154 Presentation

By: Bernardo Hernandez, Christopher Garcia, And Javier Escareno

How do Mathematicians Model Infectious Disease Outbreaks?

Presented By: Javier Escareno

Motivations

1. Being able to further predict how many may get affected during an outbreak.
2. Once a accurate model of the outbreak is built then using said model to test different methods of preventing the further spread of the outbreak to see what methods are effective in the different stages of the outbreak.
3. Help predict when peak number of cases could happen to better prepare healthcare services to be ready when a wave of infected patients may come into the regions hospitals.

Background Information

1. The information shown was originally presented by Dr. Robin Thompson on behalf of Oxford University to show how they worked on modeling Covid- 19. The video was published online on April 8th, 2020 which was the early part of the pandemic.
2. A mathematical model is a simplified representation of a real-world system, process or phenomenon using mathematical concepts, equations and symbols.

Problem Statement

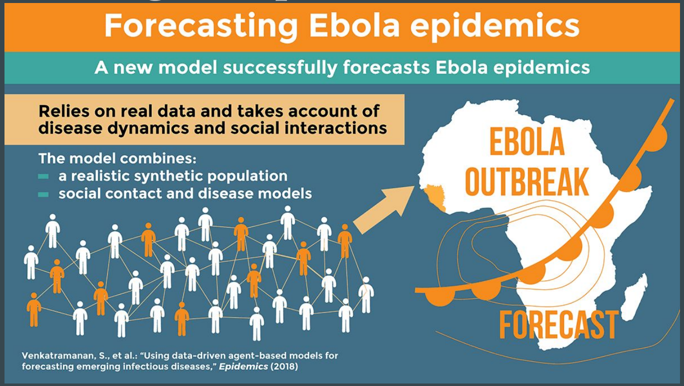
I. Imagine a world wide pandemic occurs and begins to spread world wide infecting the world population what do you do to predict how the may spread in a given area or even to and from places?

II. How could you test methods of stopping the spread of the infection and know what will and won't be effective?

Problem Statement Cont.

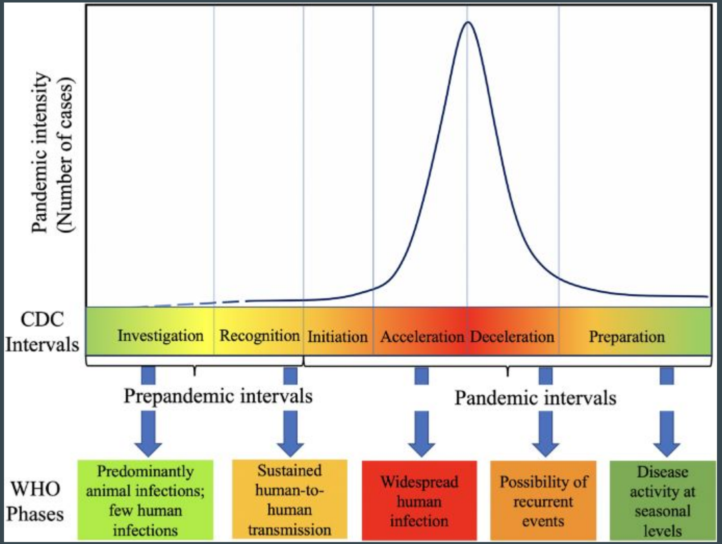
Main Objective:

1. Create a model of the outbreak you are testing and get your model to accurately have the same shape as what the current data is showing.
2. Once your model is for the most part is graphing the spread of the infection as what your IRL data is showing is to then introduce preventative measures and test their effectiveness at slowing the spread of the infection.



Approach

1. First thing to do when trying to model a outbreak is to first graph you data from the outbreak which would include number of infected over a period of time
2. Once the data is graphed then you must determine its shape as it is key in understanding what phase in its life cycle the outbreak is.



Starting simple

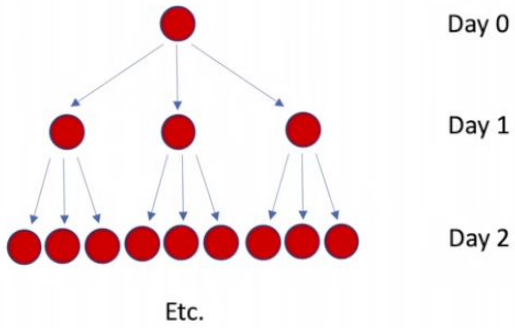
Building a simple outbreak model

Each case gives rise to 3 new cases every day

1, 3, 9… ??

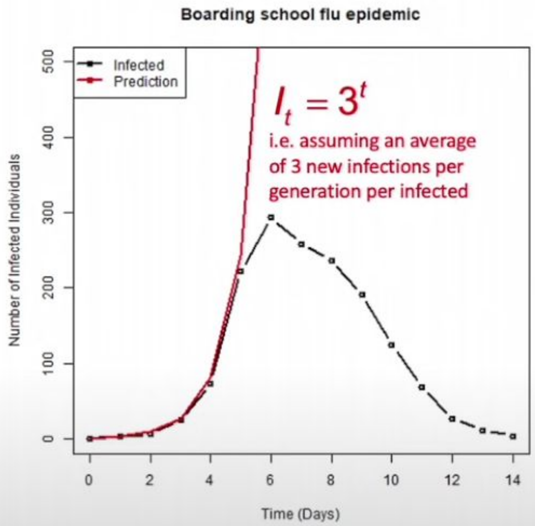
Geometric progression

$$I_t = 3^t$$



Starting Simple Cont.

Building a simple outbreak model



Question
What happens if we compare this model to the real data?

