COVID-19 Simulation

**Prepared for:**

PSA Final Project – INFO 6205 Spring 2021

**Chart

Description automatically generated with medium confidence**

**Prepared by:**

Siddhartha Raju (001084614)

Gautham Rajsimha Pulipati (001572432)

Naveen Kumar Buddhala (001582394)

Table of Contents

[Introduction 3](#_Toc69751807)

[Aim 3](#_Toc69751808)

[Project Details 4](#_Toc69751809)

[File Structure: 4](#_Toc69751810)

[**1.** **Config.cfg** 4](#_Toc69751811)

[**2.** **Config.py** 4](#_Toc69751812)

[**3.** **ConfigUtil.py** 4](#_Toc69751813)

[**4.** **Person.py** 4](#_Toc69751814)

[**5.** **PersonUtil.py** 5](#_Toc69751815)

[**6.** **DataUtil.py** 5](#_Toc69751816)

[**7.** **SimulationData.py** 5](#_Toc69751817)

[**8.** **Main.py** 5](#_Toc69751818)

[**9.** **MainFrame.py** 5](#_Toc69751819)

[Factors: 5](#_Toc69751820)

[**1.** **K factor** 5](#_Toc69751821)

[**2.** **Mask** 5](#_Toc69751822)

[**3.** **Quarantine** 6](#_Toc69751823)

[**4.** **Vaccine** 6](#_Toc69751824)

[**5.** **Days contagious** 6](#_Toc69751825)

[**6.** **Initial infected percentage** 6](#_Toc69751826)

[Implementation 6](#_Toc69751827)

[R factor 6](#_Toc69751828)

[Output 9](#_Toc69751829)

[Analysis 10](#_Toc69751830)

[Conclusion 10](#_Toc69751831)

[Unit Testing 11](#_Toc69751832)

[**1)** **Test\_Config.py** 11](#_Toc69751833)

[**2)** **Test\_ConfigUtil.py** 11](#_Toc69751834)

[**3)** **Test\_DataUtil.py** 12](#_Toc69751835)

[**4)** **Test\_person.py** 12](#_Toc69751836)

[**5)** **Test\_PersonUtil.py** 13](#_Toc69751837)

[**6)** **Test\_SimulationData.py** 13](#_Toc69751838)

# Introduction

Simulation model of a disease projects how [infectious diseases](https://en.wikipedia.org/wiki/Infectious_diseases) progress i.e how its components like transmission, spread and control of any infection grows over time and how it may strongly or weakly effect the dynamics of an epidemic. It also helps to inform [public health](https://en.wikipedia.org/wiki/Public_health) interventions.

Simulation model gives a pictoral or graphical way and it involves many factors, actors and components.

This is a new method of understanding and teaching about any disease outbreaks.

This can save money,lives and time by allowing us to test it without even applying it to real world.

Our simulation model fits for a particular pattern which is a general method involving R and K factor of the disease and several other factors of its spread.

# Aim

Simulate the spread of a virus with respect to several factors and introducing various control measures later during the simulation and finally analysing the virus.

Simulations will take into account:

* The R factor and k factor of the disease
* The usage and effectiveness of masks
* The prevalence of testing and contact tracing
* The availability and efficacy of the vaccine
* Any barriers to entry (including quarantining) into the subject area.

Building a User Interface to set several configuration values involved for respective diseases and simulating over a population and finally analysing through various graphs.

# Project Details

This disease simulation model is implemented in Python using TkInter and numpy library for probability association and matplotlib library for plotting the graph. The project consists of the following things

## File Structure:

### **Config.cfg**

* 1. The configuration file for the project can be found at the following location ~/config/config.cfg. The config file is used to set the default values for various diseases.
  2. In case no values are provided through the interface, the application will self-feed the values from this configuration file.

