NME 2021

Network Modeling for Epidemics

1

ICMs

Stochastic Individual-based Contact Models

Model frameworks

For mechanistic models

Two poles of epidemic modeling:

- Deterministic compartmental models
- Stochastic network models

In between lie all sorts of alternatives

- Deterministic individual-based models (not really a thing)
- Stochastic compartmental models (see appendix)
- Stochastic individual-based contact models ("ICMs")

Let's take a quick look at the ICMs: our poker chip example

- Before moving on to stochastic network models

What makes a model stochastic?

The transition parameters

that govern the "flows" of elements between states

- In a deterministic model these are fixed *rates*
  - Applied to aggregate stocks in the compartments
- In a stochastic model these are *probabilities*
  - Applied to individual elements

What does stochastic mean?

- In general: random, or variable
- In particular:
  - A random draw
  - From the possible range of outcome values
  - With a probability assigned to each value
- Typically, the probabilities are summarized by
  - a *probability density function (PDF)*
  - defined by one or more parameters
  - Ex.: binomial, Poisson, normal, etc.

Formalizing the poker chips

- Represent each model as an ICM
  - Identify the possible stochastic components
    - And some typical probability distribution choices
  - Identify what was stochastic in our poker chip example
    - And what we left deterministic
- NOTE: *We won't be coding these models*
  - But like DCMs, it's good to know the basics here
  - So follow the concepts, not the details

First step for all individual based models:

- Set the initial conditions
  - Create *the individual elements*
  - And assign their state
    - For poker chips, just their state of infection
    - But you can imagine assigning other attributes…
- Compared to DCMs
  - Here, elements will always be whole units (not fractional)
  - And the state of each unique element is known at each timestep

Stochastic “constant growth” model

- One transition: “infection”
  - … more like a non-infectious chronic disease incidence
- For the poker chip example:
  - a fixed, deterministic rate of new cases
- To make this model stochastic
  - Draw the incidence at each step from a distribution
    - Range: positive integers (Z+), no fixed maximum (infinite population)
    - Distribution options:
      - $Poisson(\lambda)$ is the natural choice
      - Not $Binomial(n; p)$ (why not?)

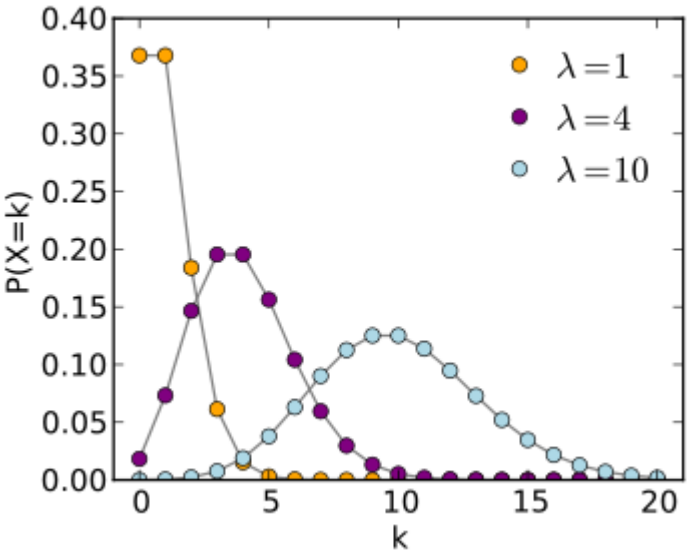
Poisson distribution

- Used for counts of events when:
  - *n* (the number of trials) is large, *not fixed*,
  - and *p* (the probability of success) is small, so the product $np = \lambda$ approximates a rate of events (e.g., per time unit, or per capita)

$$f(x;\lambda) = P(X = x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

$$\mu_X = \sigma_X^2 = \lambda$$

Note: variance=mean



Impact of stochasticity on epidemic dynamics?