Result
Not convincing…

Conclusion
Need to ***refine*** the model: add more epidemiology…

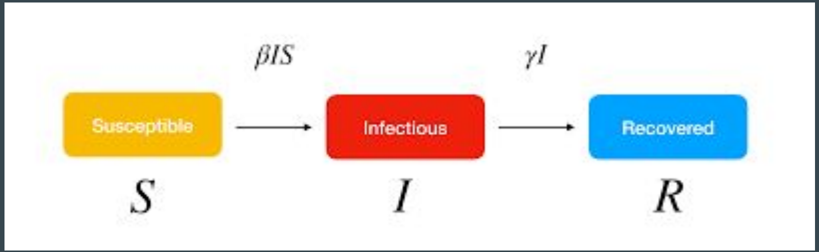
Issues with the model

- Some issues with this first iteration of the model is that the outbreak never runs out of people to infect and goes on for infinity infecting people.
- The model does not take in to account for people recovering from the disease, having asymptomatic symptoms, herd immunity, population density, among many other factors.
- Overall this model is good for representing early stage outbreaks in how infections in a population can spike in a short manner of time.

The S.I.R Model

- SIR stands for Susceptible, Infected, and Recovered, which are the three categories of individuals in a population that is susceptible to an infectious disease.
- The SIR model is a mathematical model used to study the spread of infectious diseases in a population over time.
- It assumes that the population is fixed and individuals can only move between the three categories: susceptible individuals can become infected, infected individuals can recover and become immune, and recovered individuals cannot be infected again.

The S.I.R Model Cont.

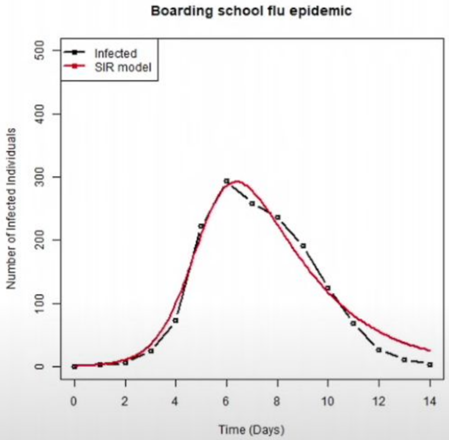


- Transmission rate (beta): Represents the probability of an infected individual passing the disease to a susceptible individual.
- Recovery rate (gamma): Represents the rate at which infected individuals recover and become immune.
- Initial number of infected individuals (I0): Represents the number of individuals initially infected in the population.

S.I.R Model Still Cont.

Building models of outbreaks | Using models to inform control during outbreaks
Shape | **Simple model** | Concepts | Extensions | Early | Middle | End

Building a simple outbreak model



Question
What happens if we fit the SIR model to the data?

Answer
Not at all bad for a model with just two parameters (β and μ)

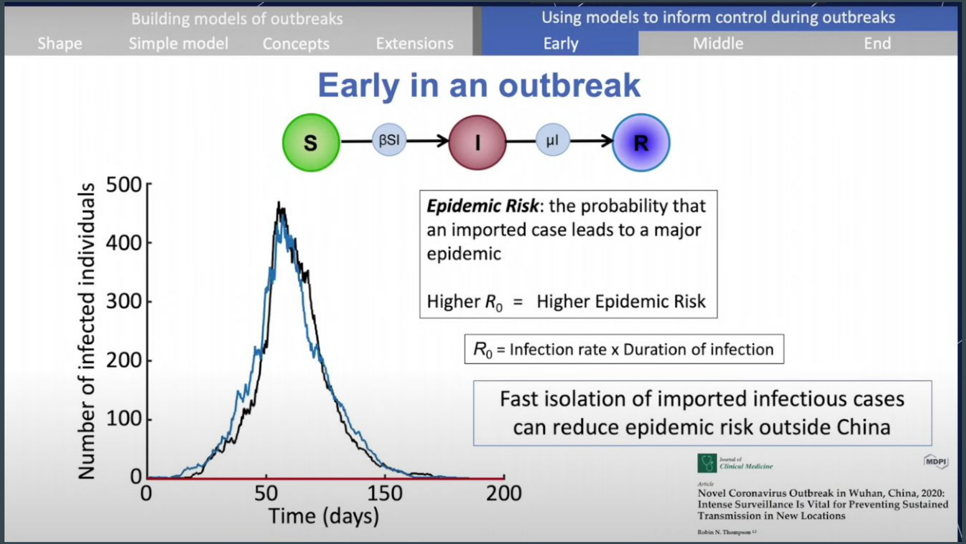
Important conclusion
Including recoveries allows us to capture overall "shape"

Extending the S.I.R Model

1. Remember infectious disease outbreaks are inherently random therefore not every outbreak is going to have smooth perfect curves so use a stochastic model to keep the randomness in your model.
2. Making sure to include individuals who may be infected but may not yet be symptomatic.
3. Making sure to be aware that different age groups of a society will contract, spread and heal from the outbreak differently.
4. The last extension of the model is taking into account the population density/ spatial structure of the population being modeled.

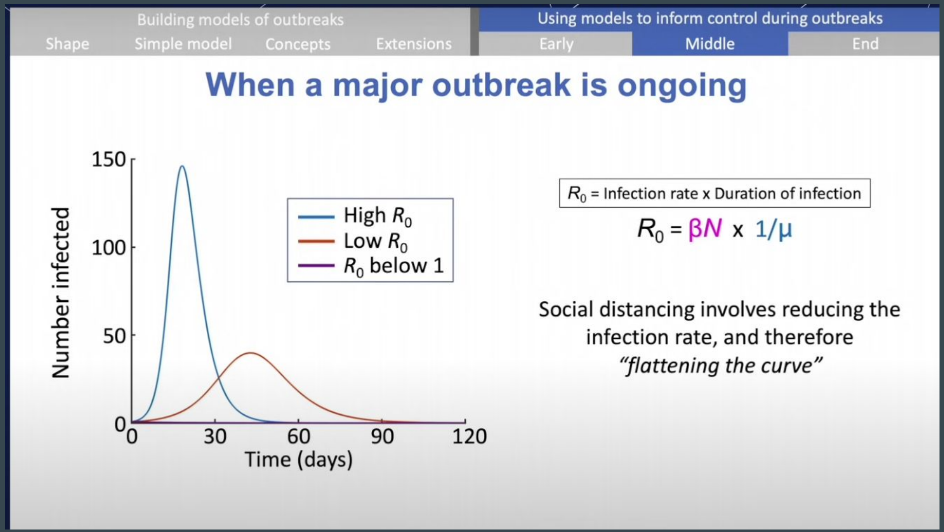
What to do once you model is built?

