UNIWERSYTET JAGIELLOŃSKI
W KRAKOWIE

WYDZIAŁ MATEMATYKI I INFORMATYKI

Projekt

“Generator liczb losowych”

Yurii Titov

Informatyka, II rok

Kraków, 2021

Wstęp

Cel projektu : implementacja generatorów liczb losowych o wybranych rozkładach oraz sprawdzenie ich jakości.

Projekt polega na zaimplementowaniu generatora liczb pseudolosowych (dalej oznaczony G) o rozkładzie równomiernym bez wykorzystania dostępnych funkcji systemowych czy bibliotek dla generatorów liczb losowych oraz implementacji na podstawie wspomnianego generatora G generatorów liczb pseudolosowych o rozkładach: jednostajnym z przedziału (0, 1) (dalej oznaczony J), Bernoulliego (ozn. B), Poissona (ozn. P), dwumianowym (ozn. D), wykładniczym (ozn. W) oraz normalnym (ozn. N). Po implementacji powinny być wykonane testy jakości generatorów liczb losowych.

W ramach projektu będą badane następujące zjawiska :

- losowość próby wygenerowanej przy pomocy zaimplementowanego generatora o odpowiednim rozkładzie (testy serii)

- zgodność próby wygenerowanej przy pomocy zaimplementowanego generatora z oczekiwanym rozkładem (testy χ^2)

- zgodność wartości oczekiwanej próby wygenerowanej przy pomocy zaimplementowanego generatora z analitycznie obliczoną wartością oczekiwaną dla danego rozkładu (test zgodności EX)

- zgodność wariancji próby wygenerowanej przy pomocy zaimplementowanego generatora z analitycznie obliczoną wariancją dla danego rozkładu (test zgodności Var X)

- niezależność kolejnych elementów próby wygenerowanej przy pomocy zaimplementowanego generatora od poprzednich (wyznaczanie współczynników autokorelacji)

Implementacja

Generator G jest zaimplementowany jako generator liniowy mieszany oparty na arytmetyce modularnej. Taki generator tworzy ciąg liczb według następującego schematu:

$$X_n = (a \cdot X_{n-1} + c) \mod m$$

Generowane liczby są od góry ograniczone przez m. Przyjmuje $m = 2^{32}$ (32-bitowa reprezentacja liczby). Zatem generowane liczby znajdują się w przedziale $[0, 2^{32}-1]$. Z kolei parametry a oraz c są równe odpowiednio 1664525 i 1013904223. Wartości mnożnika (parametr a) i przyrostu (parametr c) dla przypadku 32-bitowej reprezentacji liczby uzyskano w badaniach D. Knuth i H.W. Lewis[1].

```
def generate(self):  # parametry wzialem od D. Knuth i H.W. Lewis.
    self.next = (1664525 * self.next + 1013904223) % self.m  # generetor G

    return self.next
```

Ziarnem generatora G jest liczba $X_0 = 1$.

Generator J o rozkładzie jednostajnym na przedziale (0, 1) jest zaimplementowany z wykorzystaniem generatora G , dokonując następującego przekształcenia :

$$U_i = \frac{X_i}{m}$$

gdzie X jest elementem ciągu X_n generowanego przez G , a $m = 2^{32}$. Zatem $U_i \in [0, 1)$

```
def gen_unif(self):
    return self.generate() / self.m  # generator J
```

Generator liczb o rozkładzie Bernoulliego B, parametryzowany parametrem p, jest realizowany wspomagając się generatorem J oraz za pomocą następującej metody Monte Carlo :

$$X = \begin{cases} 1 \ \ gdy \ \ U < p \\ 0 \ \ gdy \ \ U \geq p \end{cases}$$

gdzie U – zmienna losowa o rozkładzie jednostajnym na (0, 1) , generowana przez J. Wtedy

P(1) = P(U<p) = p.

```python
def gen_bernulli(self, p):  # metoda podzialu (0,1) na przedzialy
    attempt = self.gen_unif()

    if attempt < p:  # generator B
        return 1
    else:
        return 0
```

Generator liczb o rozkładzie dwumianowym D, parametryzowany parametrami n (ilość prób Bernoulliego) oraz p (prawdopodobieństwo sukcesu dla próby Bernoulliego), jest zaimplementowany przy wykorzystaniu generatora B. Ponieważ rozkład dwumianowy jest dyskretnym rozkładem prawdopodobieństwa opisującym liczbę sukcesów k w ciągu n niezależnych prób, z których każda ma stałe prawdopodobieństwo sukcesu równe p, wystarczy zamodelować n takich prób przy użyciu generatora B i zsumować ilość sukcesów.

```python
def gen_binominal(self, n, p):  # n prob bernulli o prawdopodob p
    sum = 0
    for i in range(0, n):  # generator D
        sum += self.gen_bernulli(p)

    return sum
```

Generator liczb o rozkładzie Poissona P, parametryzowany parametrem λ, jest realizowany przy użyciu generatora J oraz za pomocą metody odwrotnej dystrybuanty dla rozkładu dyskretnego. Czyli

- najpierw generuje liczbę U o rozkładzie jednostajnym na (0, 1)

- obliczamy X = min{x ϵ S : F(x) > U}, gdzie S to zbiór możliwych wartości X.

Ponieważ jest to rozkład dyskretny o wartościach ≥ 0 , dystrybuantę (skumulowaną funkcję rozkładu) F(x) obliczamy wprost, sumując prawdopodobieństwa P(X=x). Czyli

$$F(x) = \sum_{i=0}^{x} P(X = i)$$

Z kolei

$$P(X = i) = \frac{e^{-\lambda}\lambda^i}{i!}$$

Obliczona według wzoru X = min{x ϵ S : F(x) > U} wartość X jest zwraca przez generator P.

```python
def gen_poisson(self, lambd):  # metoda odwrotnej dystrybuanty dla rozkładu dyskretnego
    U = self.gen_unif()

    x = 0
    e_lambd = np.e ** (-lambd)
    distribuant = e_lambd  # generator P
    while distribuant <= U:
        x += 1
        distribuant += (e_lambd * (lambd ** x)) / (math.factorial(x))

    return x
```

Generator liczb o rozkładzie wykładniczym W, parametryzowany λ, jest zaimplementowany z wykorzystaniem generatora J i za pomocą metody odwrotnej dystrybuanty dla rozkładu ciągłego.

