

# Raport z badania metod generowania zmiennych losowych

Adam Łojowski

Marzena Rybak

Kwiecień 2022

# 1 Wstęp

Raport ten zawiera analizę metod generowania zmiennych losowych, które pojawiły się na wykładzie. Dodatkowo w raporcie został również krótko omówiony liniowy generator kongruentny ze względu na częste używanie liczb pseudolosowych podczas generowania zmiennych badanymi metodami. Celem poniższego raportu jest porównanie metod oraz empiryczne doświadczenie ich skuteczności na podstawie wykresów gęstości oraz dystrybuant.

## 2 Liniowy generator kongruentny

Poniższe metody generowanie liczb o zadanym rozkładzie korzystają z faktu, że dany jest rozkład jednostajny. Uzyskanie losowej liczby przez komputer jest niemożliwe-generatory charakteryzujące się największą losowością najczęściej wykorzystują dane z przyrody charakteryzujące się dużą zmiennością i trudne do przewidzenia, aczkolwiek są to liczby całkowicie zdeterminowane. Aby badać zjawiska losowe wystarczy jednak jedynie umiejętność tworzenia liczb pseudolosowych-liczb nie będących losowymi wartościami, ale posiadającymi większość pożądanych w badaniach cech. Jednym z najczęściej stosowanych sposobu generacji liczb z rozkładu jednostajnego jest metoda generowania liczb pseudolosowych korzystająca z arytmetyki modularnej. Przy odpowiednim dobraniu parametrów, uzyskujemy liczby charakteryzujące się wystarczającą losowością do dalszych operacji.

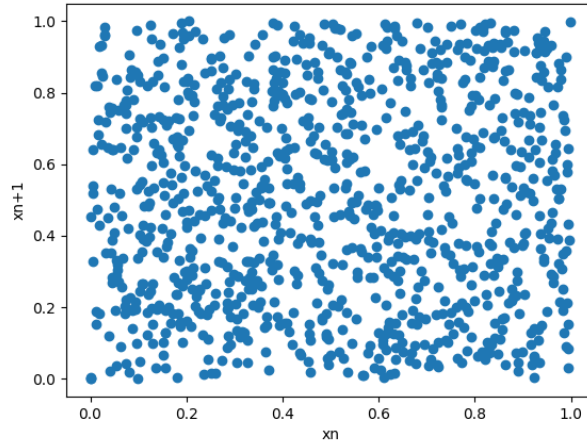
Metodę tą opisuje wzór

$$x_n = (ax_{n-1} + c) \bmod m,$$

gdzie  $a$ ,  $m$ ,  $c$  są liczbami całkowitymi, większymi od zera, takimi że  $a < m$ ,  $x_0 < m$ . Aby uzyskać liczby losowe z zakresu od 0 do 1 otrzymane wartości  $x_n$  należy podzielić przez  $m$ .

### 2.1 Przykład dobrze dobranych parametrów

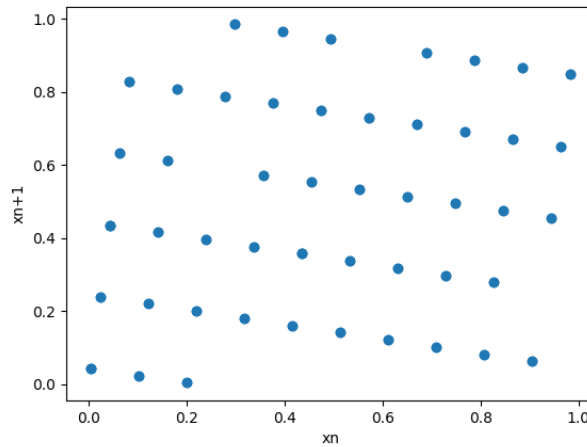
Przy dobrze dobranych parametrach nie widać bezpośrednio zależności między kolejnymi wartościami. Wykres porównujący kolejne wyrazy stworzonego ciągu liczb pseudolosowych wydaje się chaotyczny oraz nieprzewidywalny. Mimo pełnego determinizmu kolejnych liczb, liczby te zachowują wszystkie interesujące nas własności losowe.



Rysunek 1: Porównanie  $x_n$  i  $x_{n+1}$  dla parametrów  $x_0 = 0$ ,  $a = 234451$ ,  $m = 2^{64}$ ,  $c = 7$

## 2.2 Przykład źle dobranych parametrów

Dla źle dobranych parametrów da się zauważyć pewną regularność kolejnych punktów. Punkty na poniższym rysunku tworzą pewną dość regularną siatkę i dużo z nich nakłada się na siebie (rysunek został wygenerowany dla 1000 wartości). Sam fakt, że na rysunku nie widać regularność nie jest jednoznaczne z tym, że parametry są dobrze dobrane. Pewne regularności mogą być zauważalne przy badaniu trzech kolejnych wyrazów ciągu liczb pseudolosowych (wykres 3D) podczas, gdy w 2D żadne regularności nie są widoczne.

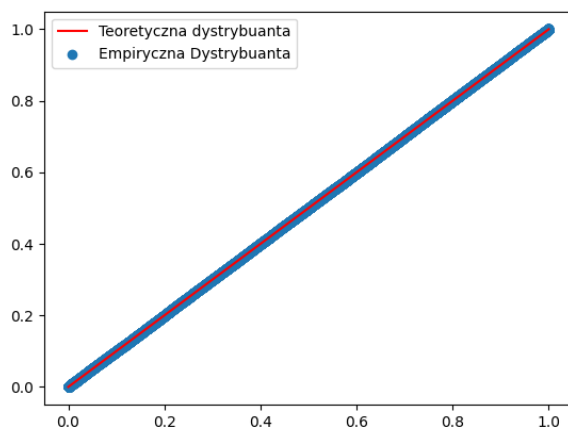


Rysunek 2: Porównanie  $x_n$  i  $x_{n+1}$  dla parametrów  $x_0 = 0$ ,  $a = 10$ ,  $m = 2^8 - 1$ ,  $c = 1$

## 2.3 Analiza

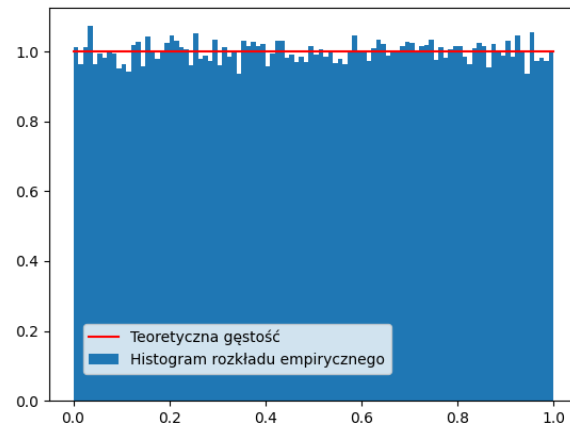
### 2.3.1 Porównanie otrzymanego rozkładu z teoretycznym rozkładem jednostajnym

Na poniższym rysunku nie da się zauważyć istotnych różnic w dystrybuancie teoretycznej i empirycznej co pozwala nam wnioskować, że wylosowane liczby faktycznie są z rozkładu jednostajnego.



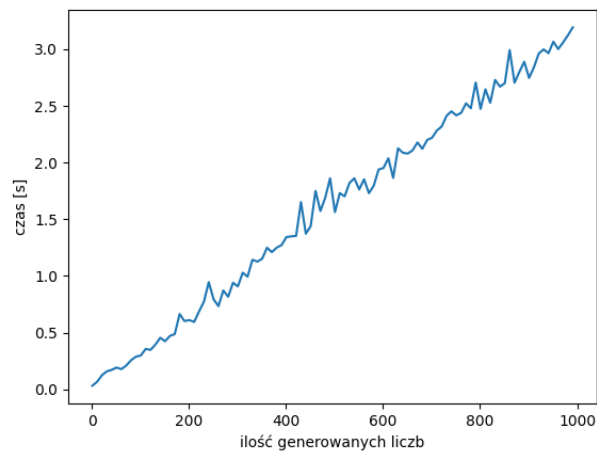
Rysunek 3: Porównanie dystrybuanty teoretycznej rozkładu jednostajnego oraz empirycznej dla parametrów  $x_0 = 0$ ,  $a = 234451$ ,  $m = 2^{64}$ ,  $c = 7$

Również przy badaniu gęstości empirycznej nie widać znaczących różnic w porównaniu z gęstością teoretyczną. Dla większej ilości wylosowanych liczb lub szerszych słupków histogramu otrzymalibyśmy wartości bardziej zbliżone do wartości teoretycznych niż na rysunku 4, jednak zbliżenie do oczekiwanej wartości na rysunku 4 jest wystarczające, aby przypuszczać, że dane empiryczne są z rozkładu jednostajnego.