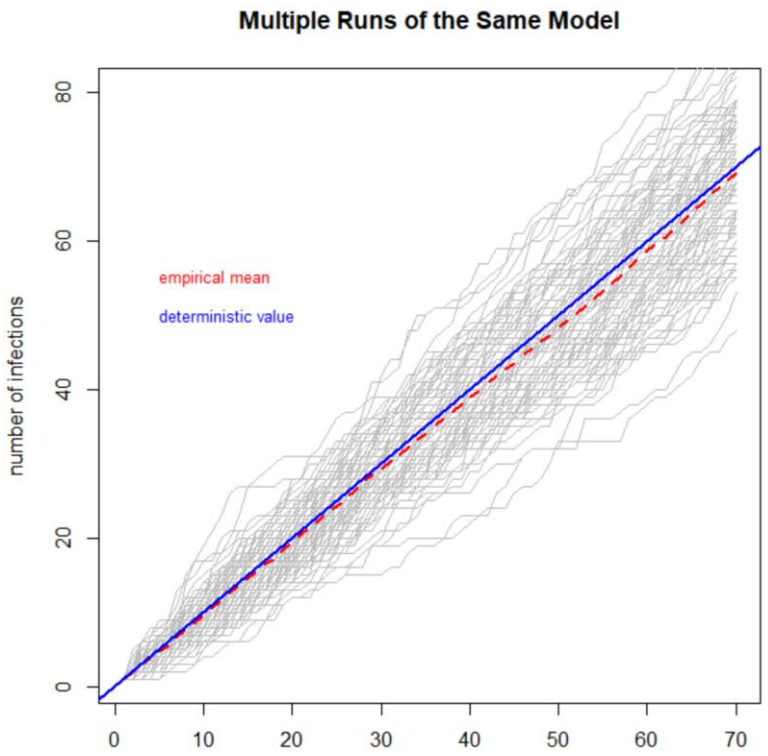
Using the Poisson distribution:

- Expected number of new cases each day is λ
  - With poker chips we had a deterministic 1 new case per day
  - With stochastic model we set $\lambda = 1$, the average number of new cases
- Variance in new cases each day is λ
  - Standard deviation = $\sqrt{\lambda}$
- Does not change the basic shape of the time series
  - Still basically linear
  - *Just has some variation*

Comparison for constant growth

Deterministic

Stochastic



Stochastic I model

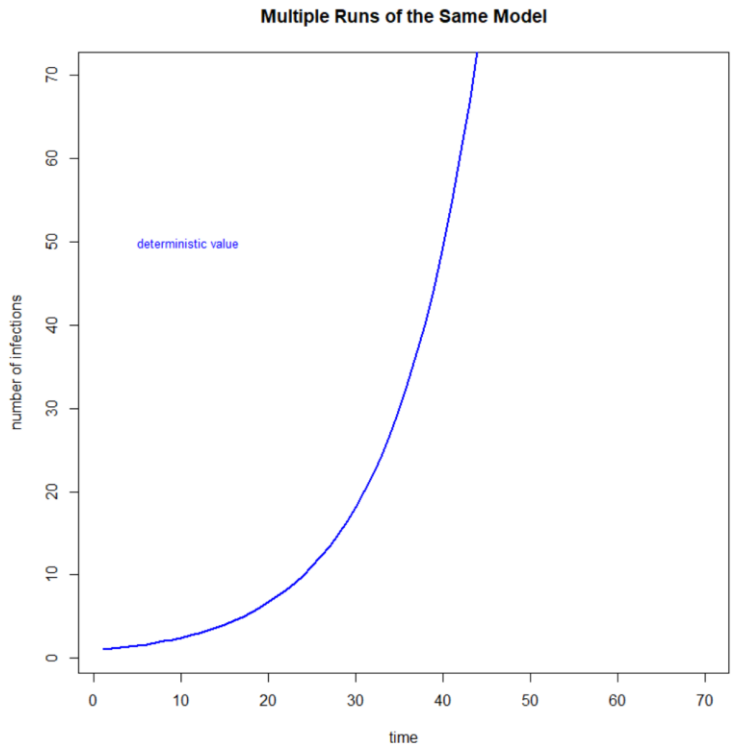
- Still just one transition: "infection"
  - But now incidence depends on prevalence
- For the poker chip example,
  - A prevalence-dependent deterministic rate
- To make this model stochastic
  - Draw the incidence at each step from a distribution
    - Range: Z+, no fixed maximum (infinite population)
  - But now the rate parameter is time-dependent (depends on I(t))
    - Distribution options:
      - $Poisson(\lambda \boldsymbol{i}(\boldsymbol{t}))$ is again the natural choice, for the same reason
      - $i(t)$ is an integer, not an aggregate, possibly fractional, value

