Echoes of the Spanish Lady: Navigating the Maze of Epidemic Models

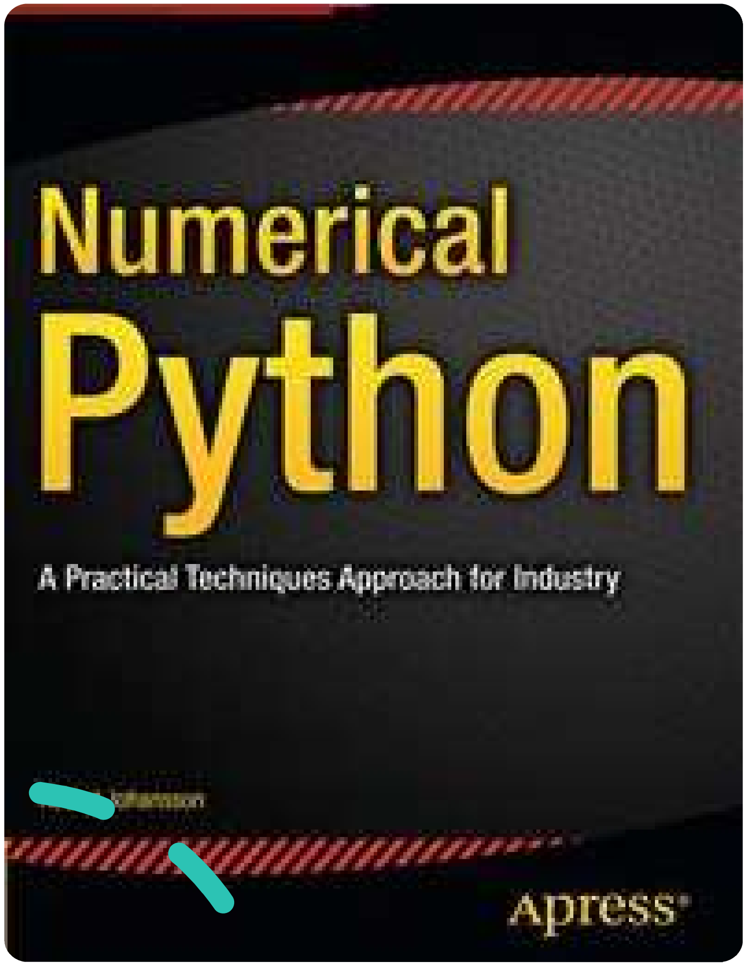
A Look at the 1918 Flu Epidemic Models

Project Details June 7 Update

June 7 Update

Adding structure and Planning Goal for each week

Becoming more comfortable with GitHub, Jupyter, Python while working toward goals



Additional Resources

Numerical Python Robert Johansson

The Extrapolation of First Order Methods for Parabolic Partial Differential Equations

J.D. Lawson and J. Ll. Morris SIAM 1978

Recent Progress of the combinatorial diameter of polytopes and simplical complexes

Francisco Santos Jul 2013

Computational Science Project Schedule

Week of June 7

- Defining the project scope and goals.
  - Clarity and feasibility of the project scope and goals.
  - Completeness of the project proposal

For June 21

- Literature review and research on existing methods
  - Breadth and depth of the literature review
  - Relevance of the literature to the project
  - Evidence of understanding the existing methods

For June 30

- Developing the computational method/algorithm:
  - Quality and completeness of the implementation
  - Evidence of testing and debugging of the method/algorithm
  - Evidence of understanding and implementation of the method/algorithm

Future Goals and Objectives

July 14

- Testing and optimization of the method/algorithm:
  - Evidence of understanding and implementation of optimization techniques
  - Evidence of testing and debugging of the optimization techniques
  - Scalability and computational efficiency of the method.

July 28

- Verification and validation of the method/algorithm:
  - Evidence of understanding and implementation of verificiation and validation techniques
  - Evidence of testing and debugging of the optimization techniques
  - Quality and completeness of the results

August 11

- Final Report/Presentation Preparation
  - Clarity and organization of the final report/presentation
  - Quality and completeness of the visual aids
  - Effectiveness of the delivery

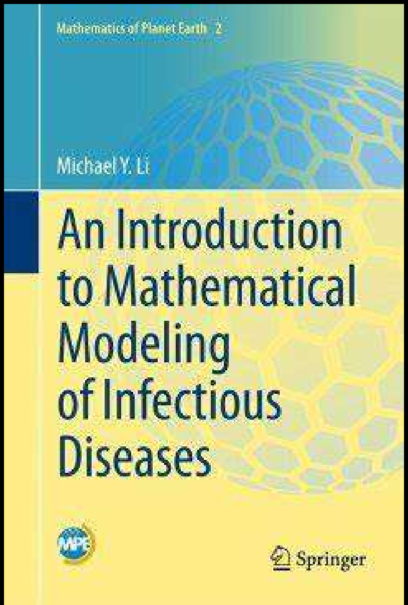
Next Steps

Anaconda/Jupyter Notebook

Workflow (developing a working procedure)

Echoes of the Spanish Lady: Navigating the Maze of Epidemic Models

A Look at the 1918 Flu Epidemic Models



Mathematics of Planet Earth 2

Michael Y. Li

An Introduction to Mathematical Modeling of Infectious Diseases

MPE

Springer

Mathematical Modeling of Infectious Diseases

Keywords: Infectious diseases, Mathematical modeling, Epidemiology, Public health

Project Focus

1. **Leveraging cellular automata for simulating epidemic models, specifically the SIR (Susceptible, Infected, Recovered) and SEIR (Susceptible, Exposed, Infected, Recovered) models.**

2. **The goal is to compare the efficiency and accuracy of these models with traditional techniques like Markov Chains, to develop more efficient and practical tools for simulating disease spread.**

3. **Additionally, the project aims to explore the use of machine learning for model refinement and seeks potential applications of these models beyond public health, such as in game development.**

Project Description & Scientific Problem

1. The scientific problem being addressed in this project is the efficient and accurate simulation of epidemic models, specifically the SIR and SEIR models, using parallel processing techniques and cellular automata.

