Random Variable Generation

Mu Niu

2024-05-12

Random Variable Selection:

- Inverse CDF: Beta(5,1) $= \frac{\Gamma(5+1)}{\Gamma(5)\Gamma(1)} x^{5-1}(1-x)^{1-1}, x \in (0,1)$
- Transformation of Random Variables: $t_2 = \frac{\Gamma(\frac{2+1}{2})}{\sqrt{2\pi}\Gamma(\frac{2}{2})}(1+\frac{x^2}{2})^{-\frac{2+1}{2}}, x \in (-\infty, \infty)$
- Accept-Rejection: Gamma(2,2) $= \frac{1}{\Gamma(2)}(2x^2)e^{-2x}x^{-1}, x \in (0,\infty)$

RVG Methods Application and Code Development:

Inverse CDF Method:

1. Derive the inverse function $F_X^{-1}(u)$

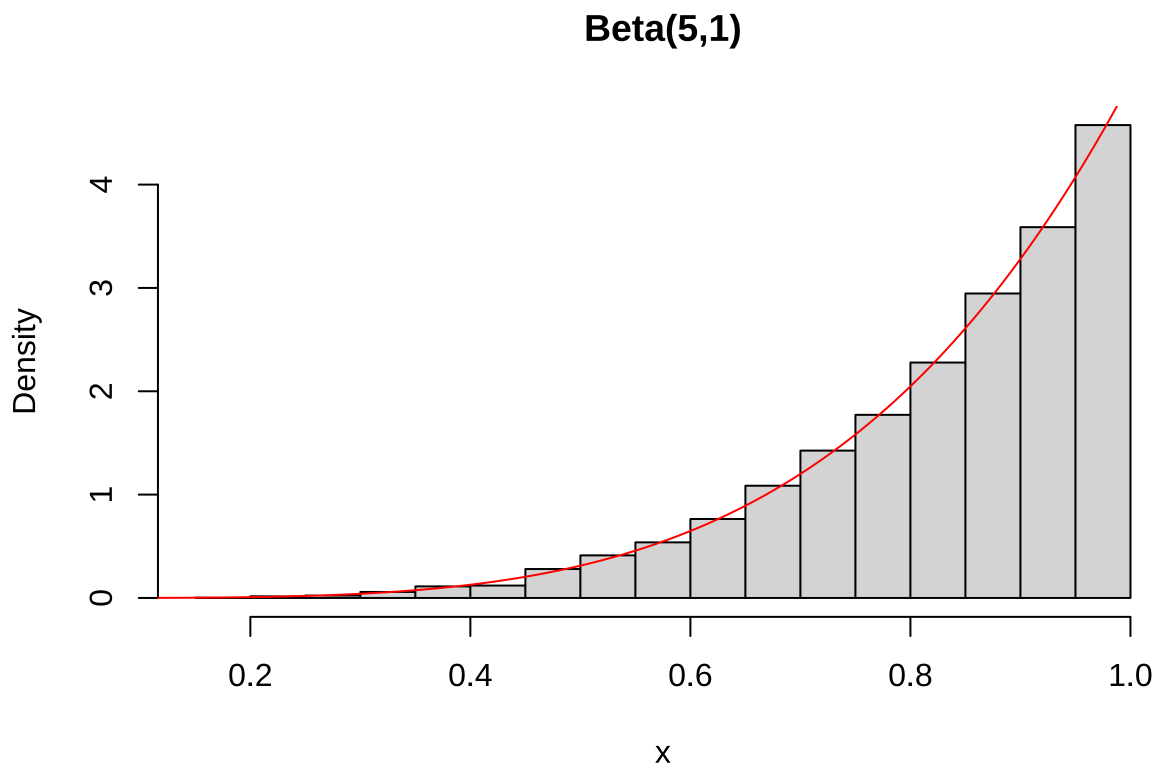
$$
\begin{aligned}
Beta(5,1): f(x,5,1) &= \frac{\Gamma(5+1)}{\Gamma(5)\Gamma(1)} x^{5-1}(1-x)^{1-1} \\
&= 5x^4 \\
CDF: F(x,5,1) &= \int_0^x 5x^4 dx \\
&= x^5 \\
\text{Inverse CDF}: F_X^{-1}(u) &= u^{\frac{1}{5}}
\end{aligned}
$$

2. Write function to compute $F_X^{-1}(u)$

```r
# sample from CDF
u_inverse <- runif(10000)
# deliver to inverse CDF
x_inverse <- u_inverse^(1/5)
```

3. Histogram and density plot

```r
# density histogram of samples from inverse CDF
hist(x_inverse, prob = TRUE, xlab = 'x', main = 'Beta(5,1)')
# plot real beta(5,1)
y_inverse <- seq(0, 1, by = 0.0001)
lines(y_inverse, dbeta(y_inverse, 5, 1), col = 'red')
```



The histogram and density plot above suggests that the empirical and theoretical distributions approximately agree.