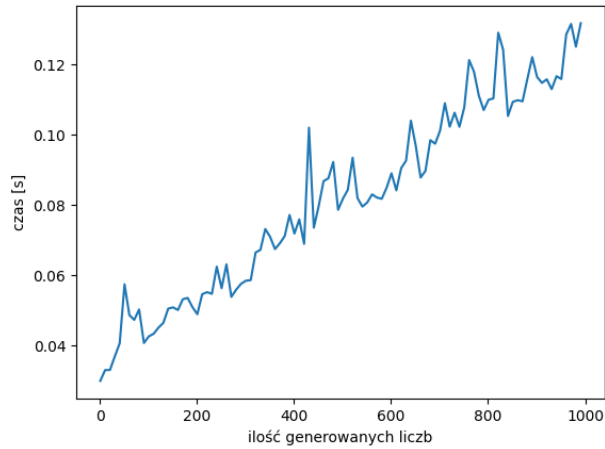


Rysunek 4: Porównanie gęstości teoretycznej rozkładu normalnego z histogramem otrzymanych liczb pseudolosowych dla parametrów  $x_0 = 0$ ,  $a = 234451$ ,  $m = 2^{64}$ ,  $c = 7$

### 2.3.2 Czas działania



Rysunek 5: Czas działania zaimplementowanego generatora liczb dla 10000 powtórzeń



Rysunek 6: Czas działania wbudowanego generatora liczb w Pythonie-*numpy.random.uniform* dla 10000 powtórzeń

Porównując powyższe wykresy otrzymujemy, że funkcje wbudowane są około 20 razy lepsze. Może to wynikać z faktu korzystania z innej metody lub lepszej optymalizacji na poziomie implementacji biblioteki *numpy* bazującej na kodzie w C, który jest znacząco szybszym językiem w porównaniu z Pythonem.

### 3 Metoda odwracania dystrybucji

Metodą odwracania dystrybucji możemy generować zmienne o danej dystrybucji, którą jesteśmy w stanie odwrócić. Możemy ją stosować zarówno do rozkładów dyskretnych jak i ciągłych. Poniżej algorytmy dla rozkładu dyskretnego i ciągłego.

#### Algorytm dla rozkładu dyskretnego

- Generujemy  $U \sim U(0, 1)$ .
- Wyznaczamy  $j \in \mathbb{N}$  takie, że  $\sum_{i=1}^{j-1} p_i < U \leq \sum_{i=1}^j p_i$ .
- Zwracamy  $x = x_j$ .

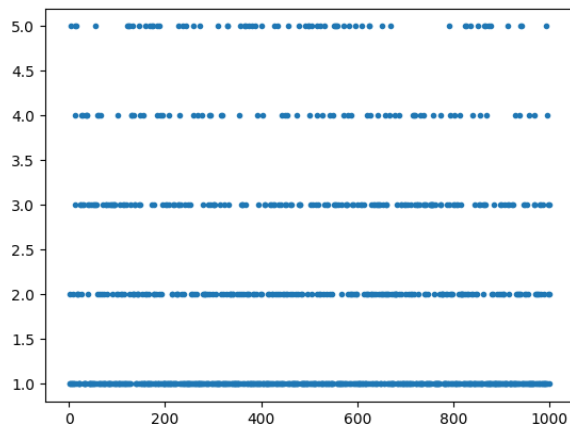
#### Algorytm dla rozkładu ciągłego

- Generujemy  $U \sim U(0, 1)$ .
- Wstawiamy  $X = F_X^{-1}(U)$ .

## Przykłady

### Rozkład dyskretny

Pierwszym przykładem dla metody odwracania dystrybuanty jest dyskretna zmienna losowa o rozkładzie:  $P(X = 1) = 1/2$ ,  $P(X = 2) = 1/5$ ,  $P(X = 3) = 1/6$ ,  $P(X = 4) = 1/15$ ,  $P(X = 5) = 1/15$ . Poniżej kolejne wyniki realizacji tej zmiennej metodą odwracania dystrybuanty.



Rysunek 7: Realizacje dyskretnej zmiennej losowej  $X$

### Rozkład ciągły

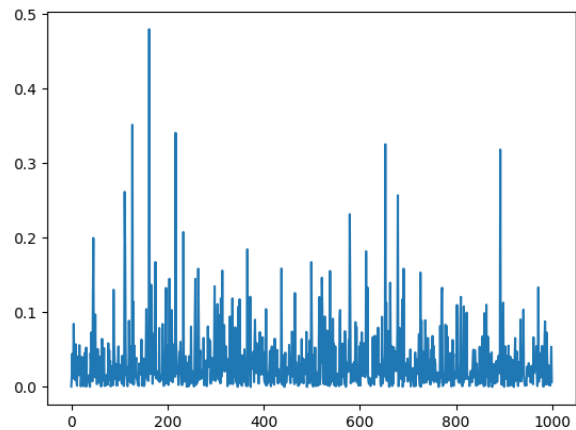
Jako przykład rozkładu ciągłego weźmiemy rozkład Pareto z parametrami  $\alpha > 0$  i  $\lambda > 0$ , którego dystrybucja dla  $x > 0$  opisana jest poniższym wzorem.

$$F(x) = 1 - \left( \frac{\lambda}{\lambda + x} \right)^\alpha.$$

Odwracając dystrybuantę rozkładu Pareto otrzymujemy funkcję odwrotną

$$F^{-1}(x) = \left( \frac{1}{\lambda - x} \right)^{\frac{\lambda}{\alpha} - 1}.$$

Poniżej realizację zmiennej z rozkładu Pareto z parametrami  $\alpha = 4$  i  $\lambda = 0.1$ .

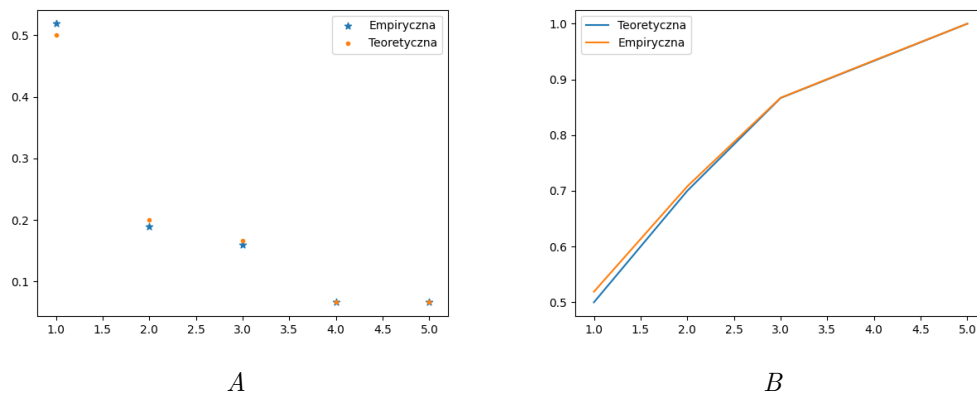


Rysunek 8: Realizacje zmiennej losowej z rozkładu Pareto z parametrami  $\alpha = 4$  i  $\lambda = 0.1$

## Analiza otrzymanych wyników

Analiza otrzymanych wyników została podzielona w zależności od typu dyskretnego lub ciągłego zmiennej losowej  $X$ .

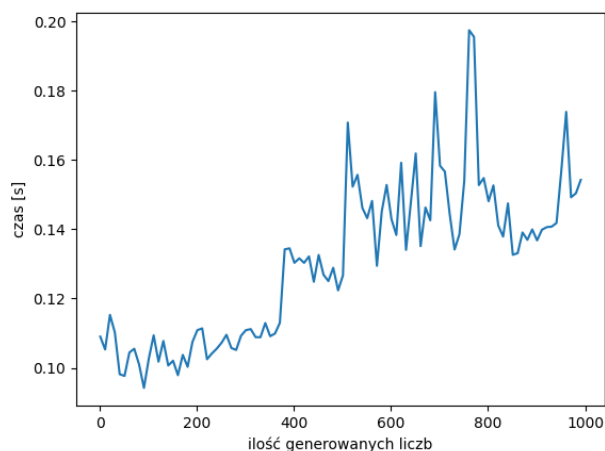
### Rozkład dyskretny



Rysunek 9: Na części A zaznaczone są prawdopodobieństwa teoretyczne oraz empiryczne otrzymania konkretnych wartości zmiennej losowej  $X$ , natomiast w części B narysowana jest dystrybuanta teoretyczna oraz empiryczna zmiennej losowej  $X$

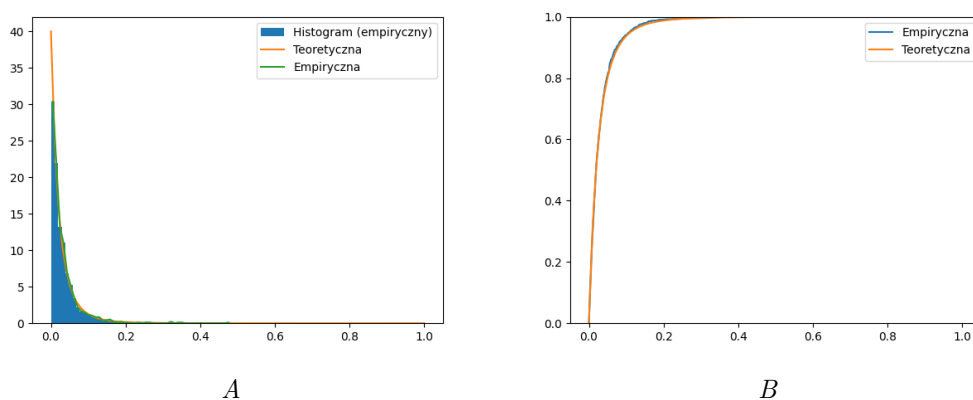


Tak jak na powyższych rysunkach widać, różnice między wartościami teoretycznymi oraz empirycznymi są niewielkie. Co daje nam empiryczne potwierdzenie skuteczności metody odwrotnej dystrybuanty.