What impact would this have?

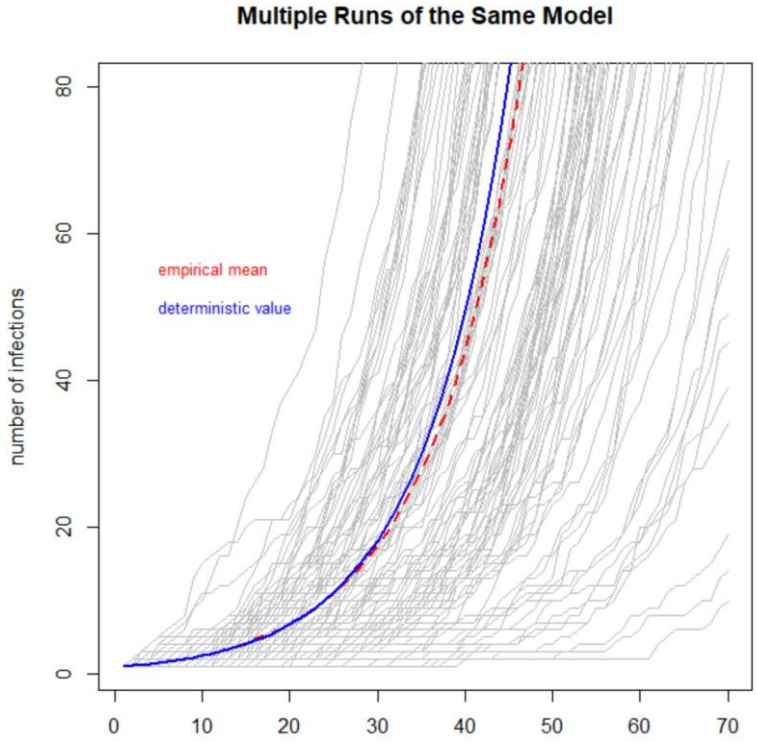
- Expected number of new cases each day is $\lambda i(t)$
  - Again translating from poker chips: We set $\lambda = 1$
    - think about this, what might λ represent now?
- Does not change the basic shape of the time series
  - Still basically exponential
  - With some variation

Comparison for I Model

Deterministic



Stochastic



Key idea: Interpreting variability

- Note how different the runs can be
  - If you saw these differences across communities
  - You might think they had wildly different underlying epidemic dynamics
- Stochastic variation can be large
  - At the beginning of an epidemic
  - Or in small populations
- Be careful not to over-interpret!

Stochastic SI model (now it gets interesting)

- Still just one transition: infection
- But now we have a finite population (the bag)
- So pop'n incidence depends on three things (at minimum):
  - $s(t)$ and $i(t)$: Drawing an SI pair at time t
  - $a(t)$: The number of acts at each time step (SI pairs drawn)
  - $m(a)$: Transmission per act

Stochastic SI model

- To make this model stochastic
- Draw one or more of the components from a distribution

Component	Attributes	Distribution*
SI pair	Draw two chips without replacement from a finite population	$Hypergeometric(1, S(t), I(t), 2)$ See appendix for derivation, approximately $Binomial$ for large n
$a(t)$	Z+, may or may not have fixed maximum	$Binomial(n/2\,, \alpha\ /(n/2))$ if max is 1 act per pair
$m(a)$	{0,1}, like a coin flip	$Bernoulli(\tau)$

* See Wikipedia for definitions

SI poker chip exercise

- What component(s) did we make stochastic?