Transformation of Random Variables Method:

According to textbook, we know that if $Z \sim N(0,1)$ and $V \sim \chi^2(n)$ are independent, then $T = \frac{Z}{\sqrt{V/n}}$ has the Student t distribution with n degrees of freedom.

Hence, to sample from Student t distribution with 2 degree of freedom:

$$\begin{aligned} t_2 &= \frac{\Gamma(\frac{2+1}{2})}{\sqrt{2\pi}\Gamma(\frac{2}{2})}(1+\frac{x^2}{2})^{-\frac{2+1}{2}} \\ &= \frac{\frac{1}{2}\sqrt{\pi}}{\sqrt{2\pi}}(1+\frac{x^2}{2})^{-\frac{3}{2}} \\ &= \frac{1}{2\sqrt{2}(1+\frac{x^2}{2})^{\frac{3}{2}}} \end{aligned}$$

1. Generate random samples z from $N(0,1)$

```
z_transform = rnorm(10000, mean = 0, sd = 1)
```

2. Generate random samples v from $\chi^2(2)$

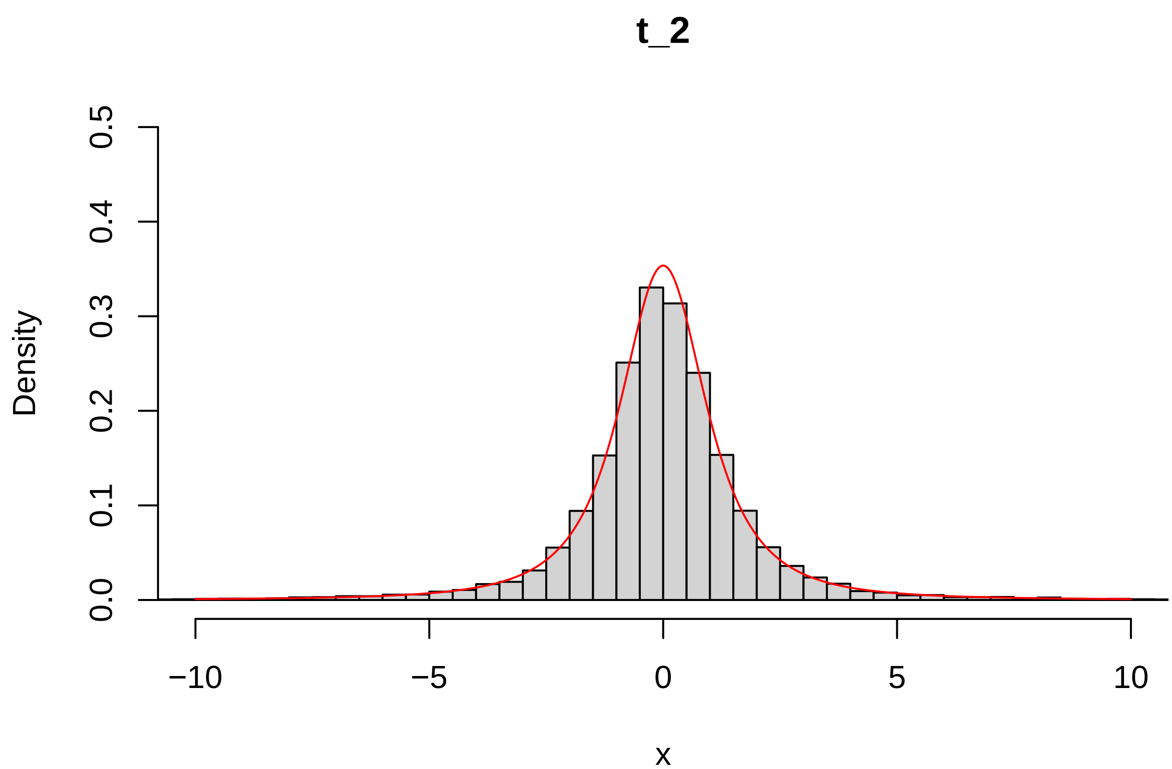
```
v_transform = rchisq(10000, df = 2)
```

3. Deliver $T = \frac{Z}{\sqrt{V/2}}$

```
t_transform = z_transform / sqrt(v_transform/2)
```

4. Histogram and density plot

```
# density histogram of samples from transformation method
hist(t_transform, prob = TRUE, xlab = 'x', main = 't_2', breaks = 500,
     xlim = c(-10,10), ylim = c(0,0.5))
# plot real t_1
y_transform <- seq(-10, 10, by = 0.002)
lines(y_transform, dt(y_transform, df = 2),col = 'red')
```



The histogram and density plot above suggests that the empirical and theoretical distributions approximately agree.