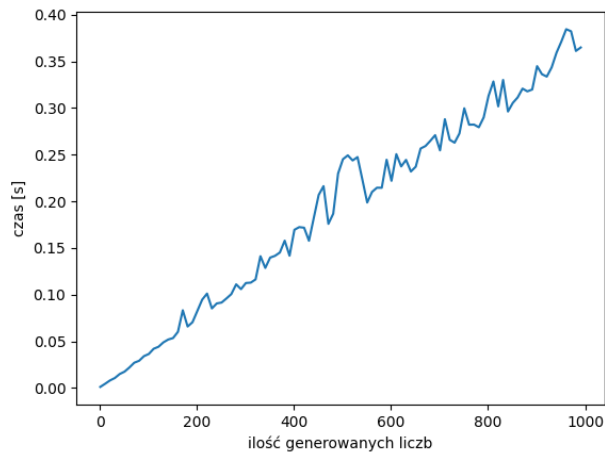


Rysunek 10: Czas działania zaimplementowanej generatora zmiennej losowej  $X$  dla 1000 powtórzeń

### Rozkład ciągły



Rysunek 11: Na części A zaznaczone są gęstości teoretyczna oraz empiryczna wraz z histogramem wygenerowanych zmiennych losowych z rozkładu Pareto, natomiast na części B narysowane są dystrybuanty empiryczna oraz teoretyczna dla tych zmiennych losowych

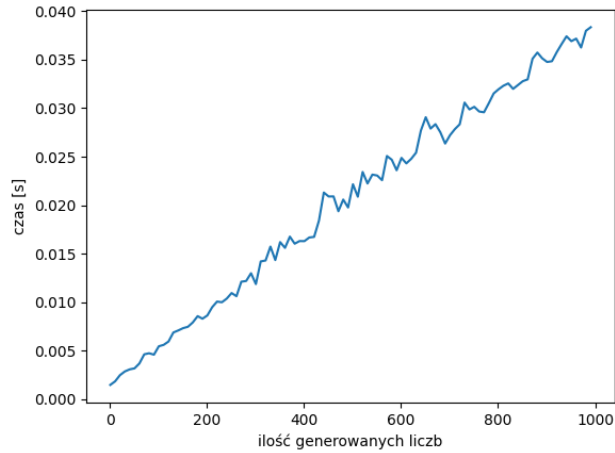


Rysunek 12: Czas działania zaimplementowanej generatora zmiennej losowej z rozkładu Pareto dla 1000 powtórzeń

Wbudowana funkcja generująca rozkład Pareto posługuje się inną jego definicją, gdzie gęstość jest określona równaniem

$$p(x) = \frac{am^a}{x^{a+1}},$$

gdzie  $a$  jest parametrem kształtu, a  $m$  parametrem skali. Do wykonania wykresu czasu przyjęte zostały parametry  $a = 2$  oraz  $m = 1$ . Mimo różnic w definicji jak i parametrach czas generowania zmiennych można uznać za porównywalny ze względu na podobną złożoność problematyki obu rozkładów.



Rysunek 13: Czas działania wbudowanego generatora zmiennej losowej z rozkładu Pareto dla 1000 powtórzeń przy użyciu funkcji wbudowanej `numpy.random.pareto`

Z powyższych wykresów możemy wywnioskować, że zaimplementowana metoda generowania zmiennych jest 10 razy wolniejsza. Przejdźmy do następnej metody sprawdzając, czy podobny wynik się powtórzy.

## 4 Metoda akceptacji-odrzućenia

### Algorytm dla rozkładu dyskretnego

- Generujemy jedną realizację  $Y$  ( $Y$  to zmienna losowa o rozkładzie  $q_i = P(Y = i)$ ,  $\sum_i q_i = 1$ ,  $q_i \geq 0$ ).
- Generujemy  $U \sim U(0, 1)$ ,  $U$  jest niezależne od  $Y$ .
- Jeśli  $U \leq \frac{p_y}{c \cdot q_y}$  to zwracamy  $X = Y$ , w przeciwnym przypadku powtarzamy procedurę ( $c$  jest stałą taką, że  $0 < c < \infty$  oraz  $\sup_{x \in \mathbb{R}} \frac{f(x)}{g(x)} \leq c$ ).

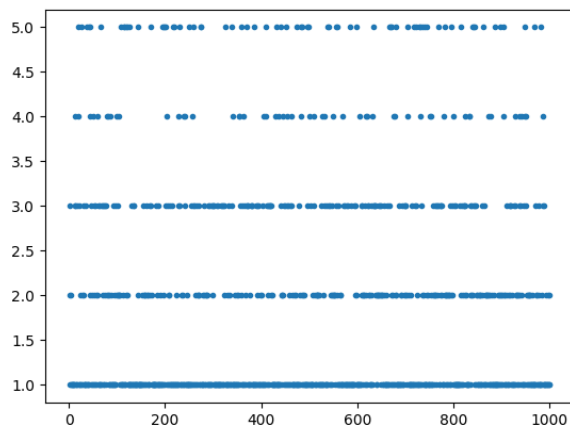
### Algorytm dla rozkładu ciągłego

- Generujemy  $Y$ .
- Generujemy  $U \sim U(0, 1)$ ,  $U$  jest niezależne od  $Y$ .
- Jeśli  $U \leq \frac{f(Y)}{c \cdot g(Y)}$  to zwracamy  $X = Y$ , w przeciwnym przypadku powtarzamy procedurę ( $c$  jest stałą taką, że  $0 < c < \infty$  oraz  $\max_i \frac{p_i}{q_i} \leq c$ ).

## Przykłady

### Rozkład dyskretny

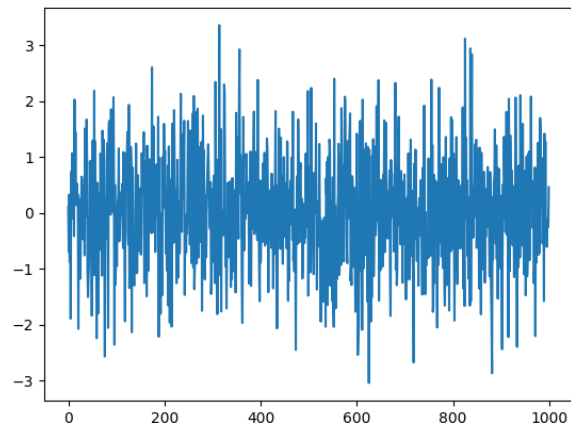
Przykład dla metody akceptacji-odrzućenia jest dyskretna zmienna losowa o rozkładzie takim samym jak w przypadku metody odwrotnej dystrybucyjnej:  $P(X = 1) = 1/2$ ,  $P(X = 2) = 1/5$ ,  $P(X = 3) = 1/6$ ,  $P(X = 4) = 1/15$ ,  $P(X = 5) = 1/15$ . Poniżej kolejne wyniki realizacji tej zmiennej metodą akceptacji-odrzućenia.



Rysunek 14: Realizacje dyskretnej zmiennej losowej  $X$

### Rozkład ciągły

Poniższy przykład dotyczy generowania zmiennej losowej z rozkładu normalnego używając funkcji pomocniczej z rozkładu wykładniczego.

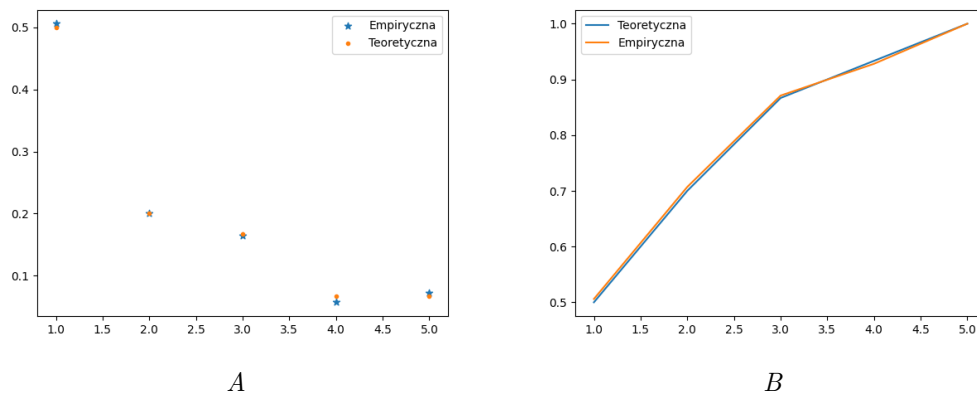


Rysunek 15: Realizacje dyskretnej zmiennej losowej  $X$

## Analiza otrzymanych wyników

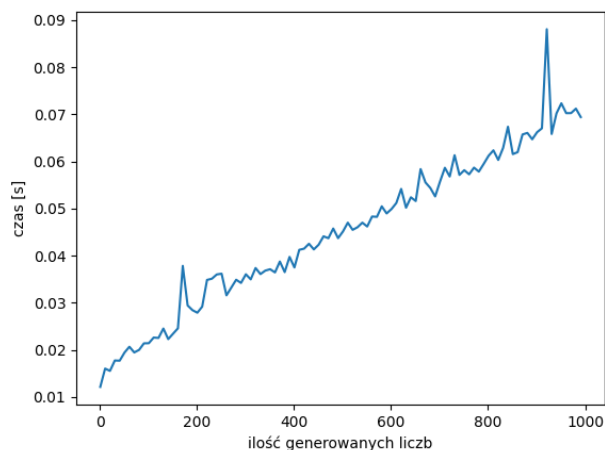
Analiza otrzymanych wyników została podzielona w zależności od typu dyskretnego lub ciągłego zmiennej losowej  $X$ .