2. Traditional techniques like Markov Chains can be computationally intensive and might not scale efficiently. The project aims to overcome the challenge of refining these models to better fit real-world observations without overfitting.

3. Ultimately, the project seeks to develop practical tools for simulating disease spread that can also be applied in diverse fields such as game development.

Scientific Conclusions

Through this project, we hope to achieve several scientific conclusions:

1. Demonstrate the effectiveness of parallel processing and cellular automata in simulating the SIR and SEIR epidemic models, potentially showcasing an alternative to traditional methods like Markov Chains that can be more efficient and scalable.

2. Determine the accuracy of these simulations compared to real-world data and understand the balance between model refinement and overfitting.

3. Explore the practicality and applicability of these simulation tools beyond academic research, such as their use in public health planning or game development.

4. Provide insights into the potential integration of machine learning techniques for automatic model refinement and adaptive simulation techniques.

5. Establish a groundwork for future research, such as the inclusion of more complex factors in epidemic models (like carrier states or vaccination rates) and the potential for a reverse-engineering approach to model selection.

Simulation & Data Analysis

The simulation in this project will involve the use of cellular automata to model the spread of an infectious disease based on SIR and SEIR models. We'll represent the population as a two-dimensional grid, where each cell corresponds to an individual and their state: Susceptible, Exposed, Infected, or Recovered. The initial state of each cell will be randomly assigned to reflect a realistic distribution of population states at the start of a disease spread.

This simulation will run on a parallel computing environment, leveraging technologies like MPI and OpenMP. By assigning different portions of the grid to multiple processes, we aim to enhance computational efficiency.

Once the simulation is running, it will apply the rules defined by the SIR and SEIR models for each time step. Each individual's state will evolve based on these rules, and the aggregate state of the population will be captured and analyzed.

In terms of data analysis, we will introduce the concept of an 'error function' that assesses the difference between the simulated data and real-world observations. This approach will allow us to refine the parameters of our model iteratively to improve its predictive accuracy while preventing overfitting.

Ultimately, we aim to generate a rich dataset that provides insight into the temporal dynamics of disease spread and the comparative efficiency and accuracy of SIR and SEIR models when implemented in parallel computing environments.

Algorithms & Numerical Methods

Several algorithms and numerical methods are anticipated to be applied:

1. **Cellular Automata**: This will be the preliminary method for modeling the disease spread. Each cell in a grid represents an individual, and their state (Susceptible, Exposed, Infected, Recovered) evolves according to the SIR/SEIR model rules.

2. **Numerical Integration**: As the SIR and SEIR models involve ordinary differential equations, numerical integration methods might be used for their solution, especially if we wish to simulate continuous time models.

3. **Random Number Generation**: Some initial states of individuals (cells) in the grid will be assigned randomly, which will require a reliable method for generating random numbers.

4. **Error Function and Optimization Algorithms**: To refine the model parameters, we will define an error function that quantifies the difference between simulated data and real-world observations. Then, we can use optimization algorithms (e.g., gradient descent, stochastic gradient descent, etc.) to adjust parameters and minimize this error.

5. **Machine Learning Algorithms**: Depending on the direction of the research, we might employ machine learning techniques to automate model refinement, or even reverse-engineer the best-fitting model based on observed data.

Programing & Development

The project will be developed in Python and the programming work will likely involve the following:

1. **Setting Up the Simulation Environment**: This would involve setting up the cellular automata grid, initializing individual states randomly according to the SIR/SEIR model rules, and defining the transition functions for changing states. Libraries such as NumPy can be used to efficiently handle arrays representing the population grid.

2. **Implementing SIR/SEIR Models**: You would be writing functions or classes to represent the SIR and SEIR models, which will include the associated differential equations. These will be used to determine state transitions in the simulation. SciPy's integrate module can be utilized for numerically solving these equations.

3. **Running the Simulation**: This would involve iterating over the defined time steps, updating the state of each cell according to the SIR/SEIR rules at each step, and storing or outputting the state of the entire system at each step for subsequent analysis.

4. **Data Analysis and Model Refinement**: Here, we write code to analyze the results of the simulation, such as computing the error between the simulated data and real-world data. We also use this analysis to iteratively refine model parameters. Libraries like Pandas for data handling and Matplotlib for visualization can aid in this process.

5. **Machine Learning Integration**: Depending on the project's scope, we could also implement machine learning models using libraries such as Scikit-Learn or TensorFlow. These models could help automate the process of model refinement or even potentially reverse-engineer the best fitting model based on real-world data.

Python's extensive standard library and the availability of scientific computing packages like NumPy, Scipy, Pandas, and Scikit-learn make it a great choice for such a project.

Relevant Papers

Deep-Data-Driven Neural Networks for COVID-19 Vaccine Efficacy by Thomas K. Torku, Abdul Q.M. Khaliq & Khaled M. Furati
This paper develops a vaccination model with an efficacy rate, using a hybrid neural network approach for reliable daily case predictions, demonstrating through data-driven simulations that higher vaccination rates with more effective vaccines decrease the infectiousness and basic reproduction number, using data from Tennessee as a case study.

