

Metody Monte Carlo

Raport nr 1

Ksawery Józefowski

Zadanie 1: Symulacja dyskretnych zmiennych losowych (3 pkt)

1. Napisz funkcję realizującą generator liczb losowych o rozkładzie dwupunktowym: $\mathbb{P}(X = 1) = p$, $\mathbb{P}(X = 0) = 1 - p$, gdzie $p \in [0, 1]$.
2. Napisz funkcję realizującą generator liczb losowych z rozkładu dwumianowego. Narysuj histogram dla przykładowej próby z tego rozkładu.
3. Napisz funkcję realizującą generator dla rozkładu Poissona (w raporcie zawrzyj obliczenia przygotowawcze). Narysuj histogram dla przykładowej próby z tego rozkładu.

Rozwiązanie

W celu generowania próbek z rozkładu dwupunktowego wykorzystujemy metodę odwrotnej dystrybuanty. Dla tego rozkładu dystrybuanta przyjmuje postać skokową, co pozwala na prostą implementację funkcji kwantylowej:

$$F_X^{-1}(u) = \begin{cases} 1 & \text{dla } 0 \leq u \leq p \\ 0 & \text{dla } p < u \leq 1 \end{cases}$$

```
dwupunktowy <- function(n, p) {  
  u <- runif(n)  
  x <- ifelse(u <= p, 1, 0)  
  return(x)  
}
```

Dla rozkładu dwumianowego wykorzystujemy fakt, że można go przedstawić jako sumę n niezależnych zmiennych o rozkładzie dwupunktowym:

$$X = \sum_{i=1}^n X_i, \quad X_i \sim Dwupunktowy(p)$$

```
dwumianowy <- function(n, k, p) {  
  wyniki <- numeric(n)  
  for(i in 1:n) {  
    wyniki[i] <- sum(dwupunktowy(k, p))  
  }  
  return(wyniki)  
}
```

Rozważamy zmienną losową X o rozkładzie Poissona z parametrem $\lambda > 0$, której funkcja prawdopodobieństwa jest dana wzorem:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, 2, \dots$$

Dystrybuanta $F(k)$ to suma prawdopodobieństw od 0 do k :

$$F(k) = P(X \leq k) = \sum_{i=0}^k \frac{\lambda^i e^{-\lambda}}{i!}$$

Dla rozkładów dyskretnych dystrybuanta odwrotna $F^{-1}(u)$ jest zdefiniowana jako najmniejsza liczba całkowita k spełniająca warunek:

$$F(k) \geq u, \quad 0 < u < 1$$

Końcowo funkcja kwantylowa wygląda następująco:

$$F^{-1}(u) = \min\{k \in \mathbb{Z}_{\geq 0} : F(k) \geq u\}$$

My zastosujemy wzoru rekurencyjnego, aby nie liczyć dźwigni i zoptymalizować działanie funkcji:

$$P(X = k) = P(X = k - 1) \frac{\lambda}{k}$$

```
poisson <- function(n, lambda) {
  wyniki <- numeric(n)
  for(j in 1:n) {
    U <- runif(1)
    k <- 0
    F_k <- exp(-lambda)
    S <- F_k
    while(S < U) {
      k <- k + 1
      F_k <- F_k * lambda / k
      S <- S + F_k
    }
    wyniki[j] <- k
  }
  return(wyniki)
}
```

Przedstawiamy teraz wyniki dla rozkładu dwumianowego i Poissona odpowiednio na wykresach 1 i 2. Zauważmy, że wykresy pokrywają się z oczekiwanyymi wykresami rozkładów prawdopodobieństwa.

