Lista 4

Alicja Wiączkowska

2023-06-07

Zadanie 1 *Symulacja rozkładu jednostajnego*

Jeśli wektory X i Y mają rozkłady jednostajne odpowiednio na przedziałach $[a, b]$, $[c, d]$, to ich gęstości wynoszą:

$$f_X(x) = \frac{1}{b-a}, \qquad dla \quad a < x < b$$

$$f_Y(y) = \frac{1}{d-c}, \qquad dla \quad c < y < d$$

Jeśli wektor losowy (X, Y) ma rozkład jednostajny na prostokącie $[a, b] \times [c, d]$,a więc rozkład dwuwymiarowy (w którym wartości na poszczególnych osiach od siebie nie zależą), to funkcja gęstości jest postaci

$$f(x, y) = f_X(x)f_Y(y) = \frac{1}{(b-a)(d-c)}, \qquad dla \quad a < x < b \quad oraz \quad c < y < d.$$

Z tej obserwacji wynika, że aby jednostajnie wylosować punkty z prostokąta możemy skożystać z polecenia: `c(runif(1,a,b), runinf(1,c,d))`

Algorytm jednostajnego losowania punktów z figury S nie będącej prostokątem

Załóżmy, że S jest ograniczonym i zamkniętym zbiorem w pewnym prostokącie R.

1. `Wygeneruj w sposób jednostajny losowy punkt w prostokącie.`
2. `Jeżeli punkt zawiera się w S,zaakceptuj go jako żądany punkt.`
3. `Jeśli punkt nie jest zawarty w S,odrzuć go i przejdź do punktu 1.`

Poprawność algorytmu

Chcemy pokzać, że losując punkt p za pomocą powyższego algorytmu, losujemy jednotajnie ze zbioru S, czyli przy ustalonym dowolnym zbiorze $A \subseteq S$ o polu $|A|$. Pokażmy, że $\mathbb{P}(p \in A \mid p \in S) = \frac{|A|}{|S|}$ korzystając ze wzoru na prawdopodobieństwo warunkowe.

$$\mathbb{P}(p \in A \mid p \in S) = \frac{\mathbb{P}(p \in A \cap S)}{\mathbb{P}(p \in S)} = \mathbb{P}(p \in A) \div \mathbb{P}(p \in S) = \frac{|A|}{|R|} \div \frac{|S|}{|R|} = \frac{|A|}{|R|} \cdot \frac{|R|}{|S|} = \frac{|A|}{|S|} \qquad c.k.d.$$

Przykład implementacji

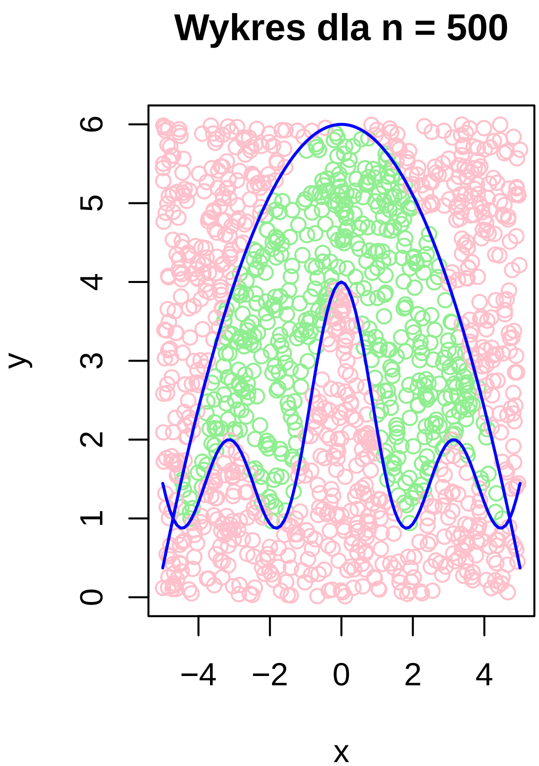
Chcemy wygenerować n=500 i n=5000 punktów w zbiorze zawartym pomiędzy wykresami funkcji