"Modeling Infectious Diseases in Humans and Animals" by Keeling, M. J., & Rohani, P. (2008, Princeton University Press).
This is not a paper but a comprehensive book that provides a thorough introduction to the mathematical modeling of infectious diseases, including both the SIR and SEIR models.

"Parallel discrete event simulation of SEIR epidemic models"
by Perumalla, K. S., & Seal, S. K. (2013, in Winter Simulations Conference (WSC)).
This paper presents an approach for accelerating the simulation of SEIR epidemic models using parallel discrete event simulation. Although you mentioned that parallel processing is not important for this project, this paper can provide useful insights on how such models are typically parallelized.

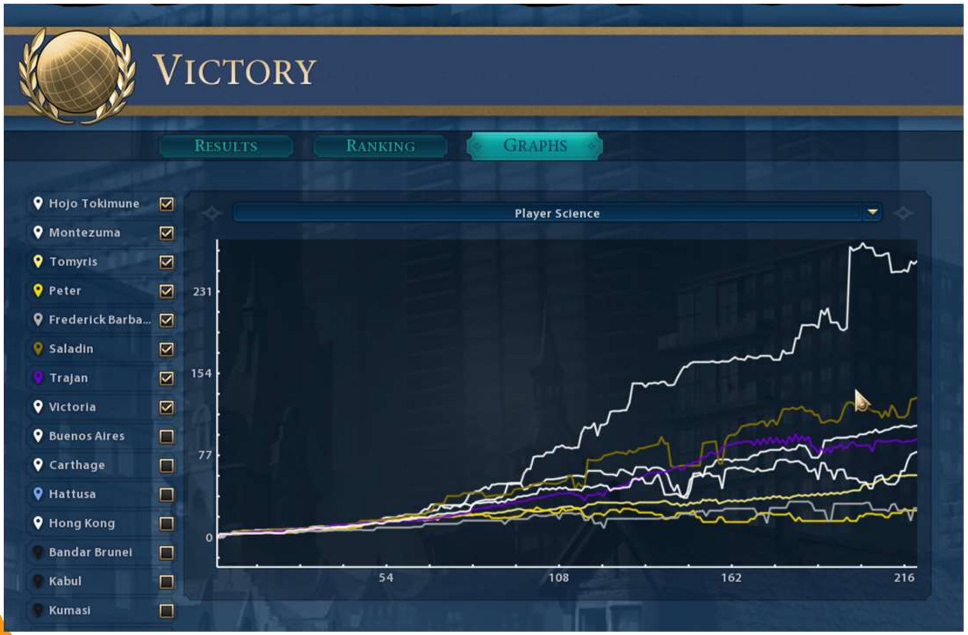
"**An efficient solution to the inverse problem of epidemic models using sequential Monte Carlo"**
by Cantwell, C. J., Ruktanonchai, N., & Tildesley, M. J. (2019, PloS One). This paper addresses the reverse problem in epidemic models, i.e., determining parameters given observed data, using a sequential Monte Carlo method. The methodology proposed could provide insights for your own reverse-engineering approach.

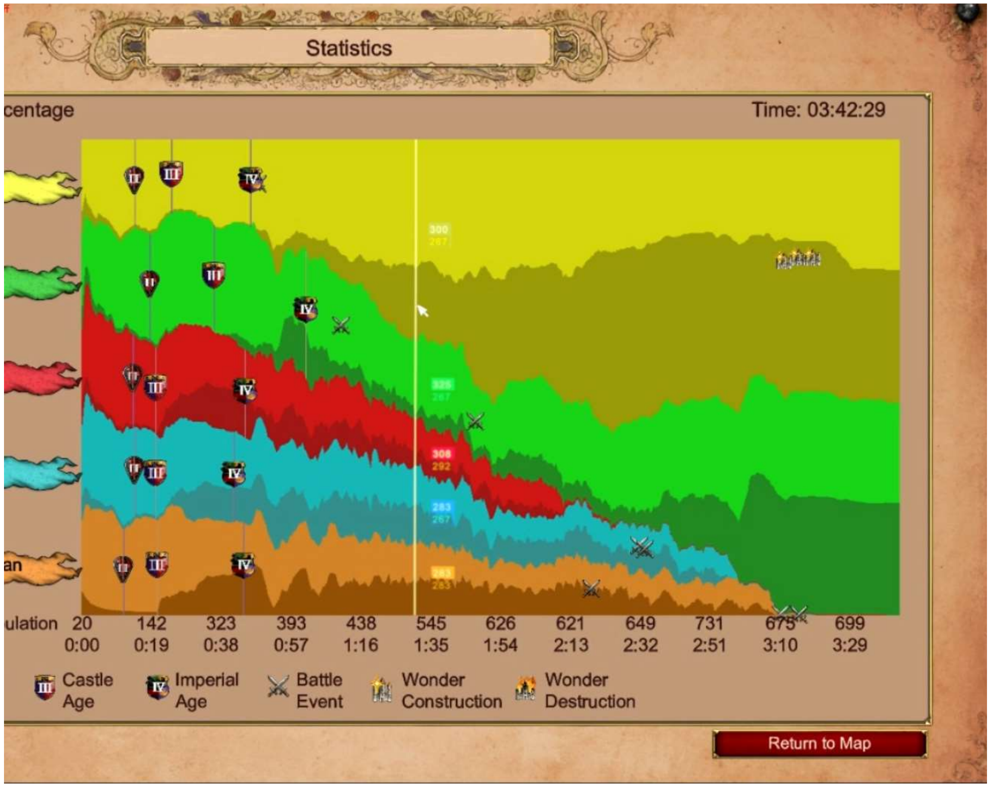
Introduction to The Spanish Lady Problem

