

Announcements:

① lab session R Thursday
14 Feb
11-12:15

② class project 1 - due Monday
25 Feb -
Friday 15 Feb click the web -
 { 5 question 20 point each
 { 2 program codes -
 chap. 2 & 3
 →



Generating discrete random variables

Math 276 *Actuarial Models*

Spring 2008 semester

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

EA Valdez
University of Connecticut - Storrs
Lecture Week 3

The inverse transform method

Generating discrete random variables

EA Valdez

- Consider a discrete r.v. X with pmf

$$P(X = x_j) = p_j, \quad j = 0, 1, \dots, \sum_i p_j = 1.$$

- To generate a value from this distribution, first we generate a random number U and set

$$X = \begin{cases} x_0, & \text{if } U < p_0 \\ x_1, & \text{if } p_0 \leq U < p_0 + p_1 \\ \vdots & \\ x_j, & \text{if } \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^j p_i \\ \vdots & \end{cases}.$$

- This is called the **inverse transform method**.
- Proof to be outlined in class.



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

$$P(X=x_j) = P\left(\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^j p_i\right)$$

$$= P\left(U < \sum_{i=0}^j p_i\right) - P\left(U < \sum_{i=0}^{j-1} p_i\right)$$

$$= \sum_{i=0}^j p_i - \sum_{i=0}^{j-1} p_i$$

$$= p_j$$

Bernoulli $p=0.8$

$$P(X=1) = 0.8$$

$$P(X=0) = 0.2$$

Generate U

e.g. 0.02 0.92 0.98 $X=1$ if $U < 0.8$
 ↓ 0 0 $X=0$ if $U \geq 0.8$



① The method can be written algorithmically as

generate a random number U

if $U < p_0$, set $X = x_0$ and STOP

if $U < p_0 + p_1$, set $X = x_1$ and STOP

if $U < p_0 + p_1 + p_2$, set $X = x_2$ and STOP

⋮

② If the X_i 's are ordered like $x_0 < x_1 < x_2 < \dots$ so that the

cdf $F(x_k) = \sum_{i=0}^k p_i$ and that



X equals x_j if $F(x_{j-1}) \leq U < F(x_j)$.

Therefore, after generating U , we determine the value of X by looking for the interval $[F(x_{j-1}), F(x_j))$ in which it lies (or equivalently finding the inverse of U).

The inverse transform
method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results
make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The
acceptance-rejection
technique

Illustrative example

The composition
approach

Example 4g

Mixture distributions

R codes

The inverse transform
method

Some remarks

Illustrative example

Coding the algorithm in R
Checking if the results
make sense

Special cases

Discrete uniform
Geometric
Poisson
BinomialThe
acceptance-rejection
technique

Illustrative example

The composition
approachExample 4g
Mixture distributions

R codes

Illustrative example

Suppose we want to simulate from the discrete distribution with

$$\begin{array}{cccc} P(X=1) & P(X=2) & P(X=3) & P(X=4) \\ p_1 = 0.20, & p_2 = 0.25, & p_3 = 0.40, & p_4 = 0.15. \end{array}$$

We do the following:

generate a random number U if $U < 0.20$, set $X = 1$ and STOPif $U < 0.45$, set $X = 2$ and STOPif $U < 0.85$, set $X = 3$ and STOPotherwise set $X = 4$.

It is suggested the following could be more efficient:

generate a random number U if $U < 0.40$, set $X = 3$ and STOPif $U < 0.65$, set $X = 2$ and STOP

0.85

if $U < 0.80$, set $X = 1$ and STOPotherwise set $X = 4$.

Coding the algorithm in R

Generating discrete random variables

EA Valdez

The function `simdiscrete.R` in R provides an algorithm for generating from a given discrete distribution.

```
# function to simulate from a given discrete distribution
# required inputs: x, probabilities, number to generate

simdiscrete <- function(x, probs, n.gen){
  xprobs <- data.frame(x=x,probs=probs)

  # sort the data in order of x, with cumulative probabilities
  xprobs.sorted <- xprobs[do.call(order, xprobs[["x"]]), ]
  cum.probs <- cumsum(xprobs.sorted[,2])
  xprobs.sorted <- cbind(xprobs.sorted,cum.probs)

  # generate and loop
  urandom <- runif(n.gen)
  sim.vector <- rep(0,n.gen)
  for(i in 1:n.gen) sim.vector[i] <- min(xprobs.sorted[,1][which(cum.probs >= urandom[i])])
  # output
  sim.vector
}
```