- Once your model is built and you feel confident in its ability to accurately model the given outbreak you should begin testing different methods of slowing down the spread of the outbreak.
- The photo is an example if social isolation for those testing positive for the disease is implemented.



What Results Should You want To See?

- Overall the result you want to see from your model after implementing different outbreak preventative measures is a flattering of your curve that shows a low number of infected over a long period of time.



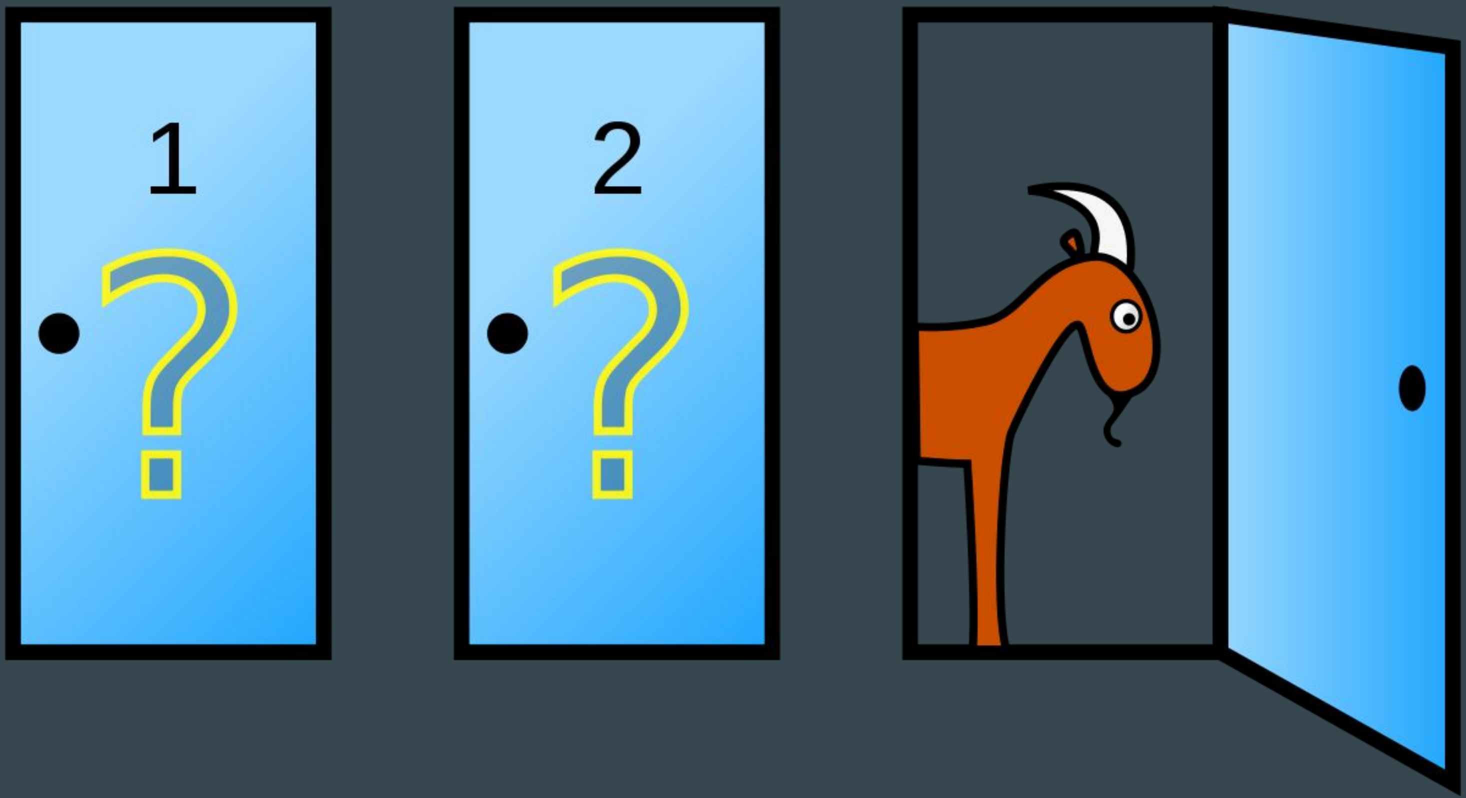
Conclusion

1. Modeling outbreaks while they are occurring have many benefits including predicting when a spike in the outbreak may occur to better prepare health care services for said rush.
2. The models are also useful in testing different preventative measures to see what is and isn't effective at flattening the curve of the outbreak.

Monty Hall Problem

Presented By Christopher Garcia

Overview


1
2

Approach

Python Implementation

1. Import required libraries (random, matplotlib, warnings)
2. Define montyHallSimulation function with parameters (numDoors, numTrials, policy, accidentalReveal)
   a. Initialize win counter (wins)
   b. Loop for each trial:
      i. Randomly choose carDoor and initialChoice
      ii. Create list of remainingDoors
      iii. If accidentalReveal, choose hostRevealedDoor from all doors except initialChoice. Otherwise, choose hostRevealedDoor from remainingDoors
      iv. Apply policy (randomSwitch or stick) to determine finalChoice
      v. If finalChoice equals carDoor, increment wins counter
   c. Calculate and return win probability
3. Set trialCounts, doorCounts, and policies lists
4. Part 1: Approximation improvement with number of iterations
   a. Loop through each policy and numDoors
      i. Loop through each numTrials
         1. Run montyHallSimulation and append win probability to winProbabilities list
      ii. Plot winProbabilities against trialCounts
   b. Save and display the plots
5. Part 2: Probability of winning for each policy with number of doors
   a. Set fixed number of trials
   b. Loop through each policy and numDoors
      i. Run montyHallSimulation and append win probability to winProbabilities list
   c. Plot winProbabilities against doorCounts
   d. Save and display the plot