### **Config.py**

* 1. This is a Singleton class, and it contains all the configuration values required to simulate the disease
  2. It has the following properties:
  3. property-name
  4. population
  5. initial-infected
  6. r-factor
  7. k-factor
  8. days-contagious
  9. mask introduction timelines
  10. mask effectiveness
  11. mask usage percentage
  12. quarantine introduction timelines
  13. quarantine effectiveness
  14. quarantine usage percentage
  15. vaccine introduction timelines
  16. vaccine effectiveness
  17. vaccine usage percentage

### **ConfigUtil.py**

* 1. This is a Singleton class, and it contains the methods to load config file and get config values based on section and key passed.

### **Person.py**

This Class contains all the person related information.

* 1. It contains the following properties along with all their respective getters and setters:
  2. id
  3. age
  4. gender
  5. x-coordinate
  6. y-coordinate
  7. vaccinated
  8. quarantined
  9. mask-usage
  10. infected
  11. recovered
  12. deceased
  13. recovery-days
  14. can-infect-others

### **PersonUtil.py**

* 1. The class contains method to perform update operations of the Person object like updating location and status quo about mask, quarantine and infection. The class is used for updating various parameters of a person object during simulation.

### **DataUtil.py**

* 1. The class contains methods to perform data related operations like counting infected people, recovered people.

### **SimulationData.py**

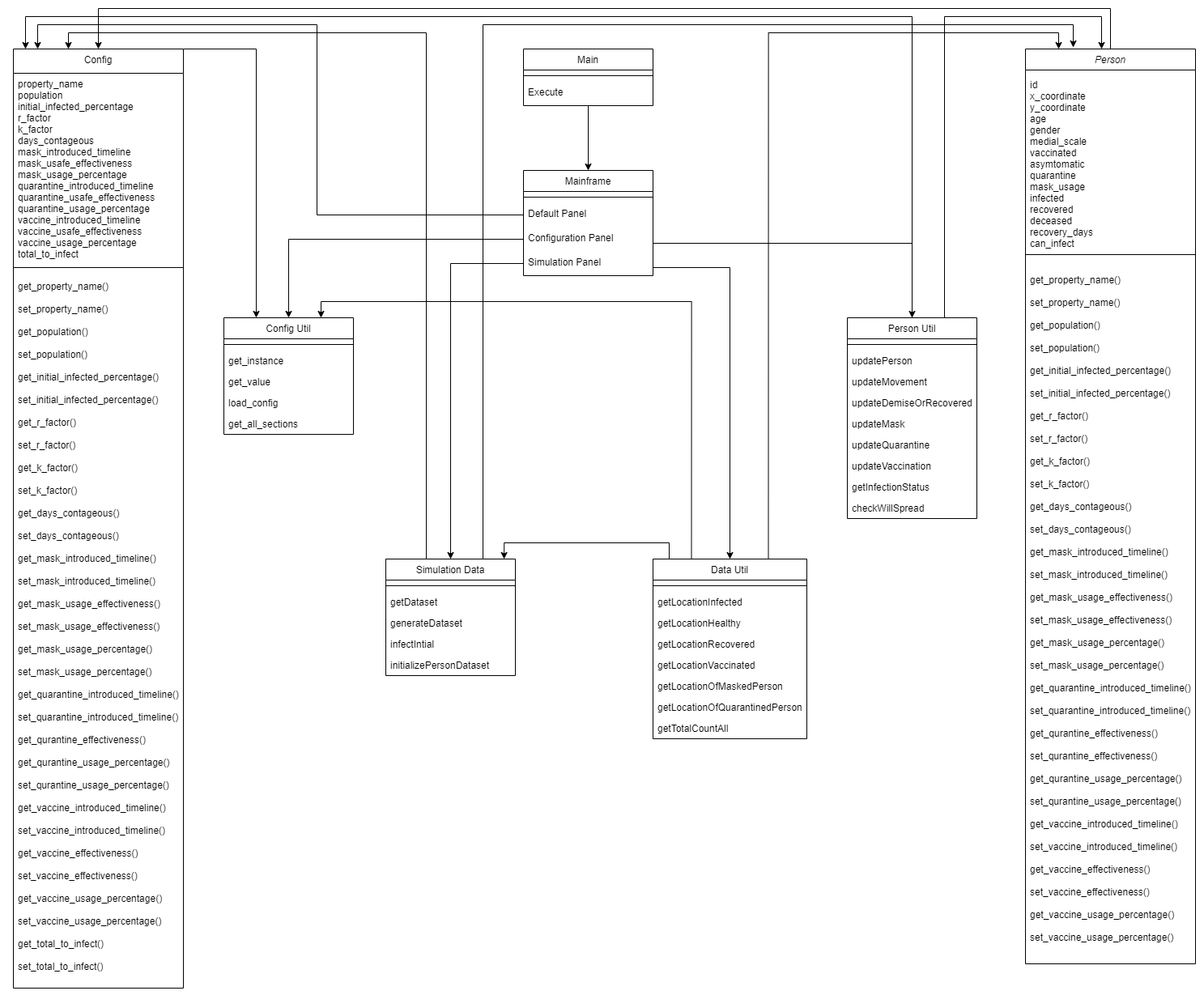
* 1. This class is used to setup the initial environment for simulation. It generates the initial dataset which is consumed during simulation for further processing.