### Rozkład dyskretny



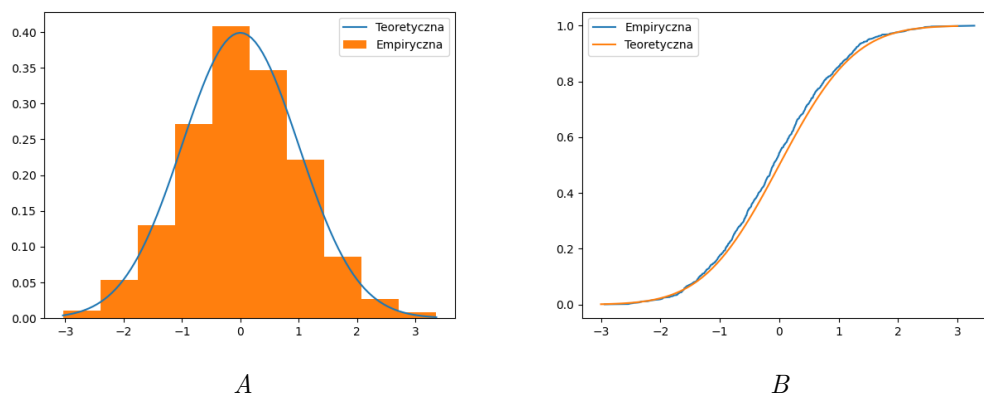
Rysunek 16: Na części A zaznaczone są prawdopodobieństwa teoretyczne oraz empiryczne otrzymane konkretnych wartości zmiennej losowej  $X$ , natomiast w części B narysowana jest dystrybuanta teoretyczna oraz empiryczna zmiennej losowej  $X$

Na rysunkach powyższym rysunku zarówno w części A jak i B nie widać większych różnic wartości empirycznej w stosunku do teoretycznej, co empirycznie potwierdza skuteczność również tej metody.



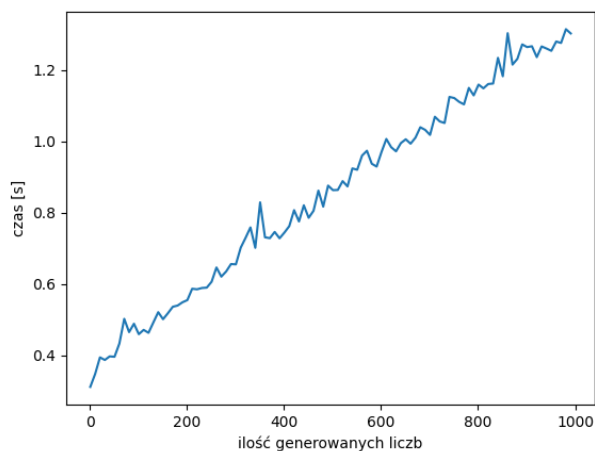
Rysunek 17: Czas działania zaimplementowanej generatora zmiennej losowej  $X$  dla 1000 powtórzeń

### Rozkład ciągły



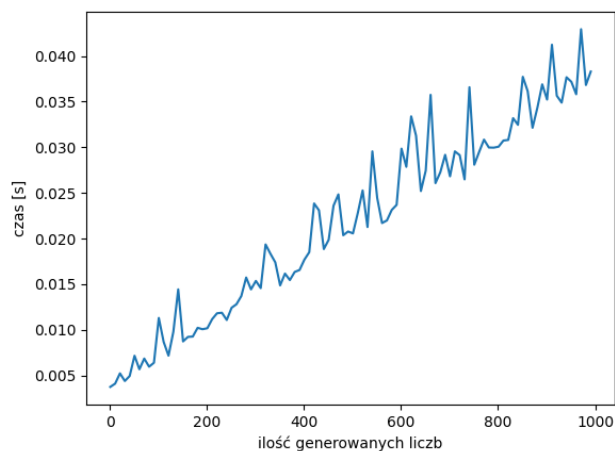
Rysunek 18: Na części A zaznaczone są gęstości teoretyczna oraz empiryczna w postaci histogramu, natomiast w części B narysowana jest dystrybucja teoretyczna oraz empiryczna zmiennej losowej z rozkładu normalnego

Czas działania zaimplementowanego generatora zmiennych losowych z rozkładu normalnego przy użyciu metody akceptacji-odrzućcia prezentują się następująco.



Rysunek 19: Czas działania zaimplementowanej generatora zmiennej losowej z rozkładu normalnego dla 1000 powtórzeń

Dla porównania poniżej czas generowania zmiennych losowych z rozkładu normalnego dla 1000 powtórzeń każdej liczebności wykonanych za pomocą funkcji wbudowanej.



Rysunek 20: Czas działania wbudowanego generatora zmiennej losowej z rozkładu normalnego dla 1000 powtórzeń przy użyciu funkcji wbudowanej *numpy.random.normal*

W tym przypadku otrzymujemy dużo gorszy wynik funkcji zaimplementowanej. Jest ona aż 30 razy wolniejsza, co pozwala nam przypuszczać niską optymalizację zaimplementowanego kodu lub dużo efektywniejszą metodę generowania zmiennych użytą w funkcji wbudowanej.

## 5 Metoda splotowa

Metoda splotowa swoją nazwę zawdzięcza własności, że gęstość sumy zmiennych losowych oraz w łatwy sposób potrafimy generować składniki tej sumy zmiennych losowych jest ich splotem. Jeśli znamy zmienne, których suma składa się na szukaną przez nas zmienną jest to bardzo efektywny sposób generowania zmiennych losowych. Problemem tej metody jest znalezienie zmiennych pomocniczych, które może być trudne w wykonaniu.

### Algorytm dla rozkładu dyskretnego oraz ciągłego

- Generujemy zmienne niezależne  $Y_1, \dots, Y_n$ .
- Wstawiamy  $X = Y_1 + \dots + Y_n$ .

### Przykłady

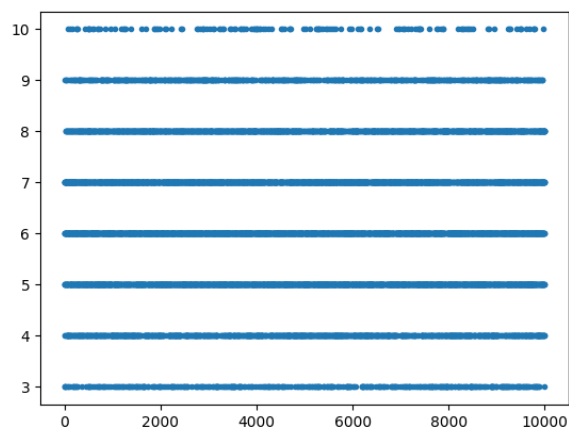
#### Rozkład dyskretny

Jako przykład dyskretny użyto splotu następujących rozkładów:

- $Y_1: P(Y_1 = 1, 2, 3) = \frac{1}{3}$ ;
- $Y_2: P(Y_2 = 1) = \frac{1}{2}, P(Y_2 = 2) = \frac{1}{3}, P(Y_2 = 3) = \frac{1}{6}$ ;
- $Y_3: P(Y_3 = 1, 2, 3, 4) = \frac{1}{4}$ .

Splot zmiennych  $Y_1, Y_2$  i  $Y_3$  można opisać za pomocą tabeli poniższej tabeli.

$i$	3	4	5	6	7	8	9	10
$P(X = i)$	$\frac{1}{24}$	$\frac{1}{9}$	$\frac{7}{36}$	$\frac{17}{72}$	$\frac{5}{24}$	$\frac{5}{36}$	$\frac{1}{18}$	$\frac{1}{72}$



Rysunek 21: Realizacje dyskretnej zmiennej losowej  $X$



### Rozkład ciągły

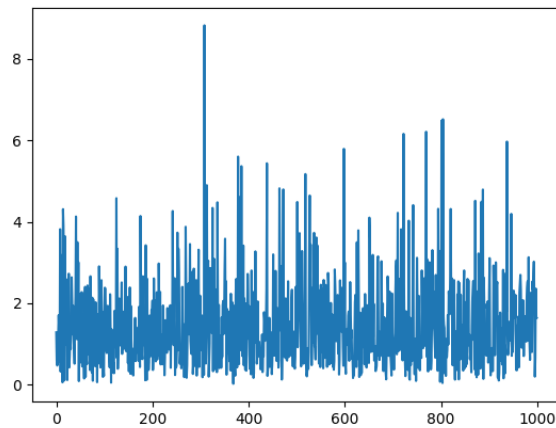
Jako przykład rozkładu ciągłego wykorzystamy rozkład zmiennej losowej będącej sumą zmiennych  $Y_1 \sim \exp(\lambda_1)$  oraz  $Y_2 \sim \exp(\lambda_2)$ . Wykorzystując, fakt, że gęstość zmiennej  $X$  jest splotem gęstości zmiennych  $Y_1$  i  $Y_2$  otrzymujemy wzór na gęstość zmiennej  $X$

$$f_X(z) = \int_{-\infty}^{\infty} f_{Y_1}(z-x)f_{Y_2}(x) \, dx = \int_0^z \lambda_1 \cdot e^{-\lambda_1(z-x)} \cdot \lambda_2 \cdot e^{-\lambda_1 x} \, dx = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2} (e^{-\lambda_2 z} - e^{-\lambda_1 z}).$$