Component	Stochastic or Deterministic?	Value or Distribution
SI pair	stochastic	Hypergeometric
$a(t)$	deterministic	Fixed at 1 per day
$m(a)$	deterministic	Fixed at 1 for all acts

Just this

SI model comparison

- You'll run this in the next lab

Finally, the SIR model

Now there are two transitions

- Infection, which drives incidence
  - As before
- Recovery, which drives the prevalence of immunity
  - For each infected case, whether it recovers at this timestep
  - $r(t)$

Stochastic SIR model

- Add another component to the SI list

Component	Attributes	Distribution
SI pair	Draw two chips without replacement from a finite population	$Hypergeometric$ Not approx. $Binomial(n, p)$ for large n, see appendix
$a(t)$	Z+, optional maximum	$Binomial$ if max is 1 act per pair
$m(a)$	{0,1}, like a coin flip	$Bernoulli(\tau)$
$r(t)$	{0,1}, a coin flip (at each time step) Or Z+ if D drawn at time of infection	$Bernoulli(\rho)$ or $Poisson(D = 1/\rho)$

SIR poker chip exercise

- What component(s) did we make stochastic?

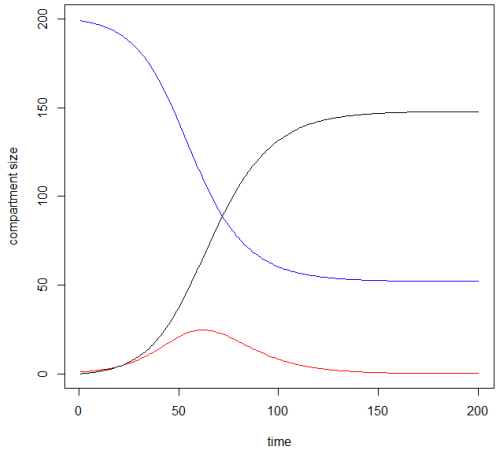
Component	Stochastic or Deterministic?	Value or Distribution
SI pair	stochastic	Hypergeometric
$a(t)$	deterministic	Fixed at 1 per day
$m(a)$	deterministic	Fixed at 1 for all acts
D	deterministic	Fixed at 10 days

Still just this

DCM SIR model (by comparison)

- The flows are:
  - $\text{Incidence(t)} = \alpha\tau * \frac{s(t)i(t)}{n}$
  - $\text{Recoveries(t)} = \rho * i(t)$

rates * aggregate values



- In both flows
  - The parameters are *rates*, not probabilities
  - Applied to aggregate compartment stocks, which may be fractional
  - The outcome flow values can also be fractional

And at each point in the time series the outcome values are always the same

How are ICMs implemented?

A simple algorithm

At least it's simple for the poker chips:

- Replicate each step
- In a line of code

ICM SIR pseudocode

```
# Initial conditions
     # create individuals
     # assign status (S, I)

# Simulate epidemic
for (at=1:num.timesteps) {
     # infection
          # draw the number of acts for that step
          # draw 1 pair of elements for each act
          # filter to just the discordant SI pairs
          # flip coin for each pair to determine transmission (or not)
          # do bookkeeping for new infections
     # recovery
          # identify infected elements
          # flip coin for each case to determine recovery
          # do bookkeeping for recoveries
}
# process output
```

ICM SIR code

- The appendix to this slideset has some actual code
  - It's the code used in the EpiModel `epiweb(icm)` shiny app
  - All fits on one page (albeit in small type)

- You'll use the `epiweb(icm)` shiny app in the next lab
  - It's a GUI, so you won't see the code
  - But now you know what's going on behind the curtain ☺