6. Part 3: Probabilities change for the new variant (accidental reveal)
   a. Loop through each policy and numDoors
      i. Run montyHallSimulation with accidentalReveal=True and append win probability to winProbabilities list
   b. Plot winProbabilities against doorCounts
   c. Save and display the plot

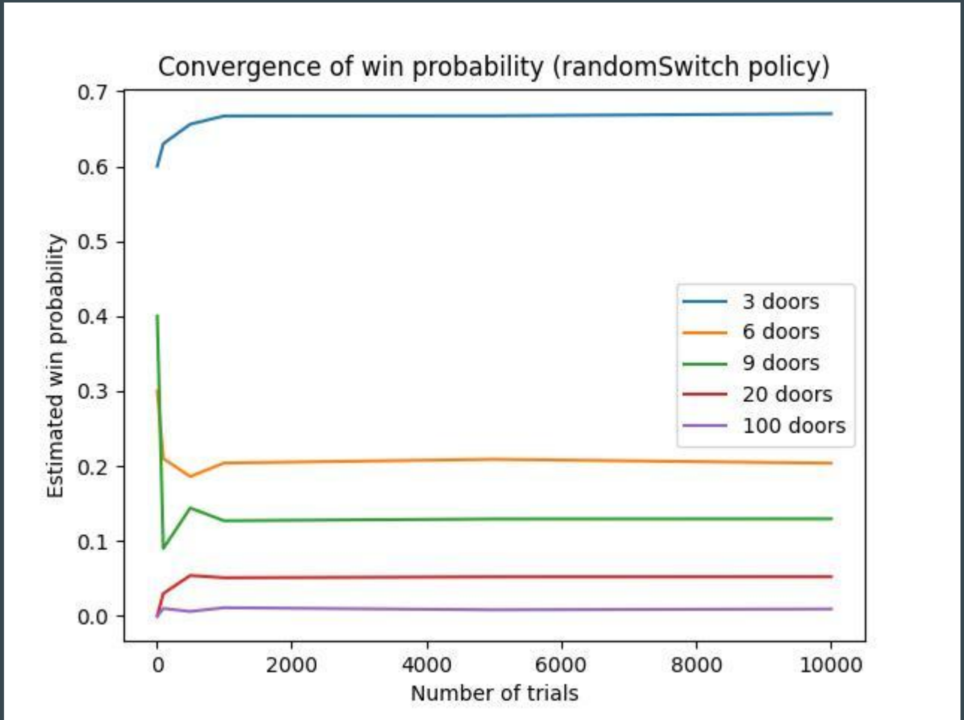
Python Implementation

1. Set up a list
2. Go through some logic with the list
3. Get results

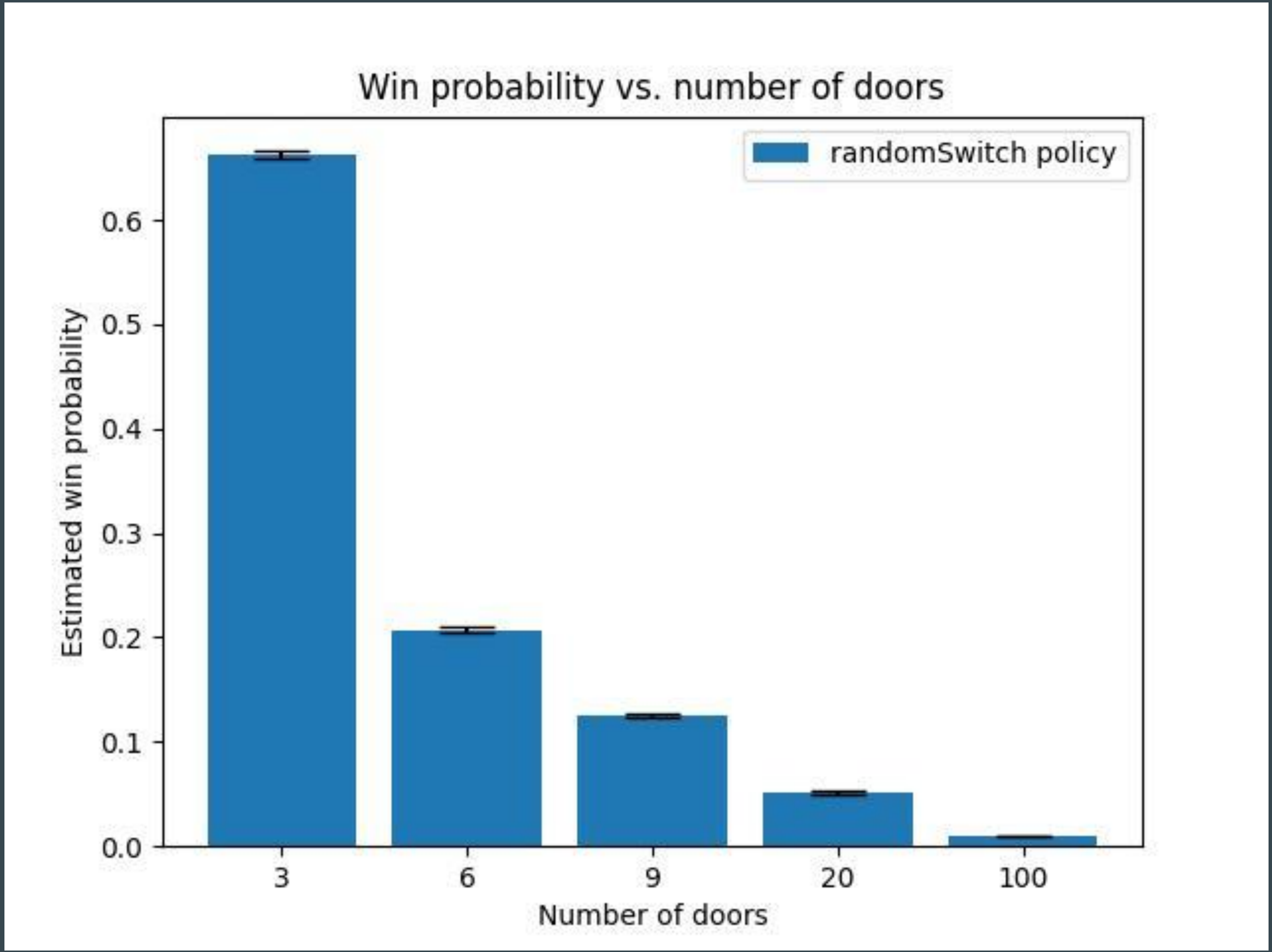
Results

Win Probability (Randomly Switch)

Number of doors: 3	67%
Number of doors: 6	20.38%
Number of doors: 9	12.96%
Number of doors: 20	5.23%
Number of doors: 100	0.93%



10, 100, 500, 1000, 5000, 10000


Win probability vs. number of doors
randomSwitch policy
Estimated win probability
0.0
0.1
0.2
0.3
0.4
0.5
0.6
3
6
9
20
100
Number of doors

Win Probability (Stick with Original)