n.gen
X
probs /

Then run the following commands:

```
> source("C:\\...\\Math276-Spring2008\\Rcodes-2008\\Week2\\simdiscrete.R")
> x <- c(9,4,0,2,8)
> probs <- c(0.50,0.10,0.05,0.15,0.20)
> sum(probs)
[1] 1
> out1 <- simdiscrete(x,probs,100)
> out1
[1] 4 8 9 9 8 9 9 2 9 8 2 9 9 2 9 8 9 2 2 2 4 9 4 9 9 9 9 9 8 8 9 8 0 9 9 9
[38] 8 2 8 2 9 9 9 2 9 9 8 0 8 9 9 9 9 8 9 9 9 9 9 9 9 4 2 9 9 2 9 9 8
[75] 4 8 9 9 8 8 9 9 9 4 8 9 9 8 0 8 4 8 9 8 9 8 9 9 8
```



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

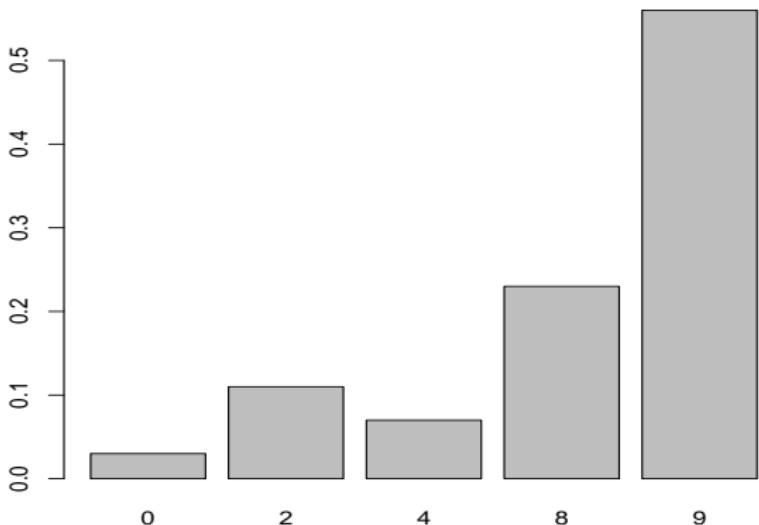


Checking if the results make sense

We can draw a barplot to check if the results make sense.

```
# this tabulates the results
> xtabs(~out1)/100
out1
 0   2   4   8   9
0.03 0.11 0.07 0.23 0.56
# this draws the plot
> barplot(xtabs(~out1)/100)
```

100



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Generating a discrete uniform random variable

Generating discrete random variables

EA Valdez



- In the discrete uniform distribution, we have equal probabilities:

$$P(X = j) = 1/n, \text{ for } j = 1, 2, \dots, n.$$

- To simulate from this distribution, we generate a random number U and then set

$$X = j \text{ if } \frac{j-1}{n} \leq U < \frac{j}{n}.$$

- This condition is equivalent to if $j-1 \leq nU < j$, that is

$$X = \text{Int}(nU) + 1,$$

$$\text{Int}(4.6) = 4$$

where $\text{Int}(x)$ is the greatest integer part of x .

integer part

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Generating a geometric random variable

Generating discrete random variables

EA Valdez



- In a geometric distribution with parameter p , we have

$$P(x = i) = pq^{i-1}, \text{ for } i \geq 1$$

- Note that the cumulative probability

$$P(X \leq j-1) = \sum_{i=1}^{j-1} P(X = i) = 1 - q^{j-1}.$$

- It can be shown that with a random number U , then

$$X = \text{Int}\left(\frac{\log(U)}{\log(q)}\right) + 1$$

is indeed geometric with parameter p .