Po kolejnych obliczeniach otrzymujemy również wzór na dystrybuantę zmiennej  $X$

$$F(z) = \frac{-\lambda_1 e^{-\lambda_2 z} + \lambda_2 e^{-\lambda_1 z} - \lambda_2 + \lambda_1}{\lambda_1 - \lambda_2}.$$

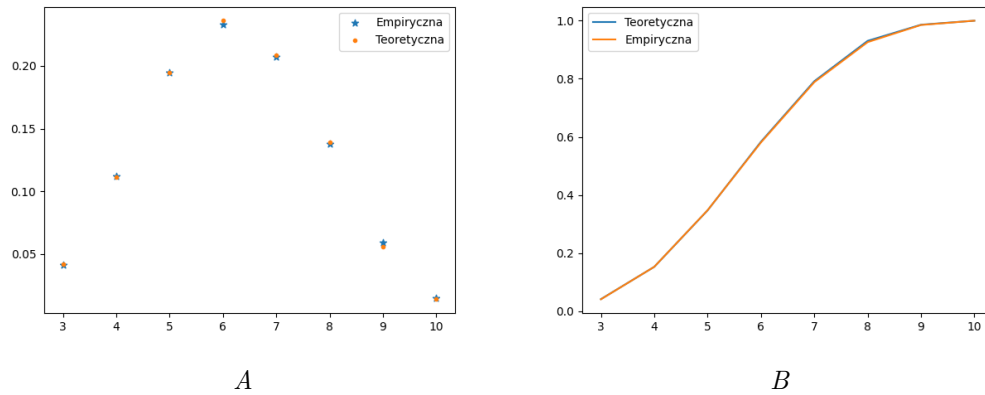
Poniższy wykres wygenerowanych zmiennych losowych został stworzony dla parametrów  $\lambda_1 = 1$  oraz  $\lambda_2 = 2$ . Do generacji liczb z rozkładu wykładniczego użyto funkcji wbudowanej `torch.distributions.exponential.Exponential`.



Rysunek 22: Realizacje zmiennej losowej  $X$

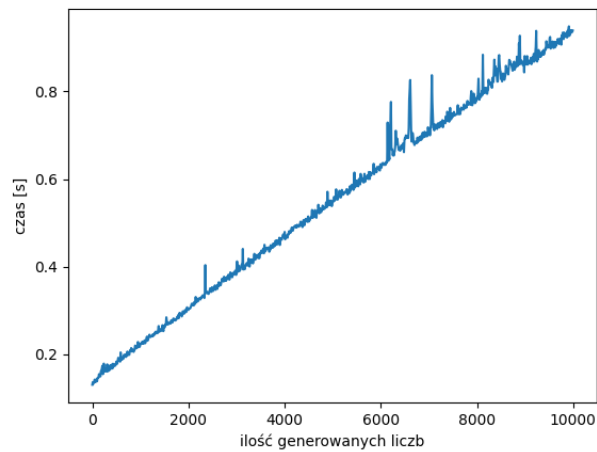
## Analiza otrzymanych wyników

### Rozkład dyskretny



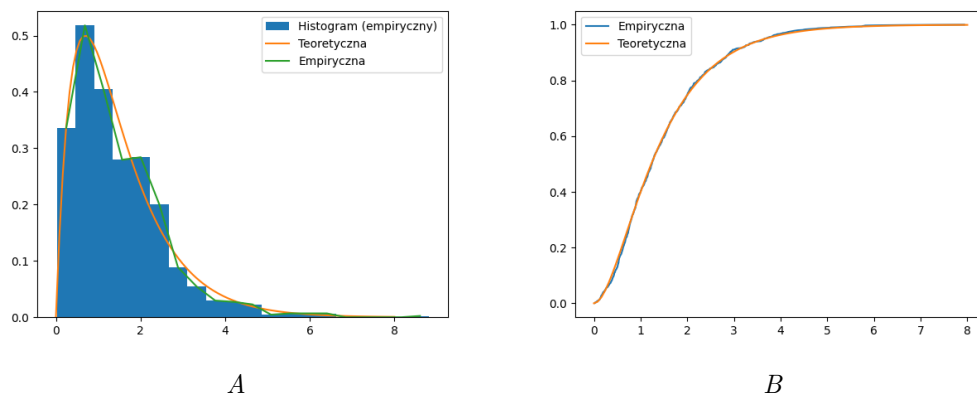
Rysunek 23: Na części A zaznaczone są prawdopodobieństwa teoretyczne oraz empiryczne otrzymania konkretnych wartości zmiennej losowej  $X$ , natomiast w części B narysowana jest dystrybucja teoretyczna oraz empiryczna zmiennej losowej  $X$

Na rysunkach powyższym rysunku zarówno w części A jak i B nie widać większych różnic wartości empirycznej w stosunku do teoretycznej, co empirycznie potwierdza skuteczność metody spłotowej.



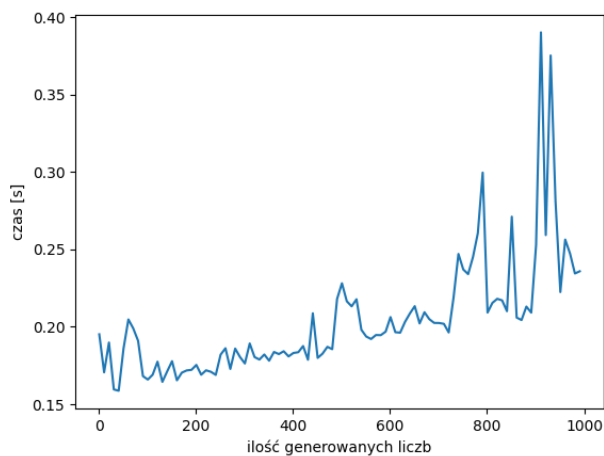
Rysunek 24: Czas działania zaimplementowanej generatora zmiennej losowej  $X$  dla 1000 powtórzeń

## Rozkład ciągły



Rysunek 25: Na części A zaznaczone są gęstości teoretyczna oraz empiryczna wraz z histogramem, natomiast w części B narysowana jest dystrybuanta teoretyczna oraz empiryczna zmiennej losowej z rozkładu będącego splotem dwóch rozkładów wykładniczych

Tak jak na powyższym rysunku widać wartości empiryczne zgadzają się z teoretycznymi. Różnice w wartościach są niewielkie. Poniżej wykres czasu działania zaimplementowanego generatora zbadanej zmiennej losowej.



Rysunek 26: Czas działania zaimplementowanej generatora zmiennej losowej dla 1000 powtórzeń

## 6 Metoda Boxa-Mullera

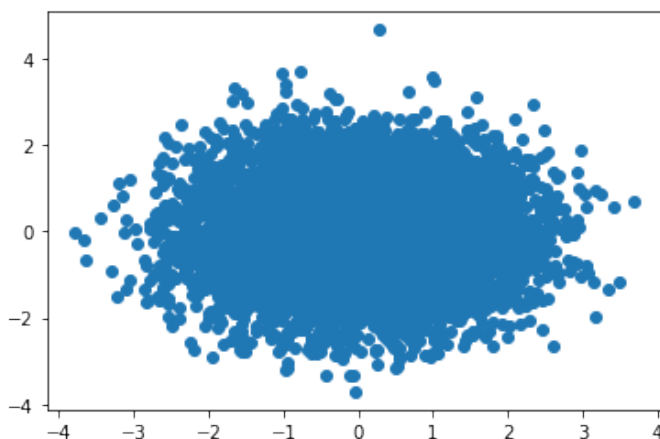
Transformacja Boxa-Mullera jest metodą generowania liczb losowych o rozkładzie normalnym, na podstawie dwóch wartości zmiennej o rozkładzie jednostajnym z przedziału  $(0, 1]$ . Wygenerowane za pomocą tej metody zmienne losowe  $X$  oraz  $Y$  są z rozkładu normalnego o odchyleniu standardowym równym 1 oraz są również niezależne od siebie.

### Algorytm

- Generujemy  $U_1, U_2$  -iid,  $U_i \sim U(0, 1)$  takie że  $U_1 \perp U_2$ .
- Wstawiamy  $X = \sqrt{-2 \ln U_1} \cos(2\pi \cdot U_2)$  oraz  $Y = \sqrt{-2 \ln U_1} \sin(2\pi \cdot U_2)$ .

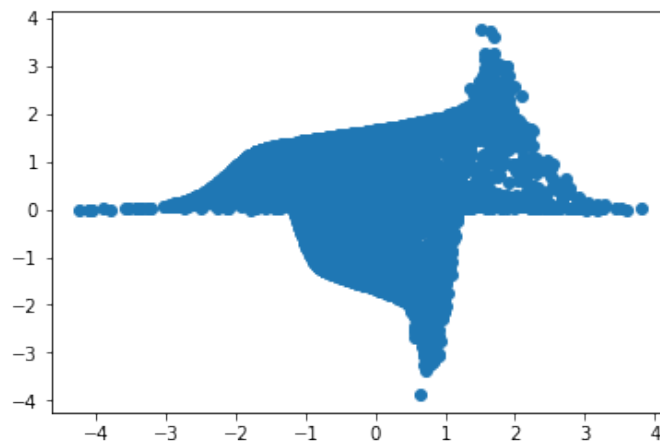
### Przykłady

Poprawnie zaaplikowana metoda Boxa-Mullera generuje dwie zmienne  $X$  i  $Y$ , które są od siebie niezależne. Dlatego też dla wygenerowanych wartości, punkty o współrzędnych  $(X, Y)$  powinny rozrzucać się po wykresie w sposób losowy.



