

# Testing COVID-19 Models

Getting Important Work Done in a Hurry

Christian Wiswell

PNSQC 2020



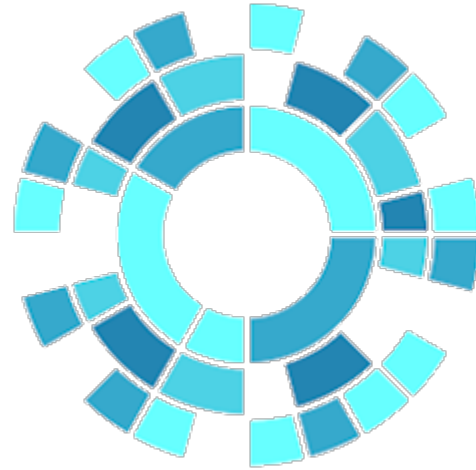
# Background: Me

- Seasoned Test Professional
- Different Employers
- Then IDM...



# Background: Institute for Disease Modeling

- 10 years or so
- 100+ papers
- 100 people

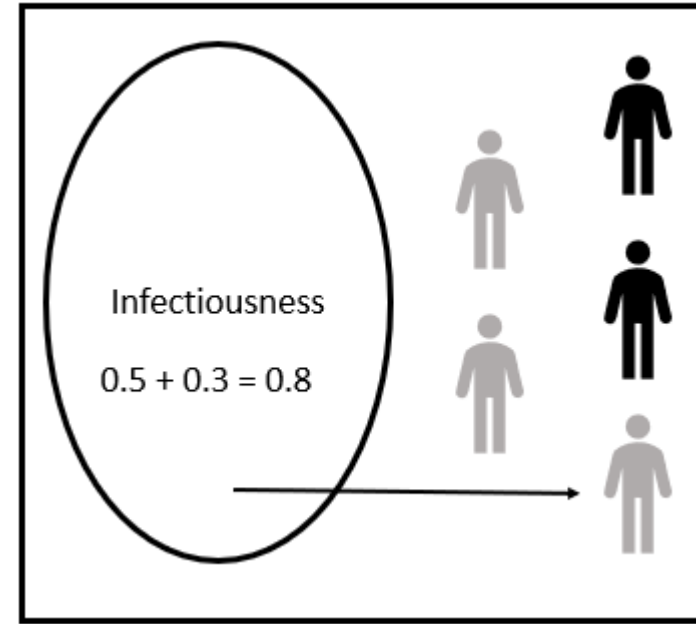
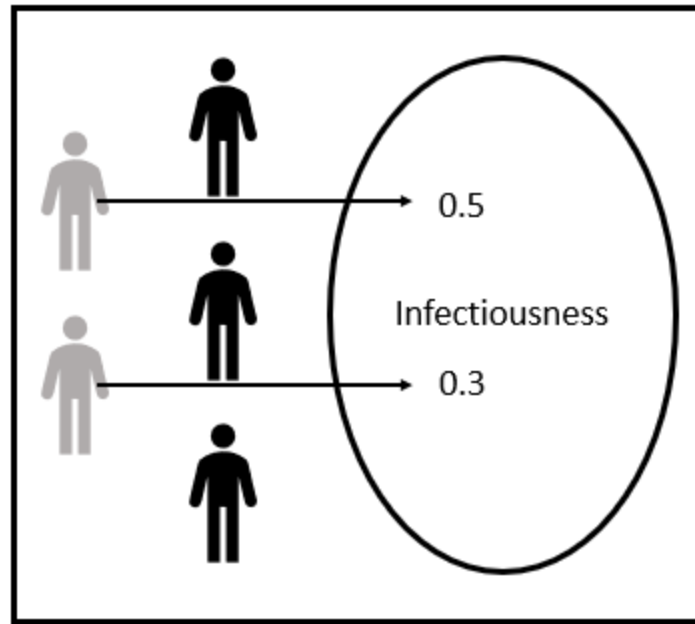


# Background: Epidemiological MODeling

- EMOD
- Stochastic, Agent-based model
- Richly Configurable with JSON



# EMOD: Transmission pools



# Testing Scientific Software

- True behavior unknown
- Agent-based software, simulation-based reports
- Results are stochastic by design