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Generate $U \in \text{set } X \leq j$ if

$$\underbrace{1 - q^{j-1}}_{P(X \leq j-1)} \leq U < \underbrace{1 - q^j}_{P(X \leq j)}$$

$$U \sim \text{Uniform}(0,1)$$
$$1-U \sim \text{Uniform}(0,1)$$

$$q^j < 1-U \leq q^{j-1}$$

$$X = \min\{j \mid q^j < 1-U\}$$

$$= \min\{j \mid j \log q < \log(1-U)\}$$

$$= \min\{j \mid j \geq \frac{\log(1-U)}{\log q}\}$$

$$= \text{Int}\left(\frac{\log(1-U)}{\log q}\right) + 1 + \boxed{\text{Int}\left(\frac{\log(U)}{\log q}\right) + 1}$$

Generating a Poisson random variable

$X = 0, 1, 2, \dots, \infty$

number of claims \sim Poisson

insurance

- For the case of the Poisson, we exploit the recursion property

$$P(X=i+1) = \frac{\lambda}{i+1} p_i, \quad \text{for } i \geq 0.$$

- The following steps can then be followed to generate from a Poisson with parameter λ :

Step 1: generate a random number U .

Step 2: set $i = 0$, $p = e^{-\lambda}$ and $F = p$.

Step 3: if $U < F$, set $X = i$ and STOP.

Step 4: set $p = \lambda p / (i + 1)$, $F = F + p$, and $i = i + 1$.

Step 5: return to Step 3.

$$P_0 = e^{-\lambda}$$

- Note that F is indeed the cdf $F(i) = P(X \leq i)$.
- It can be shown that the average number of searches grows with the square root of λ . (proof to be discussed!)

$$P(X=0) = e^{-\lambda}$$

$$\lambda = 1$$



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

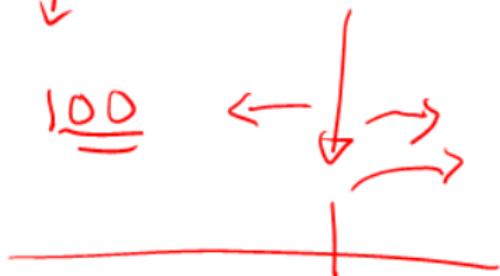
Generate

$$U = .55$$

Int(x)



$$\underline{1.00}$$



$$1.00 = .45$$

$$\sqrt{x}$$

R routine for generating a Poisson

Generating discrete random variables

EA Valdez



- Note that the code `rpois` is available in R. Type `help(Poisson)`.
- You can write your own routine following the previous steps, such as `simPoisson.R`:

```
# function to generate from a Poisson with parameter lambda

simPoisson <- function(n.gen,lambda){
  urandom <- runif(n.gen)
  sim.vector <- rep(0,n.gen)
  for(j in 1:n.gen){
    i <- 0
    p <- exp(-lambda)
    F <- p
    while(urandom[j] >= F) {
      p <- lambda*p/(i+1)
      F <- F+p
    }
    i<-i+1
  }
  sim.vector[j] <- i
}
# output
sim.vector
```