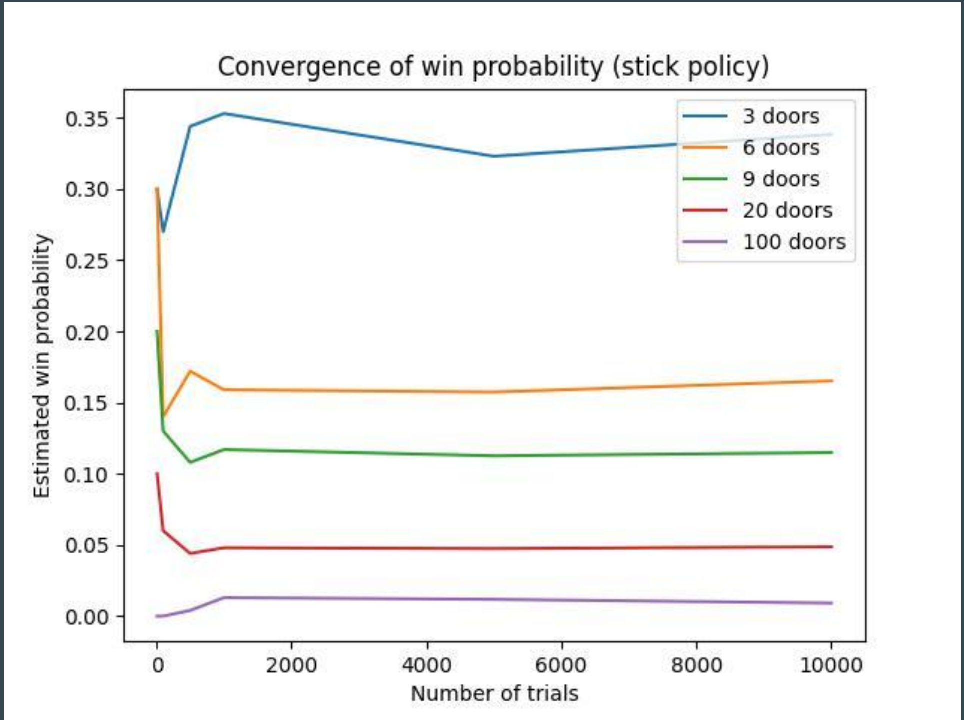
Number of doors: 3: 33.84%

Number of doors: 6: 16.52%

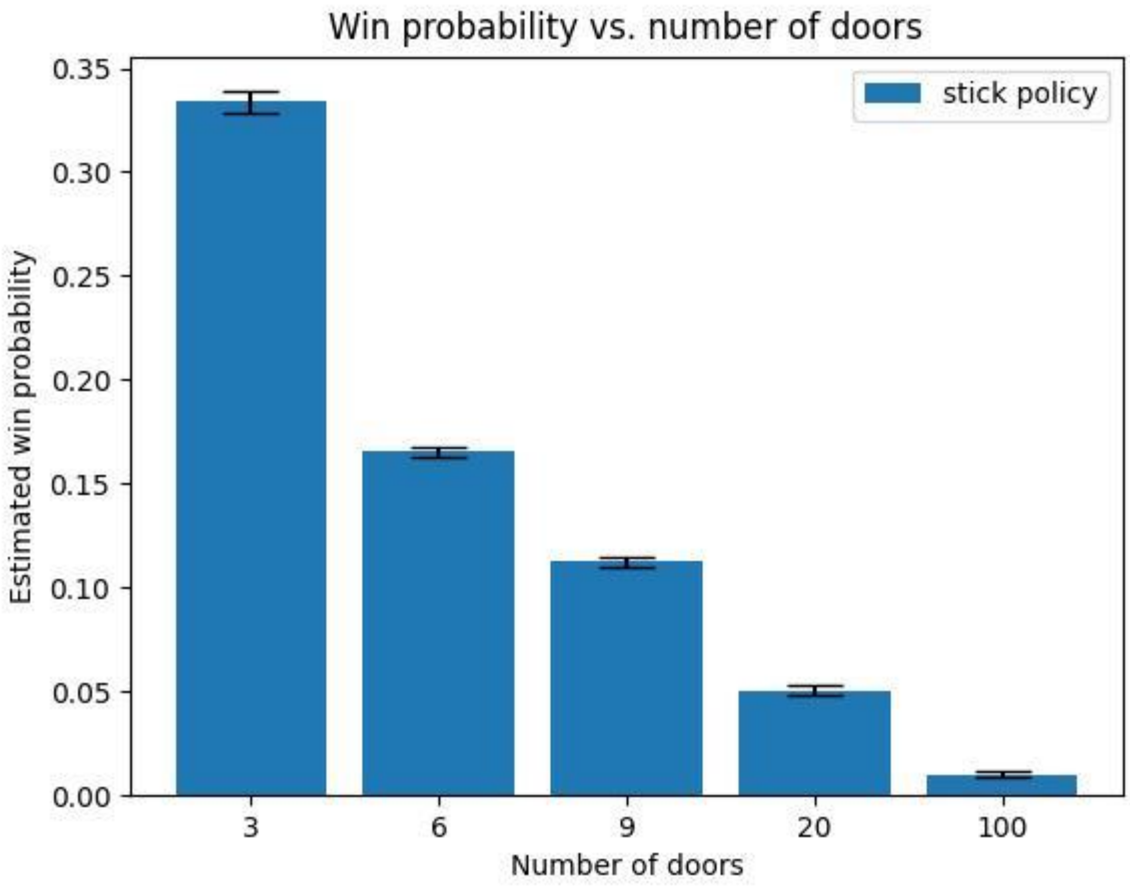
Number of doors: 9 11.49%

Number of doors: 20 4.87%

Number of doors: 100 0.92%



10, 100, 500, 1000, 5000, 10000


Win probability vs. number of doors
stick policy
Estimated win probability
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
3
6
9
20
100
Number of doors

Win Probability vs Number of Doors

Policy: randomSwitch

Number of doors: 3, Win probability: 66.82%

Number of doors: 6, Win probability: 20.20%

Number of doors: 9, Win probability: 12.39%

Number of doors: 20, Win probability: 5.18%

Number of doors: 100, Win probability: 0.94%