- We'll move on to network models after the lab
  - After a break for mid-day

Summary

- Stochastic ICMs replace
  - aggregate stocks with individual elements
  - fixed rates with draws from a probability distribution
    - There can be a mix of rates and probabilities

- Key benefits
  - Insight into the inherent variability in a process
    - Highest at the beginning of an epidemic, and in small populations
  - More control over “heterogeneities”
    - In both elements and transition processes

To the ICM lab

With EpiModelWeb

Appendices

1. Stochastic compartmental models
2. Hypergeometric distribution derivation
3. Stochastic iCM code from `epiweb(icm)`

1. Stochastic Compartmental Model

All about the transitions

Stochastic compartmental model

- This is not a common framework
- But it's useful for understanding the continuum
  - From purely deterministic, to purely stochastic
- And it does provide one way to generate variability in DCM outputs
  - Variability helps you quantify your uncertainty

Stochastic compartmental model

- How can you make a compartmental model stochastic?
  - By making the transition rate parameters or "flows" in the model stochastic

- Consider a simple proportional growth model
  - **States**: only I is tracked; population has an infinite number of susceptibles
  - **Transition rate parameters**: only β, the average growth rate of infection

- As a compartmental model, this would be:

$$i(t+1) = i(t) + \beta i(t) \qquad \text{so:}$$

$$incidence(t) = i(t+1) - i(t) = \beta i(t)$$

Stochastic compartmental model

Deterministic	Stochastic
$incidence(t) = \beta i(t)$	$P(incidence(t) = k) = P(k \mid \beta, i(t))$
Fixed rate of new infections per prevalent case $i(t)$	Stochastic number of new infections per prevalent case $i(t)$
determined by a rate β	drawn from a probability distribution with an expected value (mean) of $\beta i(t)$

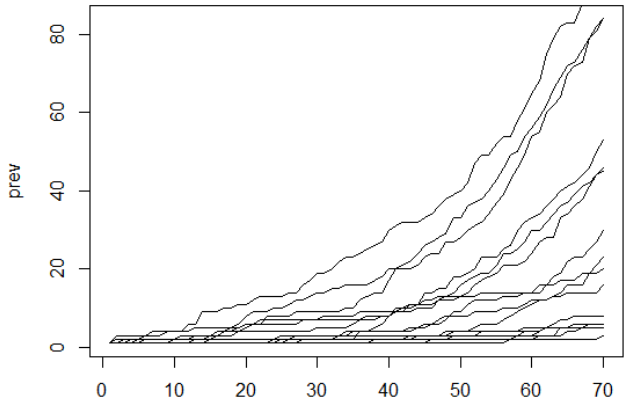
Stochastic compartmental model

- What does $P(k \mid \beta, i(t))$ equal?
  - Depends on the model you choose for the probability distribution P(●)
  - Probability of what? That the count of new infections $= k$ at time t

- So what kind of distributions are appropriate?
  - proper probability distributions ($\sum_k P(k) = 1$)
  - For discrete random variables (k takes integer values only)
  - non-negative integers only

- So the Poisson distribution is appropriate here too

Stochastic compartmental model



Each line represents a different realization of the epidemic trajectory for β = 0.05,

with $incidence(t)$ a stochastic draw from a Poisson($\lambda = \beta i(t)$) distribution

- This is one way to add stochasticity to a compartmental model
  - Provides a means to quantify the potential variation in outcomes
- But note that we are still only counting aggregates – there are no explicitly represented individuals

2. Hypergeometric distribution

Deriving the probability of choosing an SI pair

See Wikipedia for a good overview

What is the probability of an SI pair?