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

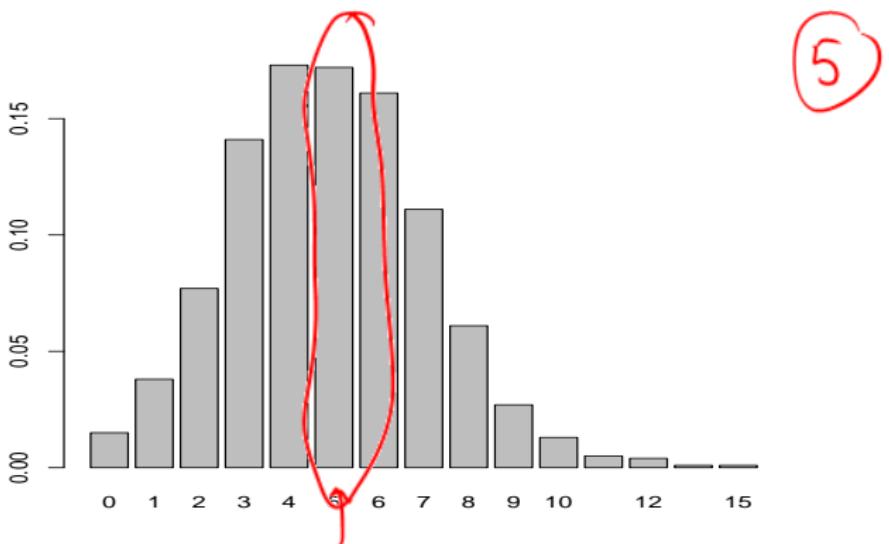
Example 4g

Mixture distributions

R codes

Executing and summarizing the results

```
> out1 <- simPoisson(1000,5)
> out1
[1] 4 6 3 6 5 5 4 2 5 5 4 6 3 3 6 9 7 7 7 2 9 7 2 9 3 7 3 6 6 8
[30] 4 2 6 8 1 5 6 6 5 4 4 3 7 3 4 3 3 3 4 3 4 7 1 9 0 4 3 5 2
[59] 5 5 8 7 5 6 5 7 6 8 5 4 6 3 2 6 4 3 3 4 4 6 4 9 4 3 7 3 4
[88] 6 6 4 10 4 0 7 6 8 8 6 5 5 7 8 7 5 8 5 5 5 3 5 3 5 0 7 4 8
.....
[987] 6 3 7 5 7 6 5 4 7 3 6 7 4 4
> mean(out1)
[1] 4.902
> xtabs(~out1)/1000
out1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 15
0.015 0.038 0.077 0.141 0.173 0.172 0.161 0.111 0.061 0.027 0.013 0.005 0.004 0.001 0.001
> barplot(xtabs(~out1)/1000)
```



Generating discrete random variables

EA Valdez



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Generating Binomial random variables

0, 1, 2, ..., n
↑
first

- Just as in the Poisson case, we exploit the recursion property for the Binomial distribution:

chap 2

$$P(X = i + 1) = \frac{n - i}{i + 1} \frac{p}{1 - p} P(X = i).$$

n=100

- The following steps can then be followed to generate a Binomial random variable with parameters n and probability of success p :

Step 1: generate a random number U .

Step 2: set $c = p/(1 - p)$, $i = 0$, $\text{pr} = (1 - p)^n$, and $F = \text{pr}$.

Step 3: if $U < F$, set $X = i$ and STOP.

Step 4: reset $\text{pr} = [c(n - i)/(i + 1)]\text{pr}$, $F = F + \text{pr}$, and $i = i + 1$.

Step 5: return to Step 3.

✓ *SimBinom(n.gen, n, p)*

- As an exercise, try to write an R routine for generating a Binomial random variable following the above steps.
- Another approach to simulate from a $\text{Binomial}(n, p)$ is to use the interpretation that it is equal to the number of success in n independent Bernoulli trials.

Generating discrete random variables

EA Valdez



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Bernoulli (p) $\rightarrow \begin{cases} X=1 & \text{w.p. } p \\ X=0 & \text{w.p. } 1-p \end{cases}$

$$X_1 + X_2 + \dots + X_n = X$$

$$p = 0.5$$

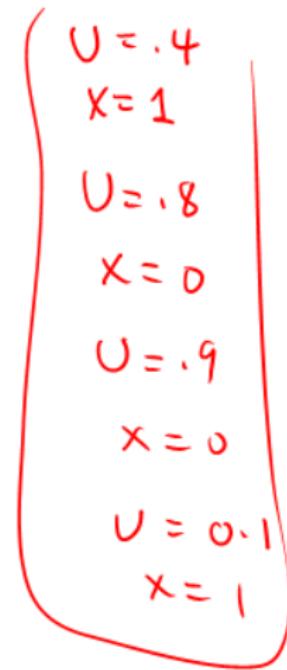
all Bernoulli

$\sim \text{Binomial}(n, p)$

$$n=100 \quad X_1 \quad X_2 \quad \dots \quad X_{100}$$

add them all up-

Practice: try to write a code using this algorithm



The acceptance-rejection technique

rejection method

- Suppose we wish to simulate from a discrete distribution with mass function $\{p_j, j \geq 0\}$. $\rightarrow X \sim p_j$
- Suppose we have an efficient method to simulate from $\{q_j, j \geq 0\}$ where

$$\frac{p_j}{q_j} \leq c, \text{ for all } j \text{ such that } p_j > 0,$$

$$p_j \leq cq_j$$

Y

where c is a fixed positive constant.

- The acceptance-rejection method is as follows:

Step 1: simulate Y with mass function $\{q_j\}$.

Step 2: generate a random number U .

Step 3: if $U < p_Y/cq_Y$, set $X = Y$ and STOP.

Step 4: else return to Step 1.

- We must prove that the random variable generated comes from the distribution $\{p_j\}$. We will prove in class.



