3. Randomness and probability: Stochastic thinking

Last time

- Modelling
- Dynamics of *deterministic* systems with *deterministic* rules (no randomness)
- $I_{n+1} = I(x_n)$

Last time

- Modelling
- Dynamics of *deterministic* systems with *deterministic* rules (no randomness)
- $I_{n+1} = I(x_n)$
- `for` loops, `Vector`s
- Plotting, interactive visualization

Goals for today

- Adding data to a `Vector`
- Fundamental ideas of **probability**
- Discrete **random variables**
- Understanding via calculation + visualization
- **Variability** of random variable: probability distribution

Adding data to a `Vector`

- Last time defined zero vector using `zeros(N)`
- Often instead want to add more data to a vector

Adding data to a `Vector`

- Last time defined zero vector using `zeros(N)`
- Often instead want to add more data to a vector

- Create vector using `xs = [x0]`
- Add one data point using `push!(xs, x)`

Randomness and probability

Why randomness / stochasticity?

- Many things in world behave **predictably**
- E.g. dropping a ball onto the ground
- So use **deterministic** model

Why randomness / stochasticity?

- Many things in world behave **predictably**
- E.g. dropping a ball onto the ground
- So use **deterministic** model

- Other processes are **random** (or *seem* so)
- E.g. time to recover from infection

Randomness as uncertainty

- Even many deterministic systems behave unpredictably after a short time – **chaos**
- E.g. Lorenz system – model of weather

Randomness as uncertainty

- Even many deterministic systems behave unpredictably after a short time – **chaos**
- E.g. Lorenz system – model of weather
- **Brownian motion** (1827): particle immersed in water
- **Model** as bouncing balls
- Deterministic, chaotic many-body dynamical system
- Simulation
- Randomness $\equiv$ *unknown information* in dynamical processes.

Recovery from infection

- Simple model of recovery from infection:

Recovery from infection

- Simple model of recovery from infection:
- Constant recovery “rate”

Recovery from infection

- Simple model of recovery from infection:
- Constant recovery “rate”
- Model as constant **probability** p to recover each day

Recovery from infection

- Simple model of recovery from infection:
- Constant recovery “rate”
- Model as constant **probability** p to recover each day
- Models fact that peoples’ immune systems behave differently
- Affected by temperature, nutrition, dust in house etc.

Some relevant questions

- How simulate this **stochastic process**?

Some relevant questions

- How simulate this **stochastic process**?
- How long will infection last?

Some relevant questions

- How simulate this **stochastic process**?
- How long will infection last?
- Does it make sense to talk about an *average* infection duration?

Some relevant questions

- How simulate this **stochastic process**?
- How long will infection last?
- Does it make sense to talk about an *average* infection duration?

- Is this, in the end, a “good model”?
- Does it reproduce the data?

Computing using randomness

- How generate randomness on computer?

Computing using randomness

- How generate randomness on computer?
- Answer 1: Computers are deterministic, so *we can't*!

Computing using randomness

- How generate randomness on computer?
- Answer 1: Computers are deterministic, so *we can't*!
- Answer 2: Start-up sequence of computer generates "entropy" = unpredictable bits

Computing using randomness

- How generate randomness on computer?
- Answer 1: Computers are deterministic, so *we can't*!
- Answer 2: Start-up sequence of computer generates "entropy" = unpredictable bits
- Answer 3: Use real physical process, e.g. noise from electronic circuit or atmospheric noise: www.random.org

Pseudo-random numbers

- Answer 4: Generate “random-looking” sequences using deterministic process
- E.g. “linear congruential generator” (1970s):
- $x_{n+1} = (ax_n + b)$ mod m

```
const a = UInt(6364136223846793005)  # unsigned integer
const b = UInt(1442695040888963407)

my_rand_int(x) = a*x + b

x = UInt(3)
for i in 1:10
    global x = myrandint(x)
    y = x / typemax(UInt)  # convert to interval [0, 1)
    @show y
```

Random numbers in Julia

- Uses sophisticated modern random number generator: Mersenne twister
- `rand()` generates number distributed **uniformly** between 0 and 1

Random numbers in Julia

- Uses sophisticated modern random number generator: Mersenne twister
- `rand()` generates number distributed **uniformly** between 0 and 1
- How generate event **with probability** $\frac{1}{2}$?

