CS368 MATLAB Programming

Lecture 9

Young Wu

Based on lecture slides by Michael O'Neill and Beck Hasti

March 30, 2022

Random Choice

Quiz

- Select a random choice.
- $A : A$
- $B : B$
- $C : C$
- $D : D$
- $E : E$

Q1

Pseudo Randomness

Math

- Truly random numbers are difficult or impossible to generate.
- A sequence of pseudo-random numbers is a deterministic sequence with complicated pattern that looks random to users who do not know the pattern therefore cannot predict the next number in the sequence.

Random Number Generator

Math

- A simple pseudo-random number generator is the Linear Congruential Generator (LCG).

1. Start with a seed x_0
2. Compute $x_{n+1} = (ax_n + c) \mod m$ for some m, a, c unknown to the user.

- The resulting sequence $\frac{x_0}{m}, \frac{x_1}{m}, \ldots$ is approximately uniformly distributed between 0 and 1, including 0, not including 1.
- For example, Java uses $m = 2^{48}$, $a = 25214903917$, and $c = 11$
- MATLAB uses another more complicated algorithm called Mersenne Twister.

Random Number Generation

Code

- *rand()* generates a uniform random number between 0 and 1, including 0, not including 1.
- *rand() * u* generates a uniform random number between 0 and u, including 0, not including u.
- *rand() * (u − l) + l* generates a uniform random number between l and u, including l, not including u.
- *rand(n, m)* creates an $n \times m$ matrix of random numbers between 0 and 1.

Integer Random Numbers

Code

floor(rand() · n) + 1

- *randi(n)* generates a uniform random integer between 1 and *n*, including 1 and *n*.
- *randi([l, u])* generates a uniform random integer between *l* and *u*, including *l* and *u*.
- *randperm(n)* generates a random permutation of *1:n*.
- *randperm(n, k)* with $k \leqslant n$ generates a random sample from *1:n* of size *k*, sampled without replacement.

v shuffle

v(randperm(n))

size(v)

n, k

Random Variable, Discrete

Math

- A discrete random variable is a random variable that takes on a finite (or countable) number of values with positive probabilities.
- The probability that the random variable X takes on value x in $\{1, 2, 3, ...\}$ is denoted by $f(x) = \mathbb{P}\{X = x\}$. The function f is called the probability mass function.
- A random number generated based on the probabilities specified by f is called a realization of the X.

Cumulative Distribution Functions, Discrete

Math

- The cumulative probability that the random variable X takes on value less than x is denoted by $F(x) = \mathbb{P}\{X \leqslant x\} = \sum_{i=1}^{x} \mathbb{P}\{X = i\}$. The function F is called the cumulative distribution function (CDF).
- The CDF can be efficiently computed using a for loop.

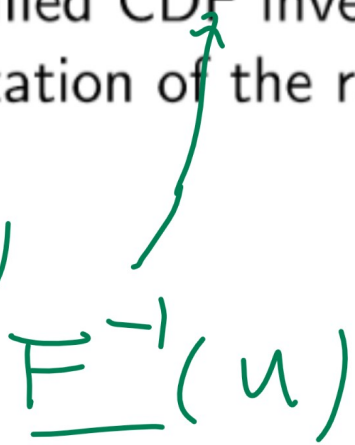
1. $F(1) = \mathbb{P}\{X = 1\}$.
2. $F(x) = F(x-1) + \mathbb{P}\{X = x\}, x > 1$.

Inverse Transform Sampling

Math

- The inverse transform sampling (also called CDF inversion method) can be use to generate a realization of the random variable X.

1. Generate $u \sim \text{Uniform}(0, 1)$.
2. Compute CDF of X, call it $F(x)$.
3. Find the largest x such that $F(x) < u$.



For Loop, Review

Code

- Use a for loop to compute the CDF based on the probabilities stored in a vector p, where $p_i = \mathbb{P}\{X = i\}, i = 1, 2, ..., n$.

