Random Numbers & Simulations

stat 480
Heike Hofmann
Outline

- True and Pseudo Random Number Generators
- Tests for Randomness
- Simulations
- Permutation Tests
Famous Quotes

• “The generation of random numbers is too important to be left to chance.”
  – Robert R. Coveyou

• “Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin.”
  – John von Neumann
Truly Random Number Generators

- e.g. monitoring of physical processes
- example: www.random.org
- speed?
- reproducibility?
Your Turn

• Go to www.random.org

• Find out, how random numbers are “generated” there.

• Draw 5 random integer numbers between 1 and 100.

• What is the speed of the service? cost? reproducibility?
Random (?) Sequences ...

- 2, 4, 6, 8, 10, ...
- 0, 3, 8, 15, 24, ...
- 3, 1, 4, 1, 5, 9, 2, ...
- 1, 11, 21, 1211, 111221, 312211, ...
- 1, 24, 57, 0, 53, 16, ...
Pseudo-Random Number Generators

- Pre-determined function that produce “random looking” numbers
- Speed?
- Reproducibility?
Random Number Generators (RNGs)

- RNGs produce a stream of random numbers \( U_1, U_2, U_3, \ldots \)
- Usually, \( U_1, U_2, U_3, \ldots \sim \text{U}[0,1] \), i.e. every number between 0 and 1 has the same “chance” of being chosen.
- \( U_1, U_2, U_3, \ldots \) are independent
- truly random vs pseudo random
Linear Congruential Method

• seed $x_0$

• $x_{i+1} = (a \times x_i + c) \mod m$

• define $u_i = x_i / m$

• different choices of $a, c, m$ result in different properties of RNG
Your Turn

- In R, write function
  \[ \text{lcg}(n, \text{seed}, m, a, c) \]
  that produces a sequence of \( n \) pseudo random numbers from a linear congruential generator with parameters \((m, a, c)\) starting with the specified seed.

- Try out \( \text{lcg}(10, 1, 100, 71, 53) \times 100 \)

- For \( m = 2^{31}, a = 65539, c = 0 \) and seed = 171717 produce 5000 random numbers.

- How do you “test” for randomness?
“Testing” for Randomness

- Test suites, e.g. DieHard (G. Marsaglia), Dieharder (R. Brown)

- General idea:
  Pick distributional property,
  compute theoretical behavior & distribution,
  generate sequence from RNG,
  do hypothesis test

- e.g. #0s and #1s should be equal in binary representation
Aims

• Learn how to simulate data to:
  • Test your statistical intuition
  • Perform power calculation
  • Experiment with a new technique on known data
• Learn how to use functions to reduce duplication
Basics of simulation

• Want to:
  • generate random numbers from known distribution
  • want to repeat the simulation multiple times
Generating random numbers in R

- `runif` (uniform), `rpois` (poisson), `rnorm` (normal), `rbinom` (binomial), `rgamma` (gamma), `rbeta` (beta)

- First argument for all is \( n \), number of samples to generate

- Then parameters of the distribution (always check that the distribution is parameterized the way you expect)
Your turn

• Generate 100 numbers ~ N(0, 1)
• Generate 50 numbers ~ N(10, 5)
• Generate 1000 numbers ~ Poisson(50)
• Generate 10 numbers ~ Beta(0.1, 0.1)
• Generate 30 numbers ~ Uniform(0, 10)
Your turn

Monte Carlo Integration:
What is the area of a circle of radius $r=1$?

- Generate two vectors $x$ and $y$ as samples from random uniform $U[-1,1]$
- Calculate vector hit that is true, if $x^2+y^2 < 1$
- Compute the ratio of number of hits compared to number of tries
- Calculate the area from the previous ratio - what is the size of the circle?