### **Main.py**

* 1. This is the main class which is executed at the program execution

### **MainFrame.py**

* 1. This class consumes the data provided from Simulation Data and use it to visualize UI which contains Tkinter screens and graph for the user interface.



1 - Class diagram

## Factors:

The following factors have been considered while designing the disease simulation

### **K factor**

* 1. The dispersion parameter, it tells exactly how many are infected by each person i.e. in case of COVID-19 10-15 % of the population is responsible for 80% of the infections.
  2. These people are called super-spreaders. Whereas others, may be quarantined and infect none.

### **Mask**

* 1. 3 factors for mask are considered: introduction timeline, effectiveness, and percentage usage
  2. **Introduction timeline** specifies number of days after which the masks were introduced i.e. after certain time of start of simulation
  3. **Effectiveness** specifies how much does mask help us to protect from getting infected on a scale of 100, 0 signifies mask has no impact on in the virus to be infected
  4. **Percentage Usage** specifies how many from the current population are using the masks

### **Quarantine**

* 1. 3 factors for quarantine are considered: introduction timeline, effectiveness, and percentage usage
  2. **Introduction timeline** specifies number of days after which the people started to get quarantined i.e. after certain time of start of simulation
  3. **Effectiveness** specifies how much does quarantine help us to protect from getting infected on a scale of 100, 100 signifies if a person is quarantined, he cannot get infected
  4. **Percentage Usage** specifies how many from the current population are quarantining after getting infected

### **Vaccine**

* 1. 3 factors for vaccine are considered: introduction timeline, effectiveness, and percentage usage
  2. **Introduction timeline** specifies number of days after which the people started to get vaccinated i.e. after certain time of start of simulation
  3. **Effectiveness** specifies how much does vaccination help us to protect from getting infected on a scale of 100, 100 signifies if a person is vaccinated, he cannot get infected
  4. **Percentage Usage** specifies how many from the current population are vaccinated

### **Days contagious**

* 1. No. of days the person is infected, after catching the infection

### **Initial infected percentage**

* 1. Percentage of total population to be infected at the beginning of simulation

# Implementation

Factors considered:

## R factor

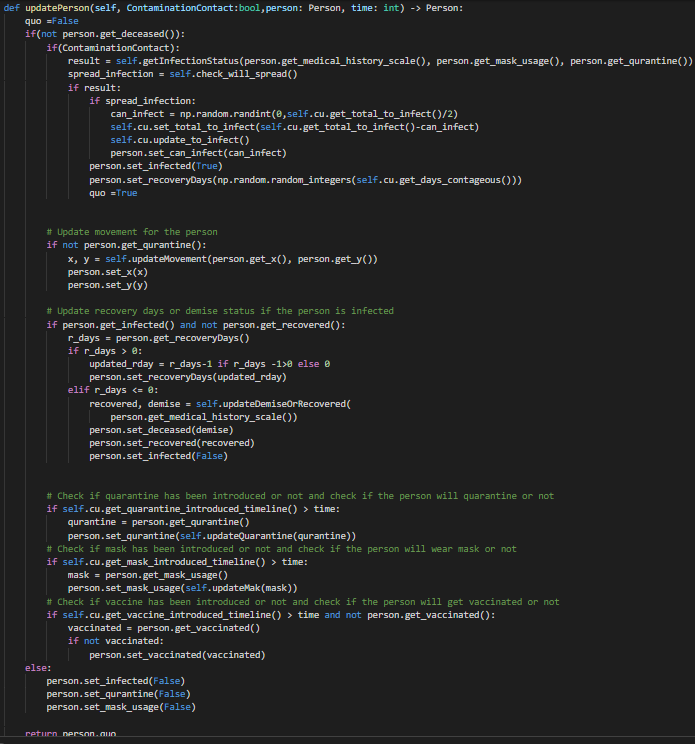
The average number of infections by one infected person.

**GUI**

There is a main panel which has 2 buttons: Set configuration and Start Simulation

These 2 buttons when clicked opens 2 other panels, one has all the default set configuration details which can be reset as per the user, while the other starts the simulation process, shows the space where the population is and how the infections take place. The panel also shows the graphs to understand how the virus is being spread as per the factors set.

For every cycle-time, for the complete dataset of persons, the below function is being called which is the main algorithm.



This function does the complete work of updating a person for the current cycle-time.

Its computational order:

If the person is alive

If the person is in close contact of any other infected person

If yes : checking their infection status whether the person will get infected

If yes: infecting and updating how many others the person can infect (K factor)

Setting randomly for how many days the person will remain infected

If the person is not quarantined

Updating the person’s movement

If the person is infected and not yet recovered: Updating the person’s recovery day’s left

If recovery days are completed: checking if the person will come under the probability of dead

If mask has been introduced: checking if the person will wear it

If quarantine has been introduced: checking if the person will quarantine

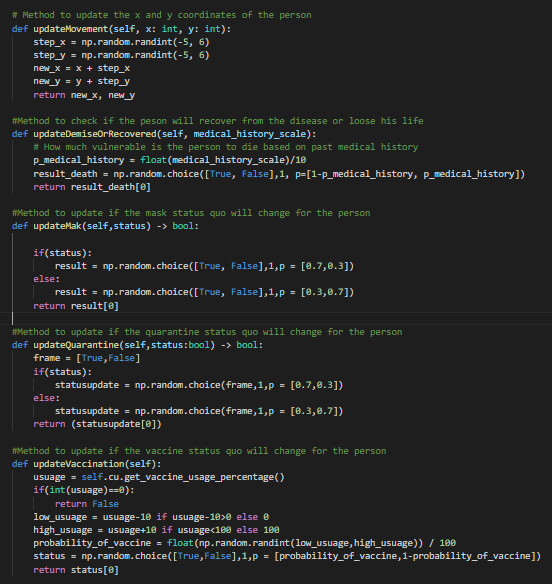
If vaccine has been introduced: checking if the person has taken the vaccination

Below are the various functions where we compute all the above-mentioned operations.

Every value required is taken from the config, set by the user, and randomizing the values in its surroundings.