Zgodnie z twierdzeniem, jeżeli X jest ciągłą zmienną losową o dystrybuancie $F_X(x)$ oraz $U = F_X(x)$, to U jest zmienną o rozkładzie jednostajnym na przedziale (0,1). Zatem postępujemy za algorytmem :

- najpierw generuje liczbę U o rozkładzie jednostajnym na (0, 1)

- obliczamy $X = F^{-1}(U)$, tzn. rozwiązujemy równanie F(X) = U dla X. Wtedy X ma dystrybuantę F(x).

Dla rozkładu wykładniczego w takim razie mamy :

$$F(x) = \int_0^x \lambda e^{-\lambda t}\, dt = 1 - e^{-\lambda x} \quad , \quad x > 0$$

$$F(X) = U$$

$$1 - e^{-\lambda X} = U$$

$$e^{-\lambda X} = 1 - U$$

$$\ln e^{-\lambda X} = \ln(1 - U)$$

$$-\lambda X = \ln(1 - U)$$

$$X = -\frac{\ln(1 - U)}{\lambda}$$

Obliczone w taki sposób X jest zwracane przez generator.

```python
def gen_exponential(self, lambd):  # metoda odwrotnej dystrybuanty
    U = self.gen_unif()

    x = np.log(1 - U) / (-lambd)  # generator W
    return x
```

Generator liczb losowych o rozkładzie Normalnym N, parametryzowany parametrami µ (wartość oczekiwana) oraz σ (odchylenie standardowe), jest realizowany przy użyciu generatora J za pomocą metody Boxa-Mullera.

Zgodnie z transformacją Boxa-Mullera, jeżeli

$$U_1 \sim Unif(0,1)$$

$$U_2 \sim Unif(0,1)$$

$$X = \sqrt{-2\ln(1-U_1)}\cos(2\pi U_2)$$

$$Y = \sqrt{-2\ln(1-U_2)}\sin(2\pi U_1)$$

to

$$X \sim N(0,1)$$
$$Y \sim N(0,1)$$

Jeżeli chcemy zmienne X i Y o rozkładzie normalnym standaryzowanym przekształcić na zmienne o rozkładzie N(µ, σ), musimy postępować następująco :

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

W implementacji korzystam tylko z jednej tak wygenerowanej zmiennej :

```
# wykonany za pomoca metody Boxa-Mullera
def gen_normal(self, mi, sigma):
    U1 = self.gen_unif()
    U2 = self.gen_unif()  # generator N

    Z1 = np.sqrt(-2 * np.log(U1)) * np.cos(2 * np.pi * U2)
  # Z2 = np.sqrt(-2 * np.log(U2)) * np.sin(2 * np.pi * U1)

    x = Z1 * sigma + mi

    return x
```

Każdy pojedynczy generator liczb pseudo-losowych o odpowiednim rozkładzie jest zaimplementowany jako oddzielna metoda klasy Generator.

Eksperymenty

Testy zgodności wartości oczekiwanej i wariancji

Wartość oczekiwaną i wariancję dla wygenerowanej próby każdego rozkładu będzie estymować według następujących wzorów:

$$\overline{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \overline{\mu})$$

gdzie x_i to kolejne elementy ciągu wygenerowanego przez odpowiedni generator liczb pseudo-losowych, N - ilość elementów ciągu.

- Testowanie generatora G

Generator G losuje liczby o rozkładzie równomiernym z przedziału od 0 do 2^{32} , zatem estymowane wartości μ oraz σ^2 będziemy porównywać z spodziewanymi wartościami wartości oczekiwanej i wariancji, obliczonymi według następujących wzorów :

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{n^2-1}{12}$$

gdzie a = 0 , b = 2^{32} , n = b – a + 1.

Wyniki (wielkość próby : 10000 liczb) :

```
wartość estymatora wartości oczekiwanej : 2136141087.8536
oczekiwana EX : 2147483648.0
========================
wartość estymatora wariancji: 1.5527873945783204e+18
oczekiwana wariancja : 1.5372286735249572e+18
```

Jak widzimy, estymatory zadowalająco przybliżają spodziewane wartości.

- Testowanie generatora J

Generator G losuje liczby o rozkładzie jednostajnym z przedziału od 0 do 1 , zatem estymowane wartości μ oraz σ^2 będziemy porównywać z spodziewanymi wartościami wartości oczekiwanej i wariancji, obliczonymi według wzorów analogicznych do zdefiniowanych dla generatora G, gdzie a=0, b=1.

Wyniki testów :

```
wartość estymatora wartości oczekiwanej : 0.49735910442043096
oczekiwana EX : 0.5
========================
wartość estymatora wariancji: 0.08417677333049606
oczekiwana wariancja : 0.0833
```

Wartości estymatorów są zgodne ze spodziewanymi.

- Testowanie generatora B

Generator B losuje liczby o rozkładzie Bernoulliego, zatem estymowane wartości μ oraz σ^2 będziemy porównywać z spodziewanymi wartościami wartości oczekiwanej i wariancji, obliczonymi według wzorów:

$$\mu = p$$

$$\sigma^2 = p(1-p)$$

gdzie p – parametr rozkładu Bernoulliego.

Wyniki testu:

```
Wpisz wartosc parametru p dla rozkładu Bernulliego (0<=p<=1):
0.4
========================
wartość estymatora wartości oczekiwanej : 0.4059
oczekiwana EX : 0.4
========================
wartość estymatora wariancji: 0.2396823782378216
oczekiwana wariancja : 0.24
```

```
Wpisz wartosc parametru p dla rozkładu Bernulliego (0<=p<=1):
0.7
========================
wartość estymatora wartości oczekiwanej : 0.7005
oczekiwana EX : 0.7
========================
wartość estymatora wariancji: 0.2115051005100642
oczekiwana wariancja : 0.21000000000000002
```

Wartości estymatorów są zgodne ze spodziewanymi.

- Testowanie generatora P

Generator P losuje liczby o rozkładzie Poissona, zatem estymowane wartości μ oraz σ^2 będziemy porównywać z spodziewanymi wartościami wartości oczekiwanej i wariancji, obliczonymi według wzorów:

$$\mu = \sigma^2 = \lambda$$

gdzie λ – parametr rozkładu Poissona.