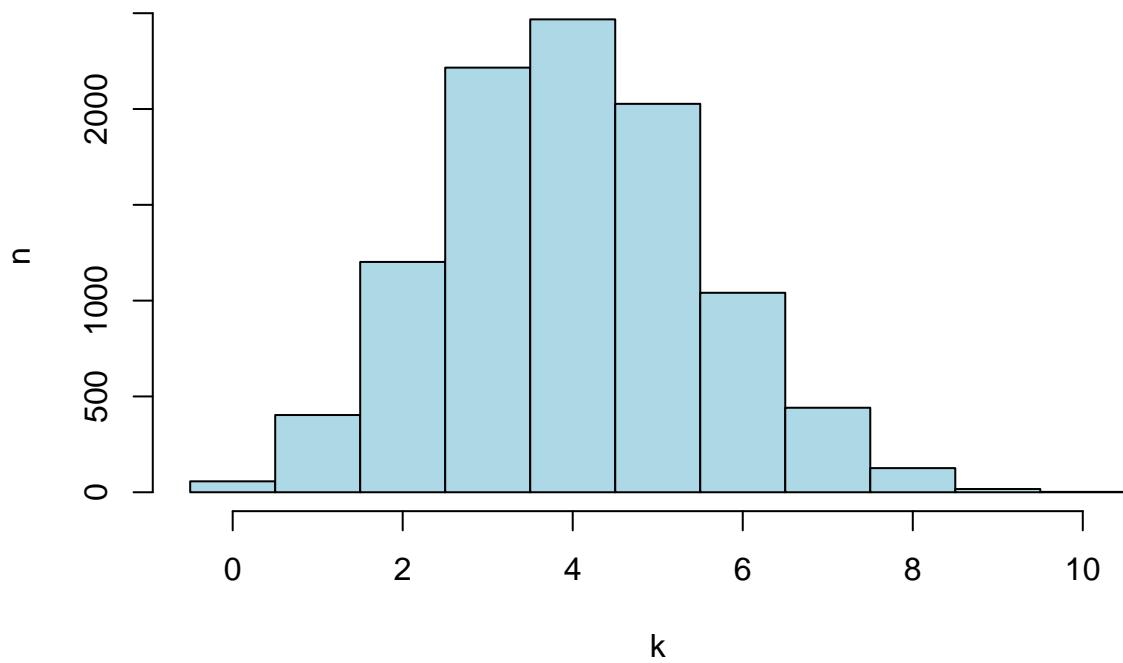


Figure 1: Wykres rozkładu dwumianowego dla $p = 0.4$

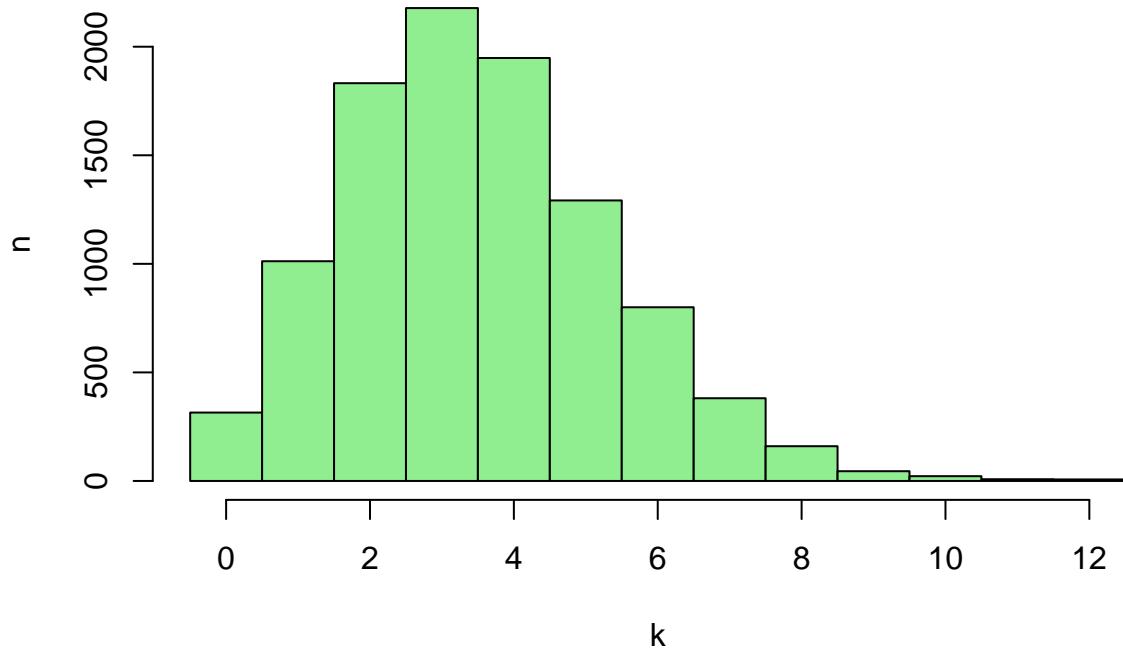


Figure 2: Wykres rozkładu Poissona dla $\lambda = 3.5$

Zadanie 2: Symulacja ciągłych zmiennych losowych (4 pkt)

1. Zaimplementuj omówiony na wykładzie generator rozkładu wykładniczego. Narysuj histogram dla próby o rozmiarze $n = 2000$ wygenerowanej przy jego pomocy i nałoż na niego krzywą gęstości rozkładu wykładniczego.
2. Wyznacz dystrybuantę odwrotną dla rozkładu Weibulla o funkcji gęstości

$$f(x) = \frac{k}{\lambda^k} x^{k-1} e^{-(x/\lambda)^k} \mathbf{1}_{(0,+\infty)}(x), \quad x \in \mathbb{R},$$

gdzie $k > 0$, $\lambda > 0$ (w raporcie zawrzyj odpowiednie obliczenia). Wykorzystaj ją do napisania funkcji realizującej generator liczb losowych z rozkładu Weibulla. Narysuj histogram dla próby o rozmiarze $n = 2000$ wygenerowanej przy jego pomocy i nałoż na niego krzywą gęstości rozkładu Weibulla.

3. Napisz funkcję realizującą generator liczb losowych z rozkładu Laplace'a z parametrem $\lambda > 0$ o gęstości:

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}, \quad x \in \mathbb{R}.$$

Narysuj histogram dla próby o rozmiarze $n = 2000$ wygenerowanej przy jego pomocy i nałoż na niego krzywą gęstości rozkładu Laplace'a.

Rozwiązanie

Zmienną losową o rozkładzie wykładniczym generujemy metodą odwrotnej dystrybuanty. Dla rozkładu wykładniczego o parametrze $\lambda > 0$ dystrybuanta ma postać:

$$F(x) = 1 - e^{-\lambda x}$$

Odwracając ją otrzymujemy funkcję kwantylową o postaci:

$$F(x)^{-1} = -\frac{1}{\lambda} \ln(1 - u)$$

```
wykladniczy <- function(n, lambda) {  
  u <- runif(n)  
  x <- -log(1 - u) / lambda  
  return(x)  
}
```

Chcemy wyznaczyć dystrybuantę odwrotną rozkładu Weibulla, więc najpierw wyliczamy dystrybuantę z podanej gęstości:

$$F(x) = \int_0^x f(t) dt = \int_0^x \frac{k}{\lambda^k} t^{k-1} e^{-(t/\lambda)^k} dt$$

Po obliczeniu całki wychodzi:

$$F(x) = 1 - e^{-(x/\lambda)^k}, \quad x \geq 0$$

Aby znaleźć funkcję odwrotną $F^{-1}(u)$, przyjmujemy $u = F(x)$:

$$u = 1 - e^{-(x/\lambda)^k}$$

Rozwiążujemy równanie względem x :

$$e^{-(x/\lambda)^k} = 1 - u$$

$$-(x/\lambda)^k = \ln(1 - u)$$

$$x^k = -\lambda^k \ln(1 - u)$$

$$x = \lambda(-\ln(1-u))^{1/k}$$

Stąd dystrybuanta odwrotna ma postać:

$$F^{-1}(u) = \lambda(-\ln(1-u))^{1/k}, \quad 0 < u < 1$$

```
weibull <- function(n, k, lambda) {
  u <- runif(n)
  x <- lambda * (-log(1 - u))^(1/k)
  return(x)
}
```

Dla rozkładu Laplace'a korzystamy z dystrybuanty odwrotnej o wzorze:

$$F^{-1}(u) = \begin{cases} \frac{1}{\lambda} \ln(2u) & \text{dla } 0 < u < \frac{1}{2} \\ -\frac{1}{\lambda} \ln(2(1-u)) & \text{dla } \frac{1}{2} \leq u < 1 \end{cases}$$

```
laplace <- function(n, lambda) {
  u <- runif(n)
  x <- numeric(n)
  idx_l <- which(u < 0.5)
  idx_up <- which(u >= 0.5)
  x[idx_l] <- log(2 * u[idx_l]) / lambda
  x[idx_up] <- -log(2 * (1 - u[idx_up])) / lambda
  return(x)
}
```