Accept-Rejection Method

To sample from Gamma(2,2) distribution:

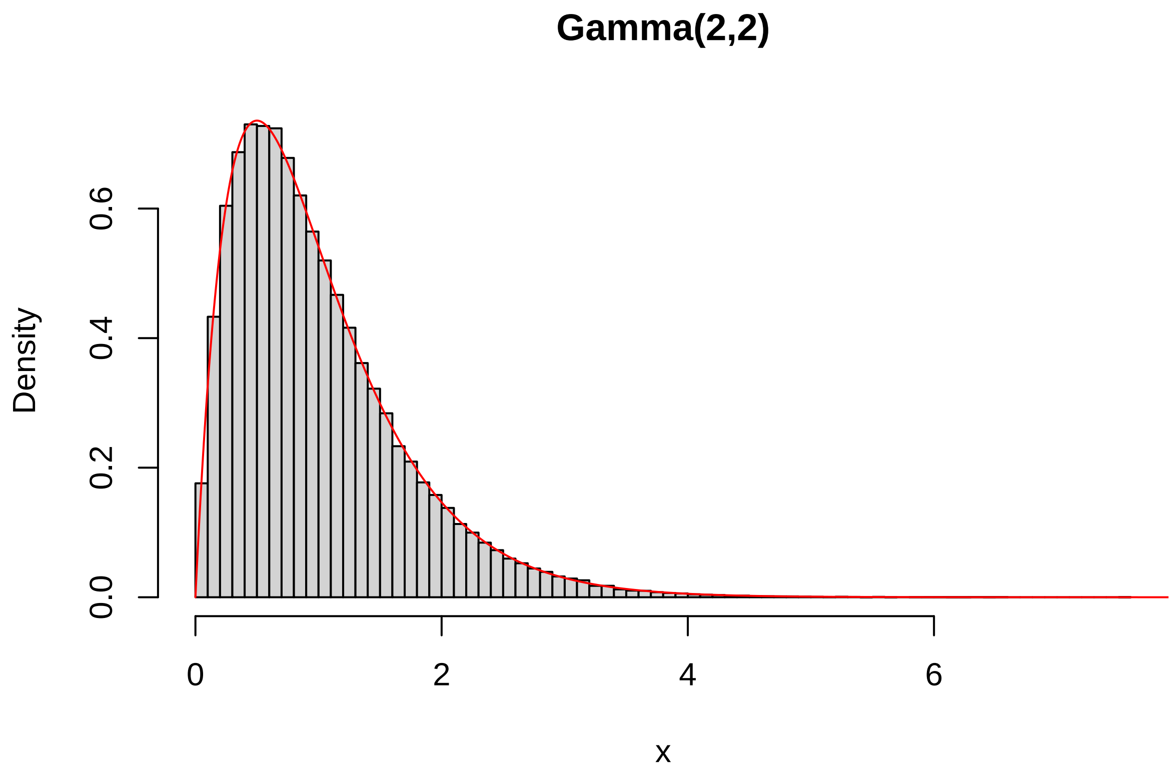
$$
\begin{aligned}
&\text{Target Distribution } f(x): \\
&\qquad \text{Gamma}(2,2) = \frac{1}{\Gamma(2)} 2x^2 e^{-2x} x^{-1} \\
&\qquad\qquad = 2x \cdot e^{-2x} \\
&\text{Proposal Distribution } g(x): \\
&\qquad \text{Gamma}(2,1) = \frac{1}{\Gamma(2)} x^2 e^{-x} x^{-1} \\
&\qquad\qquad = x \cdot e^{-x} \\
&\implies \frac{f(x)}{g(x)} = \frac{2x \cdot e^{-2x}}{x \cdot e^{-x}} \\
&\qquad = 2e^{-x} \leq 2 \text{ for all } x \in (0, \infty) \\
&\implies c = 2 \\
&\text{So a random } x \text{ from } g(x) \text{ is accepted if: } \frac{f(x)}{c \cdot g(x)} = \frac{2x \cdot e^{-2x}}{2x \cdot e^{-x}} \\
&\qquad = e^{-x} > u
\end{aligned}
$$