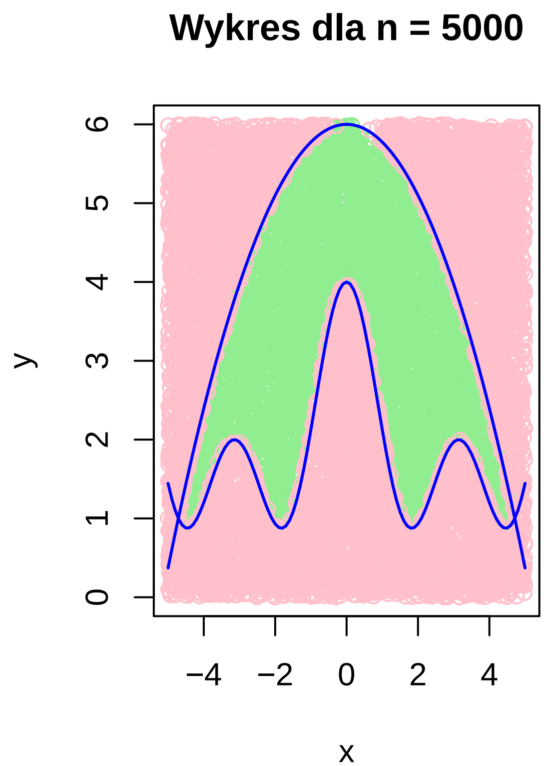
$$f(x) = 6 - \frac{20x^2}{9\pi^2} \qquad oraz \qquad g(x) = \cos(x) + \cos(2x) + 2$$

Zauważmy, że zbiór wartości funkcji f to $(-\infty, 6]$, a funkcji g zawiera się w przedziale $[0, 4]$.Obszar pomiędzy tymi funkcjami będzie nieograniczony, jeśli nie przyjmiemy ograniczenia osi odciętych lub założenia, że f ma być granicą górną, a g dolną.
Załóżmy drugą opcję. Wówczas zbiór $\{x : f(x) \geq g(x)\}$ zawiera się w przedziale $[-5, 5]$ (nieco szerszym niż jest to konieczne). Wówczas prostokątem ograniczającym będzie $R = [-5, 5] \times [0, 6]$

```
f<-function(x){6 - (20*x^2)/(9*pi^2)}
g<-function(x){cos(x)+cos(2*x)+2}

generuj<-function(n){
  dobre_x<-c()
  dobre_y<-c()
  zle_x<-c()
  zle_y<-c()

  while(length(dobre_x) < n){
    los_x<-runif(n-length(dobre_x),-5,5)
    los_y<-runif(n-length(dobre_y),0,6)
    zgodnosc<-(f(los_x)>=los_y & g(los_x)<=los_y)

    dobre_x<-c(dobre_x,los_x[zgodnosc])
    dobre_y<-c(dobre_y,los_y[zgodnosc])

    zle_x<-c(zle_x,los_x[!zgodnosc])
    zle_y<-c(zle_y,los_y[!zgodnosc])
  }

  plot(dobre_x,dobre_y,col='lightgreen',main=paste("Wykres dla n =",n),
       xlim=c(-5,5),ylim=c(0,6), xlab="x", ylab="y")
  points(zle_x,zle_y,col='pink')
  curve(f,col='blue',add=T,lwd=1.5)
  curve(g,col='blue',add=T,lwd=1.5)


  return(n/(n+length(zle_x)))
}
```

```
par(mfrow=c(1,2))
p_500<-generuj(500)
p_5000<-generuj(5000)
```





Ponieważ znamy pole prostokąta ograniczającego, możemy obliczyć przybliżone pole figury S korzystając ze wzoru:

$$|S| = |R| \cdot \mathbb{P}(p \in A \mid p \in S) \approx |R| \cdot \frac{n}{n+z},$$

gdzie przy wylosowaniu $n+z$ punktów z prostokąta R, liczba n opisuje ilość punktów należących do figury S, a z ilość punktów należących do zbioru R/S. Dokładność przybliżenia jest tym większa im więcej wylosowano punktów, zgodnie z Prawem Wielkich Liczb.