Wyniki symulacji dla $n = 2000$ z porównaniem do krzywej gęstości rozkładów przedstawiamy poniżej odpowiednio na histogramach 3, 4 i 5:

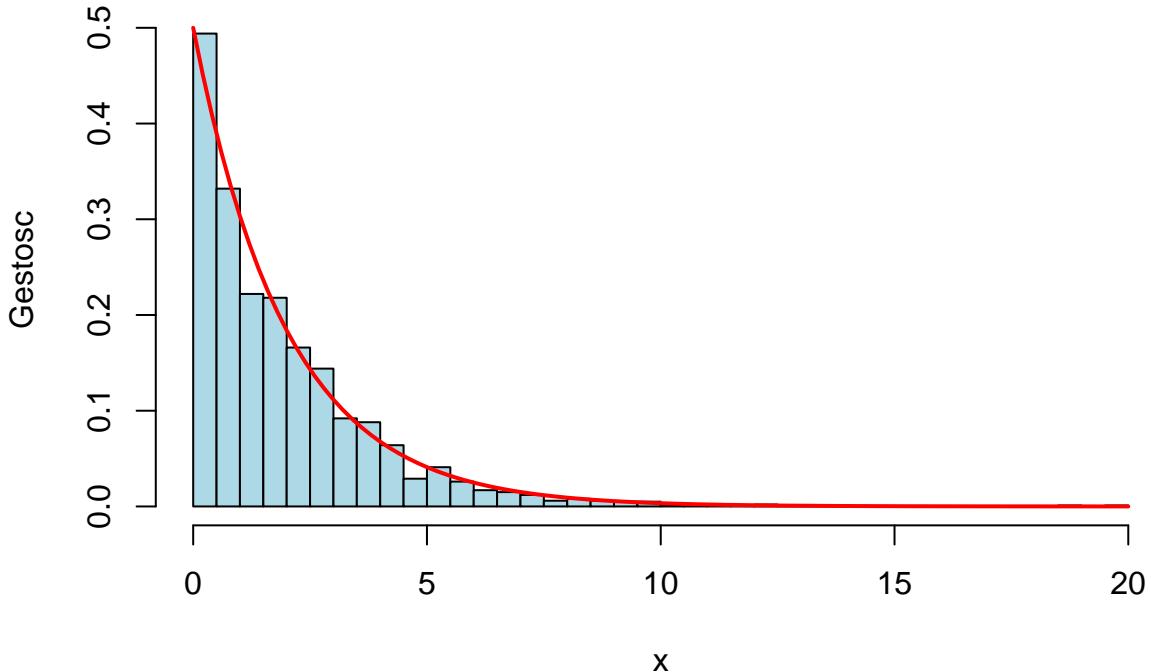


Figure 3: Wykres rozkładu wykładniczego dla $\lambda = 0.5$

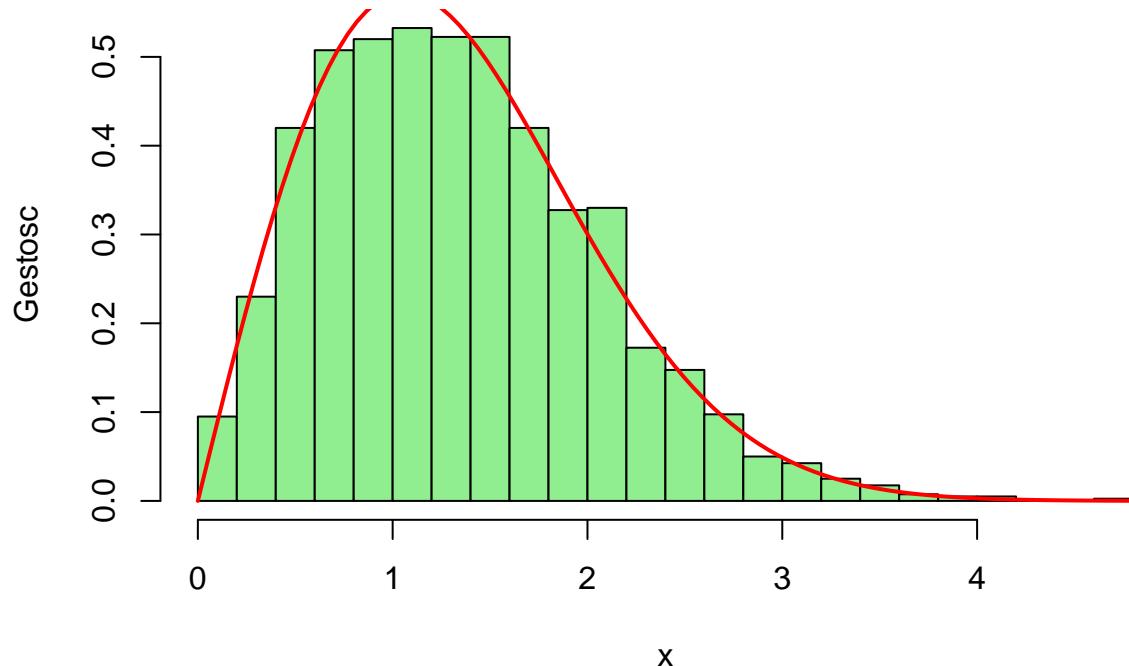


Figure 4: Wykres rozkładu Weibull'a dla $\lambda = 1.5$

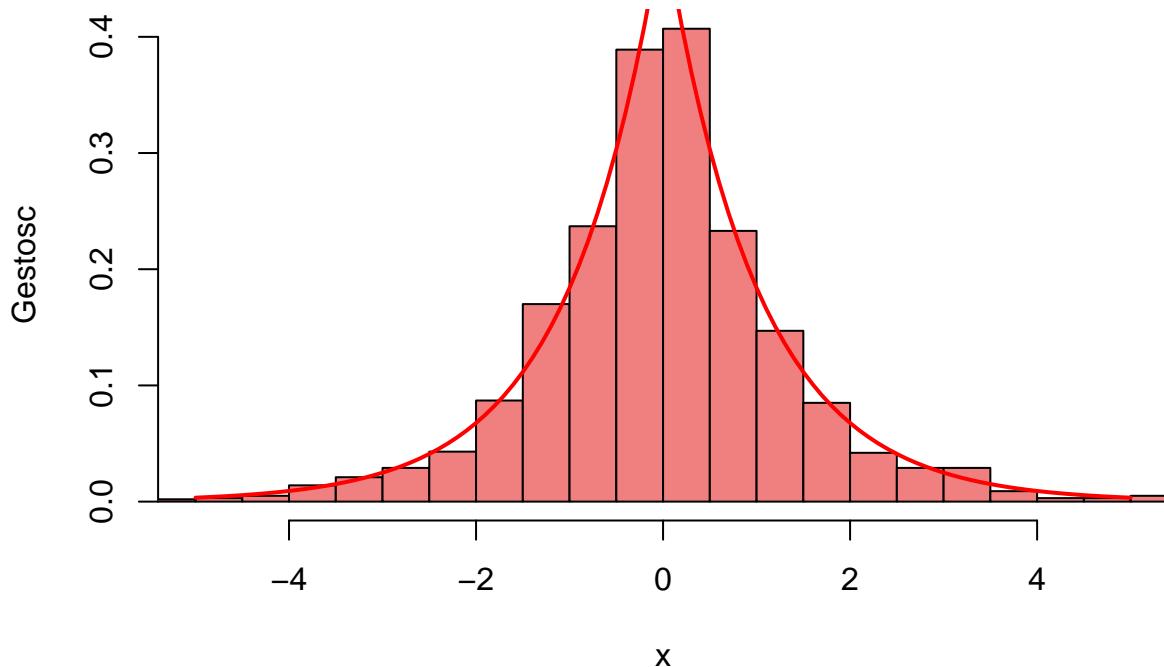


Figure 5: Wykres rozkładu Laplace'a dla $\lambda = 1$

Zadanie 3: Generowanie rozkładu normalnego metodą Boxa-Müllera (4 pkt)

Jak wiadomo z wykładu, jeżeli U_1, U_2 są dwiema niezależnymi zmiennymi z rozkładu jednostajnego na $[0, 1]$, to zmienne

$$X_1 := \sqrt{-2 \ln(U_2)} \cos(2\pi U_1), \quad X_2 := \sqrt{-2 \ln(U_2)} \sin(2\pi U_1),$$

są dwiema niezależnymi zmiennymi o standardowym rozkładzie normalnym.

1. Napisz funkcję realizującą generator par liczb losowych z rozkładu normalnego (dla zadanych parametrów μ i σ) wykorzystujący powyższą transformację Box-Mullera
2. Narysuj histogram z nałożoną odpowiednią funkcją gęstości dla próby rozmiaru $n = 5000$ wygenerowanej przy pomocy napisanego generatora.
3. Wygeneruj 2000 realizacji par (X_1, X_2) . Wykorzystaj funkcję

```
jointplot <- function(x,y){  
  df <- data.frame(x,y)  
  scatter <- ggplot(df, aes(x=x,y=y))+geom_bin2d() +  
    scale_fill_continuous(type = "viridis") +  
    theme_bw()  
  hist_right <- ggplot(df)+geom_histogram(aes(y), fill="#69b3a2",  
                                         color='darkblue')+  
    coord_flip()  
  hist_top <- ggplot(df)+geom_histogram(aes(x), fill="#69b3a2",  
                                         color='darkblue')  
  
  grid.arrange(hist_top, scatter, hist_right, ncol=2, nrow=2,  