Code Implementation:

```r
sample <- 0
y_ar <- numeric(100000)

while (sample < 100000) {
  # generate a random u from the Uniform(0, 1) distribution
  u <- runif(1)
  # generate a random x from the proposal distribution Gamma(2,1)
  x <- rgamma(1, shape = 2, rate = 1)
  if (exp(-x) > u) {
    # we accept x
    sample <- sample + 1
    y_ar[sample] <- x
  } }

# density histogram of samples from accept_rejection method
hist(y_ar, prob = TRUE, xlab = 'x', main = 'Gamma(2,2)', breaks = 100)
# plot real Gamma(2,2)
y <- seq(0, 10, by = 0.001)
lines(y, dgamma(y, shape = 2, rate = 2), col = 'red')
```



The histogram and density plot above suggests that the empirical and theoretical distributions approximately agree.

Write Up:

Random Variable Selection and Reasons for Selection

For this project, I selected three random variables for simulation: the Beta(5,1) distribution, the Student's t-distribution with 2 degrees of freedom, and the Gamma(2,2) distribution. The choice of the Beta(5,1) distribution was strategic because setting parameter $b = 1$ simplifies the term $(1 - x)^{b-1}$, which makes the calculation for CDF and inverse CDF easier. The Student's t-distribution with 2 degrees of freedom was selected from a textbook. I am interested in generating samples from this distribution because of its importance in statistics. It has a similar shape to Normal Distribution but has heavier tails. It is used for estimating population parameters for small sample sizes or unknown variances. Finally, I chose the Gamma(2,2) distribution to explore the accept-rejection method, pairing it with the Gamma(2,1) as the proposal distribution. This combination was selected because they have similar shape and it simplifies the calculation of the scaling constant c. Also, Gamma distribution is one of the most important topics in the Bayesian Data Analysis course I took last quarter, so I think it would be meaningful to practice sampling from Gamma distribution.

Description of Each RNG Method Used

The **Inverse CDF Method** is a widely used technique in statistical sampling, especially useful for generating random samples from a specified distribution when its inverse CDF can be explicitly defined. This method involves two primary steps. First, generate a random number u from a uniform distribution over the interval [0, 1]. Second, plug u into the inverse CDF, $F^{-1}(u)$, where F^{-1} is the function that reverses the effects of the CDF of the desired distribution. The result, $x = F^{-1}(u)$, will be a random variate that follows the target distribution. This method is particularly efficient when the inverse CDF is easy to compute.

The **Transformation of Random Variables Method** relies on using algebraic transformations of random variables from a known distribution to obtain a new variable with a desired distribution. This method is based on the properties of probability distributions and their relationships. For example, if a variable Z is normally distributed as $N(0,1)$, and another variable V independently follows a chi-squared distribution with n degrees of freedom, then the variable $T = \frac{Z}{\sqrt{V/n}}$ will follow a Student's t-distribution with n degrees of freedom. This method is useful as it can effectively constructs new distributions through known ones.

The **Accept-Rejection Method** is used when sampling directly from the target distribution is challenging, and it can be computationally more intensive than other methods. The process starts by selecting a proposal distribution that is easy to sample from and closely approximates the target distribution. Random samples are generated from this proposal distribution, and for each sample, a test is conducted to determine whether the sample should be accepted or rejected. This test involves comparing a uniformly distributed random number u against the ratio of the target probability density function at the sample point to the proposal PDF, scaled by a constant c. A sample is accepted if u is less than or equal to this ratio. The efficiency of the method depends on how closely the proposal distribution approximates the target distribution, as a closer fit leads to a higher acceptance rate and less computational waste.