W implementacji algorytmu losowaliśmy punkty, aż do uzyskania zadanej liczby punktów należących do figury S.
Skoro $|R| = [5-(-5)] \cdot [6-0] = 60$, to wykonana symulacja zwraca nam wartości:

- $|S| \approx 22.9$ przy $n = 500$
- $|S| \approx 23.99$ przy $n = 5000$

Zadanie 2 *Metoda transformacji zmiennych losowych*

Z wykorzystaniem wbudowanej funkcji losującej wartości z rozkładu $Exp(1)$ możemy uzyskać wartości z rozkładów chi-kwadrat, gamma lub beta.

Rozkład chi-kwadrat

Niech $X_1, X_2, ..., X_n$ będą i.i.d. o wspólnym wykładniczym $Exp(1)$. Wtedy $Y = 2\sum_{i=1}^{n} {X_i}^2 \sim \chi^2_{2n}$ ma rozkład chi-kwadrat z $2n$ stopniami swobody.

Wygernerujmy 1000 realizacji rozkładu χ^2_6.

```
chi2<-function(r,n){
  # r - ilosc realizacji; 2n- ilość stopni swobody
  #Tworzymy macierz, której kolumnami są zmienne losowe (X_1, ... ,X_n)

  X<-matrix(rexp(r*n),ncol=n)
  Y<-numeric(r)
  for(i in 1:r){
    Y[i]<-2*sum(X[i,])
  }
  return(Y)
}

opracowanie<-function(wek){
  V<-var(wek)
  opr<-c((summary(wek)[c(1,6,4,3)]),V)
  names(opr)<-c('minimum',"maksimum", "średnia", "mediana", "wariancja")
  d<-data.frame(opr)
  colnames(d)<-('wartość')
  knitr::kable(d)
  }
```

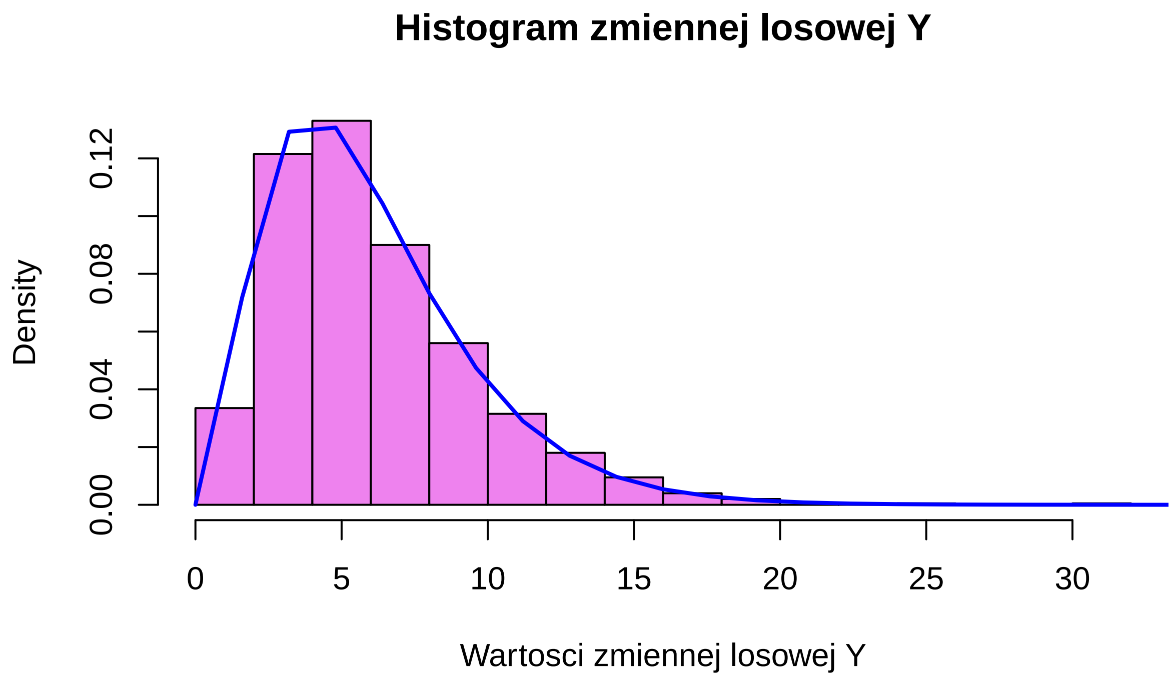
W tabeli przedstawione są najważniejsze statystyki zmiennej losowej $Y \sim \chi^2_6$

```
Y<-chi2(1000,6/2)
opracowanie(Y)
```

	wartość
minimum	0.364887
maksimum	30.153550
średnia	6.098071
mediana	5.360305
wariancja	12.617953