Andrew Becker

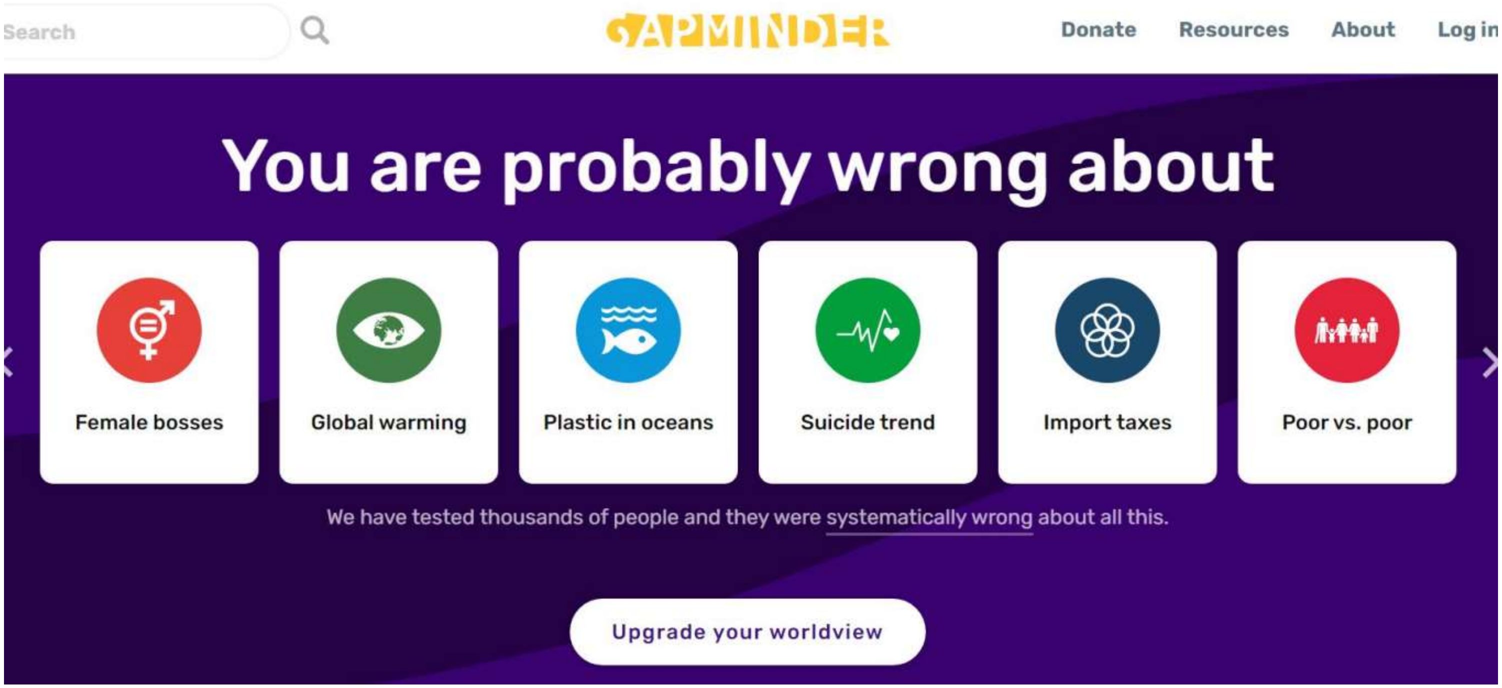
Video Games as Models

Civilization & Age of Empires Game Over Screen Graphs



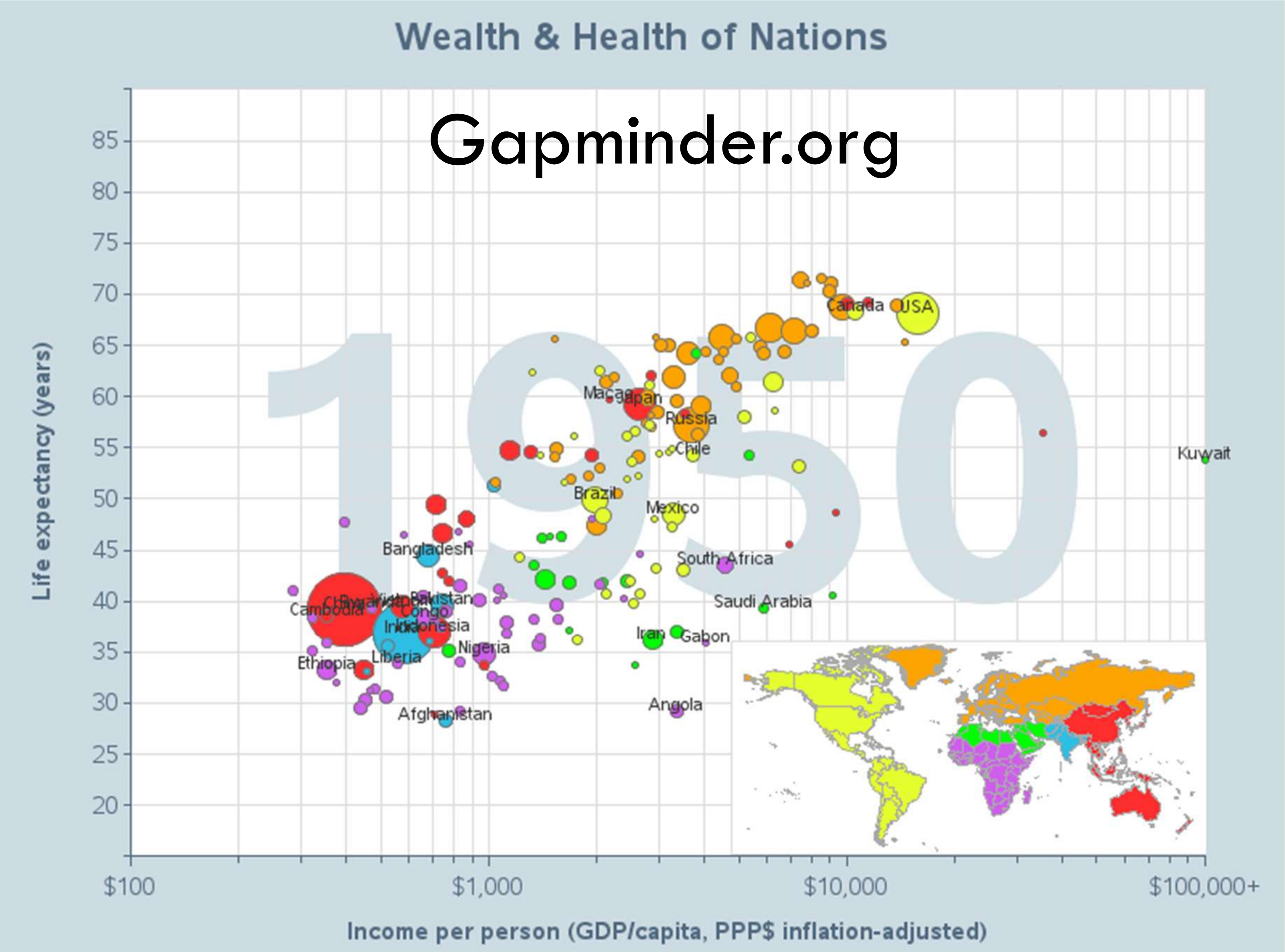