- For each draw:
  - Fixed $N\ (= S(t) + I(t))$
  - Draw one chip, then the second without replacement
    - Think of S as "success" and I as "failure"
    - Possible outcomes: SS, II, SI (depending on (t))
- The Hypergeometric distribution
  - It's not Binomial, because you draw without replacement
  - So the draws are dependent

Hypergeometric derivation

Probability of the outcome: $\frac{event\ count}{sample\ space}$

1. Enumerate the sample space:
   - With 10 marbles, how many ways to pick 3? $\binom{10}{3}$
2. Count how many outcomes meet the condition (1R, 2G)?
   - How many ways to pick 1 of the 6 reds? $\binom{6}{1}$
   - How many ways to pick 2 of the 4 greens? $\binom{4}{2}$
   - How many ways for both of these to happen? $\binom{6}{1}\ \binom{4}{2}$

So the probability is defined by: $\mathrm{h}(r=1; N=10, n=3, R=6) = \frac{\binom{6}{1}\binom{4}{2}}{\binom{10}{3}}$

Hypergeometric PMF

- General form:

$$\mathrm{h}(x; N, n, K) = \frac{\binom{K}{x}\binom{N-K}{n-x}}{\binom{N}{n}}$$

x = number of outcomes of interest (red balls drawn)

K = total number of possible outcomes of that type (6 red balls in urn)

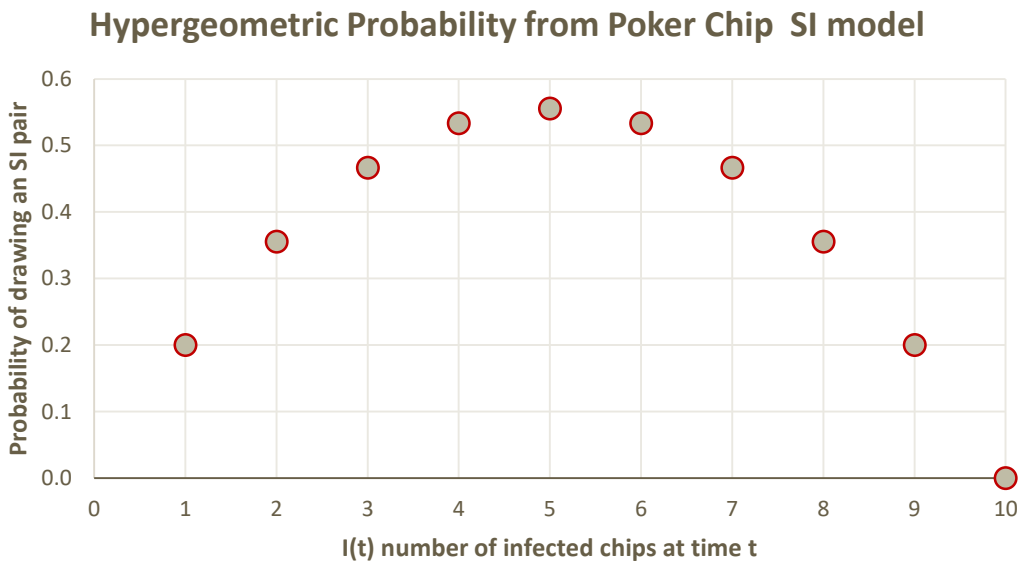
N = population of individual outcomes (total balls in urn)

n = number of outcomes sampled (number of balls drawn)

In the poker chip SI exercise

- We drew one pair each day
  - $N = 10 = S(t) + I(t)$
  - Draw 2, one chip from each state

$$h(s = 1;\ 10, S(t), 2) = \frac{\binom{S(t)}{1}\binom{I(t)}{1}}{\binom{10}{2}}$$



As you saw, the probability of drawing an SI pair changed (stochastically) as the epidemic progressed

Key point

The signature incidence curve for the SI model
Matches the curve for the hypergeometric draw

The contact process
generates the shape of the incidence curve

- So the assumptions we make there are particularly important
  - Finite population (leads to depletion of S)
  - Random mixing (the SI draws, hypergeometric)

What about the ICM SIR model?