Poniżej znajduje się histogram zmiennej losowej $Y \sim \chi^2_6$ z dorysowaną funkcją gęstości dla rozkładu χ^2_6. Zauważmy, że kształt histogramu dobrze przybliża funkcję. Nie mamy zatem podstaw do uznania przedstawionej metody losowania za niewłaściwą.

```
hist(Y, xlab="Wartości zmiennej losowej Y", col="violet",
     main="Histogram zmiennej losowej Y",freq=F )
curve(dchisq(x,6),from=0 , to=160 , add=T,col='blue', lwd=2 )
```



Rozkład gamma

Niech $X_1, X_2, ..., X_n$ będą i.i.d. o wspólnym wykładniczym $Exp(1)$. Wtedy $Y = \beta \sum_{i=1}^{n} X_i \sim \mathcal{G}(n, \beta)$ ma rozkład gamma z parametrami n i β.

Wygernerujmy 1000 realizacji rozkładu $\mathcal{G}(2, 6)$.

```
Gamma<-function(r,n,beta){
  # r - ilosc realizacji
  #Tworzymy macierz, której kolumnami są zmienne losowe (X_1, ... ,X_n)

  X<-matrix(rexp(r*n),ncol=n)
  Y<-numeric(r)
  for(i in 1:r){
    Y[i]<-beta*sum(X[i,])
  }
  return(Y)
}
```

W tabeli przedstawione są najważniejsze statystyki zmiennej losowej $Y \sim \mathcal{G}(2, 6)$

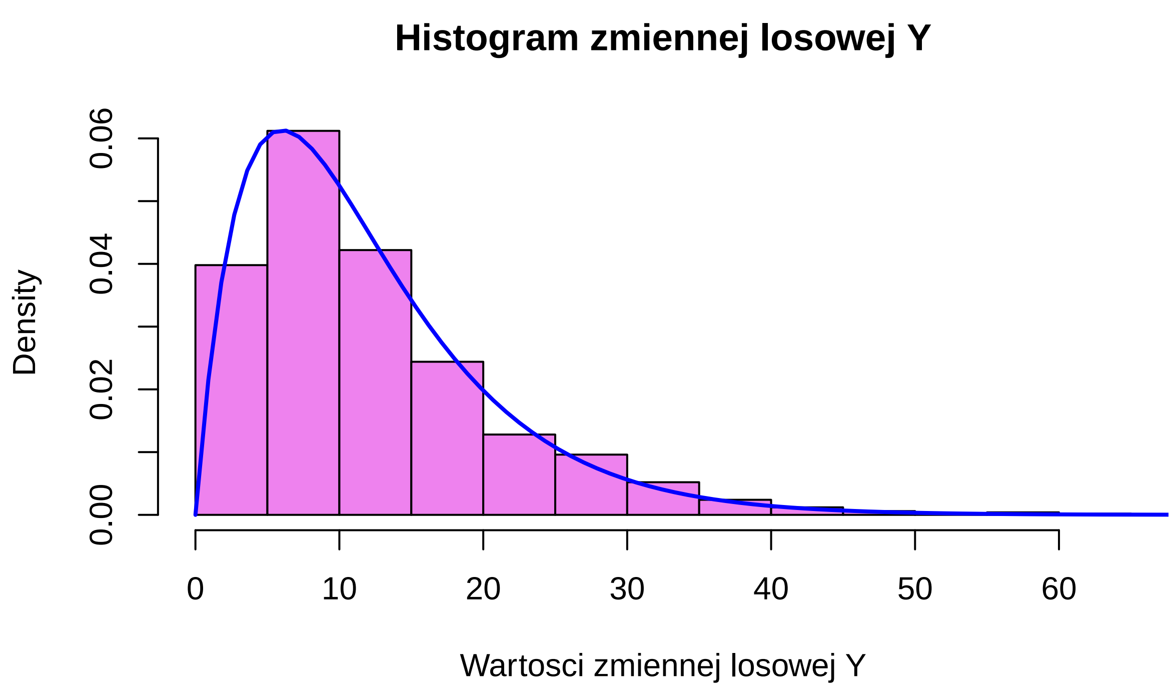
```
Y<-Gamma(1000,2,6)
opracowanie(Y)
```

	wartość
minimum	0.0569722

	wartość
maksimum	61.9552918
średnia	12.1714451
mediana	9.9402041
wariancja	80.0484101