Rysunek 27: Wykres punktowy dla współrzędnych  $(X, Y)$

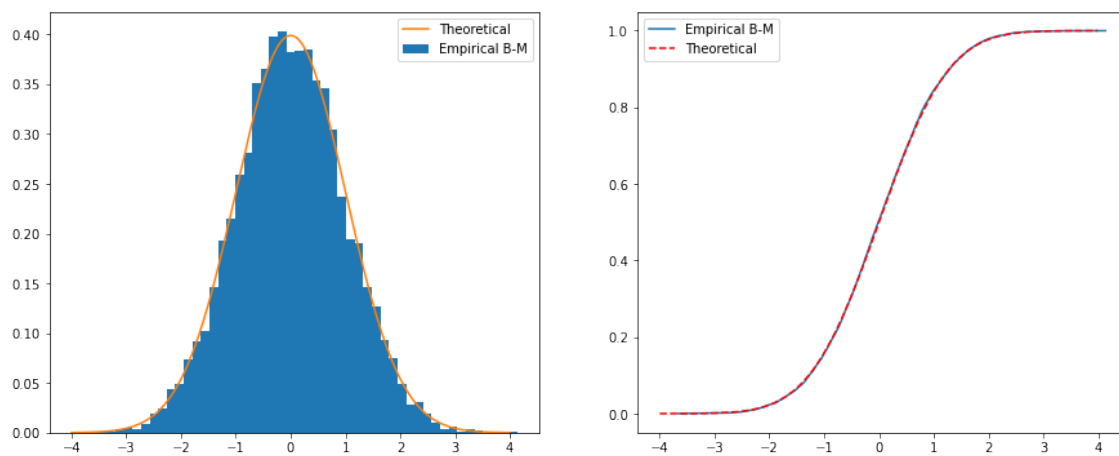
Nieuważny student mógłby jednak popełnić drobny błąd generując zmienne ze wzoru  $X = \sqrt{-2 \ln U_1} \cos(2\pi U_2)$  oraz  $Y = \sqrt{-2 \ln U_2} \sin(2\pi U_1)$ . Proszę zwrócić uwagę na to, gdzie we wzorze występują zmienne  $U_1$  oraz  $U_2$ ! Okazuje się, że ta drobna zmiana nie wpływa szczególnie na otrzymane wykresy gęstości oraz dystrybuant dla obu zmiennych losowych  $X, Y$  jednak wykres punktowy dla współrzędnych  $(X, Y)$  pokazuje, że dla tak zadanego wzoru, zmienne te nie są już niezależne od siebie. Wyszukanie takiego błędu może być trudne, więc powinno się zwracać uwagę na najmniejsze niuanse w trakcie aplikowania metod generowania zmiennych losowych.



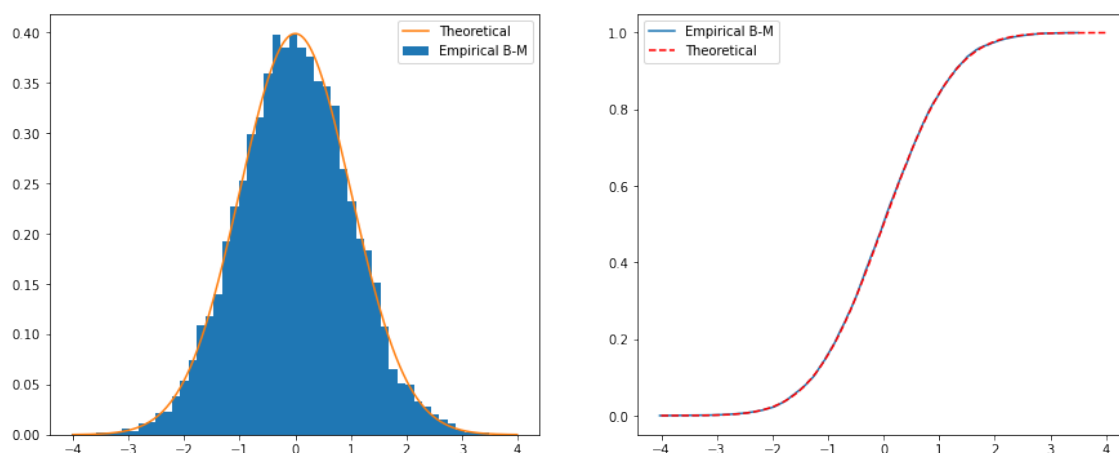
Rysunek 28: Wykres punktowy dla współrzędnych  $(X, Y)$  zadane błędnym wzorem. Punkty układają się w kształt krzyża, co daje nam powody do myślenia, że zmienne  $X, Y$  nie są już od siebie niezależne.

### Analiza otrzymanych wyników

Wykresy histogramów oraz dystrybuant dla zmiennych losowych  $X, Y$  pokrywają się w znacznym stopniu z teoretycznymi wyliczeniami.

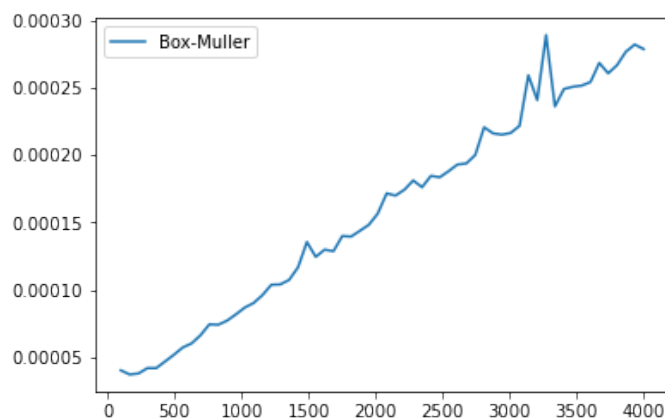


Rysunek 29: Gęstość zmiennej losowej  $X$  oraz jej dystrybuanta wraz z wykresami teoretycznymi

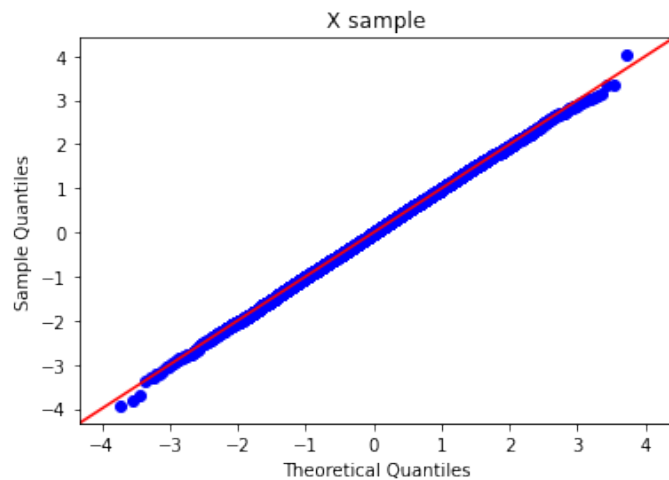


Rysunek 30: Gęstość zmiennej losowej  $Y$  oraz jej dystrybuanta wraz z wykresami teoretycznymi

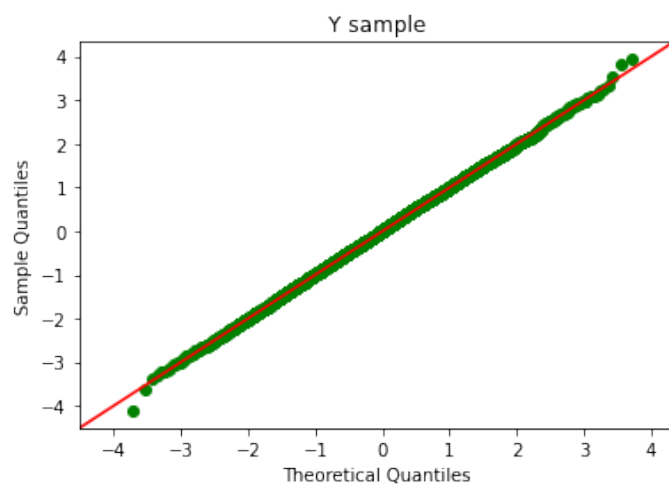
Warto mieć na uwadze fakt, że otrzymamy praktycznie tak samo wyglądające wykresy w przypadku, kiedy popełnimy błąd we wzorze, który został omówiony powyżej.



Rysunek 31: Czas działania metody Boxa-Mullera w zależności od ilości generowanych wartości.



Rysunek 32: QQplot zmiennej losowej X dla metody Boxa-Mullera.



Rysunek 33: QQplot zmiennej losowej Y dla metody Boxa-Mullera.