Wyniki testów:

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
4
========================
wartość estymatora wartości oczekiwanej : 3.99654
oczekiwana EX : 4.0
========================
wartość estymatora wariancji: 4.016877261176282
oczekiwana wariancja : 4.0
========================
```

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
0.5
========================
wartość estymatora wartości oczekiwanej : 0.5008
oczekiwana EX : 0.5
========================
wartość estymatora wariancji: 0.5026834108341507
oczekiwana wariancja : 0.5
========================
```

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
10
========================
wartość estymatora wartości oczekiwanej : 9.99476
oczekiwana EX : 10.0
========================
wartość estymatora wariancji: 10.062071310313458
oczekiwana wariancja : 10.0
```

Wartości estymatorów są zgodne ze spodziewanymi.

- Testowanie generatora D

Generator D losuje liczby o rozkładzie dwumianowym, zatem estymowane wartości μ oraz σ^2 będziemy porównywać z spodziewanymi wartościami wartości oczekiwanej i wariancji, obliczonymi według wzorów:

$$\mu = np$$

$$\sigma^2 = np(1-p)$$

gdzie n, p – parametry rozkładu dwumianowego.

Wyniki testu:

```
Wpisz wartosc parametru p dla rozkładu dwumianowego (0<=p<=1):
0.7
Wpisz wartosc parametru n dla rozkładu dwumianowego (n>0):
20
========================
wartość estymatora wartości oczekiwanej : 13.99076
oczekiwana EX : 14.0
========================
wartość estymatora wariancji: 4.203270988710098
oczekiwana wariancja : 4.200000000000001
```

```
Wpisz wartosc parametru p dla rozkładu dwumianowego (0<=p<=1):
0.3
Wpisz wartosc parametru n dla rozkładu dwumianowego (n>0):
30
========================
wartość estymatora wartości oczekiwanej : 8.99623
oczekiwana EX : 9.0
========================
wartość estymatora wariancji: 6.334885459753684
oczekiwana wariancja : 6.3
========================
```

Wartości estymatorów są zgodne ze spodziewanymi.

-Testowanie generatora W

Generator W losuje liczby o rozkładzie wykładniczym, zatem estymowane wartości μ oraz σ^2 będziemy porównywać z spodziewanymi wartościami wartości oczekiwanej i wariancji, obliczonymi według wzorów:

$$\mu = \frac{1}{\lambda}$$

$$\sigma^2 = \frac{1}{\lambda^2}$$

gdzie λ – parametr rozkładu wykładniczego.

Wyniki testu:

```
Wpisz wartosc parametru lambda dla rozkładu wykładniczego (lambda>0):
1.5
========================
wartość estymatora wartości oczekiwanej : 0.6651953532035513
oczekiwana EX : 0.6666666666666666
========================
wartość estymatora wariancji: 0.4451454424172031
oczekiwana wariancja : 0.4444444444444444
========================
```

```
Wpisz wartosc parametru lambda dla rozkładu wykładniczego (lambda>0):
0.5
========================
wartość estymatora wartości oczekiwanej : 1.9955860596106485
oczekiwana EX : 2.0
========================
wartość estymatora wariancji: 4.006308981754767
oczekiwana wariancja : 4.0
```

Wartości estymatorów są zgodne ze spodziewanymi.

-Testowanie generatora N

Generator N losuje liczby o rozkładzie normalnym z parametrami μ oraz σ, zatem estymowane wartości μ oraz $σ^2$ będziemy porównywać z odpowiednio przekształconymi parametrami rozkładu.

Wyniki testu:

```
Wpisz wartosc parametru mi (wartosc oczekiwana) dla rozkładu normalnego :
0
Wpisz wartosc parametru sigma (odchylenie standardowe) dla rozkładu normalnego:
1
========================
wartość estymatora wartości oczekiwanej : 0.004170826764798692
oczekiwana EX : 0.0
========================
warość estymatora wariancji: 1.0034548887435095
oczekiwana wariancja : 1.0
```

```
Wpisz wartosc parametru mi (wartosc oczekiwana) dla rozkładu normalnego :
2
Wpisz wartosc parametru sigma (odchylenie standardowe) dla rozkładu normalnego:
0.5
========================
wartość estymatora wartości oczekiwanej : 2.0020854133824395
oczekiwana EX : 2.0
========================
warość estymatora wariancji: 0.2508637221858769
oczekiwana wariancja : 0.25
```

Wartości estymatorów są zgodne ze spodziewanymi.

Testy zgodności rozkładu χ^2

W tym teście badamy hipotezę, że generowana zmienna losowa ma dystrybuantę F(x).

Generujemy ciąg n liczb X_1, X_2, … X_n. Dokonujemy podziału zbioru wartości zmiennej losowej X na k przedziałów $[a_0, a_1]$, … , $[a_i , a_{i+1}]$, …, $[a_{k-1}, a_k]$, gdzie $F(a_0)=0$, $F(a_k) = 1$, i = 0, 1, …, k. (W praktyce często a_0 – najmniejsza wylosowana liczba, a_k-największa wylosowana liczba). Sprawdzamy ile z wygenerowanych liczb mieści się w każdym z przedziałów. Ich liczbę oznaczamy n_i.

Statystyka testu χ^2 wyraża się wzorem[2] :

$$\chi_s^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i}$$

gdzie p_i – to prawdopodobieństwo wylosowania liczby z i-tego przedziału, s – to ilość stopni swobody (s = ilość przedziałów – ilość parametrów rozkładu - 1).

Dla dużego n statystyka ta ma rozkład χ^2 o s stopniach swobody. Uznajemy, że generowana próba jest zgodna z badanym rozkładem, jeżeli statystyka testu jest mniejsza od wartości krytycznej dla wybranego poziomu istotności α oraz stopni swobody. W naszym przypadku przyjmujemy α = 0.05. Wartości krytyczne wyznaczamy za pomocą tablicy rozkładu χ^2.

Hipoteza zerowa : rozkład próby jest zgodny z rozkładem badanego generatora.

Fragment tablicy rozkładu :

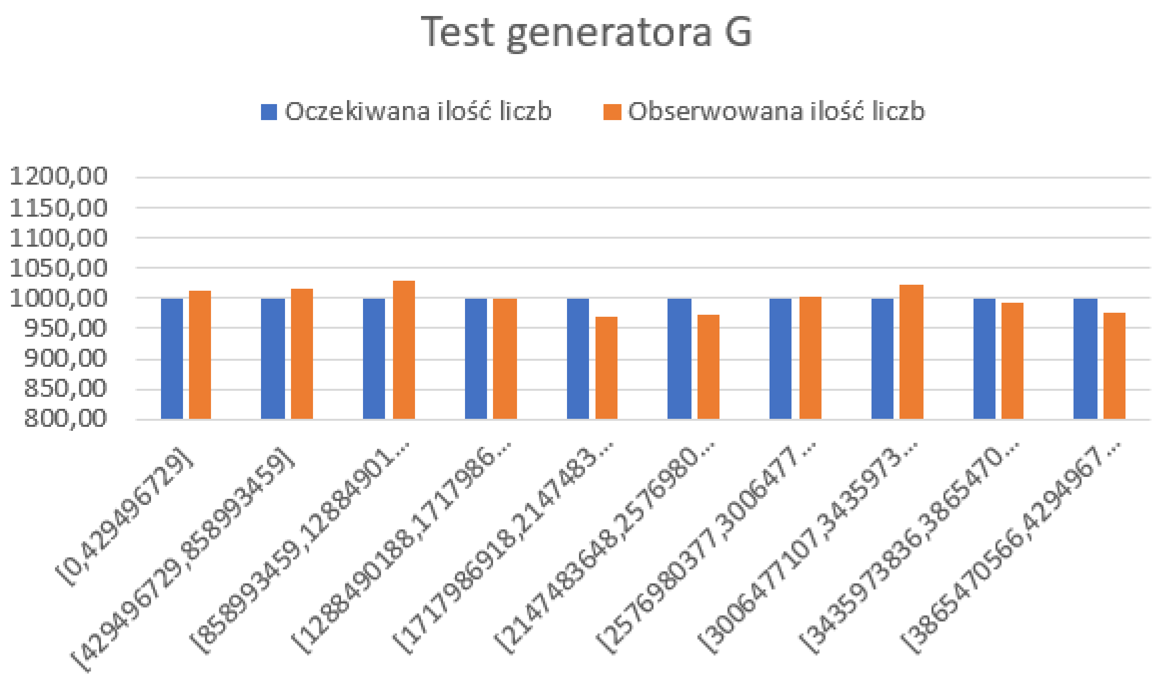
s\ p	**0,05**	**s\ p**	**0,05**	**s \ p**	**0,05**	**s \ p**	**0,05**
1	3,841	**8**	15,51	**15**	25,00	**22**	33,92
2	5,991	**9**	16,92	**16**	26,30	**23**	35,17
3	7,815	**10**	18,31	**17**	27,59	**24**	36,41
4	9,488	**11**	19,67	**18**	28,87	**25**	37,65
5	11,07	**12**	21,03	**19**	30,14	**26**	38,88
6	12,59	**13**	22,36	**20**	31,41	**27**	40,11
7	14,07	**14**	23,68	**21**	32,67	**28**	41,34

- Testowanie generatora G :

Dzielimy na 10 przedziałów , generator nie posiada parametrów , zatem s = 10-1 = 9. Więc statystyka testu powinna być mniejsza od 16.92. Wynik :

