Stochasticity

EES 4760/5760

Agent-Based & Individual-Based Computational Modeling

Jonathan Gilligan

Class #19: Wednesday Mar. 21 2018

Live web page: https://ees4760.jgilligan.org/Slides/Class_18
PDF: https://ees4760.jgilligan.org/Slides/Class_18/EES_4760_5760_Class_18_Slides.pdf

Model to Download

- Download Page:
  - ees4760.jgilligan.org/downloads/stochasticity_class_18/
- Zip File: class_18_models.zip, which contains:
- Stochastic Business-Investor model:
  - NetLogo model: business_investor_class_18.nlogo
  - Testing library: jg-tif.nls

Vignette: Participatory Agent-Based Modeling for Decision Support

Managing Groundwater in Chicago

- C. Hoch *et al*., Planning Theory & Practice **16**, 319 (2015),
- J. Radinsky *et al*., J. Environ. Planning & Management **60**, 1296 (2017).

- Regional groundwater plan
  - Requires coordination among independent suburbs
  - Stakeholders: homeowners, developers, lenders, realtors, businesses, elected officials
- ABM as a tool to explore sustainability under different policies
  - Teach planners to use models
  - Method: Explore impacts on groundwater levels of different policy interventions for land and water use.
  - Research Question: How does interaction with models affect thinking of participants, especially when they discover that their favorite policies do not appear to be sustainable?

Response to Working with Model

- Model did help planners identify and grapple with misleading assumptions, but they were reluctant to discard those assumptions and instead raised objections about model.
  - If model predictions conflicted with preconceptions, tendency to criticize model.
  - Ease of working with model gave impression it was a “toy”
- Cognitive bias is prevalent and difficult to overcome.
- Best practices with ABMs:
  - Moving from *talking about tools* to *using tools* to *talking about planning*
  - Moving from one-dimentional to multi-dimensional/multi-level thinking
  - Moving from one pattern of argumentation to flexible exploration of multiple lines.
- Many challenges
- Some promising results
- Further research needed

Stochasticity:
Why do we use random numbers?

- To “inject ignorance” into a model:
  - We want to represent some kind of variability *but*
  - We do not want all the details of what causes the variability

```
ask patches [set profit 1000 + (random 1000)]
ask turtles [ if random-float 1.0 < mortality-prob [die] ]
```

