou et al., 2004). If the user selects "Random", we will randomly generate the rate parameters from a uniform distribution. If the user selects rate parameters that lie outside of our constraints, they will receive a message in the third tab letting them know the issue which has occurred.

The third column allows the user to choose whether they would like the infectivity rate to be a constant.

The default selection is “Yes”, which includes a slider for the infectivity fraction,  $k$ . If the user selects “No”, the slider disappears and assumes the infectivity fraction follows the formula  $\beta(t) = (kE + I)$  (Zhou et al., 2004).

3. Results: This will render an animated plot based on the inputs that the user selects on the parameters tab. The default plot will give a curve of the number of exposed individuals per day over the number of days. We have included a check box input on the left to allow the user to display the different curves for the other classes (infected, quarantined, diagnosed and recovered) based on their research needs.

A table will render and show the user the day for which the maximum number of people in the check boxed groups occurs. The groups which are displayed correspond to what the user checks in the input.

## Functions

- `get_diagnosed(e, i, q, j, r, epsilon, lambda, delta, theta, sigma, gamma, k, b)`: takes in numeric parameters based the number of people in the five different groups and the rate at which patients flow from one group to another. This function returns a data frame of the number of patients in each group for each for each passing day between 1 and 400 using the equations given by Zhou et al., 2004.
- `run_animation(dataframe)`: takes in a data frame filtered based on the population and returns an interactive plot of days against number of patients by group.
- `accumulate_by(data, variable)`: takes in a data parameter and variable parameter to accumulate by. The function returns accumulated data with an additional frame variable.
- `checkcondition(deathrate, epsilon, lambda, theta, gamma)`: takes in numeric variables based on the rate parameters. If the parameters are not within acceptable rate parameters, we will regenerate new parameter values from a uniform distribution. The function returns numeric parameters that fit the model.
- `create_table(dataframe)`: takes in a data frame parameter and returns a data frame which displays the day where we observed the highest membership for each group.

## Citation

Zhou, Y., Ma, Z., & Brauer, F. (2004). A discrete epidemic model for SARS transmission and control in China. *Mathematical and Computer Modelling*, 40(13), 1491– 1506. <https://doi.org/10.1016/j.mcm.2005.01.007>