Poniżej umieszczono histogram zmiennej losowej $Y \sim \mathcal{G}(2,6)$ z dorysowaną funkcją gęstości dla rozkładu $\mathcal{G}(2,6)$. Również w tym przypadku kształt histogramu dobrze przybliża funkcję, zatem nie ma podstaw do uznania tej metody losowania za niepoprawną.

```
hist(Y, xlab="Wartości zmiennej losowej Y", col="violet",
     main="Histogram zmiennej losowej Y",freq=F )
curve(dgamma(x,2,1/6),from=0 , to=90 , add=T,col='blue', lwd=2 )
```



Rozkład beta

Niech $X_1, X_2, ..., X_n$ będą i.i.d. o wspólnym wykładniczym $Exp(1)$. Wtedy
$Y = \beta \frac{\sum_{i=1}^{a} X_i}{\sum_{i=1}^{a+b} X_i} \sim \mathcal{G}(n, \beta)$ ma rozkład gamma z parametrami a i b.

Wygernerujmy 1000 realizacji rozkładu $\mathcal{B}(2,6)$.

```
Beta<-function(r,a,b){
  # r - ilosc realizacji
  #Tworzymy macierz, której kolumnami są zmienne losowe (X_1, ... ,X_n)
  n=a+b
```

```
  X<-matrix(rexp(r*n),ncol=n)
  Y<-numeric(r)
  for(i in 1:r){
    Y[i]<-(sum(X[i,1:a]))/(sum(X[i,1:n]))
  }
  return(Y)
}
```

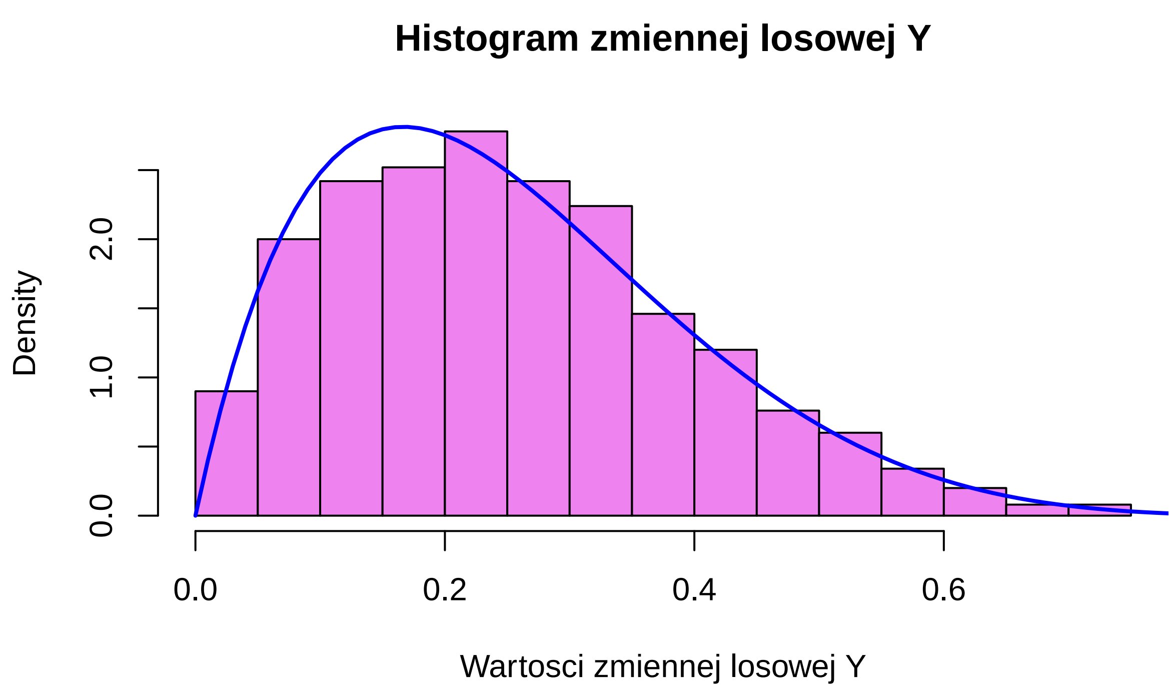
W tabeli przedstawione są najważniejsze statystyki zmiennej losowej $Y \sim \mathcal{G}(2,6)$

```
Y<-Beta(1000,2,6)
opracowanie(Y)
```

