
covid-19-simulations Documentation

Arjun Savel, Adam Bittenson

Apr 01, 2021

| | | |
|----------|-----------------------------------|-----------|
| 1 | About covid-19-simulations | 1 |
| 2 | Installing from source | 3 |
| 3 | API | 5 |
| 3.1 | zero_d | 5 |
| 3.2 | one_d | 6 |
| 3.3 | two_d | 7 |
| | Python Module Index | 9 |
| | Index | 11 |

CHAPTER 1

About covid-19-simulations

We're seeking to explore problems related to the ongoing COVID-19 pandemic. While an agents-based approach limited by computational tractability will be inherently inaccurate to some degree, we hope that we may be able to gain intuition for questions such as:

- How effective are lockdowns at different points in the spread of disease?
- To what extent might we be under-reporting cases?
- Does quarantining at-risk individuals exclusively for a period of time reduce overall harm?

If you find something that doesn't work or doesn't make sense, please file an issue on GitHub. If you're interested in helping out with the project, we'd love to have you on board! Get in contact (asavel@berkeley.edu) or just fork the repository and start working — we'll review your pull request when you're ready!

CHAPTER 2

Installing from source

This code is developed on [GitHub](#). If you received the code as a tarball or zip, skip to below the `git clone` line. I would recommending performing the below steps in a fresh conda environment.

```
python3 -m pip install -U pip
python3 -m pip install -U setuptools setuptools_scm pep517
git clone https://github.com/arjunsavel/covid-hacks-2020.git
cd covid-hacks-2020
pip install -r requirements.txt
```


3.1 zero_d

`covid_19_simulations.zero_d.infect(df, trans_rate, day_name)`

Simulates a single day of infection. Each infected person

NOTE: a 0 counts as infected, while a 1 is healthy.

Inputs:

df [(pandas DataFrame) object holding all values of infected people. Each] column of “infected day _” corresponds to a different day, with “_” being some integer or float. The “name” column assigns a name to each object, independent of index. In the infected columns, a 0 counts as infected, while a 1 is healthy.

trans_rate [(float) rate of transmission between individuals. infection] is performed in a probabilistic manner, casting it as a draw from a binomial distribution with a rate of 1 - trans_rate.

day_name [(float or int) the day of this infection, used to create a new] column in the dataframe tracking the day’s infections.

Outputs:

df [(pandas DataFrame) object, same as the input, but with a new column] holding this day’s infected results.

`covid_19_simulations.zero_d.simulate(N, trans_rate, t_steps, N_initial)`

Simulates an infection run.

Inputs: N : (int) number of individuals in the system. trans_rate : (float) rate of transmission between individuals. infection

is performed in a probabilistic manner, casting it as a draw from a binomial distribution with a rate of 1 - trans_rate.

t_steps : (int) number of time steps (“days”) to consider. N_initial : (int) number of initially infected individuals.

Outputs:

df [(pandas DataFrame) object holding all values of infected people. Each] column of “infected day _” corresponds to a different day, with “_” being some integer or float. The “name” column assigns a name to each object, independent of index. In the infected columns, a 0 counts as infected, while a 1 is healthy.

3.2 one_d

`covid_19_simulations.one_d.animate_histogram(df, title)`
animates a histogram.

adapted from https://matplotlib.org/gallery/animation/animated_histogram.html

`covid_19_simulations.one_d.infect1D(df, trans_rate, day_name, thresh, power)`
Simulates a single day of infection in 1D.

NOTE: a 0 counts as infected, while a 1 is healthy.

Inputs:

df [(pandas DataFrame) object holding all values of infected people. Each] column of “infected day _” corresponds to a different day, with “_” being some integer or float. The “name” column assigns a name to each object, independent of index. In the infected columns, a 0 counts as infected, while a 1 is healthy.

trans_rate [(float) rate of transmission between individuals. infection] is performed in a probabilistic manner, casting it as a draw from a binomial distribution with a rate of 1 - trans_rate.

day_name [(float or int) the day of this infection, used to create a new] column in the dataframe tracking the day’s infections.

Outputs:

df [(pandas DataFrame) object, same as the input, but with a new column] holding this day’s infected results.

`covid_19_simulations.one_d.simulate1D(N, trans_rate, t_steps, N_initial, thresh, power, distrib_pop, distrib_infec, kwargs_for_pop={}, kwargs_for_infec={})`

Simulates an infection run in 1D.

Inputs: N : (int) number of individuals in the system. trans_rate : (float) rate of transmission between individuals. infection

is performed in a probabilistic manner, casting it as a draw from a binomial distribution with a rate of 1 - trans_rate.

t_steps : (int) number of time steps (“days”) to consider. N_initial : (int) number of initially infected individuals. thresh : (float) distance less than which infection is transmitted at the trans_rate;

that is, less than which this function returns a value of 1. At a distance greater than this, this function returns 1/distance^power.

power [(float) Greater than 0. Power to which the multiplier falls off if the distance] is greater than some threshold.

distrib_pop : (func) distribution function to determine how individuals are initialized. distrib_infec : (func) distribution function to determine how initial infections are initialized. kwargs_for_pop : (dict) keyword arguments passed to the distrib_pop distribution type.