```
statystyka testu chi-kwadrat : 4.0540000000000145
```

Na wykresie przedstawiono oczekiwaną i obserwowaną ilość liczb w każdym z przedziałów (wielkość próby – 10000 liczb):



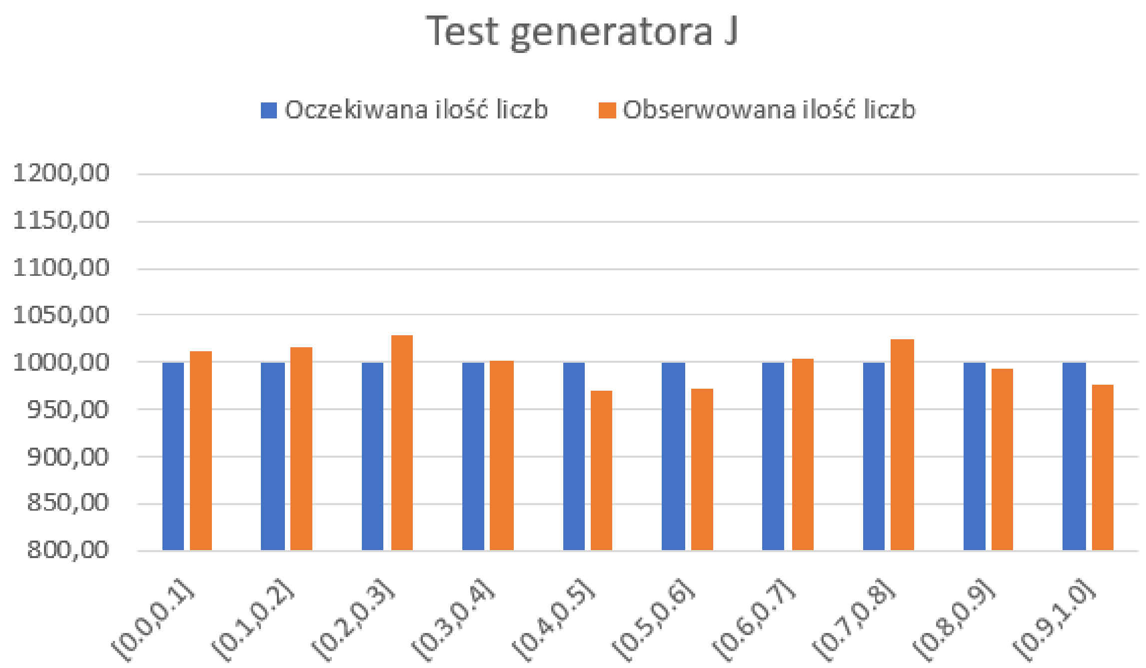
Możemy twierdzić o zgodności generowanego przez generator G ciągu liczb z rozkładem równomiernym na przedziale od 0 do 2^{32}. Nie mamy podstaw do odrzucenia hipotezy zerowej.

- Testowanie generatora J :

Dzielimy na 10 przedziałów , generator nie posiada parametrów , zatem s = 10-1 = 9. Więc statystyka testu powinna być mniejsza od 16.92. Wynik :