Common uses of stochasticity

- Initialization

```
set fish-length random-normal 50 length-std-dev
```

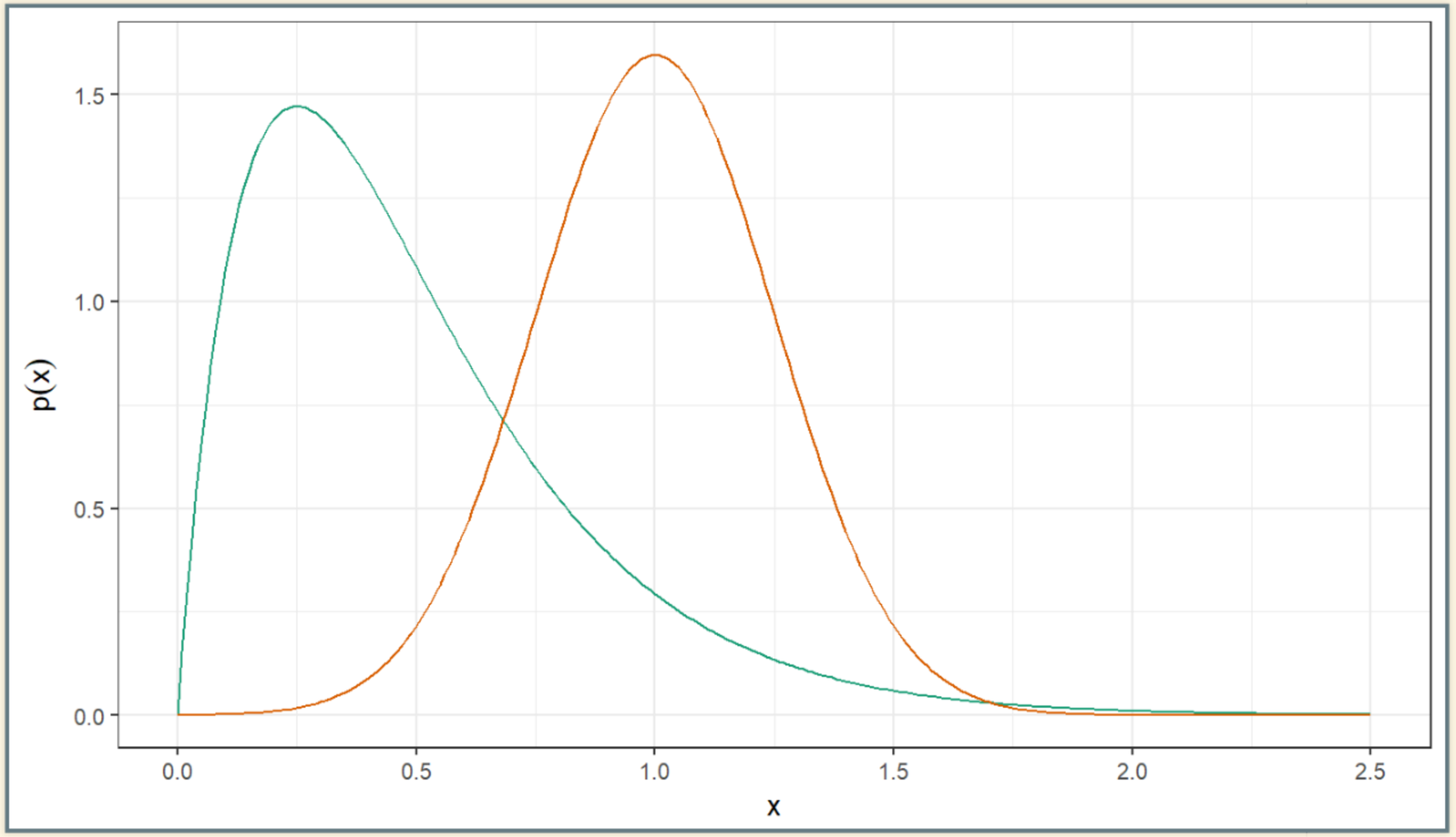
- In submodels

```
ifelse random-float 1.0 < q
[ uphill elevation ]
[ move-to one-of neighbors ]
```

Guidance for Stochasticity

- **Do** use stochasticity to initialize model differently on different runs
  - Makes sure that effects you see are not *artifacts* of a specific initialization
- **Do** use stochasticity to simplify representation of very complex processes
  - If wild dogs live an average of 5 years:
    - instead of a detailed submodel that determines exactly when each dog will die,
    - let dogs die at random with a 20% probability of dying each tick.
- **Don't** use too much stochasticity
  - If you put too many different sources of randomness into your models every run may be *so different* you can't discover any general properties.