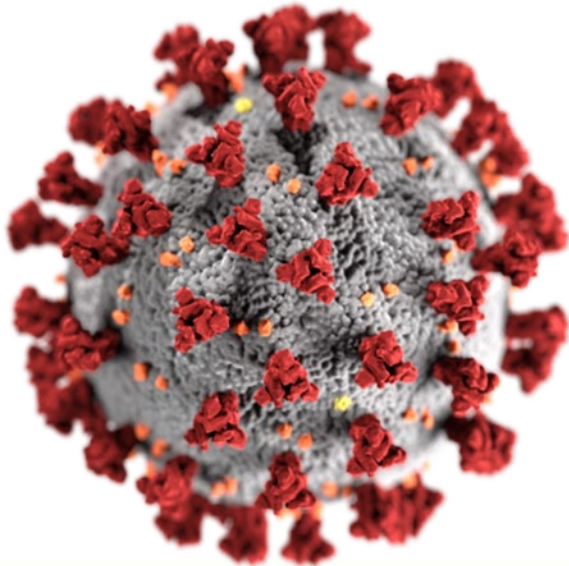
# Testing EMOD

- Configurability, not “truth”
- Heavyweight logging of individuals
- Statistical test techniques with random seeds



# Challenge: COVID-19

- Warnings in November 2019
- Modeling scenarios incompatible
- New scientist, new methods





# Covasim: Out of my Comfort Zone

- Unknown dependencies
- “Just generate coverage numbers”

EMOD	Covasim
Configuration classes and files	Properties of a python object
100s of regression scenarios	Some sample scripts
100+ mathematically verified feature tests	10ish plotting scripts
Exhaustive model documentation	A short README.md



# Compromises

- I can run the model like you.
- You can document parameters.
- I can live with “relative validation”

```
import covasim as cv
sim = cv.Sim()
sim.run()
sim.plot()
```



# Test implementation

- Parameters -> Support classes
- Outline many tests
- Implement with relative validation



# Test case 1: Random seed

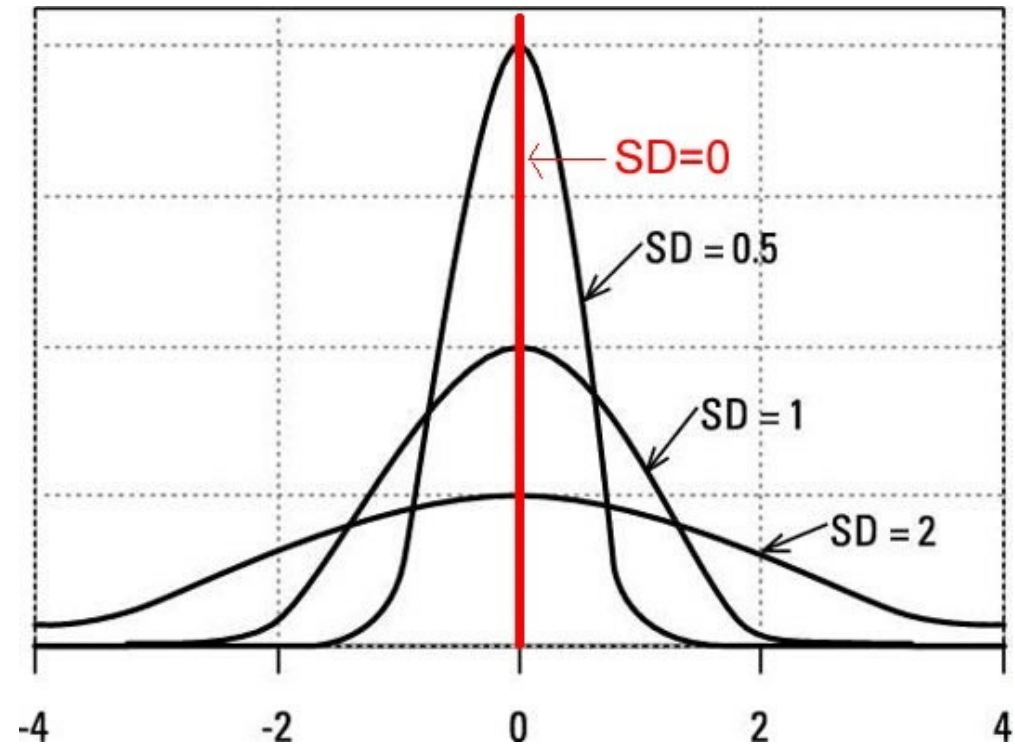
- Verifies reproducibility
- Enables static regressions

```
def test_random_seed(self):  
    """  
    Run two simulations with the same seed  
    and one with a different one. Something  
    randomly drawn (number of persons infected  
    day 2) is identical in the first two and  
    different in the third  
    """
```



# Test case 2: Variance

- `test_exposure_to_infectiousness_delay_deviation_scaling`
- Loop through `std_devs`
- Check first, highest, last
- Compare to next higher



# Test case 3: Duration without Variance

- Builds on previous test
- With `std_dev 0`, loop through mean
- No infectious until target, then all



# Testing scientific software revisited

- True behavior -> Configurability
- Stochasticity -> Statistical tests / Relative validation
- Reportability -> Heavyweight debugging / Careful crafting



# Conclusions

- You are the expert for software
- Your SME is the expert for science
- Find your anchor point and collaborate