```
statystyka testu chi-kwadrat : 4.053999999999931
```

Na wykresie przedstawiono oczekiwaną i obserwowaną ilość liczb w każdym z przedziałów (wielkość próby – 10000 liczb):



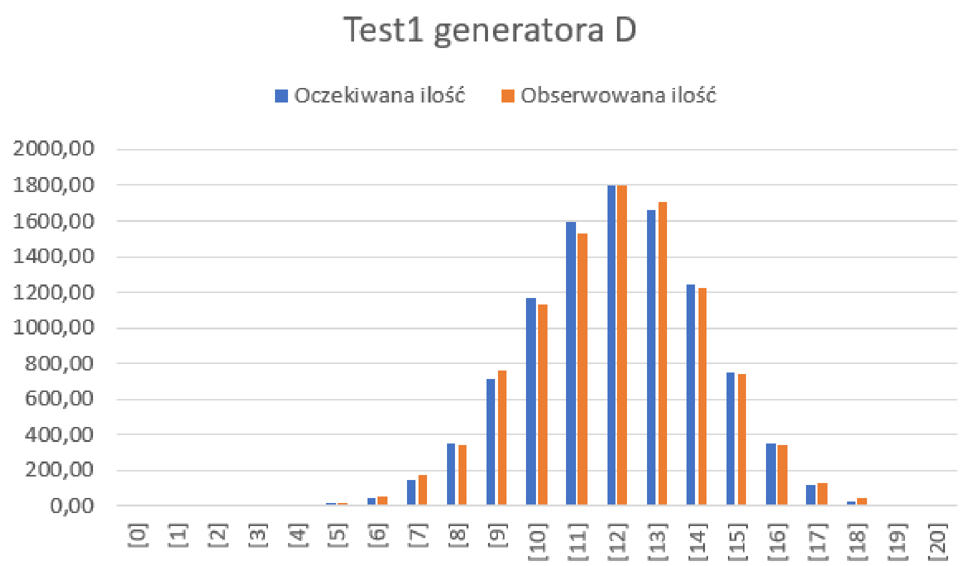
Możemy twierdzić o zgodności generowanego przez generator J ciągu liczb z rozkładem jednostajnym na przedziale od 0 do 1. Nie mamy podstaw do odrzucenia hipotezy zerowej.

- Testowanie generatora D:

Generator D posiada dwa parametry : n oraz p. Może wygenerować wartości od 0 do n, czyli w sumie n+1 różnych liczb. Każda taka liczba jest dla nas oddzielnym przedziałem (możemy tak przyjąć , ponieważ rozkład dwumianowy jest rozkładem dyskretnym). Zatem s = n+1-2-1=n-2. Przeprowadźmy 2 testy. W pierwszym przyjmujemy n=20. Więc s=18 i statystyka testu powinna być mniejsza od 28.87. Wynik:

```
Wpisz wartosc parametru p dla rozkładu dwumianowego (0<=p<=1):
0.6
Wpisz wartosc parametru n dla rozkładu dwumianowego (n>0):
20
statystyka testu chi-kwadrat : 24.616034413153688
ilość intervałów 21
ilość stopni swobody 18
```

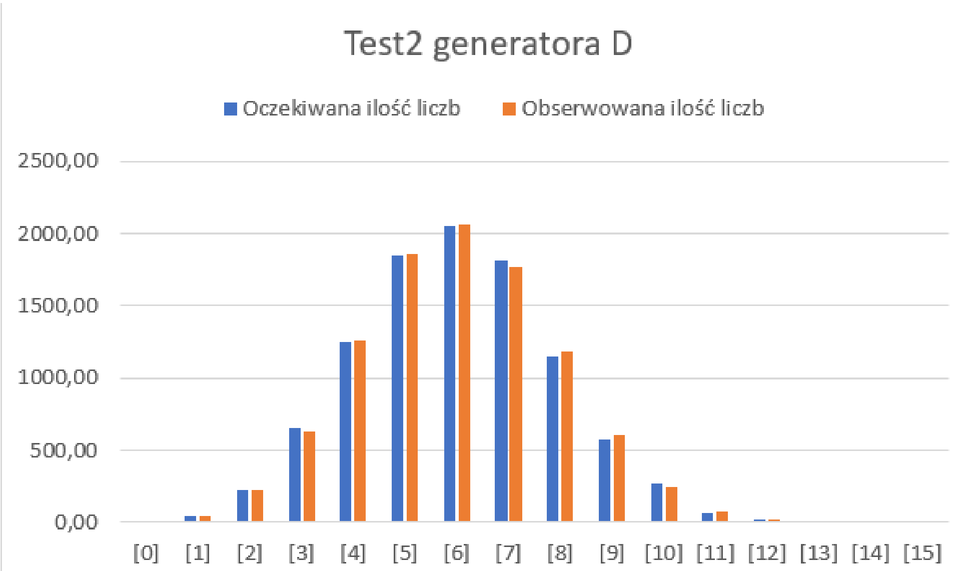
Na wykresie przedstawiono oczekiwaną i obserwowaną ilość liczb w każdym z przedziałów (wielkość próby – 10000 liczb):



W drugim teście przyjmujemy n=15. Więc s=13 i statystyka testu powinna być mniejsza od 22.36.

```
Wpisz wartosc parametru p dla rozkładu dwumianowego (0<=p<=1):
0.4
Wpisz wartosc parametru n dla rozkładu dwumianowego (n>0):
15
statystyka testu chi-kwadrat : 15.629454142550447
ilość intervałów 16
ilość stopni swobody 13
```

Na wykresie przedstawiono oczekiwaną i obserwowaną ilość liczb w każdym z przedziałów (wielkość próby – 10000 liczb):



Możemy twierdzić o zgodności generowanego przez generator D ciągu liczb z rozkładem dwumianowym. Nie mamy podstaw do odrzucenia hipotezy zerowej.

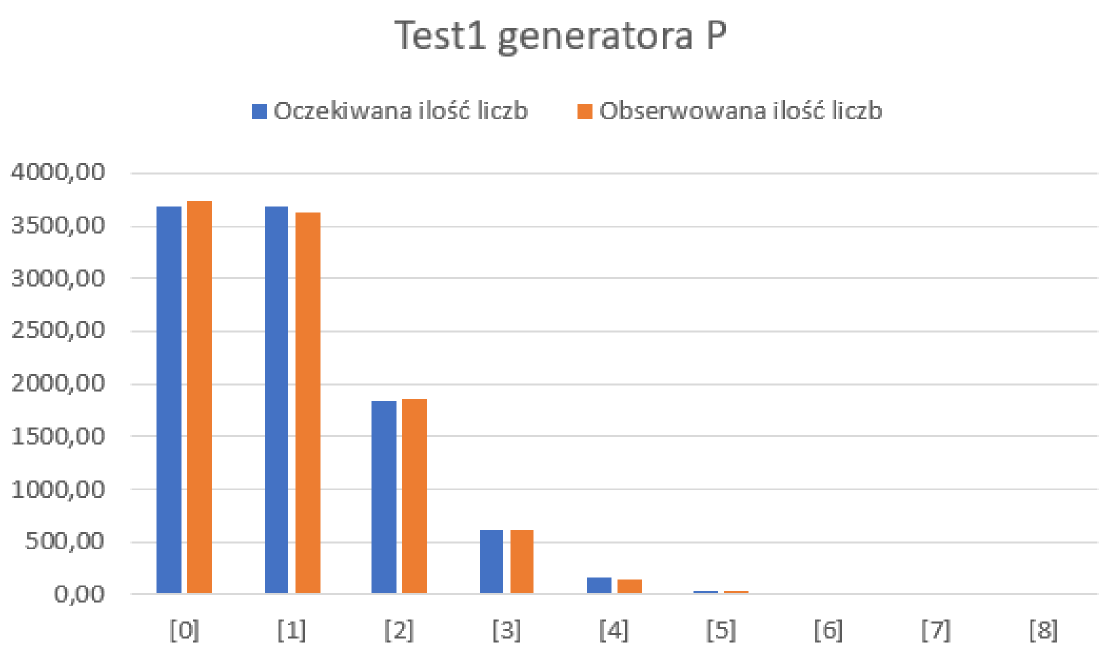
- Testowanie generatora P

Generator P posiada jeden parametr λ. Podobnie jak w przypadku generatora D jedną liczbę uznajemy za osobny przedział (rozkład Poissona jest dyskretny) i zliczmy ilość wystąpień każdej liczby naturalnej od 0 do maksymalnej wylosowanej dla danego λ.

Wynik 1 testu:

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
1
ilość interwałów 9
ilość stopni swobody 7
statystyka testu chi-kwadrat : 6.602323300173795
```