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

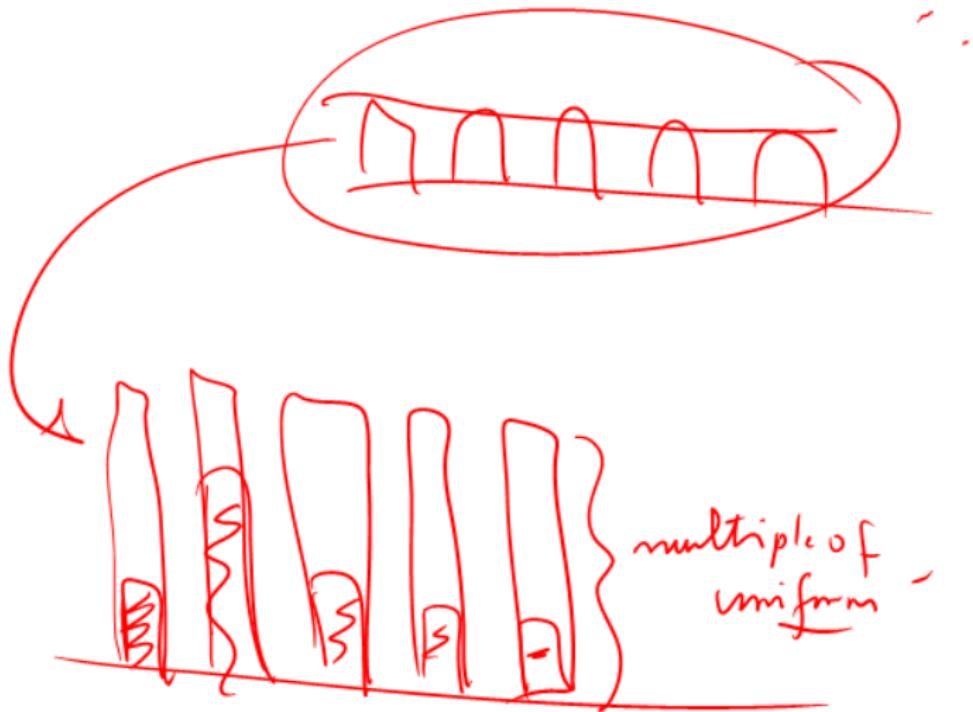
Illustrative example

The composition approach

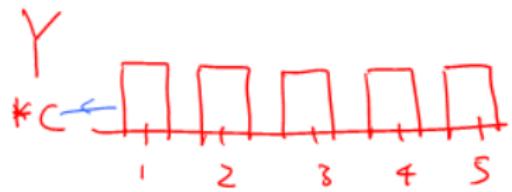
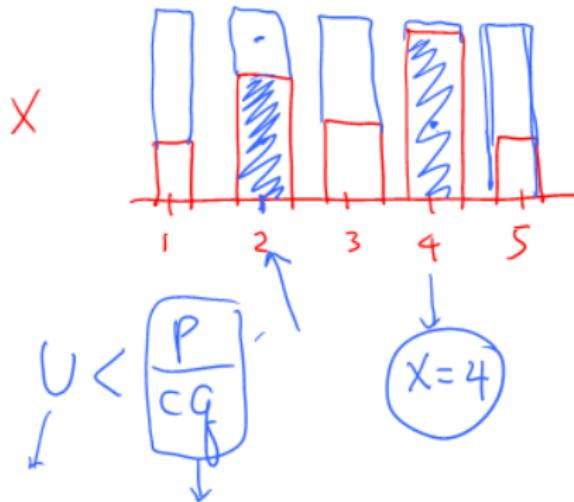
Example 4g

Mixture distributions

R codes



PROOF -
Some examples



$$\frac{P}{g} \leq c \Rightarrow \frac{P}{cg} \leq 1$$

Proof: $P(X=j) = P_j \leftarrow$ prove this $P(U < \frac{P_j}{c q_j})$

Consider

$$P(Y=j, \text{ accepted}) = P(Y=j) \underbrace{P(\text{accepted} | Y=j)}_{\frac{P_j}{c q_j}} = \frac{P_j}{c} \checkmark$$

$$\sum_j P(Y=j, \text{ accepted}) = \sum_j \frac{P_j}{c} = \frac{1}{c} = P(\text{accepted})$$
$$1 - \frac{1}{c} = P(\text{rejected})$$

$$P(X=j) = \sum_n P(j \text{ is accepted on } n^{\text{th}} \text{ iteration})$$

$$= \sum_n \left(1 - \frac{1}{c}\right)^{n-1} \frac{P_j}{c} = P_j \underbrace{\sum_n \left(1 - \frac{1}{c}\right)^{n-1} \frac{1}{c}}_{\text{Geometric } (p=\frac{1}{c})} = P_j$$