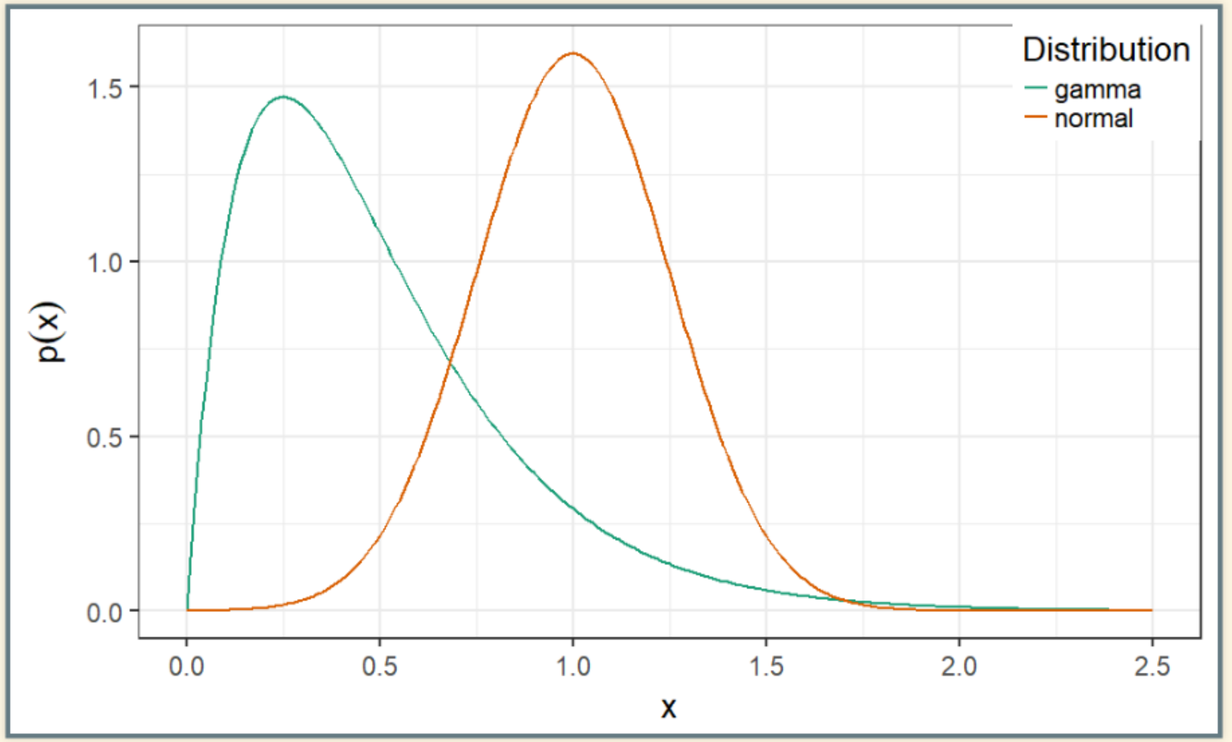
What is a Distribution?



What is a Distribution?

- In simulation programming, an algorithm that produces (pseudo)random numbers that fit a particular statistical distribution.

```
let x1 random-normal 1.0  0.25
let x2 random-gamma  2.0  4.0
```



Distributions in NetLogo

- Continuous (real-number)
  - Uniform: `random-float` *`upper-limit`*
  - Normal: `random-normal` *`mean sd`* (beware of outliers)
  - Also: `random-gamma`, `random-exponential`
- Discrete (integer):
  - Uniform: `random` *`upper-limit`*
    - 0 to *`upper-limit - 1`*
  - Poisson: `random-poisson` *`mean`*
    - `mean` = average value
  - Bernoulli (`true` or `false`): `random-float 1.0 < p`
    - `true` with probability `p`
    - See `random-bernoulli` reporter on p. 200 of the textbook.

Randomness and Pseudo-Randomness

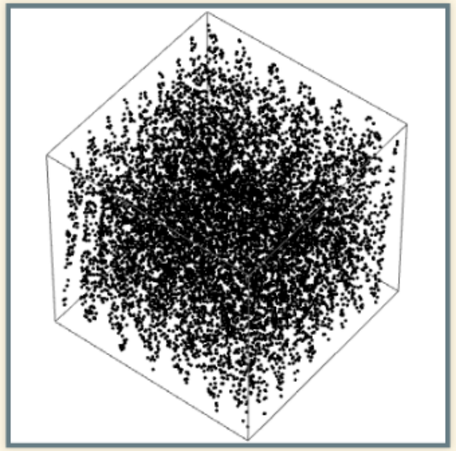
Randomness and Pseudo-Randomness

- We could get true random numbers from physical processes:
  - Radioactive decay
  - Thermal noise in electronic components
  - Chaotic systems (e.g., lava lamps)

- But often you want reproducible random numbers:
  - Pseudo-random number generators
  - Begin with a “seed” number and use it to generate a series of numbers that
    - look random to all appearances,
    - but can be reproduced by starting from the same seed.