Inspired by Gapminder.org (Hans Rosling)



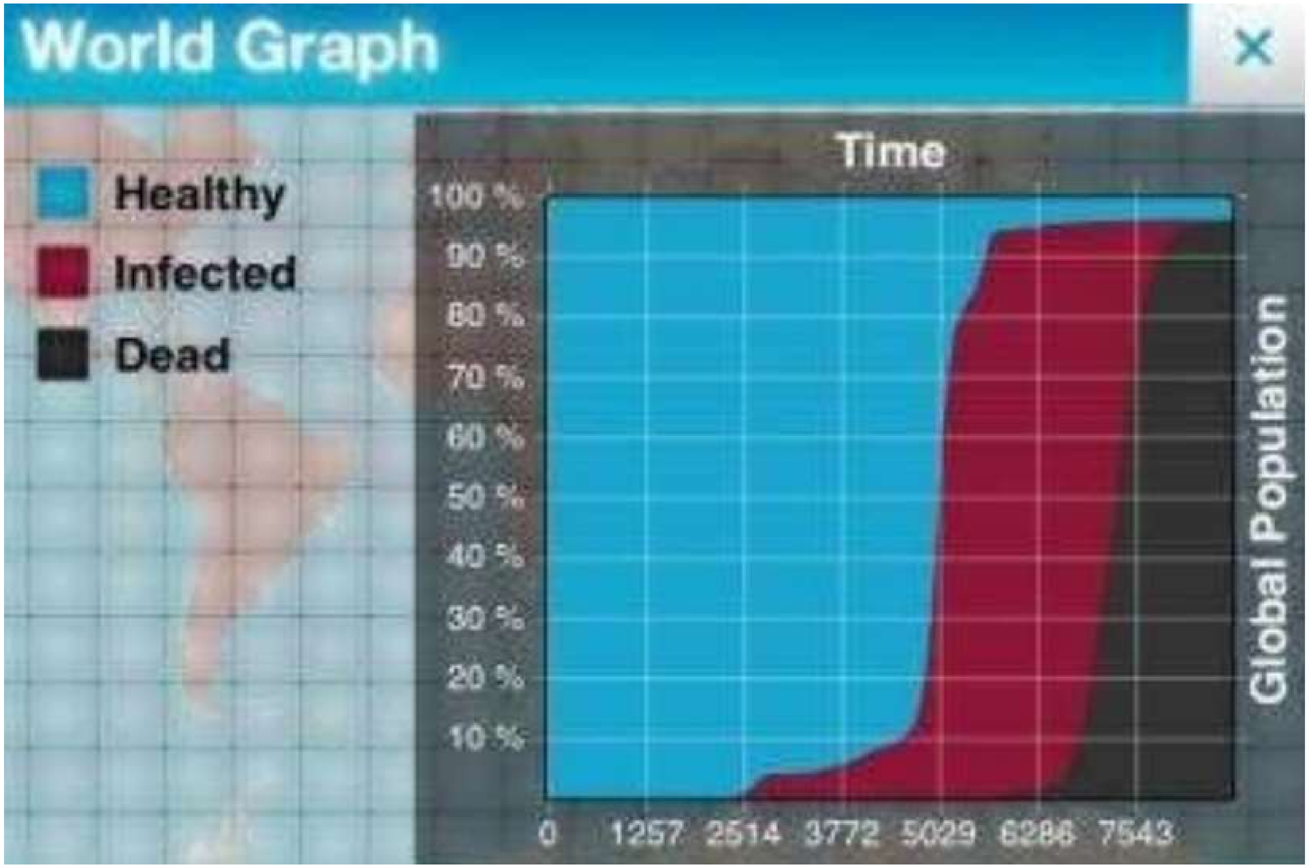
You are probably wrong about

We have tested thousands of people and they were systematically wrong about all this.