Policy: stick

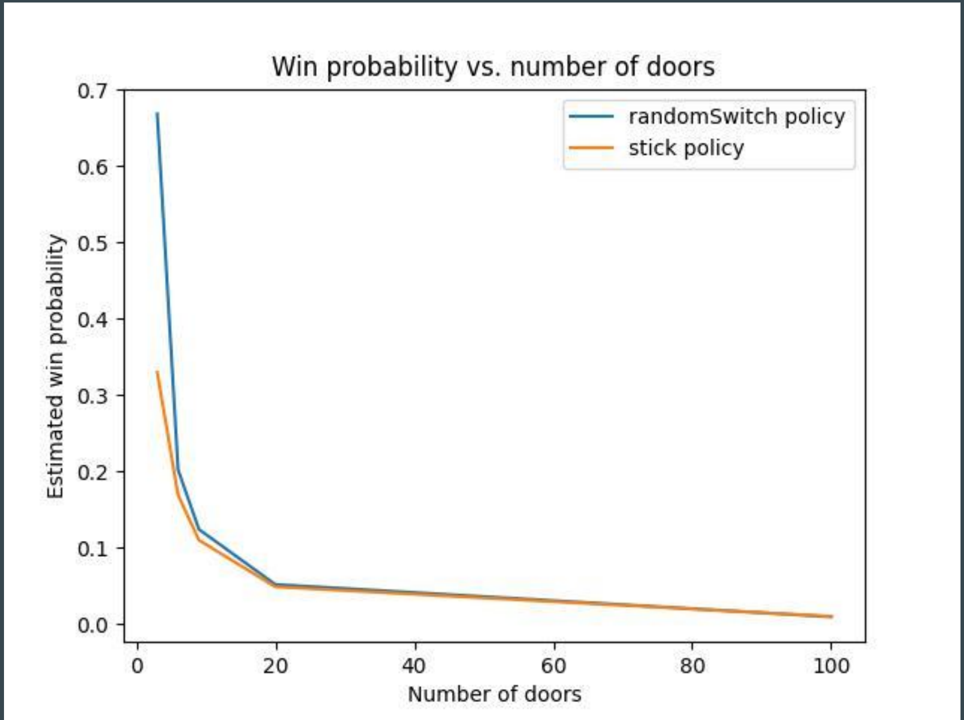
Number of doors: 3, Win probability: 32.96%

Number of doors: 6, Win probability: 16.89%

Number of doors: 9, Win probability: 10.98%

Number of doors: 20, Win probability: 4.87%

Number of doors: 100, Win probability: 1%



10,000 Trials

Banana Peel Variation

Policy: Random Switch

Number of doors: 3, Win probability: 33.31%

Number of doors: 6, Win probability: 16.71%

Number of doors: 9, Win probability: 11.05%

Number of doors: 20, Win probability: 05.09%

Number of doors: 100, Win probability: 00.92%

Policy: stick

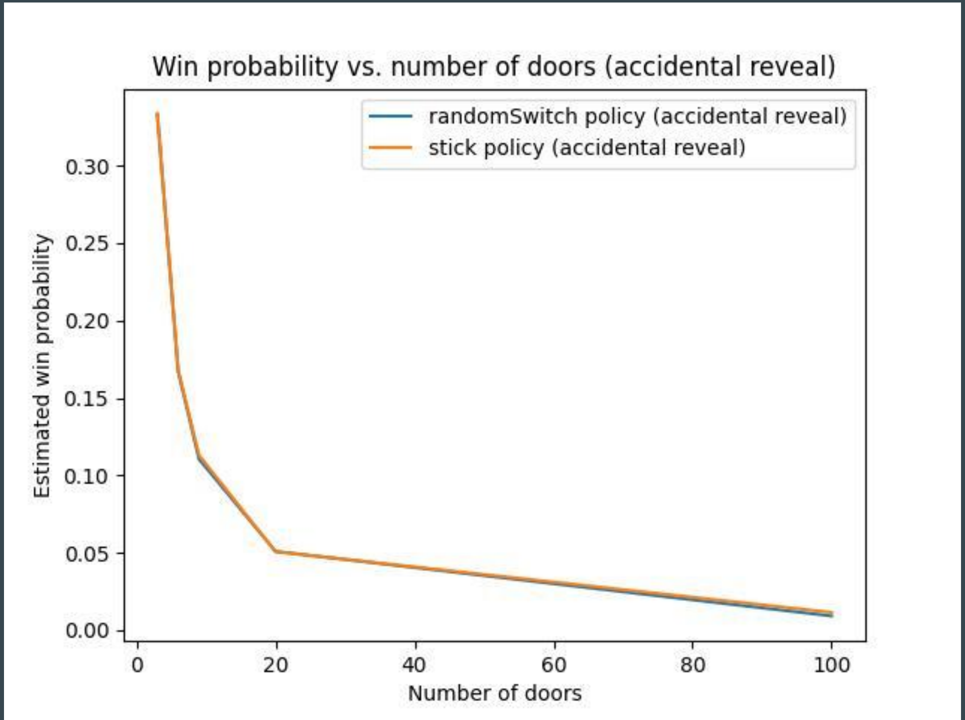
Number of doors: 3, Win probability: 33.36%