Challenges of pseudo-random numbers

- Because pseudo-random number generators use deterministic equations, there will be patterns to their output, but hopefully those patterns won't be very pronounced.
  - Output will eventually repeat exactly, but hopefully not for a very long time.
- RANDU random number generator (IBM, 1960s) turned out to be horrible:
  - Plot triplets of consecutive numbers in three-dimensions



- When told of this problem, an IBM programmer replied that the numbers were guaranteeded to be random when taken one at a time, but not if taken in groups.
- NetLogo has a high-quality random number generator.

Controlling randomness

- `random-seed` *number*
  - As long as *number* is the same, you get the same sequence of random numbers

```
to setup
    clear-all
    random-seed 32149
    ...
end
```

Controlling randomness

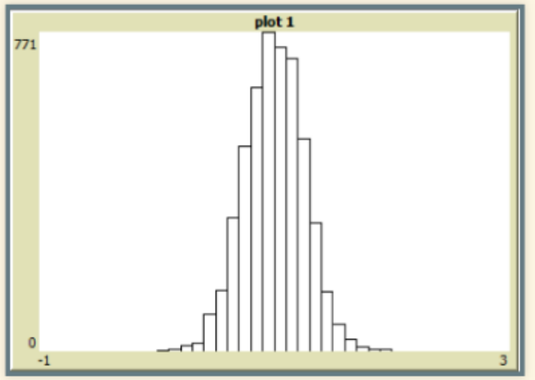
- `with-local-randomness [ commands ]`
  Runs without changing sequence of random numbers in other parts of the model

```
to move
  with-local-randomness
  [
    random-seed 63592
    ...
  ]
end
```

How can we see a distribution?

- Histograms

```
to plot-histogram-normal
  clear-all
  set-plot-pen-mode 1 ; bar mode
  set-plot-pen-interval 0.1
  set-plot-x-range -1 3
  let x (list)
  ; fill x with 5000 random numbers from a normal distribution
  repeat 5000 [ set x fput (random-normal 1.0 0.25) x]
  histogram x
end
```



Uniform distributions

- Integer: `random n` gives an integer i: $0 \leq i < n$
  - From 0 to $(n - 1)$
- Continuous: `random-float z` gives a number x: $0 \leq x < z$
  - Should we worry that $x < z$?

```
to test
  let num_draws 10000
  let max-rand 0
  repeat num_draws
  [
    let x random-float 1000
    if x > max-rand [ set max-rand x ]
  ]
  show max-rand
end
```

```
observer> test
observer: 999.9869678378017
```

Poisson distribution

- For countable things that happen at a small rate.
  - On every turn a random number of agents turn red, with an average of 5% of agents

```
ask n-of (random-poisson (0.05 * count turtles)) turtles
[
  set color red
]
```

or

```
let n random-poisson (0.05 * count turtles)
ask n-of n turtles [set color red]
```

Normal distribution

- For measurable things with an average value

```
set weight random-normal 150 20  ; weight in pounds
set height random-normal 70 2    ; height in inches
```

Stochastic Business Investors

Stochastic Business Investors

Model: ees4760.jgilligan.org/models/class_19/business_investor_class_19.nlogo

Original model:

Investors move to neighbor with highest expected utility (including own patch)

Average over 10,000 runs:

Alternative	Frequency
Higher profit, lower risk	83.3%
Higher profit, higher risk	5.4%
Lower profit, lower risk	4.9%
Lower profit, higher risk	0%
Don’t move	92.7%

- Mean wealth = $128,400
- Total wealth = $12,000,000

Stochastic Model

Original model:

Alternative	Frequency
Higher profit, lower risk	83.3%
Higher profit, higher risk	5.4%
Lower profit, lower risk	4.9%
Lower profit, higher risk	0%
Don’t move	92.7%

Stochastic model

- If there are neighbors with higher profit and lower risk:
  - 83.3% probability of moving to one of them
- Otherwise, if there are neighbors with higher profit and higher risk:
  - 5.4% probability of moving to one of them
- etc.

Compare models:

Original model:

Alternative	Frequency
Higher profit, lower risk	83.3%
Higher profit, higher risk	5.4%
Lower profit, lower risk	4.9%
Lower profit, higher risk	0%
Don’t move	92.7%

- Mean wealth = $128,400
- Total wealth = $12,000,000

Stochastic model:

???