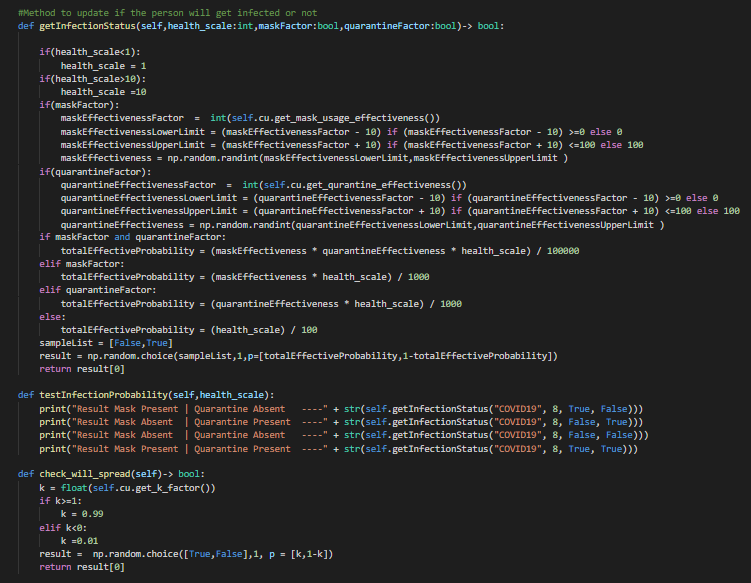
For movement: we are randomly moving the person in the space in between -5 to 5 for both the coordinates

For checking if the person dies of covid, we considered the health scale, how immune the person is at the time of infection, together with the fatality rate of the virus specified in the config.



For getting the infection status, we consider all the factors responsible, such as checking if the person has good immunity levels and is healthy, the mask factor, quarantine factor, vaccination factor, all the three separately as well as combination of them.

For checking if the person will spread, we considered the K-factor, the main factor that is responsible for the COVID-19 infection rate, i.e. the super spreaders, which is about 0.1, which says 10% of the infected people are responsible for the overall 90% of the infections.



# Output

**A picture containing diagram

Description automatically generated**

**2**- Entry Screen

**Graphical user interface, application, Word

Description automatically generated**

**3** - Configuration Screen

**Chart

Description automatically generated with medium confidence**

4- Simulation Screen

# Analysis

## Performance:

**Time Complexity** : The alogorithm worst case complexity is O(N\*T) where N is the number of people and T is the time the simulation is run.

Space Complexity : The algorithm complexity is O(N+T) where N is the number people to simulate and T is time the simulation is run. Additional T value is needed to store the state of N at time T for plotting graphs

## Simulation:

**Case 1 : Mask, quarantine and vaccine are present**

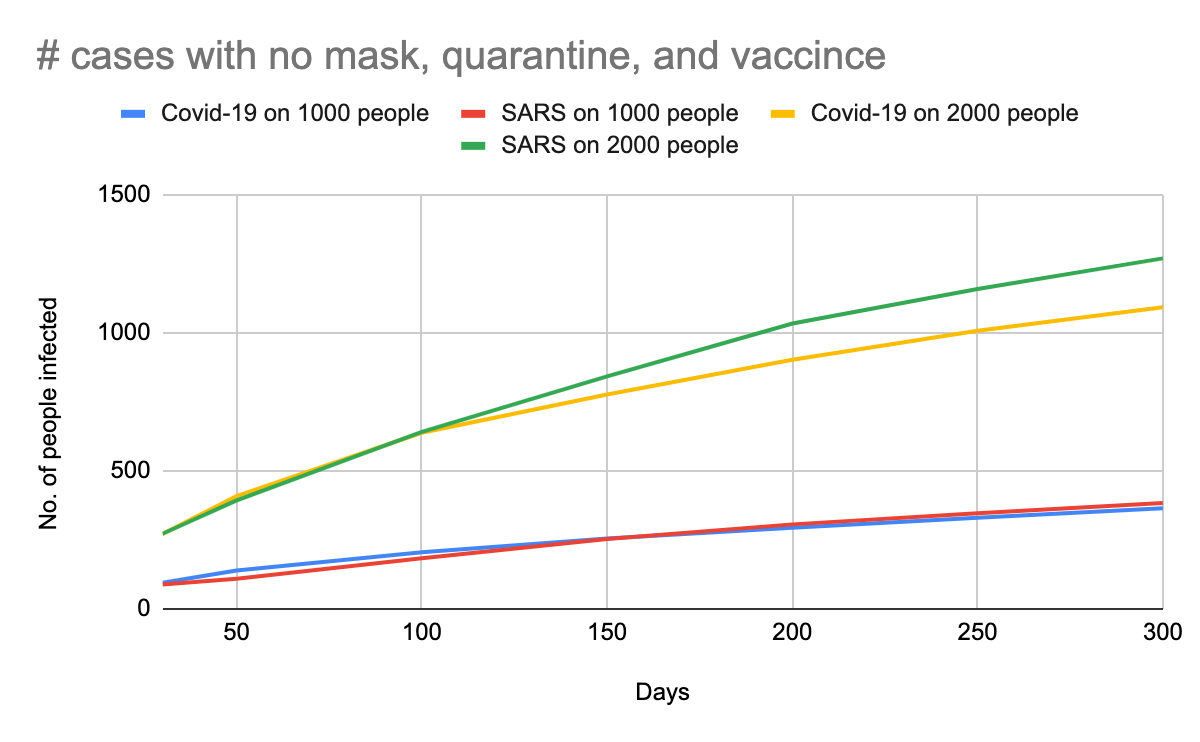
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Days | Covid-19 on 1000 people | SARS on 1000 people | Covid-19 on 2000 people | SARS on 2000 people |
| 30 | 80 | 87 | 276 | 255 |
| 50 | 109 | 107 | 384 | 392 |
| 100 | 158 | 180 | 593 | 633 |
| 150 | 201 | 236 | 764 | 813 |
| 200 | 221 | 261 | 845 | 945 |
| 250 | 232 | 287 | 911 | 1008 |
| 300 | 243 | 296 | 938 | 1059 |