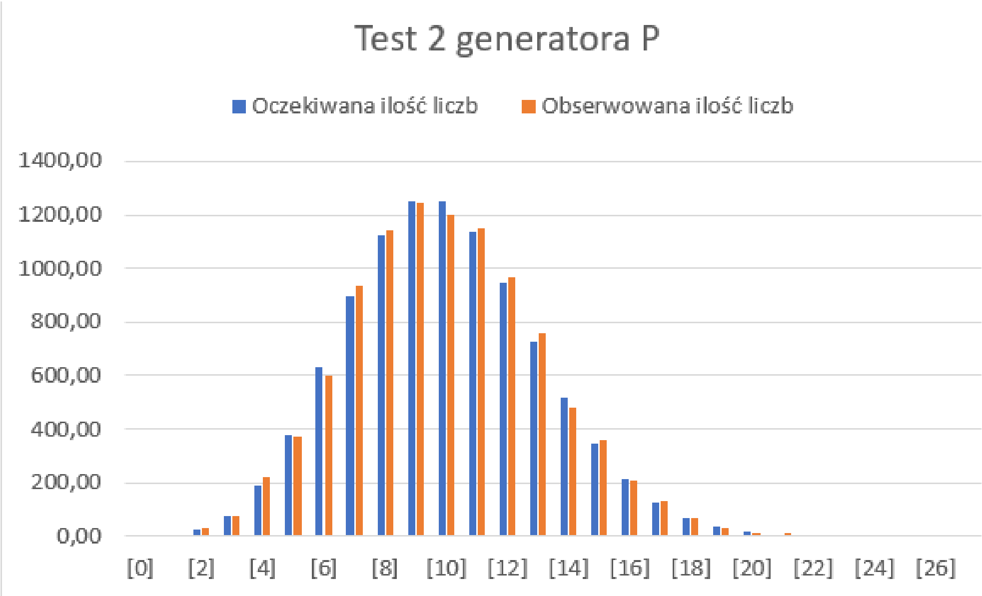
Dla 7 stopni swobody statystyka testu musi być mniejsza od 14.07. Odpowiedni wykres:



Wynik drugiego testu:

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
10
ilość interwałów 28
ilość stopni swobody 26
statystyka testu chi-kwadrat : 29.564556384215425
```

Dla 26 stopni swobody statystyka testu musi być mniejsza od 38.88. Odpowiedni wykres:



Możemy twierdzić o zgodności generowanego przez generator P ciągu liczb z rozkładem Poissona. Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora W:

Generator W posiada jeden parametr λ. Zatem s = k -2, gdzie k-ilość przedziałów. Ilość przedziałów zależy od maksymalnej wylosowanej liczby w próbie (ozn. max):

$$k = \lfloor max \rfloor + 1$$

Każdy przedział jest postaci [0, 1], [1, 2], …, [k-1, k].

Wyniki testów:

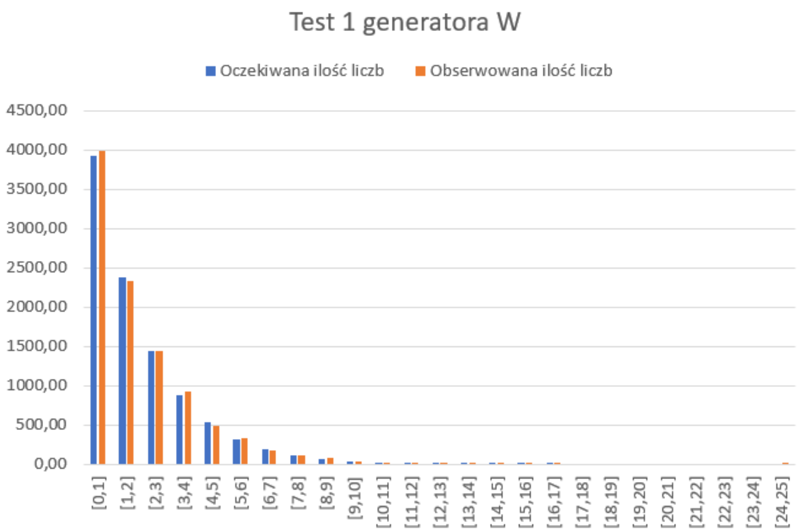
```
Wpisz wartosc parametru lambda dla rozkładu wykładniczego (lambda>0):
0.5
statystyka testu chi-kwadrat : 25.568491940077713
ilość interwałów 25
ilość stopni swobody 23
```

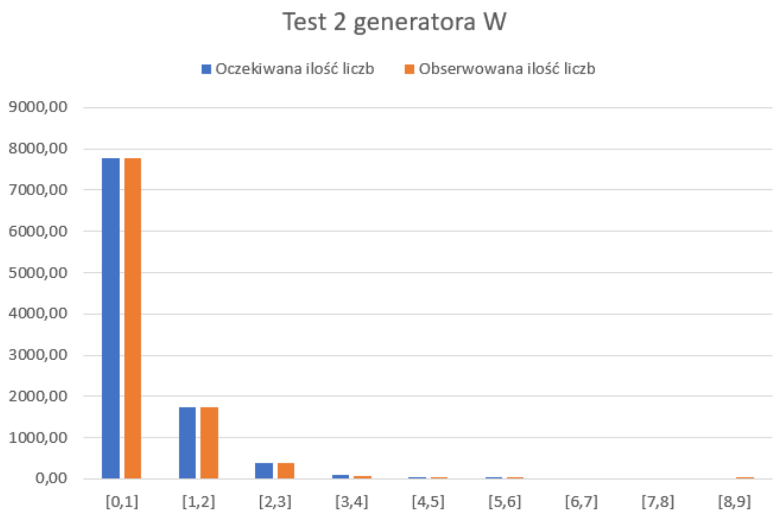
Dla 23 stopni swobody statystyka testu musi być mniejsza od 35.17.