Gapminder.org

Plague Inc., Video Game



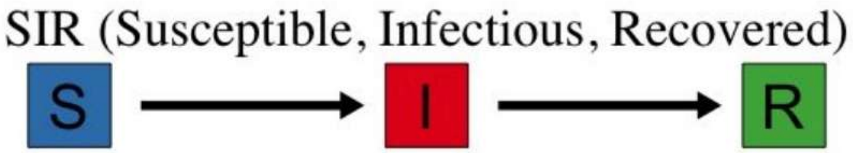
Basic SIR Model

Susceptible- set of people in the model who have not had the disease.

Infected- set of people in the model who have had the disease currently

Recovered- set of people in the model who have recovered from the disease.

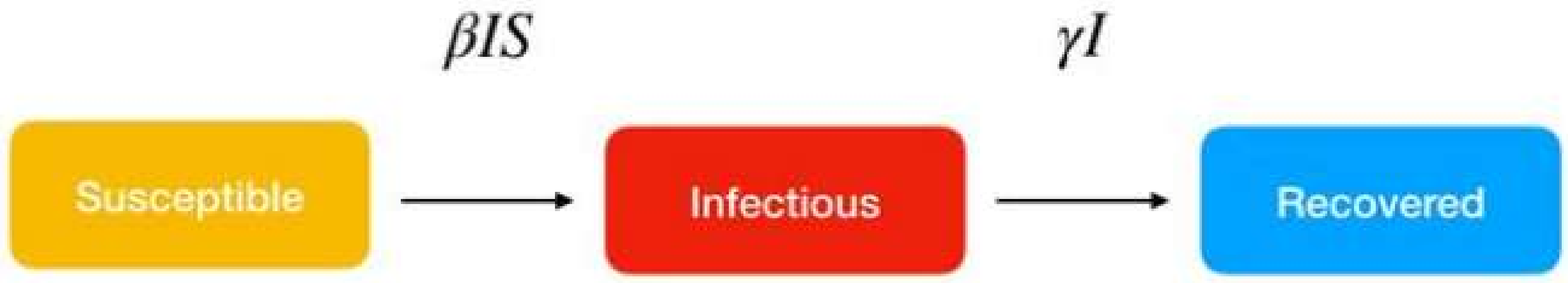
Basic SIR Model



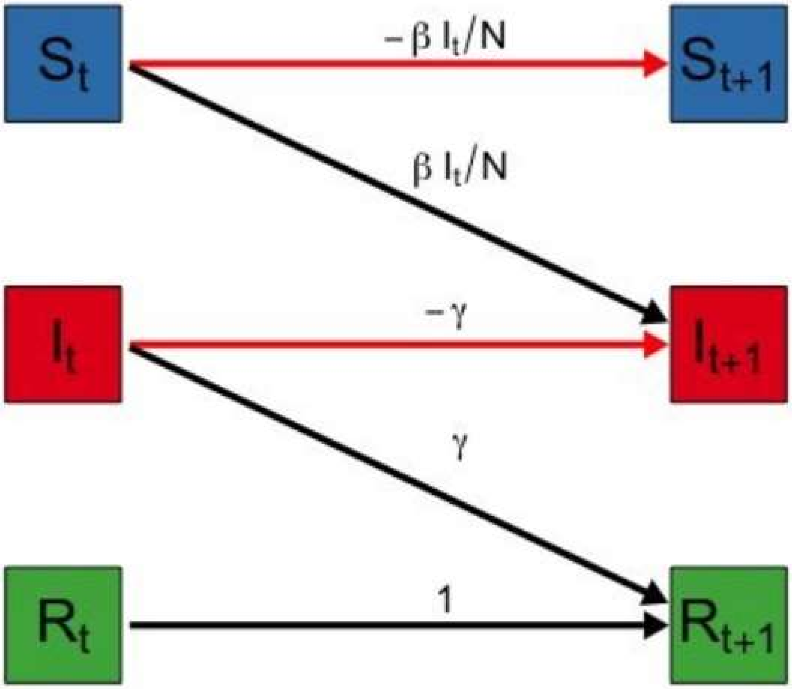
SIR (Susceptible, Infectious, Recovered)