               widths=c(4, 1), heights=c(1, 4),  
               layout_matrix = rbind(c(1, NA), c(2,3)))}
```

by narysować wykresy łączne rozkładów realizacji par zmiennych:

- X_1 i X_2 ,
 - $X_1, X_1 + X_2$,
 - $X_1 + X_2, X_1 - X_2$.
4. W każdym z powyższych (trzech) przypadków, oblicz korelacje realizacji obu zmiennych. Czy współrzędne są od siebie niezależne? Wytlumacz.

Rozwiązanie:

Funkcja `box_muller_generator` implementuje transformację Boxa-Mullera, która pozwala generować liczby z rozkładu normalnego na podstawie zmiennych z rozkładu jednostajnego. Przyjmuje trzy parametry:

- μ – wartość oczekiwana rozkładu
- σ – odchylenie standardowe
- n – liczba generowanych wartości

Algorytm działa poprzez generowanie par niezależnych zmiennych z rozkładu jednostajnego, które następnie są transformowane do rozkładu normalnego przy użyciu funkcji trygonometrycznych i logarytmicznych:

$$R = \sqrt{-2 \ln U_2}, \quad \theta = 2\pi U_1$$

$$X_1 = R \cos \theta, X_2 = R \sin \theta$$

Końcowo wartości są skalowane zgodnie z μ i σ :

$$X = \sigma \cdot X_i + \mu$$

```
box_muller_generator <- function(n, mu = 0, sigma = 1) {  
  m <- ceiling(n/2) * 2  
  U1 <- runif(m/2)  
  U2 <- runif(m/2)  
  
  R <- sqrt(-2 * log(U2))  
  theta <- 2 * pi * U1  
  
  X1 <- R * cos(theta)  
  X2 <- R * sin(theta)  
  
  result <- c(X1, X2) * sigma + mu  
  return(result[1:n])  
}
```

Poniżej przedstawiamy wykres 6 wygenerowany poprzez transformację Box-Mullera dla $n = 5000$ prób losowych z rozkładu normalnego o parametrach $\mu = 2$ i $\sigma = 3$. Czerwona linia przedstawia funkcję gęstości wspomnianego rozkładu. Jak widać na histogramie, napisany generator bardzo dobrze dopasowuje się do krzywej gęstości.