Illustrative example

$$q_j \cdot 2 \quad .2 \quad .2 \quad .2 \quad .2 \quad .2 \quad Y$$

- Suppose we want to simulate from the discrete distribution

j	1	2	3	4	5	X
p_j	0.05	0.25	0.45	0.15	0.10	

- We can use the acceptance-rejection method by choosing the discrete uniform for q and then the constant

$$c = \max \frac{p_j}{q_j} = 2.25 \leftarrow$$

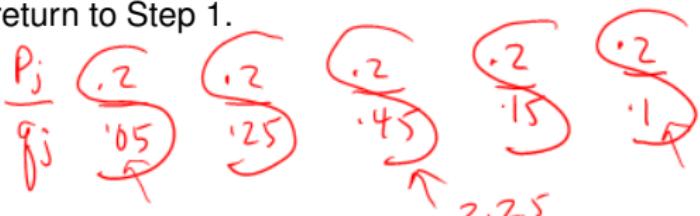
- The algorithm can then be summarized as follows:

Step 1: generate a random number U_1 and set $Y = \text{Int}(5U_1) + 1$.

Step 2: generate a second random number U_2 .

Step 3: if $U_2 < p_j/0.45$, set $X = Y$ and STOP.

Step 4: else return to Step 1.



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

$$U < \frac{P_Y}{Cg_Y} = \frac{P_Y}{2.25(-2)}$$

$$= \frac{P_Y}{-45}$$

$X \sim 1, \dots, n$

$$X = \text{Int}(nU) + 1$$

↓
random number
↳ total number

R routine for the acceptance-rejection illustration

Generating discrete random variables

EA Valdez



Name of R routine `simaccept.R`:

```
# function to illustrate the acceptance-rejection technique
✓ simaccept <- function(n.gen){
  # enter the probabilities
  probs <- c(.05,.25,.45,.15,.10)
  ✓ sim.vector <- rep(0,n.gen)
  for(j in 1:n.gen){
    u1 <- runif(1)
    y <- floor(5*u1) + 1
    u2 <- runif(1)
    while(u2 > probs[y]/0.45)
      u1 <- runif(1)
      y <- floor(5*u1) + 1
      u2 <- runif(1)
    }
    sim.vector[j] <- y
  }
  # output
  sim.vector
}
```

$$U_2 < P_Y / 0.45 \rightarrow \text{STOP}$$

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Executing and summarizing the results

```
> source("C:\\\\Math276-Spring2008\\\\Rcodes-2008\\\\Week3\\\\simaccept.R")
```

```
✓ out1 <- simaccept(1000)
```

```
> out1
```

```
[1] 4 4 3 3 3 2 2 4 2 4 5 1 2 4 3 3 3 3 2 2 4 4 2 5 1 2 2 2 3 4 3 3 5 3 1 5 2  
[38] 3 4 5 3 3 2 3 2 3 3 5 3 2 3 2 2 2 4 3 3 2 4 5 3 3 2 3 3 2 2 3 4 4 3 3 3  
.....  
[963] 1 3 3 3 2 4 3 2 5 2 3 3 3 1 3 2 2 4 3 2 2 3 2 3 3 2 3 3 2 4 2 3 4 1 3
```

```
[1000] 4
```

```
> mean(out1)
```

```
[1] 3.003
```

```
> xtabs(~out1)/1000
```