Number of doors: 6, Win probability: 16.79%

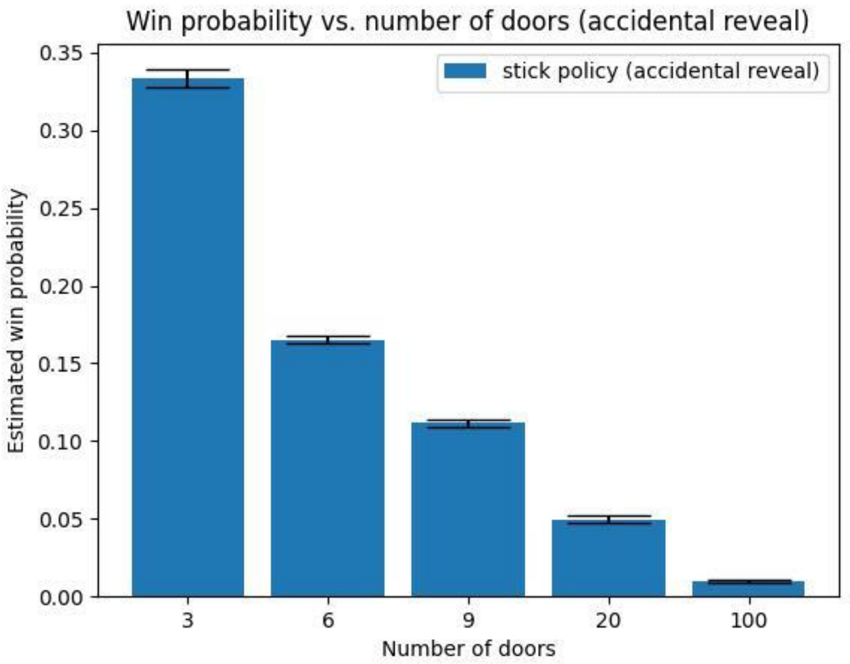
Number of doors: 9, Win probability: 11.30%

Number of doors: 20, Win probability: 5.07%

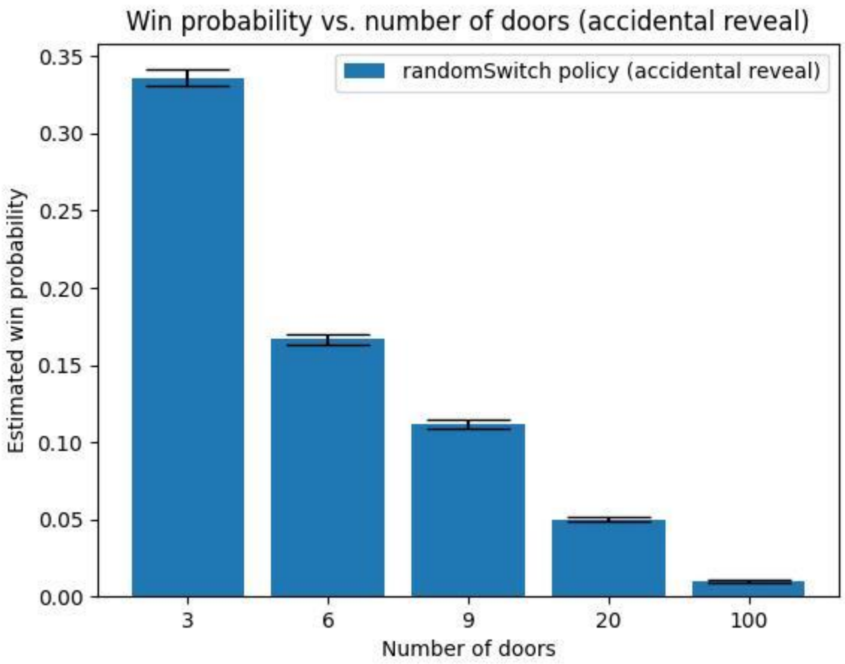
Number of doors: 100, Win probability: 1.15%



10,000 Trials



Win probability vs. number of doors (accidental reveal)
stick policy (accidental reveal)
Estimated win probability
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
3
6
9
20
100
Number of doors



Win probability vs. number of doors (accidental reveal)
randomSwitch policy (accidental reveal)
Estimated win probability
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
3
6
9
20
100
Number of doors

Takeaway

Black Jack Monte Carlo

...