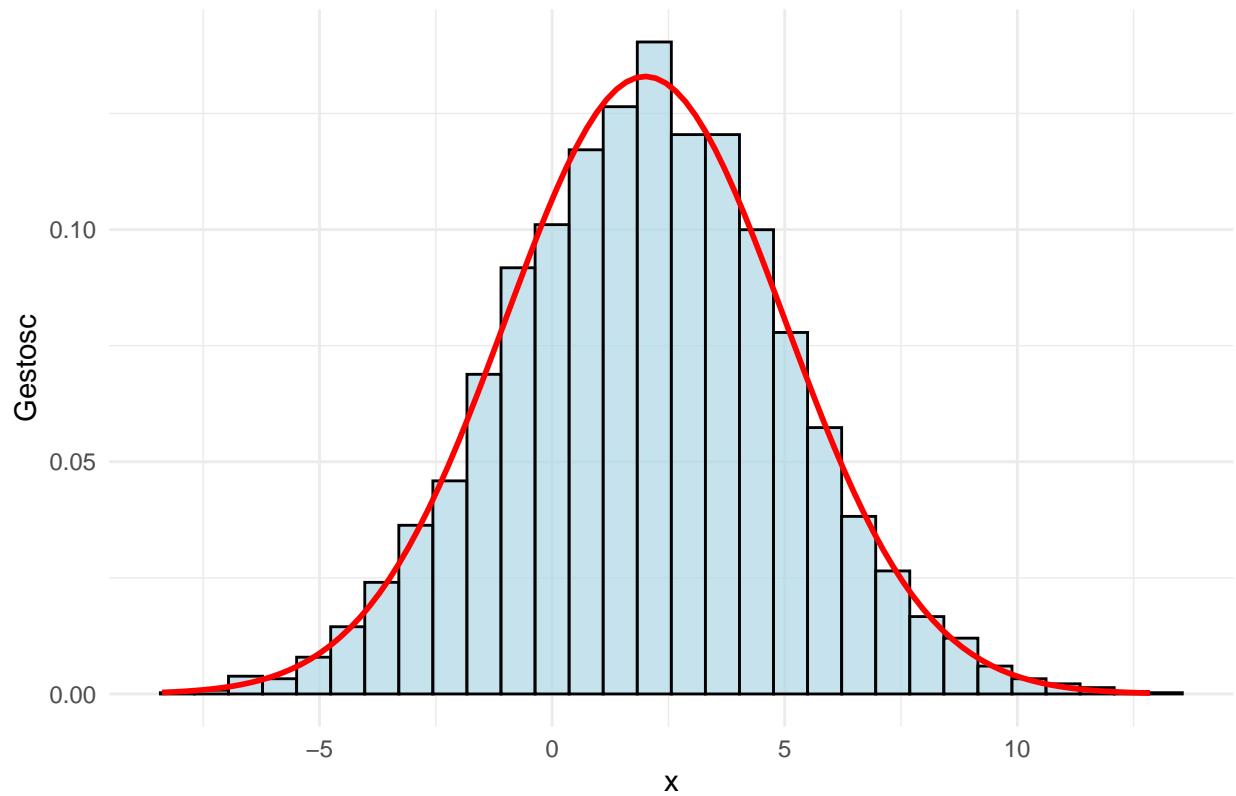


Figure 6: Wykres rozkładu normalnego generowany Box'em-Muller'em

Generujemy $n = 2000$ realizacji par. Następnie wykorzystujemy podaną funkcję do generowania wykresów 7, 8 i 9 odpowiednio dla par (X_1, X_2) , $(X_1, X_1 + X_2)$, $(X_1 + X_2, X_1 - X_2)$