1. $f = zeros(n);$
2. $f(1) = p(1);$
3. $for\ \ t = 1{:}n$
4. $\quad f(t) = f(t-1) + p(t);$
5. end

$p = [0.1 \quad 0.2 \quad 0.3 \quad 0.4]$

$cdf = [0 \quad 0 \quad 0 \quad 0]$

$0.1 \quad 0.3 \quad 0.6 \quad 1$

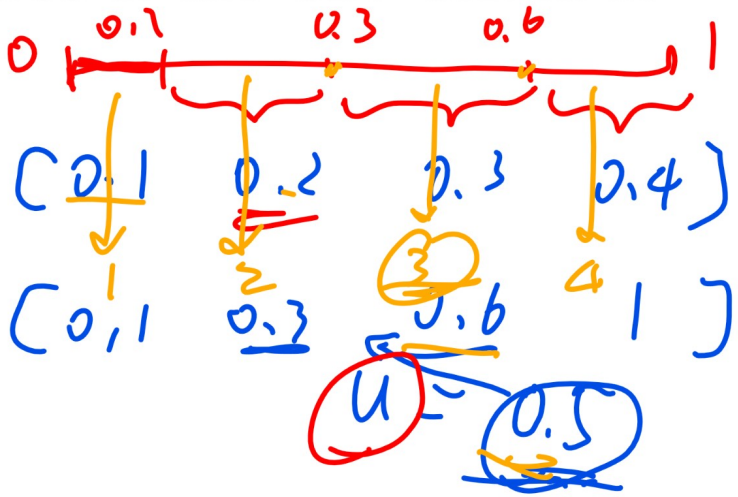
$\leq 1 \quad \leq 2 \quad \leq 3 \quad \leq 4$

- $cumsum(x)$ finds the same CDF.

While Loop

Code

- A while loop is used when the loop stops after an unknown number of iterations until some condition is met.
- Use a while loop to compute the inverse of the CDF stored in a vector f, where $f_i = \mathbb{P}\{X \leqslant i\}$.



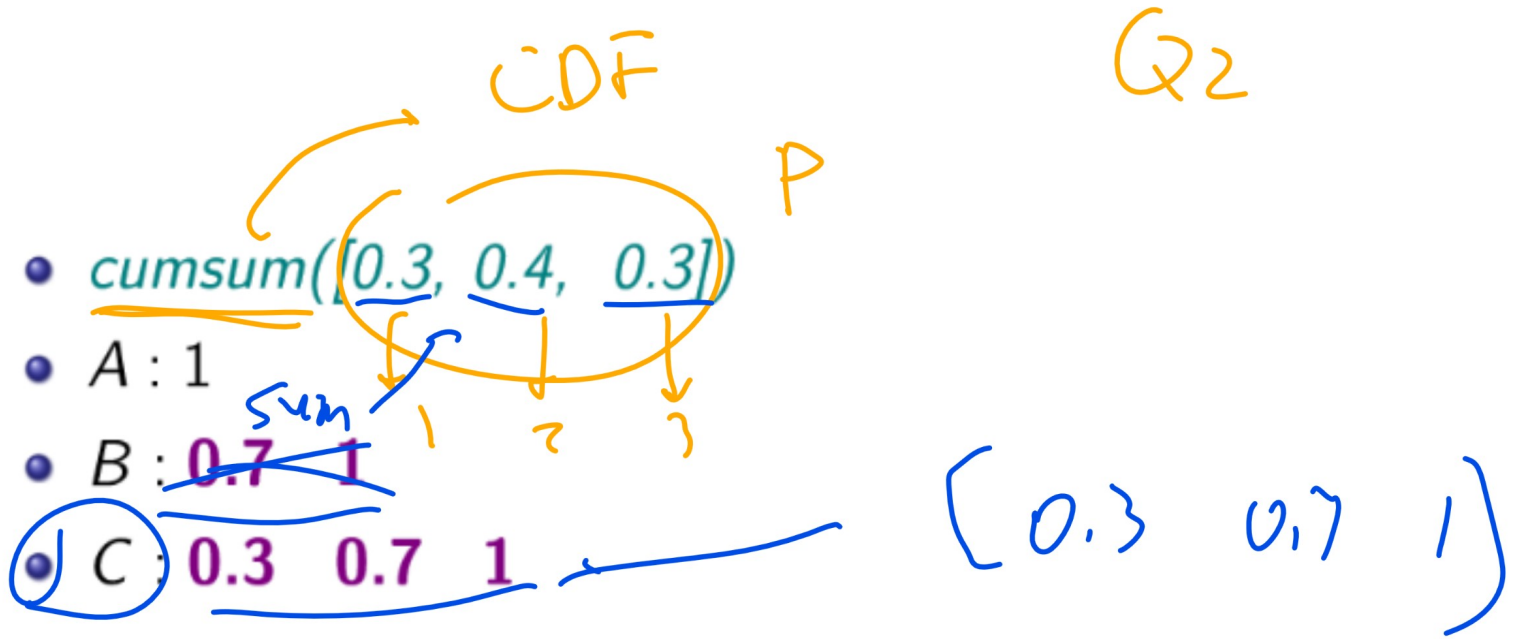
1. $t = 1;$
2. $while\ f(t) <= x$
3. $\quad t = t + 1;$
4. end

$= t$

- $sum(f <= x) + 1$ or $find(x < f,\ 1)$ finds the same inverse CDF.