Application to Chosen Variables

- Inverse CDF Method for Beta(5,1): To sample from the Beta(5,1) distribution using the inverse CDF method, we first need to derive the CDF of the Beta(5,1) distribution. $F(x; 5, 1)$, can be calculated as x^5 over the interval $x \in [0, 1]$. The inverse of this function, $F^{-1}(u)$ is $u^{1/5}$. The steps are as follows: First, generate a random number u from a uniform distribution over the interval [0, 1]. Next, deliver the random number u into the inverse CDF by raising u to the power of 1/5 to obtain x, which will be distributed according to the Beta(5,1) distribution.
- Transformation of Random Variables for t-distribution: This method uses the property that a ratio of a standard normal variable to the square root of a chi-squared variable, scaled by its degrees of freedom, follows a t-distribution with the same degrees of freedom as the chi-square variable. To sample from t_2 distribution using the transformation method, we first need to generate random samples z from Normal(0,1) distribution and generate random samples v from $\chi^2(2)$ distribution. Then, we can plug the samples z and v into the equation $T = \frac{Z}{\sqrt{V/2}}$, and the resulting samples t will be distributed according to the t distribution with 2 degrees of freedom.
- Accept-Rejection Method for Gamma(2,2): To sample from the Gamma(2,2) distribution using the accept-rejection method with Gamma(2,1) as the proposal distribution, I first calculated the pdf of our Gamma(2,2) target distribution, which is $2xe^{-2x}$, and the pdf for Gamma(2,1) proposal distribution is xe^{-x}. From the ratio $\frac{f(x)}{g(x)} = \frac{2xe^{-2x}}{xe^{-x}} = 2e^{-x}$, I determined that $2e^{-x} \leq 2 = c, x \in (0, \infty)$. Hence, a random sample from Gamma(2,1) is accepted if $\frac{f(x)}{2 \cdot g(x)} = \frac{2x \cdot e^{-2x}}{2x \cdot e^{-x}} = e^{-x} > u$, where u is a random sample from Unif(0,1) distribution. Otherwise, reject and resample until the desired number of accepted samples is reached.

Challenges Encountered and Resolutions

The primary challenge in this project was ensuring both the accuracy and efficiency of each sampling method. For the **Inverse CDF Method**, achieving precise computation of both the CDF and its inverse function was important, as any inaccuracies could lead to significant errors. To avoid potential issues, I selected a distribution whose CDF and inverse function are straightforward to calculate. This strategic choice minimized the risk of errors and streamlined the sampling process.

The **Transformation of Random Variables Method** presented fewer challenges because the transformation equation was directly provided by the textbook.

In contrast, the **Accept-Rejection Method** posed substantial challenges, particularly in the selection of an efficient proposal distribution. Initially, I selected Expo(2) as the proposal for sampling from Gamma(2,2), which is problematic because the ratio of their probability density functions made it difficult to determine a scaling constant c. To address this, I experimented with different parameters and ultimately chose Gamma(2,1) as the proposal distribution. This proposal distribution is more effective, optimizing the acceptance rate and enhancing the overall efficiency of the method.

Appendix

Textbook

Rizzo, Maria L. Statistical Computing with R. 1st ed., Chapman & Hall/CRC, 2007.

Code

```r
## Inverse CDF: Beta(5,1)
# sample from CDF
u_inverse <- runif(10000)
# deliver to inverse CDF
x_inverse <- u_inverse^(1/5)
# density histogram of samples from inverse CDF
hist(x_inverse, prob = TRUE, xlab = 'x', main = 'Beta(5,1)')
# plot real beta(5,1)
y_inverse <- seq(0, 1, by = 0.0001)
lines(y_inverse, dbeta(y_inverse, 5, 1), col = 'red')

## Transformation: t_2
z_transform = rnorm(10000, mean = 0, sd = 1)
v_transform = rchisq(10000, df = 2)
t_transform = z_transform / sqrt(v_transform/2)
# density histogram of samples from transformation method
hist(t_transform, prob = TRUE, xlab = 'x', main = 't_1', breaks = 500,
     xlim = c(-10,10), ylim = c(0,0.5))
# plot real t_1
y_transform <- seq(-10, 10, by = 0.002)
```

```r
lines(y_transform, dt(y_transform, df = 2),col = 'red')

## Accept-Rejection: Gamma(2,2)
sample <- 0
y_ar <- numeric(100000)

while (sample < 100000) {
  # generate a random u from the Uniform(0, 1) distribution
  u <- runif(1)
  # generate a random x from the proposal distribution Gamma(2,1)
  x <- rgamma(1, shape = 2, rate = 1)
  if (exp(-x) > u) {
    # we accept x
    sample <- sample + 1
    y_ar[sample] <- x
  } }

# density histogram of samples from accept_rejection method
hist(y_ar, prob = TRUE, xlab = 'x', main = 'Gamma(2,2)', breaks = 100)
# plot real Gamma(2,2)
y <- seq(0, 10, by = 0.001)
lines(y, dgamma(y, shape = 2, rate = 2), col = 'red')
```