```
Wpisz wartosc parametru lambda dla rozkładu wykładniczego (lambda>0):
1.5
statystyka testu chi-kwadrat : 10.153368336528867
ilość interwałów 9
ilość stopni swobody 7
```

Dla 7 stopni swobody statystyka testu musi być mniejsza od 14.07.

Odpowiednie wykresy do testów:





Możemy twierdzić o zgodności generowanego przez generator W ciągu liczb z rozkładem wykładniczym. Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora N

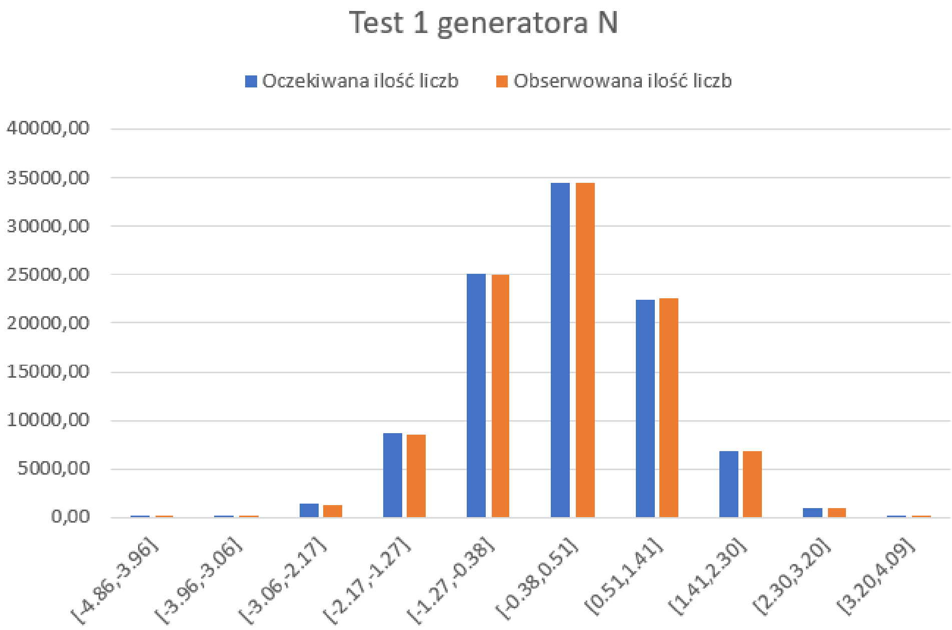
Generator N posiada dwa parametry. Dzielimy zawsze na 10 przedziałów. Więc ilość stopni swobody wynosi s=10-2-1=7. Zatem statystyka testu musi być mniejsza od 14.07. Testujemy najpierw rozkład normalny standaryzowany a potem odmienny od standaryzowanego.

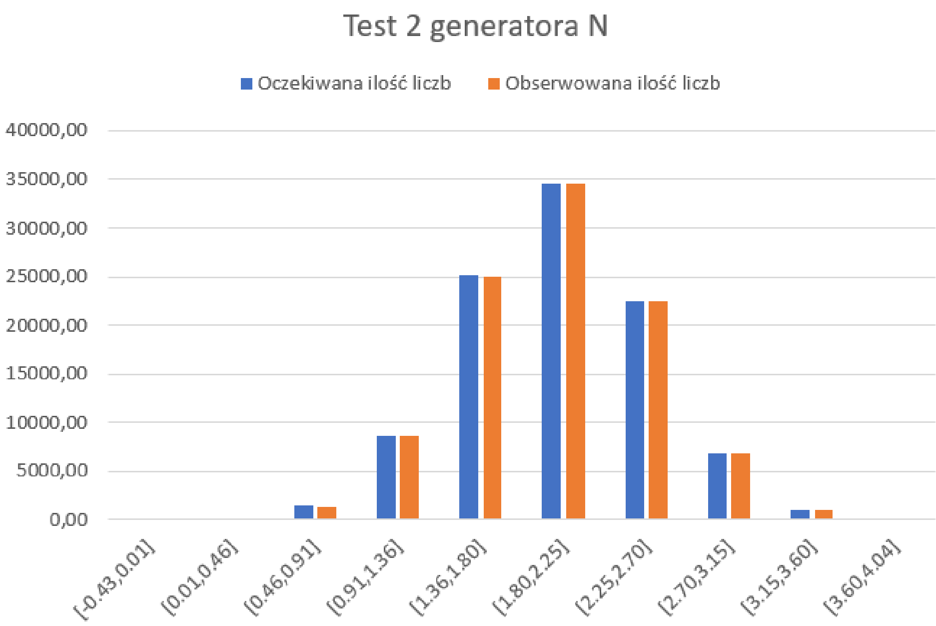
Wyniki testów (wielkość próby – 100000 liczb):

```
Wpisz wartosc parametru mi (wartosc oczekiwana) dla rozkładu normalnego :
0
Wpisz wartosc parametru sigma (odchylenie standardowe) dla rozkładu normalnego:
1
statystyka testu chi-kwadrat : 6.671839687431957
```

```
Wpisz wartosc parametru mi (wartosc oczekiwana) dla rozkładu normalnego :
2
Wpisz wartosc parametru sigma (odchylenie standardowe) dla rozkładu normalnego:
0.5
statystyka testu chi-kwadrat : 6.671839687431957
```

Wykresy dla odpowiednich testów:





Możemy twierdzić o zgodności generowanego przez generator N ciągu liczb z rozkładem normalnym. Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora B

Zmienna losowa o rozkładzie Bernoulliego może przyjmować tylko 2 wartości: 0 i 1. Właśnie te dwie liczby pełną rolę przedziałów w naszym teście. Wyniki testowania (przyjmujemy że mamy 1 stopień swobody):