Chart, line chart

Description automatically generated

**Case 2 : No Mask, No quarantine and No vaccine are present**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Days | Covid-19 on 1000 people | SARS on 1000 people | Covid-19 on 2000 people | SARS on 2000 people |
| 30 | 95 | 88 | 271 | 272 |
| 50 | 139 | 109 | 408 | 393 |
| 100 | 205 | 183 | 638 | 641 |
| 150 | 255 | 253 | 776 | 842 |
| 200 | 294 | 306 | 902 | 1033 |
| 250 | 330 | 346 | 1007 | 1158 |
| 300 | 364 | 383 | 1092 | 1269 |



# Conclusion

For time T(Days) and number of infected population N

|  |  |  |
| --- | --- | --- |
| Disease | With Mask, Quarantine and Vaccination | Without Mask, Quarantine and Vaccination |
| Covid19 – 1000 Person | N ≅ 1.087670851 \* T | N ≅ 1.145563034\* T |
| Covid19 – 2000 Person | N ≅ 1.114069835\* T | N ≅ 1.133301742\* T |
| SARS – 1000 Person | N ≅ 1.270149541\* T | N ≅ 1.382468493\* T |
| SARS – 2000Person | N ≅ 1.280048267 \* T | N ≅ 1.394761819\* T |

# 

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

# Unit Testing

### **Test\_Config.py**

A picture containing text

Description automatically generated

### **Test\_ConfigUtil.py**

Text

Description automatically generated

### **Test\_DataUtil.py**

Graphical user interface, text

Description automatically generated

### **Test\_person.py**

Graphical user interface, text, application

Description automatically generated

### **Test\_PersonUtil.py**

Graphical user interface, text, application, chat or text message

Description automatically generated

### **Test\_SimulationData.py**

Text

Description automatically generated

# Rerefences

1) https://www.doh.wa.gov/Portals/1/Documents/1600/coronavirus/WearAClothFaceCovering.pdf

2) https://royalsociety.org/-/media/policy/projects/set-c/set-covid-19-R-estimates.pdf

3) https://www.healthline.com/health/r-nought-reproduction-number#covid-19-r-0

4) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2851497/

5) https://www.cdc.gov/sars/clinical/respirators.html

6) https://jbiomedsci.biomedcentral.com/articles/10.1186/s12929-020-00695-2

7) https://sph.umich.edu/pursuit/2020posts/how-scientists-quantify-outbreaks.html