## 7 Metoda biegunowa

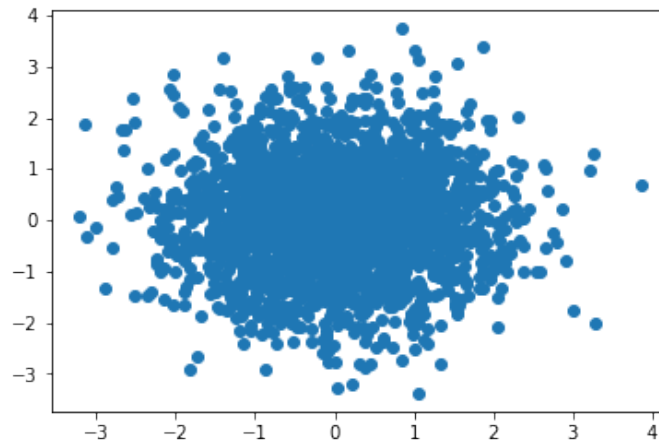
Metoda biegunowa, tak jak metoda Boxa-Mullera, jest używana do generowania par liczb losowych o rozkładzie normalnym. Postać biegunowa generuje niezależne zmienne  $V_1$  oraz  $V_2$  z rozkładu jednostajnego, które traktowane jako współrzędne  $(V_1, V_2)$  muszą znaleźć się w okręgu jednostkowym. Tak wygenerowane zmienne algorytm przekształca na niezależne od siebie zmienne z rozkładu normalnego.

## Algorytm

- Generujemy  $V_1 \sim U(-1, 1)$  oraz  $V_2 \sim U(-1, 1)$  takie że  $V_1 \perp V_2$ .
- Wyznaczamy  $R^2 = V_1^2 + V_2^2$ .
- Jeśli  $R^2 \geq 1$ , wracamy do kroku 1.
- Wstawiamy  $X = \sqrt{\frac{-2 \ln R^2}{R^2}} V_1$ ,  $Y = \sqrt{\frac{-2 \ln R^2}{R^2}} V_2$ .

## Przykłady

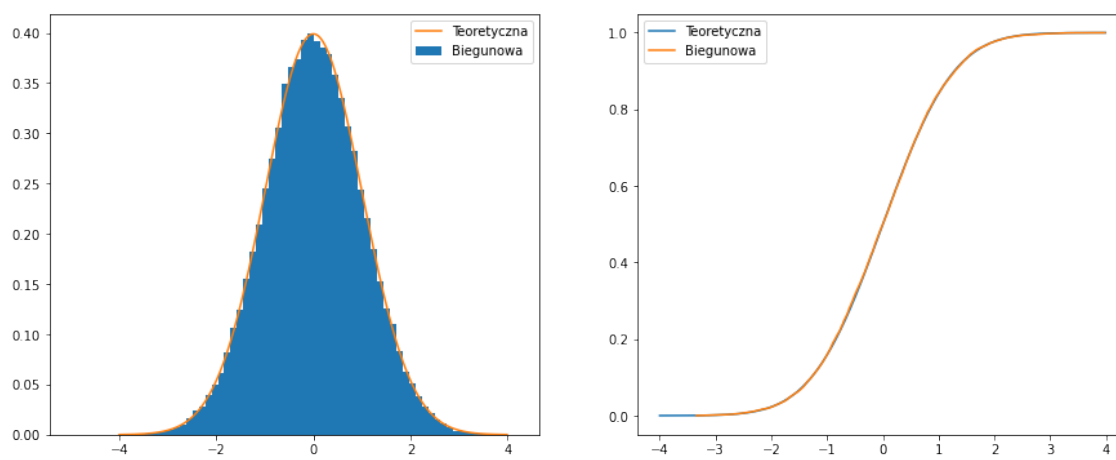
Poniższy wykres punktowy przedstawia rozłożenie punktów o współrzędnych  $(X, Y)$ . Jak możemy zauważyć, punkty te są rozłożone w sposób chaotyczny i wygląda bardzo podobnie do tego, który był przedstawiony w punkcie poświęconym metodzie Boxa-Mullera.



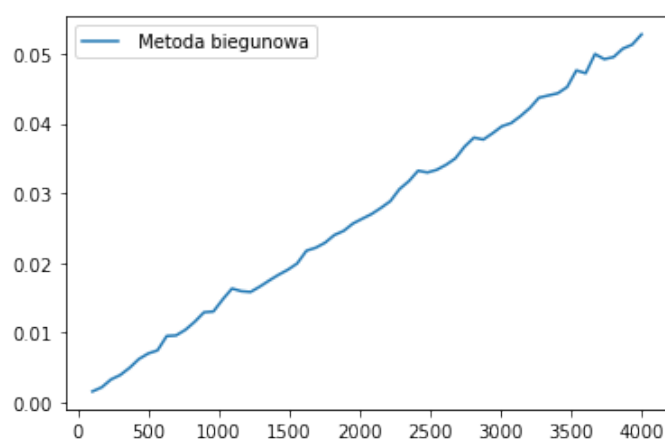
Rysunek 34: Wykres punktowy współrzędnych  $(X, Y)$ .



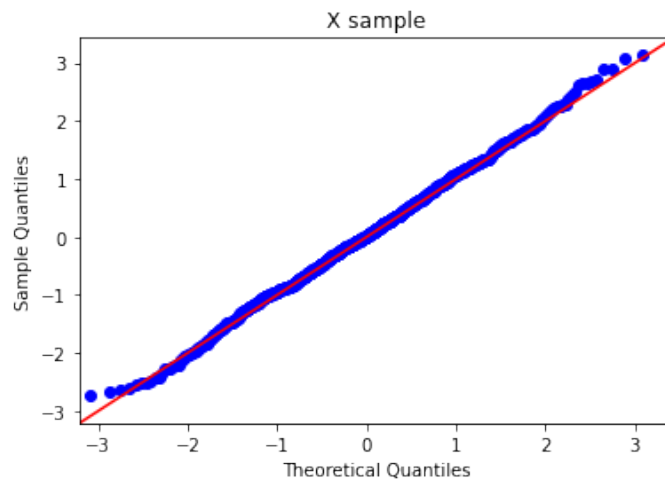
## Analiza otrzymanych wyników



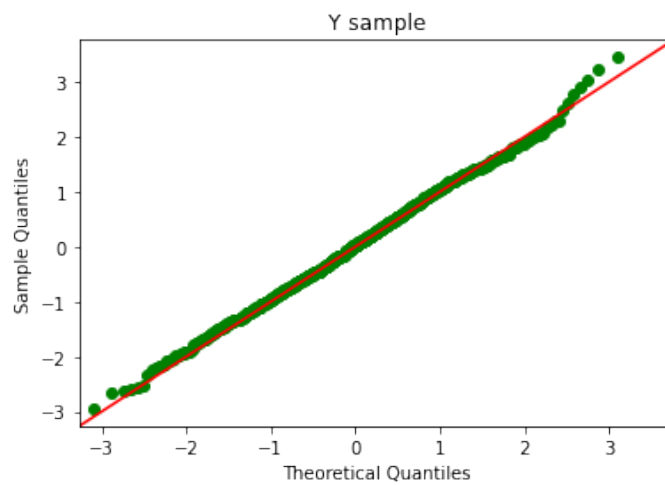
Rysunek 35: Histogram wyników metody Biegunowej wraz z zaznaczoną wartością teoretyczną



Rysunek 36: Czas działania metody Biegunowej w zależności od ilości generowanych wartości

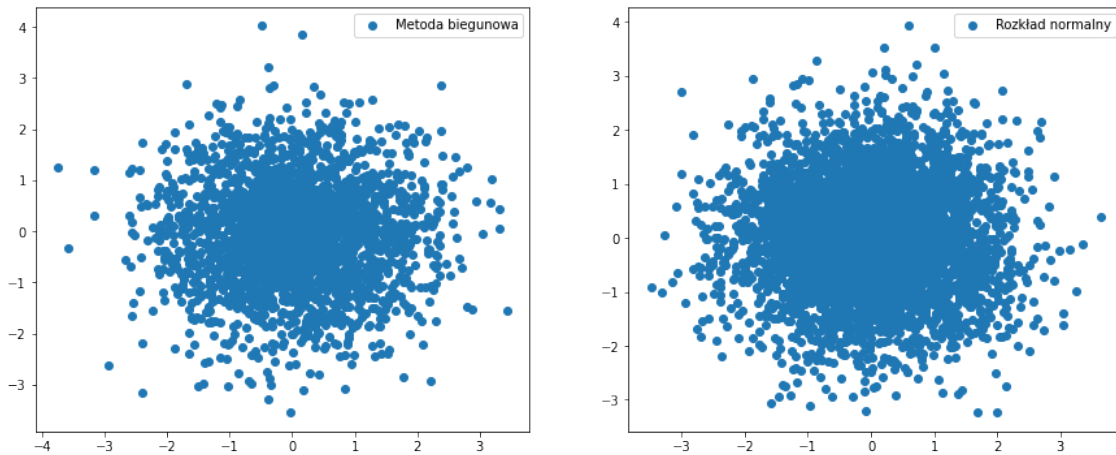


Rysunek 37: QQplot zmiennej losowej X dla metody biegunowej



Rysunek 38: QQplot zmiennej losowej Y dla metody biegunowej

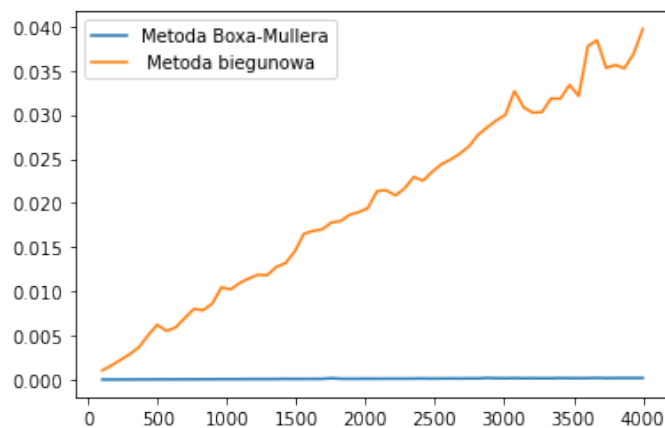
Przyjrzyjmy się wykresom punktowym dla wskazanej metody i porównajmy ją z tym, który jest zadany rozkładem normalnym, który ten właśnie rozkład ma generować.