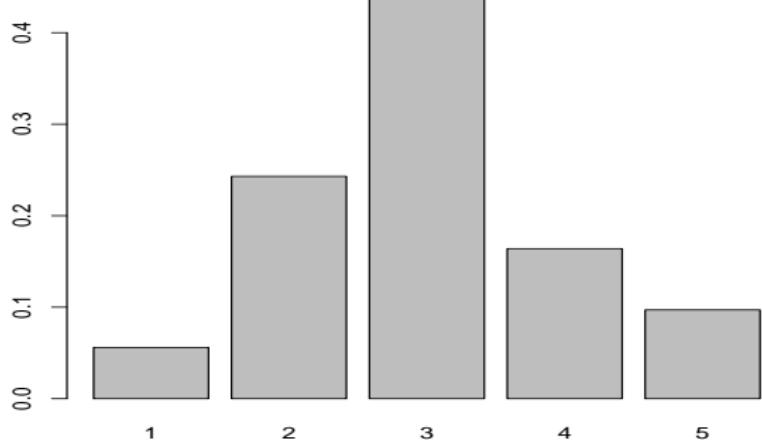
```
out1
```

1	2	3	4	5
0.056	0.243	0.440	0.164	0.097

```
> barplot(xtabs(~out1)/1000)
```

{ } 1 2 3 4 5

.05 .25 .45 .15 .10



Generating discrete random variables

EA Valdez



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

$P_j .05 .10 .70 .15$ $j \downarrow \begin{matrix} 1 \\ - \uparrow \\ 3 \\ - \uparrow \\ 5 \\ - \uparrow \\ 10 \end{matrix}$ $1 \ 2 \ 3 \ 4 \ . \ - \ 10$ $U < \underbrace{\textcircled{p}}_{cq} = 0$

will never be accepted

The composition approach

Generating discrete random variables

EA Valdez



- Consider now simulating from a distribution with mass function

$$P(X = j) = \alpha p_j^{(1)} + (1 - \alpha) p_j^{(2)}, \quad j \geq 0, \quad 0 < \alpha < 1.$$

- If X_1 and X_2 are the random variables with respective mass functions $p_j^{(1)}$ and $p_j^{(2)}$, then

$$\rightarrow X = \begin{cases} X_1, & \text{w.p. } \alpha \\ X_2, & \text{w.p. } 1 - \alpha \end{cases}$$

- One approach then to generate from this mixture distribution is:

- Step 1: generate a random number U_1 .
Step 2: generate from X_1 and X_2 distributions.
Step 3: if $U < \alpha$, set $X = X_1$.
Step 4: else if $U > \alpha$, set $X = X_2$.

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Example 4g

Can use rejection method ✓

Generating discrete random variables

EA Valdez



- Consider the example of generating X from a distribution with mass function

$$\checkmark p_j = P(X = j) = \begin{cases} \underline{0.05}, & \text{for } j = 1, 2, 3, 4, 5 \\ \underline{0.15}, & \text{for } j = 6, 7, 8, 9, 10 \end{cases}$$

- Note that this is equivalent to

$$\checkmark p_j = \underline{0.5} \underline{p_j^{(1)}} + \underline{0.5} \underline{p_j^{(2)}},$$

$\xrightarrow{x_1} \quad \xrightarrow{x_2}$

where

$$p_j^{(1)} = 0.10, \text{ for } j = 1, 2, \dots, 10 \quad \checkmark \text{Uniform}$$

and

$$p_j^{(2)} = \begin{cases} 0, & \text{for } j = 1, 2, 3, 4, 5 \\ 0.2, & \text{for } j = 6, 7, 8, 9, 10 \end{cases} \quad \checkmark \text{Uniform}$$

- Thus, first generate a random number U , and then generate from the discrete uniform over $1, \dots, 10$ if $U < 0.5$ and from the discrete uniform over $6, 7, 8, 9, 10$ otherwise.

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Step 1: Generate U_1 ; Generate X_1 , Generate X_2

Step 2: If $U_1 \leq 0.5$, Set $X = X_1$

Step 2: If $U_1 > 0.5$, Set $X = X_2$

$$\underline{P(U=.5) \approx 0}$$

X uniform 1, ..., 10

$$X = \text{Int}(10U + 1)$$

X uniform 16, ..., 20

16 17 18 19 20
·2 ·2 ·2 ·2 ·2

$$X = \text{Int}(5U) + \underline{16}$$

R routine to illustrate the composition approach

Generating discrete
random variables

EA Valdez



Name of R routine simcomp.R:

```
# function to illustrate the composition approach

simcomp <- function(n.gen){
  urandom1 <- runif(n.gen)
  urandom2 <- runif(n.gen)
  sim.vector <- rep(0,n.gen)
  for(j in 1:n.gen){
    if (urandom1[j] < 0.5) {
      sim.vector[j] <- floor(10*urandom2[j])+1
    }
    else {
      sim.vector[j] <- floor(5*urandom2[j])+6
    }
  }
  # output
  sim.vector
}
```

The inverse transform
method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results
make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The
acceptance-rejection
technique