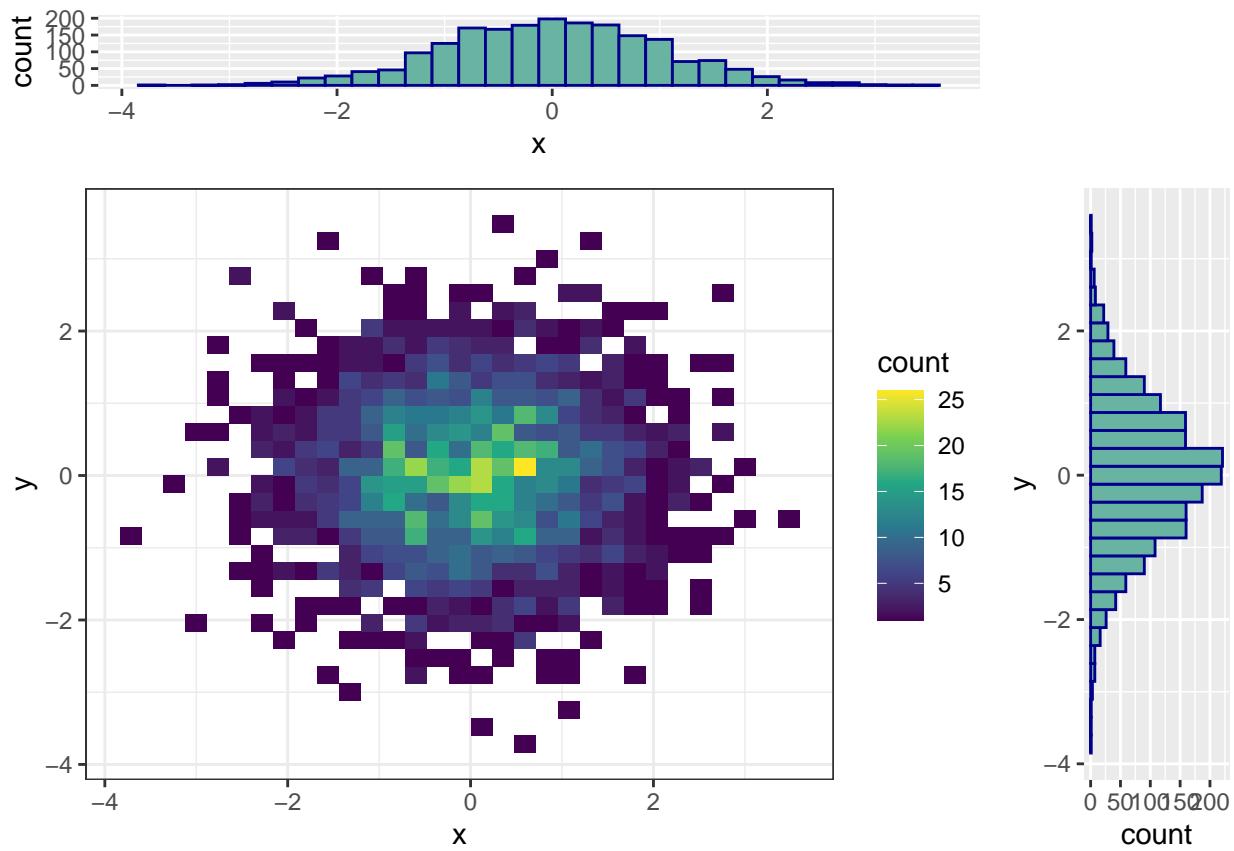


Figure 7: Symulacja dla X_1, X_2

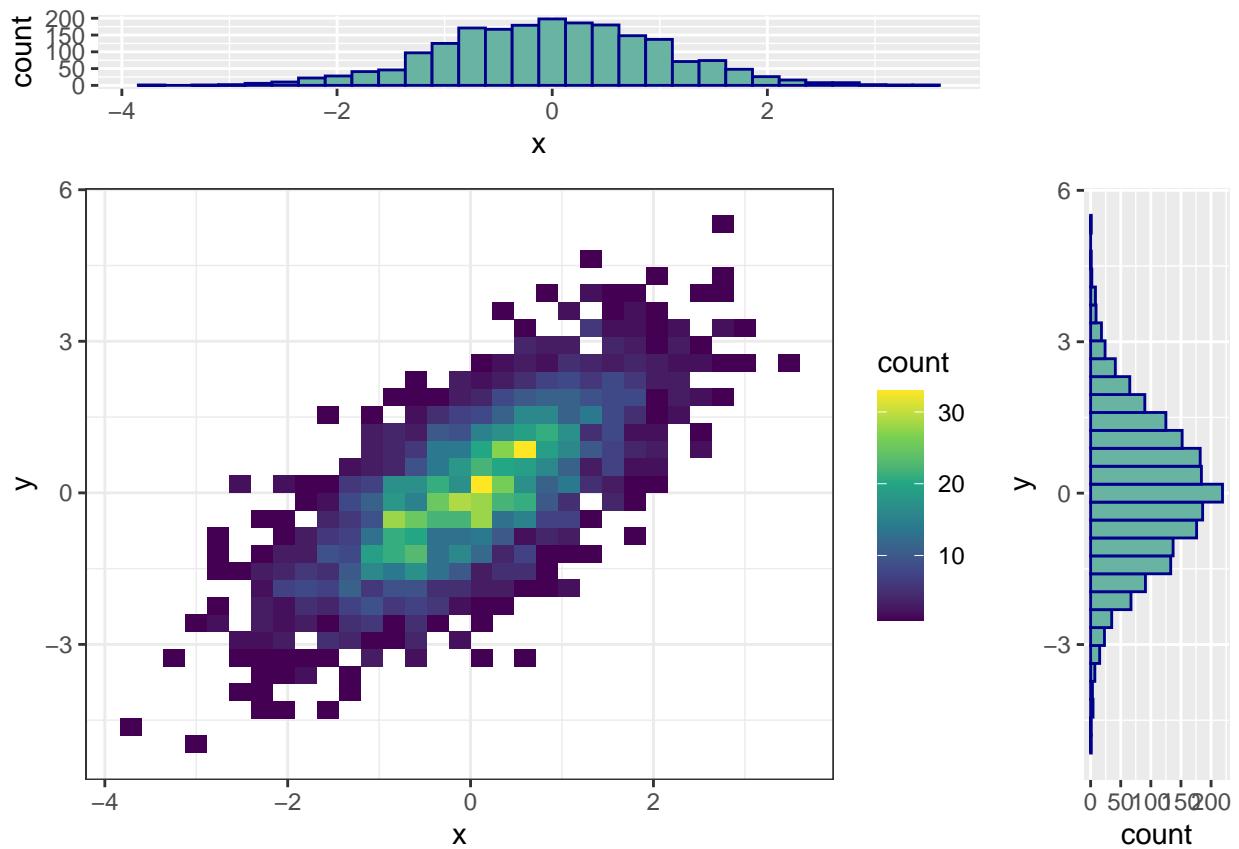


Figure 8: Symulacja dla $X_1, X_1 + X_2$

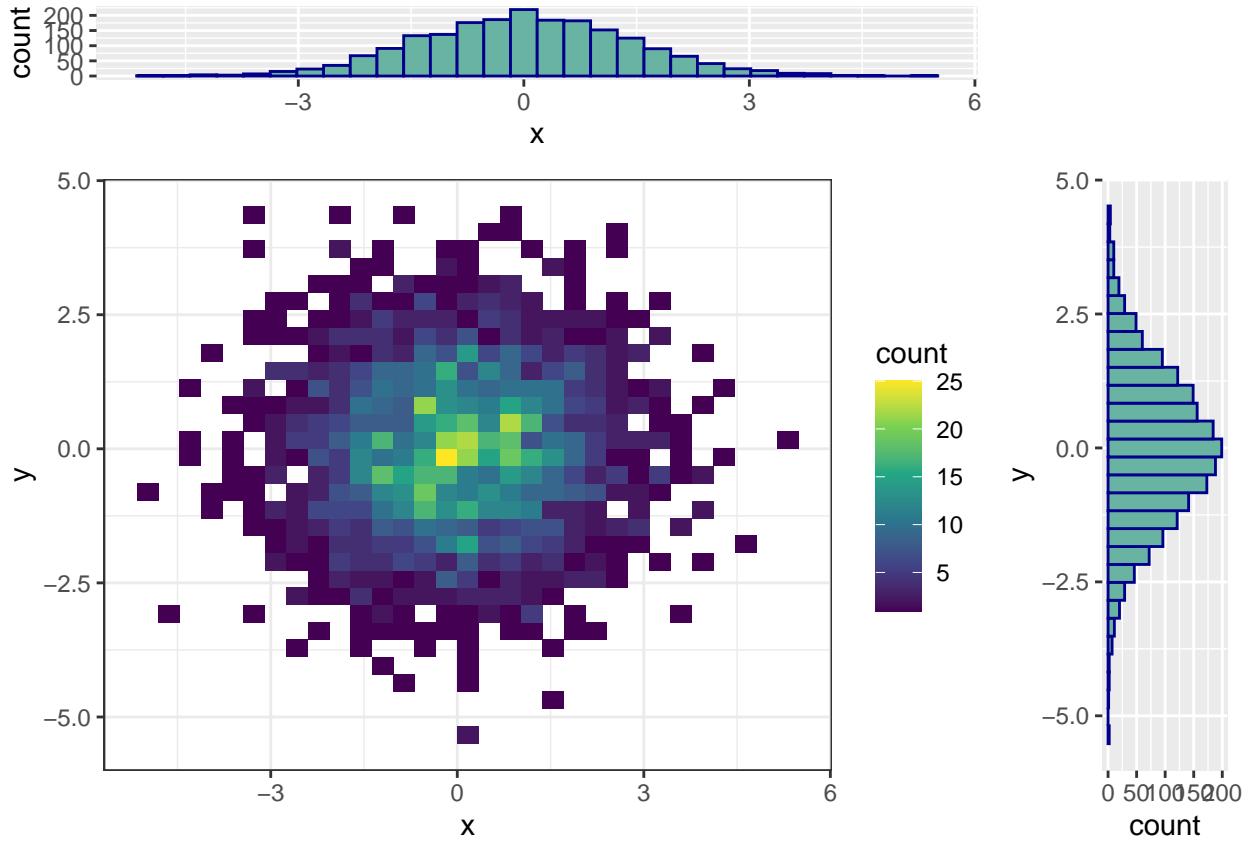


Figure 9: Symulacja dla $X_1 + X_2, X_1 - X_2$

Korelacje między zmiennymi:

Para(X_1, X_2):

```
cor(X1, X2)
```

```
## [1] -0.0005815006
```

Korelacja pierwszej pary jest bardzo bliska zeru, więc zmienne X_1, X_2 są niezależne. Wynika to bezpośrednio z metody Boxa-Mullera, która generuje niezależne zmienne losowe.

Para($X_1, X_1 + X_2$):

```
cor(X1, X1 + X2)
```

```
## [1] 0.7115682
```

Dla drugiej pary korelacja jest w wysokich dodatnich wartościach (0.7), więc $X_1, X_1 + X_2$ są zależne. Staje się tak ponieważ kombinacja liniowa wprowadza "szum", który skutkuje w zależności zmiennych.

Para($X_1 + X_2, X_1 - X_2$):

```
cor(X1 + X2, X1 - X2)
```

```
## [1] 0.01323242
```

Trzecia para wykazuje korelację bliską zeru co wskazuje na niezależność zmiennych losowych, a jest to prawda z racji ortogonalności kombinacji liniowych $X_1 + X_2$ i $X_1 - X_2$.

Zad. 4. Metoda akceptacji (3 pkt)

1. Zaproponuj alternatywny sposób generowania prób z rozkładu normalnego oparty na metodzie akceptacji. Jako rozkład majorzyzujący rozkład normalny wybierz rozkład Laplace'a i wykorzystaj generator napisany w zadaniu 2.3. Spróbuj wyznaczyć wartości stałej normującej M oraz parametru λ rozkładu Laplace'a, tak by możliwie zminimalizować liczbę odrzucanych próbek.
2. Porównaj czas generowania próby o rozmiarze $n = 10000$ za pomocą napisanego generatora oraz generatora wykorzystującego transformację Boxa-Müllera z zadania 3.