By Bernardo Hernandez

Motivation

- Which policies have a better win probability in two scenarios:
  - Infinite Deck : Each Card has an equal probability of being drawn
  - Single Deck : The deck is shuffled after every game

- Which policies decrease depending on the scenario

Black Jack Rules

- Player wins if :
  - Gets “Black Jack”
    - Ace with 10/ Jack/Queen/King
  - Gets 21
  - Player beats dealer hand
  - Dealer exceeds 21
- Player Loss if :
  - Dealer beats player’s hand
  - Player exceeds 21
  -

Policies and Black Jack Terms

- Policy 1: if your hand ≥ 17, **stick**. Else **hit**.
- Policy 2: if your hand ≥ 17 and is **hard**, **stick**. Else hit unless your hand = 21
- Policy 3: Always **stick**.
- Policy 4: if hand >=12 then **stick**
- Policy 5:if hand>=20 stick
- **Stick** = No longer draw Card
- **Hit** = Draw Card
- **Soft Hand** = player has an Ace that is valued 11
- **Hard Hand** = player does not have an Ace, or has Ace is valued at 1 since there hand exceeds 21 with it

Policies with Single Deck: Best to Worst (Win Probability)

Single Deck with Policy 4 : games won = 6203 amount of games = 10000

6203/10000 = 62.03

Single Deck with Policy 1 : games won = 5410 amount of games = 10000

5410/10000 = 54.1

Single Deck with Policy 2 : games won = 5124 amount of games = 10000

5124/10000 = 51.24

Single Deck with Policy 3 : games won = 4582 amount of games = 10000

4582/10000 = 45.82

Single Deck with Policy 5 : games won = 3288 amount of games = 10000

3288/10000 = 32.88

Policies with Infinite Deck: Best to Worst (Win probability)

- Infinite Deck with Policy 1 : games won = 4568 amount of games = 10000

  4568/10000 = 45.68
- Infinite Deck with Policy 2 : games won = 4493 amount of games = 10000

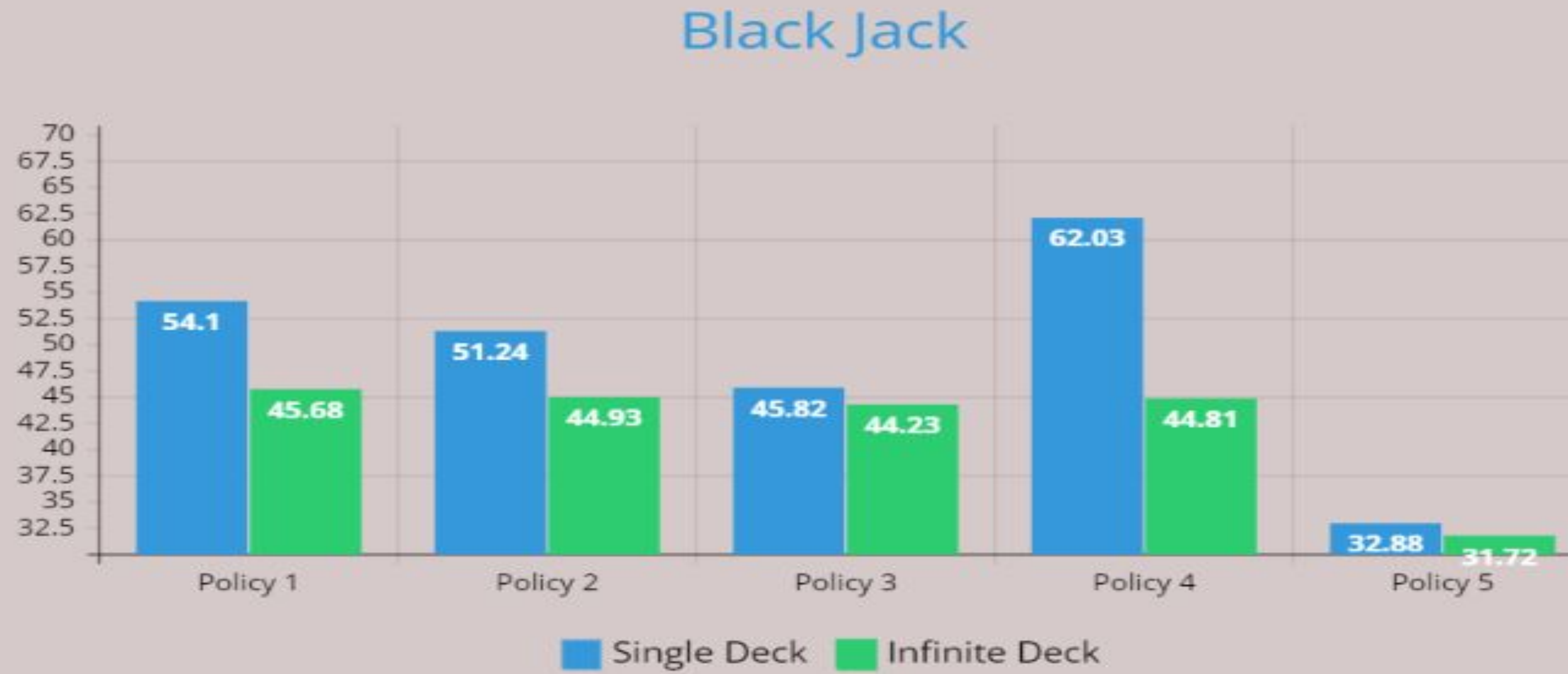
  4493/10000 = 44.93
- Infinite Deck with Policy 4 : games won = 4481 amount of games = 10000

  4481/10000 = 44.81
- Infinite Deck with Policy 3 : games won = 4423 amount of games = 10000

  4423/10000 = 44.23

- Infinite Deck with Policy 5 : games won = 3172 amount of games = 10000

  3172/10000 = 31.72



Black Jack
70
67.5
65
62.5
60
57.5
55
52.5
50
47.5
45
42.5
40
37.5
35
32.5
54.1
45.68
51.24
44.93
45.82
44.23
62.03
44.81
32.88
31.72
Policy 1
Policy 2
Policy 3
Policy 4
Policy 5
Single Deck
Infinite Deck

Conclusion : Best Policy (Single Deck and Infinite Deck)

- Policy 4: if player hand is >=12 then **stick**
- This could be because of how the dealer must play.
    - Dealer is forced to keep drawing cards even if his current hand beats the player hand
    - E.g player reaches 14 and sticks, dealer draws card and has 16
    - dealer is forced to draw again until >=17. Dealer either wins or exceeds 21(busts)
- Infinite Deck Policy 1 had a better win probability than in the Single Deck
    - In actual Casinos, more than one deck can be used