Illustrative example

The composition
approach

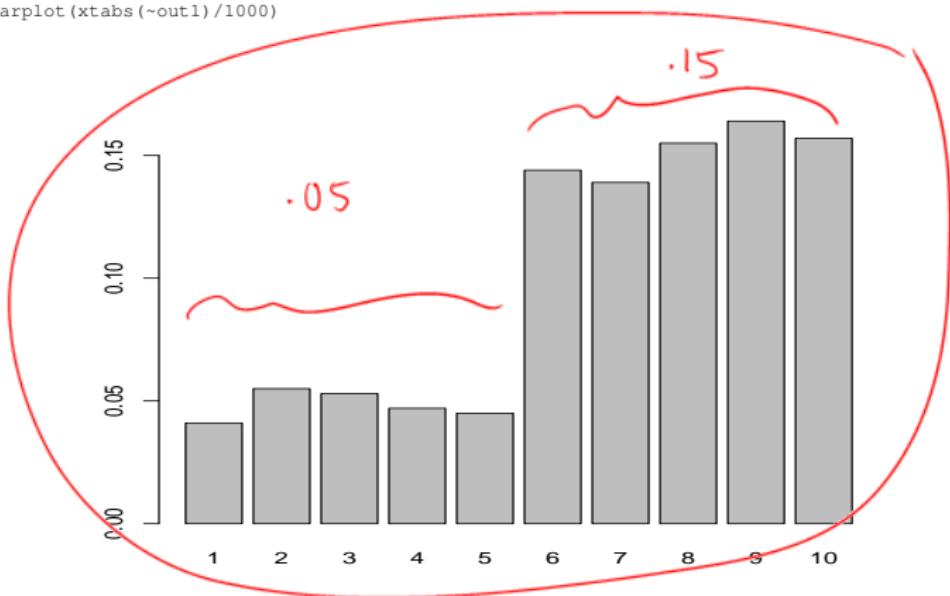
Example 4g

Mixture distributions

R codes

Executing and summarizing the results

```
> source("C:\\\\Math276-Spring2008\\\\Rcodes-2008\\\\Week3\\\\simcomp.R")
> out1 <- simcomp(1000)
> out1
 [1] 7 5 8 9 9 7 1 8 9 6 9 8 2 8 4 9 4 8 4 8 7 6 9 8
[25] 9 5 3 6 7 2 7 2 6 6 6 8 5 4 6 10 2 3 9 4 3 9 9 4
[49] 9 3 10 9 7 6 2 3 4 3 3 10 7 3 8 6 8 10 5 8 10 8 9 10
[985] 9 9 1 10 1 7 1 8 6 7 6 10 9 10 3 6
> mean(out1)
[1] 6.846
> xtabs(~out1)/1000
out1
 1   2   3   4   5   6   7   8   9   10
0.041 0.055 0.053 0.047 0.045 0.144 0.139 0.155 0.164 0.157
> barplot(xtabs(~out1)/1000)
```



Generating discrete random variables

EA Valdez



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Mixture distributions

$$\rightarrow \sum \alpha_1, \alpha_2, \boxed{\alpha_j}, \dots, \alpha_n = 1$$

x_j

- In the case where the distribution function of X is given by

$$F(x) = \sum_{i=1}^n \alpha_i F_i(x),$$

where $F_i, i = 1, \dots, n$ are distribution functions, we have what we call a **mixture** distribution.

- To simulate from such a mixture distribution,

Step 1: simulate a random variable I , equal to i with probability α_i , for $i = 1, \dots, n$.

Step 2: simulate from the distribution F_i .

- This is also called the **composition method**.



The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes

Generate U x_1, \dots, x_n

$$U < \alpha_1 \rightarrow X = x_1$$

$$U < \alpha_1 + \alpha_2 \rightarrow X = x_2$$

⋮

$$U < \alpha_1 + \alpha_2 + \dots + \alpha_j \rightarrow X = x_j$$

⋮

⋮

R codes for simulating from known discrete distributions

Generating discrete random variables

EA Valdez



- In R, there are many functions that generate discrete random variables. Most of them start with `r`.
- Here are a few of them:

`rbinom` - binomial

`rnbnom` - negative binomial

`rpois` - Poisson

`rgeom` - geometric

`rhyper` - hypergeometric

next week
we'll show how
this is done!

The inverse transform method

Some remarks

Illustrative example

Coding the algorithm in R

Checking if the results make sense

Special cases

Discrete uniform

Geometric

Poisson

Binomial

The acceptance-rejection technique

Illustrative example

The composition approach

Example 4g

Mixture distributions

R codes