Rozwiążanie:

Krok 1:

Niech

- Rozkład docelowy: $N(0, 1)$ z gęstością $f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$
- Rozkład pomocniczy: Laplace'a z gęstością $g(x) = \frac{\lambda}{2} e^{-\lambda|x|}$

Wyznaczamy stałą M taką, że $f(x) \leq M \cdot g(x)$ dla wszystkich $x \in \mathbb{R}$.

Krok 2:

Stosunek gęstości:

$$\frac{f(x)}{g(x)} = \frac{\frac{1}{\sqrt{2\pi}} e^{-x^2/2}}{\frac{\lambda}{2} e^{-\lambda|x|}} = \frac{2}{\lambda \sqrt{2\pi}} e^{-x^2/2 + \lambda|x|}$$

Maksimum funkcji $h(x) = -\frac{x^2}{2} + \lambda|x|$ otrzymujemy dla:

$$h'(x) = -x + \lambda = 0 \implies x = \lambda$$

Podstawiając:

$$M = \sup_x \frac{f(x)}{g(x)} = \frac{2}{\lambda \sqrt{2\pi}} e^{-\lambda^2/2 + \lambda^2} = \frac{2}{\lambda \sqrt{2\pi}} e^{\lambda^2/2}$$

Aby zminimalizować prawdopodobieństwo odrzucenia, minimalizujemy M względem λ :

$$\frac{dM}{d\lambda} = \frac{2}{\sqrt{2\pi}} \left(-\frac{1}{\lambda^2} e^{\lambda^2/2} + \frac{1}{\lambda} \cdot \lambda e^{\lambda^2/2} \right) = 0$$

Rozwiązanie to $\lambda = 1$. Zatem optymalny parametr to $\lambda = 1$. Optymalna stała normująca:

$$M = \frac{2}{1 \cdot \sqrt{2\pi}} e^{1/2} = \sqrt{\frac{2e}{\pi}} \approx 1.315$$

Algorytm na, którym bazować będzie kod wygląda następująco:

1. Wygeneruj X z rozkładu Laplace'a z parametrem $\lambda = 1$
2. Wygeneruj $U \sim U(0, 1)$
3. Jeśli

$$U \leq \frac{f(x)}{M \cdot g(x)} = \frac{1}{M} \cdot \frac{f(x)}{g(x)} = \frac{1}{M} \cdot \frac{2}{\lambda \sqrt{2\pi}} e^{-x^2/2 + |x|}$$

to zaakceptuj X , w przeciwnym razie odrzuć i powtóż.