The probability of choosing an SI pair changes

- Because there are more types of pairs you can draw
  - SS, SR, RR, II and SI
- So this is a multivariate hypergeometric distribution

$$\mathrm{h}(s=1, i=1;\ 10, S(t), I(t), 2) = \frac{\binom{S(t)}{1}\binom{I(t)}{1}\binom{R(t)}{0}}{\binom{10}{2}}$$

Wikipedia has good info on distributions

Article Talk Read Edit View history Search Wikipedia

Hypergeometric distribution

From Wikipedia, the free encyclopedia

In probability theory and statistics, the **hypergeometric distribution** is a discrete probability distribution that describes the probability of k successes (random draws for which the object drawn has a specified feature) in n draws, *without* replacement, from a finite population of size N that contains exactly K objects with that feature, wherein each draw is either a success or a failure. In contrast, the binomial distribution describes the probability of k successes in n draws *with* replacement.

In statistics, the **hypergeometric test** uses the hypergeometric distribution to calculate the statistical significance of having drawn a specific k successes (out of n total draws) from the aforementioned population. The test is often used to identify which sub-populations are over- or under-represented in a sample. This test has a wide range of applications. For example, a marketing group could use the test to understand their customer base by testing a set of known customers for over-representation of various demographic subgroups (e.g., women, people under 30).

Contents [hide]

1 Definition
2 Combinatorial identities
3 Application and example
3.1 Application to auditing elections
3.2 Application to Texas Hold'em Poker
4 Symmetries
5 Hypergeometric test
5.1 Relationship to Fisher's exact test
6 Order of draws
7 Related distributions
8 Tail bounds
9 Multivariate hypergeometric distribution
9.1 Example
10 See also
11 Notes
12 References
13 External links

Definition [edit]

The following conditions characterize the hypergeometric distribution:

- The result of each draw (the elements of the population being sampled) can be classified into one of two mutually exclusive categories (e.g. Pass/Fail or Employed/Unemployed).
- The probability of a success changes on each draw, as each draw decreases the population (*sampling without replacement* from a finite population).

A random variable X follows the hypergeometric distribution if its probability mass function (pmf) is given by[1]

$$P(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}},$$

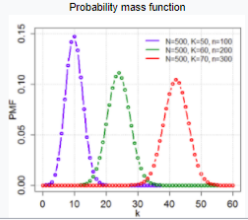
where

- N is the population size,
- K is the number of success states in the population,
- n is the number of draws,
- k is the number of observed successes,
- $\binom{a}{b}$ is a binomial coefficient.

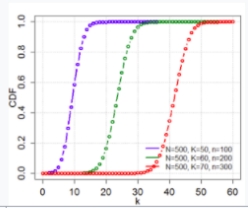
The pmf is positive when $\max(0, n + K - N) < k < \min(K, n)$.

Hypergeometric

Probability mass function



Cumulative distribution function



Parameters	$N \in \{0, 1, 2, \ldots\}$ $K \in \{0, 1, 2, \ldots, N\}$ $n \in \{0, 1, 2, \ldots, N\}$
Support	$k \in \{\max(0, n+K-N), \ldots, \min(n, K)\}$
pmf	$\frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$
CDF	$1 - \frac{\binom{n}{k+1}\binom{N-n}{K-k-1}}{\binom{N}{K}} {}_3F_2\left[\begin{matrix}1, k+1-K, k+1-n \\ k+2, N+k+2-K-n\end{matrix}; 1\right]$, where ${}_pF_q$ is the generalized hypergeometric function
Mean	$n\frac{K}{N}$
Mode	$\left\lfloor \frac{(n+1)(K+1)}{N+2} \right\rfloor$
Variance	$n\frac{K}{N}\frac{(N-K)}{N}\frac{N-n}{N-1}$

3. Code from `epiweb(icm)`

SIR model, step by step

Setup

- create individuals

```
ids <- 1:num                          # num = the initial # of individuals
```

- assign status

```
status <- rep("s", num)               # init.inum = the initial # of infecteds
status[sample(ids,
          size = init.inum)] <- "i"
```

```
> status
[1] "s" "s" "s" "i" "s" "s" "s" "i" "i" "i" "i" "s" "s" "i" "s" "s" "s"
```

Infection process

- Step 1: calculate number of acts

```
# n Acts per Time Step = fixed act rate * n/2
acts <- round(act.rate * num[at - 1] / 2)
```

- Note: this is a deterministic rate.
- How would you change this code to make it stochastic?