	wartość
minimum	0.0058658
maksimum	0.7372410
średnia	0.2553688
mediana	0.2416411
wariancja	0.0204040

Poniżej umieszczono histogram zmiennej losowej $Y \sim \mathcal{G}(2,6)$ z dorysowaną funkcją gęstości dla rozkładu $\mathcal{G}(2,6)$. Widzimy, że kształt histogramu dobrze przybliża funkcję, zatem nie mamy podstaw do uznania tej metody losowania za niewłaściwą.

```
hist(Y, xlab="Wartości zmiennej losowej Y", col="violet",
     main="Histogram zmiennej losowej Y",freq=F )
curve(dbeta(x,2,6),from=0 , to=1 , add=T,col='blue', lwd=2 )
```



Zadanie 3 *Metoda akceptacji-odrzucenia*

Chciemy wygenerować metodą akceptacji-odrzucenia 1000 realizacji z rozkładu $\mathcal{B}(2.7, 6.3)$ obierając różne rozkłady instrumentalne.
Oznaczmy przez f gęstość szukanego rozkładu $\mathcal{B}(2.7, 6.3)$. Jeśli gęstość instrumentalna g ma zgodny nośnik z f oraz istnieje stała M, dla której zachodzi $\forall x \quad \frac{f(x)}{g(x)} \leq M$ to poniższy algorytm poprawnie symuluje punkty zmiennej losowej X o zadanym rozkładzie:

1. `Generujemy punkty` $Y \sim g, U \sim \mathcal{U}(0,1)$;
2. `Jeśli` $U \leq \frac{f(Y)}{M \cdot g(Y)}$, `to zaakceptuj wylosowany punkt.`
3. `W przeciwnym przypadku odrzuć wygenerowany punkt i przejdź do punktu 1.`

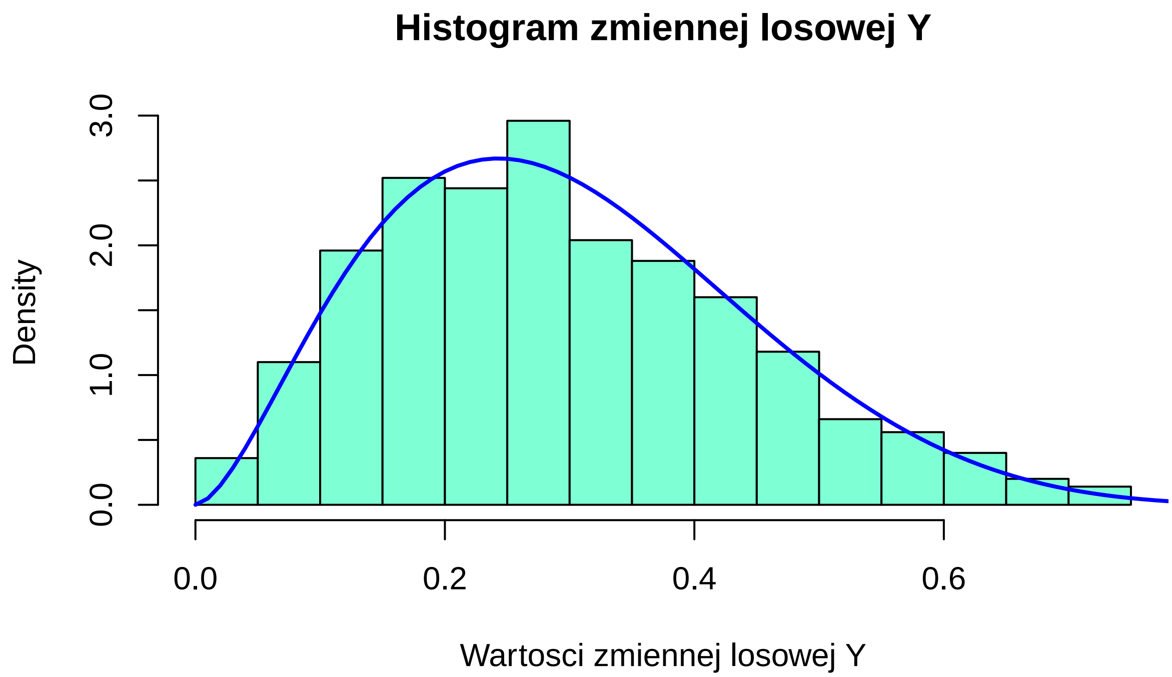
Rozkłd instrumentalny: rozkład jednostajny na odcinku [0,1]