W ten sposób uzyskane wartości będą pochodziły z rozkładu normalnego $N(0, 1)$.

```

normal_accept <- function(n, lambda=1) {
  M <- (2 / (lambda * sqrt(2 * pi))) * exp(lambda^2 / 2)
  results <- numeric(0)
  count <- 0

  while(length(results) < n) {
    x <- laplace(1, lambda)
    u <- runif(1)

    f <- (1 / sqrt(2 * pi)) * exp(-x^2 / 2)
    g <- (lambda / 2) * exp(-lambda * abs(x))
    count <- count + 1

    if (u <= f / (M * g)) {
      results <- c(results, x)
    }
  }

  return(results)
}

set.seed(37)
system.time({x_accept <- normal_accept(10000, lambda = 1)})

## użytkownik      system      upłynęło
##          0.14        0.06        0.18
system.time({x_box <- box_muller_generator(10000)})

## użytkownik      system      upłynęło
##          0          0          0

```

Pomiar czasu wykonania pokazał, że metoda Boxa–Müllera jest zdecydowanie szybsza od metody akceptacji. W przypadku próby o rozmiarze $n = 10000$ metoda akceptacji zajmuje średnio około 0,30 sekundy, podczas gdy metoda Boxa–Müllera była tak szybka, że czas jej wykonania został zarejestrowany jako zero, co wynika jedynie z ograniczonej dokładności funkcji `system.time()` dla bardzo krótkich operacji. Różnica ta jest naturalna, ponieważ metoda akceptacji wymaga generowania dodatkowych wartości, które są odrzucone, aby uzyskać próbki zgodną z rozkładem normalnym, natomiast metoda Boxa–Müllera każdorazowo generuje dwie poprawne wartości bez żadnych odrzuceń. Pomimo tej różnicy w czasie wykonania, obie metody generują próbki o rozkładzie dobrze przybliżającym rozkład normalny, co możemy zauważyć na wykresach 10 i 11 poniżej:

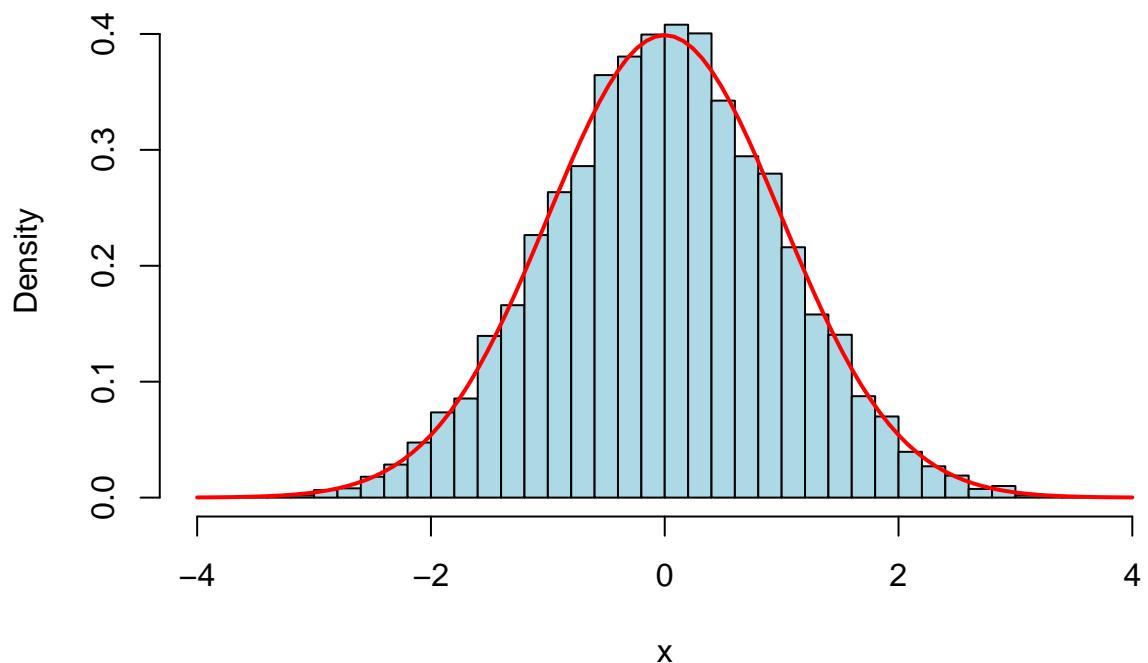


Figure 10: Metoda Akceptacji

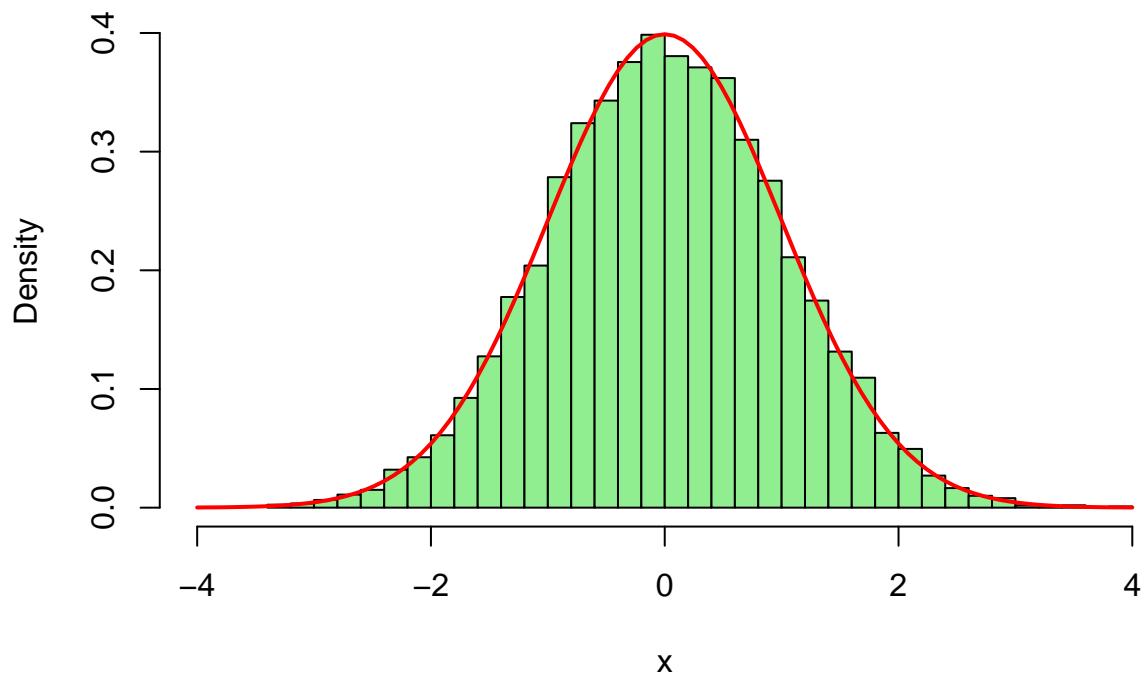


Figure 11: Metoda Box'a-Muller'a