Infection process

- Step 2: determine who has an act with whom

```
# Make edgelist of partnerships by ID number
el <- t(replicate(acts, sample(1:num, 2)))
```

```
     [,1] [,2]
[1,]   80    9
[2,]    9   59
[3,]    5   66
[4,]    4   84
```

Infection process

- Step 3: limit edge list to discordant pairs

```
# look up the status of each member of the pair
discordant <- (status[el[, 1]] == "i" & status[el[, 2]] == "s") |
         (status[el[, 1]] == "s" & status[el[, 2]] == "i")
[1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE

# create a "discordant edgelist"
del <- el[discordant == TRUE, ]
     [,1] [,2]
[1,]   80    9
[2,]    9   59
[3,]    5   66
[4,]   29   38
```

Infection process

- Step 4: determine infections

```
# Infection is a Bernoulli draw for each discordant pair
infections <- rbinom(nrow(del), 1, tprob)

> infections
[1] 1 0 0 1
```

Infection process

- Step 5: bookkeeping for infections

```
# Limit discordant edge list to pairs with incident infection
del <- del[infections == TRUE, ]

# Look up newly infected ID in each pair
susIds <- ifelse(status[del[, 1]] == "s", del[, 1], del[, 2])
newInfIds <- susIds[infections == 1]

# Update individual-level status attribute
status[newInfIds] <- "i"
```

Recovery process

```
# Identify infected (persons eligible to recover)
idsElig <- which(status == "i")
nElig <- length(idsElig)

# Draw random numbers to determine recoveries
vecRecov <- which(rbinom(nElig, 1, rec.rate) == 1)

# Do bookkeeping
if (length(vecRecov) > 0) {
    idsRecov <- idsElig[vecRecov]
    nRecov <- length(idsRecov)
    status[idsRecov] <- "r"
}
```

Wrap up

- Process output

```
# Calculate summary statistics
prevalence <- sum(status == "i")
incidence <- length(newInfIds)
```

epiweb(icm) SIR : full code

```
ids <- 1:num                                                    # initial # of individuals
status <- rep("s", num)
status[sample(ids,  size = init.inum)] <- "i"                   # initial # of infecteds

acts <- round(act.rate * num[at - 1] / 2)                       # n Acts per Time Step
el <- t(replicate(acts, sample(1:num, 2)))                      # Edgelist of partnerships by ID
discordant <- (status[el[, 1]] == "i" & status[el[, 2]] == "s") |
         (status[el[, 1]] == "s" & status[el[, 2]] == "i")      # Status lookup
del <- el[discordant == TRUE, ]                                 # Find "discordant edgelist"
infections <- rbinom(nrow(del), 1, tprob)                       # Infection is a Bernoulli draw
del <- del[infections == TRUE, ]                                # Incident pairs
susIds <- ifelse(status[del[, 1]] == "s", del[, 1], del[, 2])   # Inci ID lookup
newInfIds <- susIds[infections == 1]                            # Update individual infection status
status[newInfIds] <- "i"
idsElig <- which(status == "i")
nElig <- length(idsElig)

vecRecov <- which(rbinom(nElig, 1, rec.rate) == 1)              # Recovery is a Bernoulli draw
if (length(vecRecov) > 0) {                                     # Update individual recovery status
    idsRecov <- idsElig[vecRecov]
    nRecov <- length(idsRecov)
    status[idsRecov] <- "r"
}
prevalence <- sum(status == "i")                                # Calculate summary statistics
incidence <- length(newInfIds)
```