Za wartość M przyjmujemy maksimum gęstości rozkładu beta $\mathcal{B}(2.7, 6.3)$ i implementujemy powyższy algorytm.

```
generuj2<-function(n=1000,g,r_g, M){
  'g(x) -funkcja gęstości rozkładu instrumentalnego
  r_g(x) - funkcja losująca z x elementów z rozkładu instrumentalnego'
  f<-function(x){dbeta(x,2.7,6.3)} # funkcja szukanego rozkładu
  dobre<-c()
  zle<-c()

  while(length(dobre) < n){
    U<-runif(n-length(dobre))
    Y<-r_g(n-length(dobre))

    zgodnosc<-(U <= f(Y)/(M*g(Y)) )
    dobre<-c(dobre,Y[zgodnosc])
    zle<-c(zle,Y[!zgodnosc])
  }
  hist(dobre, xlab="Wartości zmiennej losowej Y", col="aquamarine",
     main="Histogram zmiennej losowej Y",freq=F)
  curve(dbeta(x,2.7,6.3),from=0 , to=1 , add=T,col='blue', lwd=2 )
  return(length(zle)+length(dobre))
}
```

Z wykorzystaniem powyżego algorytmu generujemy 1000 punktów, które przedstawiamy na histogramie z naniesioną na niego krzywą gęstości rozkładu $\mathcal{B}(2.7, 6.3)$. Narysowany histogram dobrze przybliża krzywą gęstości f, zatem algorytm można stosować do symulowania wartości z zadanego rozkładu beta.

```
M1 = optimize(f=function(x){dbeta(x,2.7,6.3)},interval=c(0,1), maximum=T)$objective
X<-generuj2(1000, dunif, runif,M1)
```



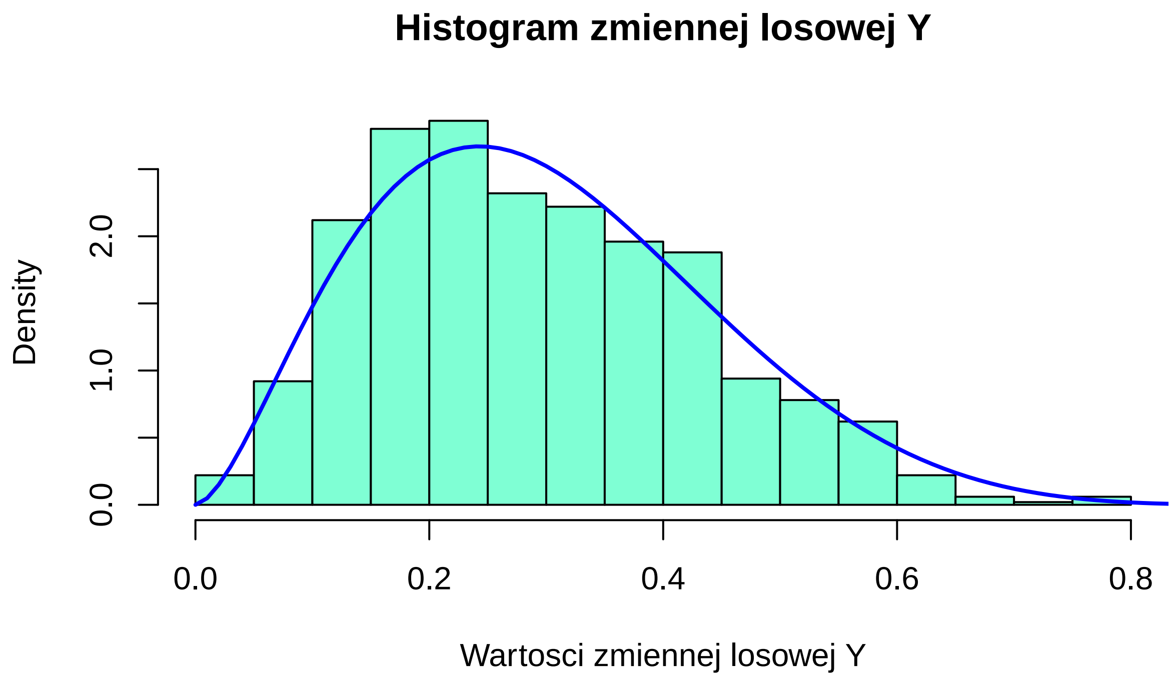
W przypadku gdy rozkładem instrumentalny jest rozkład jednostajny, do wylosowania 1000 punków z rozkładu $\mathcal{B}(2.7, 6.3)$ algorytm musiał rozpatrzeć 2622 punktów, z czego 1622 z nich algorytm odrzucił.