Random Variable, CDF

Quiz

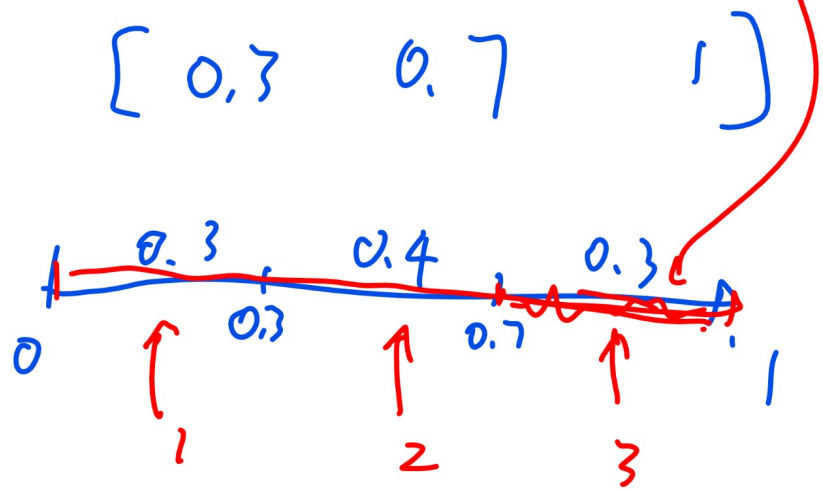


Random Variable, Realization

Quiz

1. $cdf = cumsum([0.3, 0.4, 0.3]);$
2. $rng(0);\ u = rand();$ % it has value $u = 0.8147$
3. $sum(cdf <= u) + 1$

- $A : 0$
- $B : 1$
- $C : 2$
- $D : 3$



Random Variable, Realization

Quiz

1. $cdf = cumsum([0.3, 0.4, 0.3])$
2. $rng(1);\ u = rand();$ % it has value $u = 0.4170$
3. $sum(cdf <= u) + 1$

- $A : 0$
- $B : 1$
- $C : 2$
- $D : 3$

Q4

0 1 2

[0.3 0.7 1]

sum [1 0 0] + 1

true false false

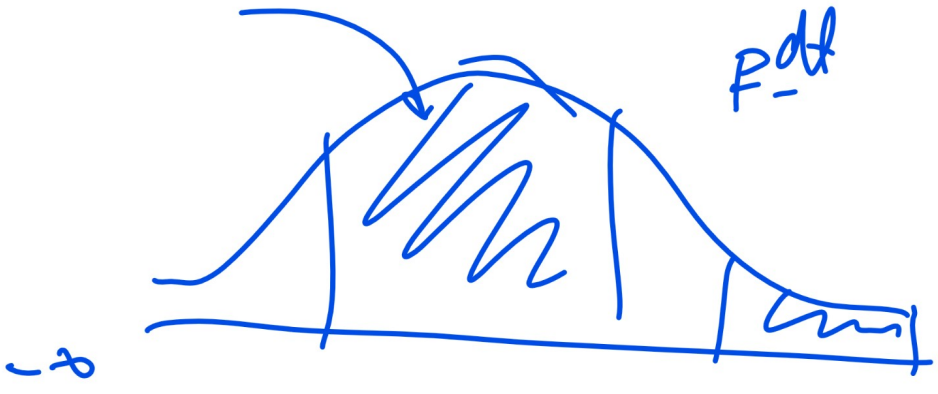
1

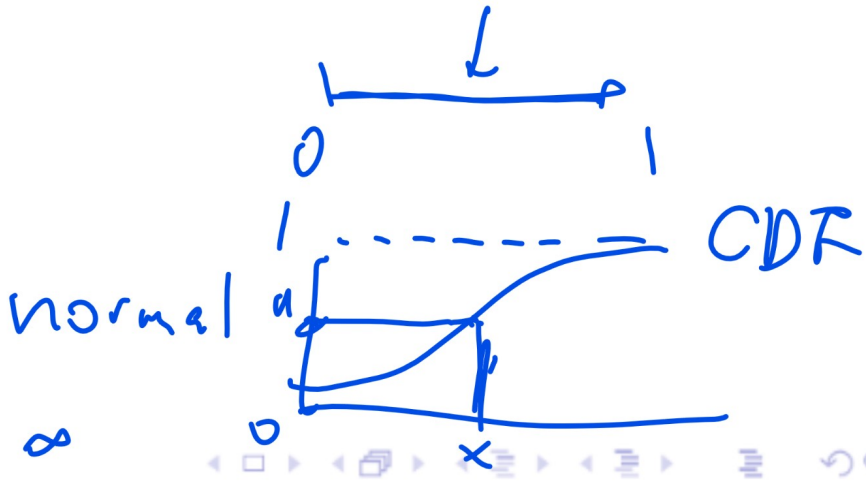
-(1

Random Variable, Continuous

Math

- A continuous random variable is a random variable that takes on uncountably infinite number of values.
- The (theoretical) probability that the random variable is equal to any number is 0.
- Inverse transform sampling can also be used to generate a random value of the variable.