Random numbers in Julia

- Uses sophisticated modern random number generator: Mersenne twister
- `rand()` generates number distributed **uniformly** between 0 and 1

- How generate event **with probability** $\frac{1}{2}$?
- **With probability** $\frac{1}{4}$?

Random numbers in Julia

- Uses sophisticated modern random number generator: Mersenne twister
- `rand()` generates number distributed **uniformly** between 0 and 1

- How generate event **with probability** $\frac{1}{2}$?
- **With probability** $\frac{1}{4}$?
- **With probability** p**?**

Generating boolean with probability p

- Want to “capture” only a certain **fraction** of all events.
- Simple algorithm to generate even with probability p:

1. Generate uniform random variate r in interval $[0, 1)$
2. If $r < p$, return `true`
3. Else return `false`

Generating boolean with probability p

- Want to "capture" only a certain **fraction** of all events.
- Simple algorithm to generate even with probability p:

  1. Generate uniform random variate r in interval $[0, 1)$
  2. If $r < p$, return `true`
  3. Else return `false`

- Called **Bernoulli trial**
- "Flipping a biased coin"

Random variables

- Think of this process as running an **experiment**
- Call X_i the output of the ith run, or **trial**

Random variables

- Think of this process as running an **experiment**
- Call X_i the output of the ith run, or **trial**
- Each X_i can take value 0 or 1
- Which value it takes is random

Random variables

- Think of this process as running an **experiment**
- Call X_i the output of the ith run, or **trial**
- Each X_i can take value 0 or 1
- Which value it takes is random

- X_i is a **random variable**
- Called a **Bernoulli random variable**

Modelling recovery

- Let's return to modelling recovery from infection
- Recover at each time step with probability p

Modelling recovery

- Let's return to modelling recovery from infection
- Recover at each time step with probability p
- Basic question: **How long does it take until I recover**?

Modelling recovery

- Let's return to modelling recovery from infection
- Recover at each time step with probability p
- Basic question: **How long does it take until I recover**?

- At each time step i generate a Bernoulli random variable with probability p of success (recovery)

Modelling recovery

- Let's return to modelling recovery from infection
- Recover at each time step with probability p
- Basic question: **How long does it take until I recover**?

- At each time step i generate a Bernoulli random variable with probability p of success (recovery)
- If $X_i = 1$, we recover
- Else keep going

Modelling recovery

- Let's return to modelling recovery from infection
- Recover at each time step with probability p
- Basic question: **How long does it take until I recover**?

- At each time step i generate a Bernoulli random variable with probability p of success (recovery)
- If $X_i = 1$, we recover
- Else keep going
- I.e. Wait until **first time** τ when value is 1
- Simple example of a **first-passage problem**

Random variables

- Time to recovery τ is also a **random variable**:
- Each time run simulation will get different **outcome**

Random variables

- Time to recovery τ is also a **random variable**:
- Each time run simulation will get different **outcome**

- What questions are we interested in?

Random variables

- Time to recovery τ is also a **random variable**:
- Each time run simulation will get different **outcome**

- What questions are we interested in?
- Average time to recovery

Random variables

- Time to recovery τ is also a **random variable**:
- Each time run simulation will get different **outcome**

- What questions are we interested in?
- Average time to recovery
- How large fluctuations are

Random variables

- Time to recovery τ is also a **random variable**:
- Each time run simulation will get different **outcome**

- What questions are we interested in?
- Average time to recovery
- How large fluctuations are
- More generally, want **frequency** of each outcome: **probability distribution**.

Computational thinking: Do the experiment!

- Computers are good at experiments like this:
  - Generate data
  - Count how many times each possible outcome occurs

Computational thinking: Do the experiment!

- Computers are good at experiments like this:
  - Generate data
  - Count how many times each possible outcome occurs
- Run experiment many times: **Monte Carlo simulation**

Variability

- We have **finite sample** from ideal **population**
- If repeat experiment, get different sample with different counts
- Plot implies die is *biased* (non-uniform) – one bar taller than others.
- But repeating calculation gives *different* results each time
- How characterize this *variability*?

Review

- Random variables have set of possible outcomes
- Probability distribution measures how frequently each outcome occurs
- Variability between different experiments measured by mean and variance