Rysunek 39: Wykresy punktowe współrzędnych  $(X, Y)$  dla metody biegunowej oraz niezależnych par punktów zadanych rozkładem normalnym z parametrami  $\mu = 0, \sigma = 1$ . W obydwu przypadkach zostało wygenerowanych 5000 zmiennych.

Możemy zauważyć podobne zagęszczenie punktów szczególnie w okolicach początku układu współrzędnych, co daje nam kolejne dowody, że metoda biegunowa generuje zmienne losowe z rozkładu normalnego.

### Porównanie czasów działania metody Boxa-Mullera oraz metody biegunowej



Rysunek 40: Porównanie czasu działania metody biegunowej oraz Boxa-Mullera

Na przedstawionym wykresie możemy wyciągnąć jasny wniosek, że metoda Boxa-Mullera góruje nad metodą Biegunową. Tak ogromna rozbieżność czasów działania algorytmu może być wynikiem

tego, że w metodzie Biegunowej w trzecim kroku musimy użyć warunku  $if$ , co znacząco wydłuża czas działania kodu.

## 8 Metoda mieszaniny rozkładów

### Algorytm

Algorytm ten zakłada, że jesteśmy w stanie efektywnie generować  $Y_i$ .

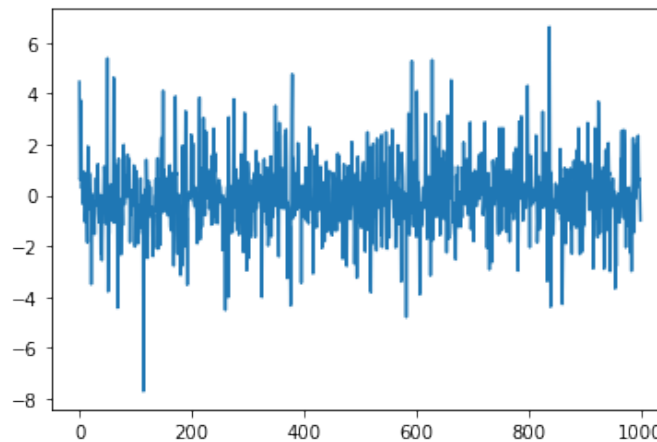
- Generujemy dyskretną zmienną losową  $I$  o rozkładzie  $P(I = i) = p_i, i = 1, 2, \dots, n. I \perp Y_i, i = 1, 2, \dots, n.$
- Generujemy zmienną losową  $Y_I$ .
- Wstawiamy  $X = Y_I$ .

### Przykłady

Z pomocą metody mieszaniny rozkładów jesteśmy w stanie wygenerować rozkład Laplace’a, znanego także jako dwustronny rozkład wykładniczy. Rozkład ten jest zadany wzorem:

$$f(x) = \frac{1}{2\lambda} e^{-\lambda|x|}, \lambda = \frac{1}{b} > 0$$

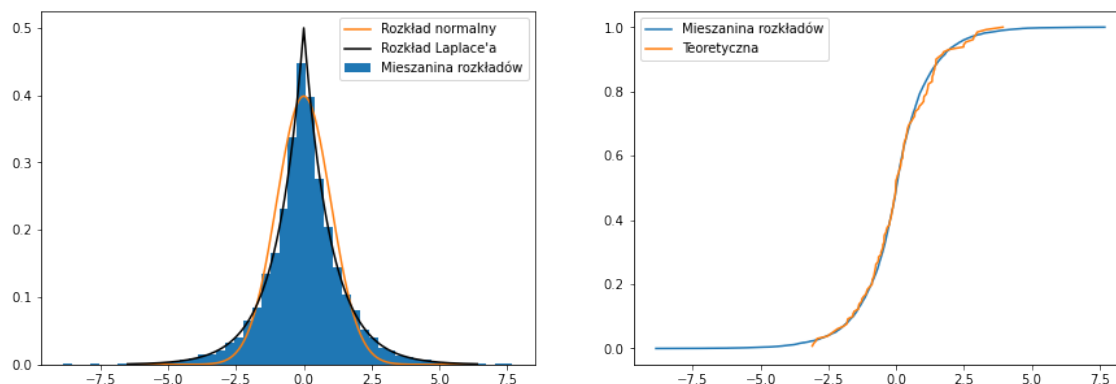
Rozkład ten można komponować z dwóch ”połówek” rozkładu wykładniczego, co tłumaczy jego drugą nazwę. W tym przykładzie jednak wykorzystano fakt, że jeśli  $X$  posiada rozkład wykładniczy, oraz  $Y \sim U(-1, 1)$ , to zmienna  $XY$  posiada rozkład Laplace’a o  $\mu = 0$  oraz skali  $b$ .



Rysunek 41: Realizacje zmiennej losowej  $X$  z rozkładu Laplace’a

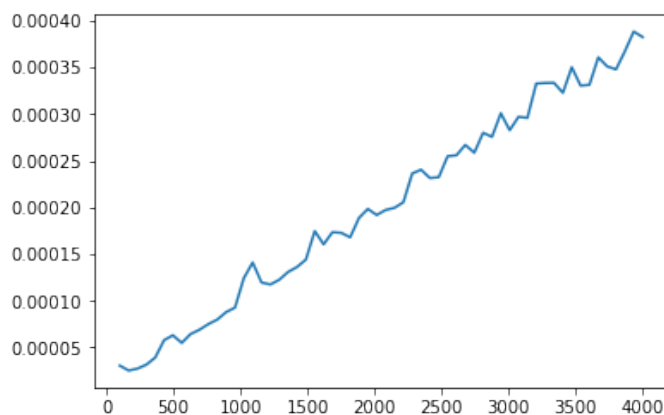
## Analiza otrzymanych wyników

Warto zwrócić szczególną uwagę na ciekawy, szpiczasty kształt wykres gęstości tego rozkładu oraz jego podobieństwo do dobrze nam już znanego rozkładu normalnego.



Rysunek 42: Gęstość zmiennej losowej  $X$  z rozkładu Laplace'a oraz jej dystrybuanta wraz z wykreśami teoretycznymi. Do porównania została wyrysowana również gęstość standardowego rozkładu normalnego

Zauważamy drobne niedokładności na wykresie dystrybuanty. Teoretyczna dystrybuanta miejscami jest kanciasta. Może to być spowodowane doбором nie najdokładniejszej metody jej generowania. W tym przypadku wartości teoretyczne zostały wygenerowane w Pythonie za pomocą `numpy.random.laplace` oraz własnej funkcji, tworzącej dystrybuantę. Jest to kolejna uwaga, którą warto mieć w pamięci w trakcie pracy z rozkładami zmiennych losowych.



Rysunek 43: Czas działania metody mieszaniny rozkładów w zależności od ilości generowanych wartości.

## 9 Porównanie powyższych metod

Podsumowanie czasu generowania zmiennych daną metodą dla 1000 zmiennych generowanych 1000 razy, gdzie  $t_d$  oznacza czas dla zmiennej dyskretnej, a  $t_c$  oznacza czas dla zmiennej ciągłej:

- metoda odwracania dystrybucyj:  $t_d \approx 0.18s$ ,  $t_c \approx 0.35s$ ;
- metoda akceptacji-odrzuć:  $t_d \approx 0.07s$ ,  $t_c \approx 1.2s$ ;
- metoda splotowa:  $t_d \approx 0.2s$ ,  $t_c \approx 0.3s$ ;
- metoda Boxa-Mullera:  $t_c \approx 0.8s$ ;
- metoda biegunowa:  $t_c \approx 15s$ ;
- metoda mieszaniny rozkładów:  $t_c \approx 0.1s$ .

Czasów tych nie należy bezpośrednio porównywać ze względu na użycie różnych bibliotek podczas implementacji, wykonywania ich na różnym sprzęcie oraz różnych rozkładów testowych. Z danych tych jednak można wyciągnąć, że zgodnie z oczekiwaniami generowanie zmiennych losowych dyskretnych jest znacząco szybsze w porównaniu do generowania zmiennych ciągłych.

## 10 Podsumowanie i wnioski

Mimo, że niektóre metody generowania zmiennych losowych możemy uznać za szybsze nie znaczy to, że są lepsze. Każda metoda ma pewne ograniczenia i trudności w stosowaniu w konkretnych przypadkach. Metodę generowania zmiennych powinniśmy dobierać uwzględniając informacje o rozkładzie, który chcemy uzyskać. Wartym rozważenia (jeśli mamy taką możliwość) jest korzystanie z funkcji wbudowanych generowania zmiennych losowych, które są dostępne dla wielu najpopularniejszych rozkładów. W każdym testowanym przez nas przypadku funkcje wbudowane okazały się lepsze od tych zaimplementowanych przez nas. Plusem korzystania z wbudowanych metod jest mniejsze prawdopodobieństwo popełnienia błędu przy implementacji oraz lepsza optymalizacja wynikająca z korzystania z bardziej skomplikowanych metod oraz szybszych funkcji.