Cumulative Distribution Functions, Continuous

Math

- The CDF of a distribution X taking values $(-\infty, \infty)$ is given by $F(x) = \mathbb{P}\{X \leqslant x\}$.
- The derivative of this function is called the probability density function $f(x) = F'(x)$, meaning $F(x) = \int_{-\infty}^{x} f(\hat{x})\, d\hat{x}$.
- More details in the next next lecture.

$F^{-1}(u)$

Simulation

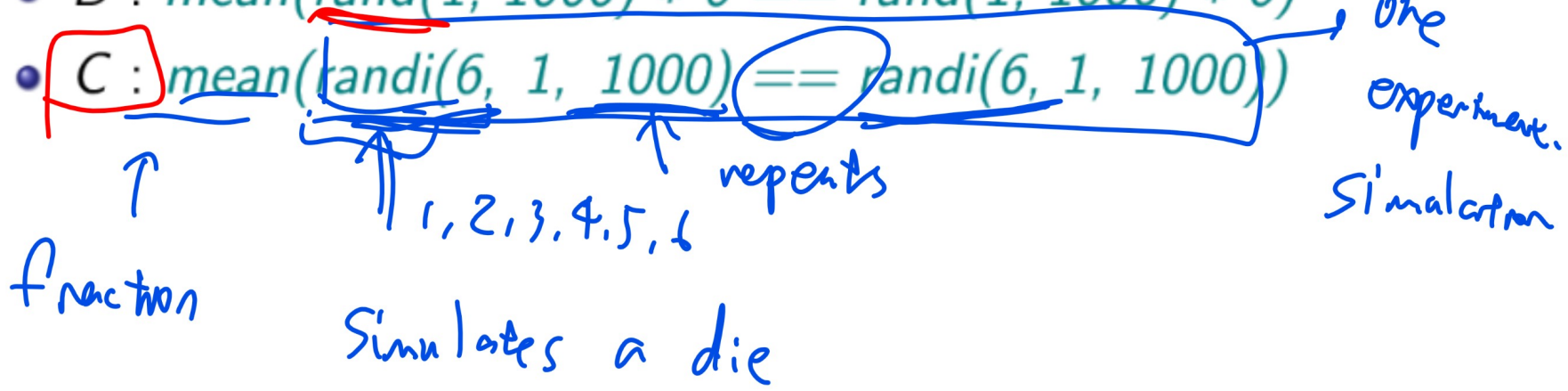
Math

- Direct computation of the probability of an event is sometimes difficult, and simulation can be used to approximate this probability by repeating the same random process a large number of times and find the fraction of times the event occurs.

Simulation, Equal

Quiz

- (Estimate the probability that the values from two dice are the same.)
- $\approx$ **0.1667**
- B : *mean(rand(1, 1000) * 6 == rand(1, 1000) * 6)*
- C : *mean(randi(6, 1, 1000) == randi(6, 1, 1000))*



Simulation, Sum

Quiz

Q6

- (Estimate the probability that the values from two dice sum up to an odd number.)
- $\approx$ **0.5**
- B : *mean(mod(randi(12, 1, 1000), 2) == 1)*
- C :

  mean(mod(randi(6, 1, 1000) + randi(6, 1, 1000), 2) == 1)