S → I → R

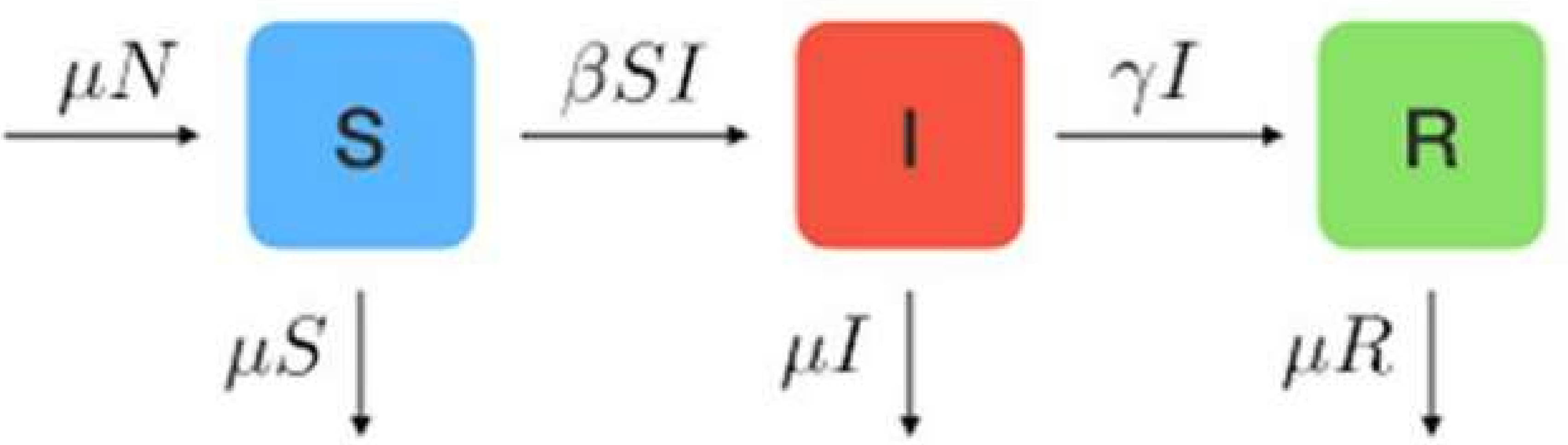
Basic SIR Model



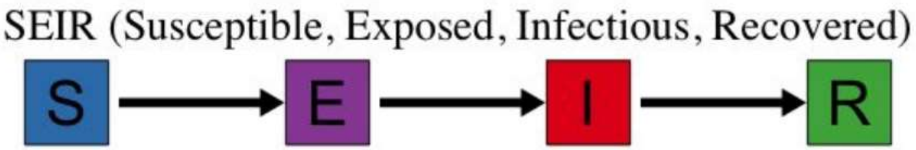
Differential Equations Model of SIR Transformations



Basic SIR Model

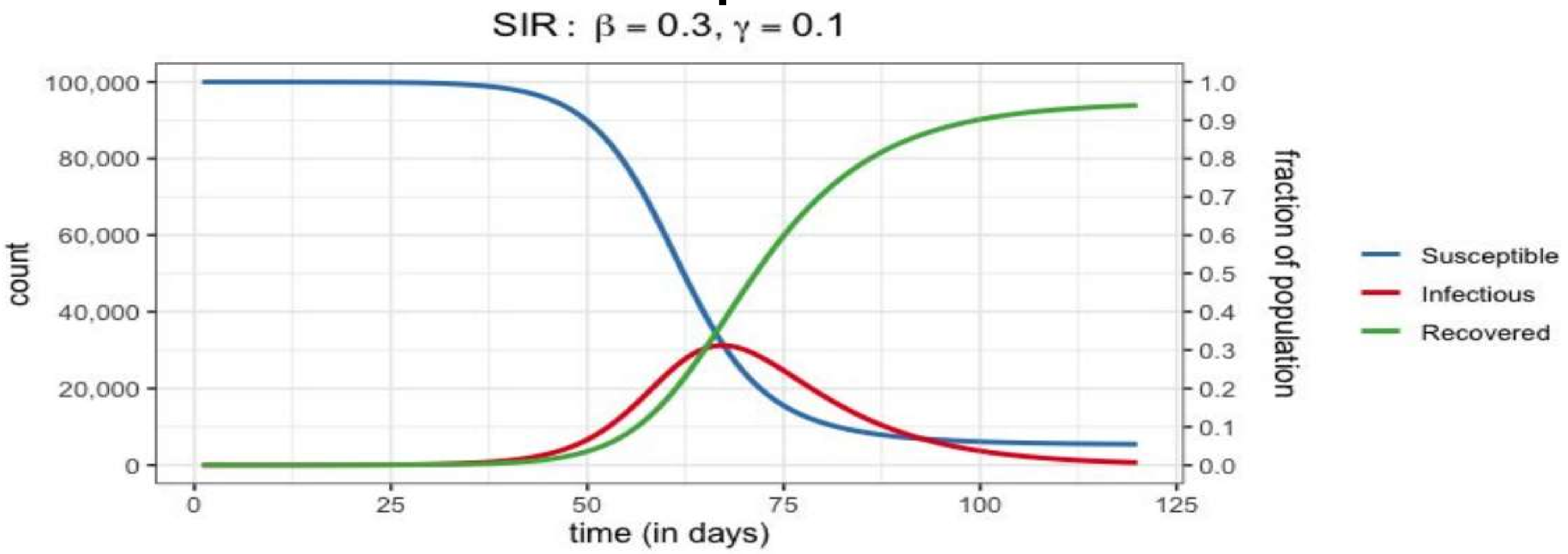


Basic SEIR Model


SEIR (Susceptible, Exposed, Infectious, Recovered)