Size not included.

kwargs_for_infec [(dict) keyword arguments passed to the distrib_infect distribution type. Size not included.

Outputs:

df [(pandas DataFrame) object holding all values of infected people. Each] column of “infected day _” corresponds to a different day, with “_” being some integer or float. The “name” column assigns a name to each object, independent of index. In the infected columns, a 0 counts as infected, while a 1 is healthy.

3.3 two_d

`covid_19_simulations.two_d.distance` (*frame, ind1, ind2*)

Just finding the distance between two rows and their x-y pairs.

`covid_19_simulations.two_d.do_multiplier` (*x, y, power, thresh, df_sorted, df2D_test, farthest_calc*)

`covid_19_simulations.two_d.find_first`

return the index of the first occurrence of item in vec

`covid_19_simulations.two_d.infect2D` (*df, trans_rate, day_name, thresh, power, df_sorted, farthest_calc*)

Simulates a single day of infection in 1D.

NOTE: a 0 counts as infected, while a 1 is healthy.

Inputs:

df [(pandas DataFrame) object holding all values of infected people. Each] column of “infected day _” corresponds to a different day, with “_” being some integer or float. The “name” column assigns a name to each object, independent of index. In the infected columns, a 0 counts as infected, while a 1 is healthy.

trans_rate [(float) rate of transmission between individuals. infection] is performed in a probabilistic manner, casting it as a draw from a binomial distribution with a rate of 1 - trans_rate.

day_name [(float or int) the day of this infection, used to create a new] column in the dataframe tracking the day’s infections.

dist_matrix : (numpy.ndarray) distance matrix holding the distances between all individuals.

Outputs:

df [(pandas DataFrame) object, same as the input, but with a new column] holding this day’s infected results.

`covid_19_simulations.two_d.initialize_pop_2D` (*N, distrib, **kwargs*)

This will change once we have the U.S. map.

distrib: size is not a thing again.

`covid_19_simulations.two_d.simulate2D` (*N, trans_rate, t_steps, N_initial, thresh, power, distrib_pop, distrib_infec, kwargs_for_pop={}, kwargs_for_infec={}*)

Simulates an infection run in 1D.

Inputs: *N* : (int) number of individuals in the system. *trans_rate* : (float) rate of transmission between individuals. *infection*

is performed in a probabilistic manner, casting it as a draw from a binomial distribution with a rate of $1 - \text{trans_rate}$.

t_steps : (int) number of time steps (“days”) to consider. **N_initial** : (int) number of initially infected individuals. **thresh** : (float) distance less than which infection is transmitted at the **trans_rate**;

that is, less than which this function returns a value of 1. At a distance greater than this, this function returns $1/\text{distance}^{\text{power}}$.

power [(float) Greater than 0. Power to which the multiplier falls off if the distance] is greater than some threshold.

distrib_pop : (func) distribution function to determine how individuals are initialized. **distrib_infec** : (func) distribution function to determine how initial infections are initialized. **kwargs_for_pop** : (dict) keyword arguments passed to the **distrib_pop** distribution type.

Size not included.

kwargs_for_infec [(dict) keyword arguments passed to the **distrib_infec** distribution type. Size not] included.

Outputs:

df [(pandas DataFrame) object holding all values of infected people. Each] column of “infected day _” corresponds to a different day, with “_” being some integer or float. The “name” column assigns a name to each object, independent of index. In the infected columns, a 0 counts as infected, while a 1 is healthy.

C

`covid_19_simulations.one_d`, 6
`covid_19_simulations.two_d`, 7
`covid_19_simulations.zero_d`, 5

A

`animate_histogram()` (in module `covid_19_simulations.one_d`), 6

C

`covid_19_simulations.one_d(module)`, 6

`covid_19_simulations.two_d(module)`, 7

`covid_19_simulations.zero_d(module)`, 5

D

`distance()` (in module `covid_19_simulations.two_d`), 7

`do_multiplier()` (in module `covid_19_simulations.two_d`), 7

F

`find_first` (in module `covid_19_simulations.two_d`), 7

I

`infect()` (in module `covid_19_simulations.zero_d`), 5

`infect1D()` (in module `covid_19_simulations.one_d`), 6

`infect2D()` (in module `covid_19_simulations.two_d`), 7

`initialize_pop_2D()` (in module `covid_19_simulations.two_d`), 7

S

`simulate()` (in module `covid_19_simulations.zero_d`), 5

`simulate1D()` (in module `covid_19_simulations.one_d`), 6

`simulate2D()` (in module `covid_19_simulations.two_d`), 7