mod(X, 2) == 1

Q? → 1 is possible

1→12

1→6 1→6 ≥1 ≥1

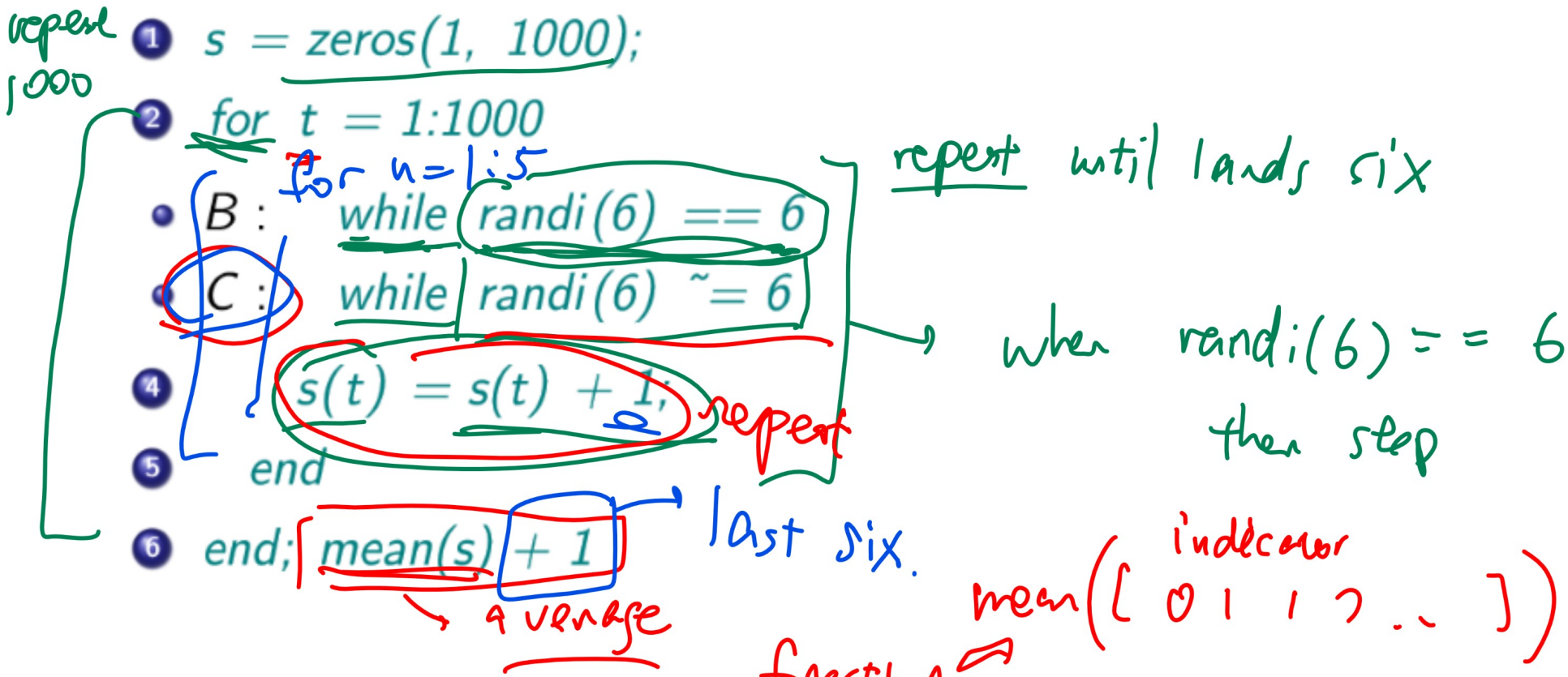
1 is not possible

Simulation, Geometric

Quiz

- (Estimate the average number of times it takes to throw a die until it lands six.)
- ≈ 6

```
1  s = zeros(1, 1000);
2  for t = 1:1000
B:   while randi(6) == 6
C:   while randi(6) ~= 6
4      s(t) = s(t) + 1;
5    end
6  end; mean(s) + 1
```



Simulation, Geometric Again

Quiz

- (Estimate the average number of times it takes to throw a loaded die with probabilities $[0.1 \quad 0.2 \quad 0.4 \quad 0.2 \quad 0 \quad 0.1]$ until it lands six.)
- $\approx$ **10**

```
1  s = zeros(1, 1000);
2  for t = 1:1000
B :   while rand() <= 0.9
C :   while rand() > 0.9
4       s(t) = s(t) + 1;
5     end
6  end; mean(s) + 1
```

Simulation, Geometric General

Quiz

[illegible]@cs.wisc.edu
wu48@wisc.edu

ID: wu48

1. $s = zeros(1, 1000);$
2. $for\ t = 1:1000$
   - The following conditions are more general (with $cdf = [0.1\ 0.2\ 0.4\ 0.2\ 0\ 0.1]$)
   - $while\ sum(cumsum(cdf) <= rand()) + 1 == 6$
   - $while\ find(rand() < cumsum(cdf), 1) == 6$
4. $s(t) = s(t) + 1;$
5. end
6. $end;\ mean(s) + 1$

PS

CDF inversion

Reproducibility

Code

- The same code can produce a different output every time it is excuted.
- In order to make a simulation reproducible, the best practice is to always set a seed at the beginning of the simulation using *rng(seed)*.

Blank Slide