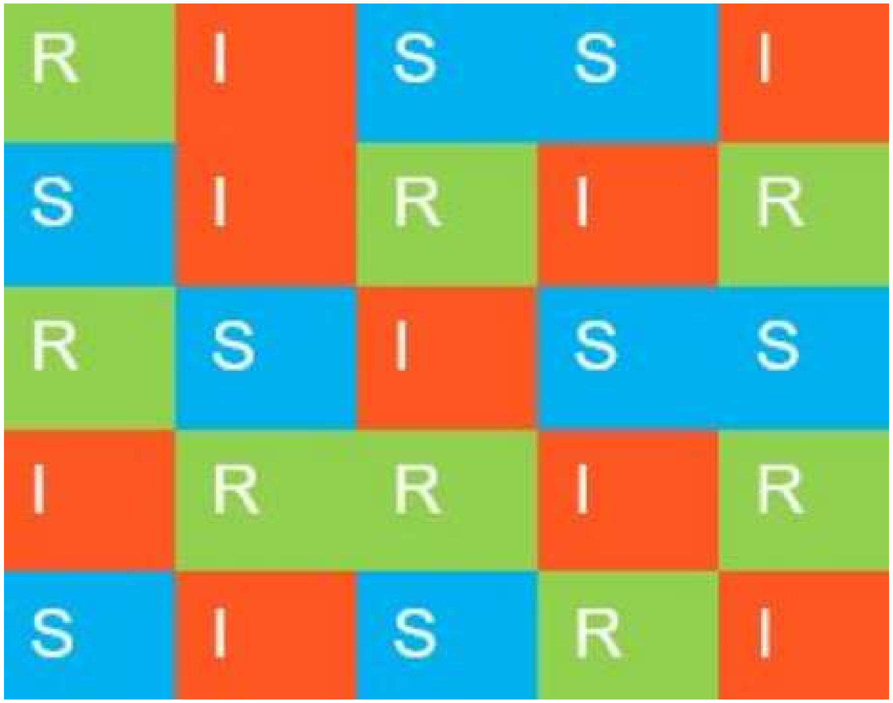
E (Exposed) - Set of people in the model who have been exposed to the disease, and do not show symptoms.

SIR Example over Time



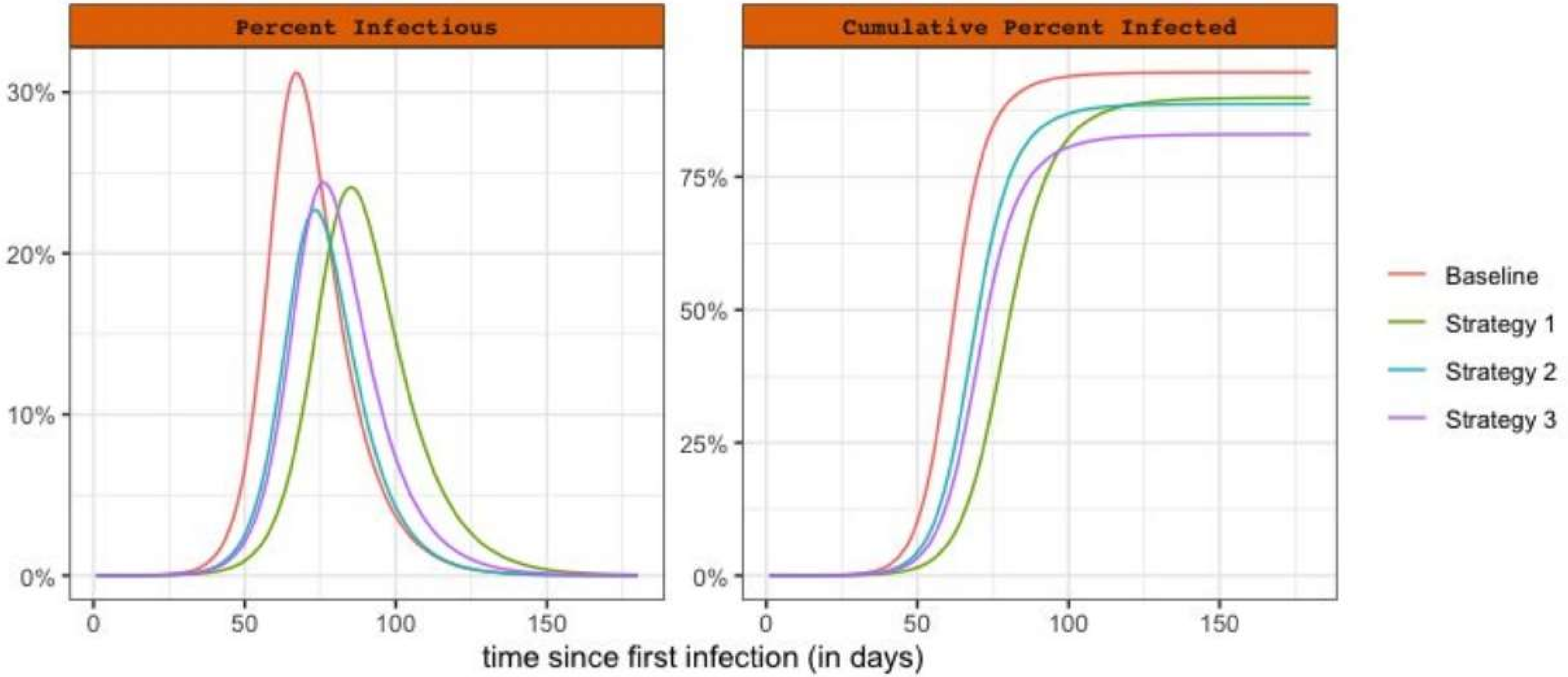
- Graphical representation on the SIR model (Susceptible, Infected and Recovered) on the right, and bellow their structure and dynamics in flow charts.

Oversimplified (simple forced probability) Model



R	I	S	S	I
S	I	R	I	R
R	S	I	S	S
I	R	R	I	R
S	I	S	R	I

Models Derived through Different Strategies


Percent Infectious
Cumulative Percent Infected
time since first infection (in days)
Baseline
Strategy 1
Strategy 2
Strategy 3