Rozkłd instrumentalny: $\mathcal{B}$(2,6)

Wartość M obliczamy jako $M = \max\{\frac{f(x)}{g(x)} : x \in [0, 1]\}$

Z wykorzystaniem algorytmu generujemy 1000 punktów, które przedstawiamy na histogramie z naniesioną na niego krzywą gęstości rozkładu $\mathcal{B}(2.7, 6.3)$. Histogram przybliża krzywą gęstości f, zatem algorytm można stosować do symulowania wartości z zadanego rozkładu beta.

```
M2 = optimize(f=function(x){dbeta(x,2.7,6.3)/dbeta(x,2,6)},interval=c(0,1),
              maximum=T)$objective
X<-generuj2(1000, function(x){dbeta(x,2,6)}, function(x){rbeta(x,2,6)},M2)
```



W tej metodzie również otrzymaliśmy symulację 1000 realizacji z rozkładu $\mathcal{B}(2.7, 6.3)$. Jest ona jednak znacznie bardziej optymalna niż w przypadku gdy rozkładem instrumentalnym był $\mathcal{U}(0,1)$, bowiem do wykonania zadania algorytm musiał wygenerować jedynia 1675 punktów.

Dobór optymalnego rozkładu instrumentalnego

Wykażmy formalnie, że aby stosunek $\frac{f(x)}{g(x)}$ był funkcją ograniczoną na $\mathbb{R}$, gdy f jest gęstością rozkładu $\mathcal{B}(\alpha, \beta)$, a g rozkładu $\mathcal{B}(a, b)$, muszą zachodzić nierówności $a \leq \alpha$ i $b \leq \beta$. Tak więc najlepszym wyborem, dla naturalnych a i b, jest $a = \lfloor \alpha \rfloor$ oraz $b = \lfloor \beta \rfloor$.

Jako gęstości rozkład beta, funkcje gęstości mają postać

$$f(x) = c_{\alpha,\beta} \cdot x^{\alpha-1}(1-x)^{\beta-1}$$

$$g(x) = c_{a,b} \cdot x^{a-1}(1-x)^{b-1}$$

$$gdzie\ stała \quad c_{A,B} = \frac{1}{\int_0^1 u^{A-1}(1-u)^{B-1}du}$$

$$\frac{f(x)}{g(x)} = \frac{c_{\alpha,\beta} \cdot x^{\alpha-1}(1-x)^{\beta-1}}{c_{a,b} \cdot x^{a-1}(1-x)^{b-1}} = \frac{c_{\alpha,\beta}}{c_{a,b}} \cdot x^{\alpha-a}(1-x)^{\beta-b} = C \cdot (x^{\alpha-a} - x^{\alpha-a+\beta-b})$$

funkcja ta jest ograniczona , gdy granice przy $x \to \infty$ i $x \to -\infty$ są skończone, co zachodzi gdy wykładniki $\alpha - a$ i $\beta - b$ są liczbami z przedziału$(0,1)$. Zatem musi być spełnione $a \leq \alpha$ i $b \leq \beta$.