```
Wpisz wartosc parametru p dla rozkładu Bernulliego (0<=p<=1):
0.3
ilość zer 6942
ilość jedynek 3058
statystyka testu chi-kwadrat : 1.601904761904762
wartość krytyczna dla poziomu istotnośći 0.05 oraz 1 stopnia swobody wynosi 3.841
```

```
Wpisz wartosc parametru p dla rozkładu Bernulliego (0<=p<=1):
0.8
ilość zer 1971
ilość jedynek 8029
statystyka testu chi-kwadrat : 0.5256249999999869
wartość krytyczna dla poziomu istotnośći 0.05 oraz 1 stopnia swobody wynosi 3.841
```

Możemy twierdzić o zgodności generowanego przez generator B ciągu liczb z rozkładem Bernoulliego. Nie mamy podstaw do odrzucenia hipotezy zerowej.

<u>Testowanie losowości generowanego ciągu przez wyznaczenie współczynników autokorelacji</u>

Funkcja autokorelacji opisuje zależność elementów ciągu od wyrazów poprzednich. Jest ona zdefiniowana następująco[3]

$$R_r = \frac{1}{(N-r)\sigma^2}\sum_{i=1}^{N-r}(X_i-\mu)(X_{i+r}-\mu)$$

Gdzie X_i – kolejne elementy ciągu wygenerowanego przez odpowiedni generator liczb pseudo-losowych, N – ilość liczb w próbie, r – wybrane przesunięcie.

R_r powinno być bliskie 0, bo oznacza to brak korelacji pomiędzy elementami ciągu, czyli rozkład liczb pseudolosowych jest stochastyczny. Wyznaczymy współczynniki autokorelacji R_{100} dla ciągów 1000 liczb generowanych przez każdy zaimplementowany generator.

Hipoteza zerowa: elementy ciągu są niezależne jeden od drugiego, generator jest stochastyczny.

Generator B:

```
Wpisz wartosc parametru p dla rozkładu Bernulliego (0<=p<=1):
0.3
========================
wartość współczynnika autokorelacji dla przesuniecia 100 : 0.002115853347832633
oczekiwana wartość współczynnika autokorelacji: 0
```

Generator G:

```
========================
wartość współczynnika autokorelacji dla przesuniecia 100 : 0.007797481180751632
oczekiwana wartość współczynnika autokorelacji: 0
```

Generator J:

```
wartość współczynnika autokorelacji dla przesuniecia 100 : 0.007797481180751632
oczekiwana wartość współczynnika autokorelacji: 0
```

Generator D:

```
Wpisz wartosc parametru p dla rozkładu dwumianowego (0<=p<=1):
0.7
Wpisz wartosc parametru n dla rozkładu dwumianowego (n>0):
20
========================
wartość współczynnika autokorelacji dla przesuniecia 100 : -0.0037107682050093052
oczekiwana wartość współczynnika autokorelacji: 0
```

Generator P:

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
4
========================
warość współczynnika autokorelacji dla przesuniecia 100 : -0.0025175142928206993
oczekiwana wartość współczynnika autokorelacji: 0
```

Generator W:

```
Wpisz wartosc parametru lambda dla rozkładu wykładniczego (lambda>0):
1.5
========================
wartość współczynnika autokorelacji dla przesuniecia 100 : 0.0063245400290527975
oczekiwana wartość współczynnika autokorelacji: 0
```

Generator N:

```
Wpisz wartosc parametru mi (wartosc oczekiwana) dla rozkładu normalnego :
0
Wpisz wartosc parametru sigma (odchylenie standardowe) dla rozkładu normalnego:
1
========================
warość współczynnika autokorelacji dla przesuniecia 100 : 0.003668678975471308
oczekiwana wartość współczynnika autokorelacji: 0
```

Dla każdego rozkładu współczynnik autokorelacji jest bliski zera, więc nie mamy podstaw do odrzucenia hipotezy zerowej.

Testy serii – badanie losowości próby

Test serii jest prostym narzędziem do badania losowości próby. Przeprowadzamy go dla każdego generatora w następujący sposób[4]:

- Generujemy ciąg n liczb pseudolosowych (x_i) przy pomocy odpowiedniego generatora

- Dla rozpatrywanego ciągu danych statystycznych obliczamy medianę m_e według następującego wzoru:

$$m_e = \begin{cases} x_{\frac{n+1}{2}} & n - nieparzyste \\ \frac{1}{2}\left(x_{\frac{x}{2}} + x_{\frac{x+2}{2}}\right) & n - parzyste \end{cases}$$

gdzie (x_1, x_2, … , x_n) – posortowana rosnąco próba.

- Elementom próby (nieposortowanej) przypisujemy symbol 0, jeżeli element jest mniejszy do m_e, oraz 1, jeżeli jest on większy od m_e. Wartości równe m_e nie rozpatrujemy. Serie to podciągi złożone z jednakowych symboli.

Statystyka testu to ilość serii. Oznaczymy ją U.

Niech n_0 – ilość zer, a n_1 – ilość jedynek po przypisaniu próbie odpowiednich symboli. Wtedy zbiór krytyczny testu ma postać:

$$K = (-\infty, k_1] \cup [k_2, +\infty)$$

Gdzie k1 odczytujemy z tablicy rozkładu serii dla n_0 i n_1 oraz poziomu istotności $\frac{\alpha}{2}$, a k2 odczytujemy z tablicy rozkładu serii dla n_0 i n_1 oraz poziomu istotności 1 - $\frac{\alpha}{2}$. Przyjmujemy α = 0.05.

Hipoteza zerowa – elementy próby mają charakter losowy.

Jeżeli U ϵ K – odrzucamy hipotezę zerową, w przeciwnym razie - nie mamy podstaw do odrzucenia hipotezy zerowej.

Tabele rozkładu serii (są one symetryczne):

										α=0,025										
n_2	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**n_2**
2	2																			**2**
3	2	2																		**3**
4	2	2	2																	**4**
5	2	2	2	2																**5**
6	2	2	2	3	3															**6**
7	2	2	2	3	3	3														**7**
8	2	2	3	3	3	4	4													**8**
9	2	2	3	3	4	4	5	5												**9**
10	2	2	3	3	4	5	5	5	6											**10**
11	2	2	3	4	4	5	5	6	6	7										**11**
12	2	2	3	4	4	5	6	6	7	7	7									**12**
13	2	2	3	4	5	5	6	6	7	7	8	8								**13**
14	2	2	3	4	5	5	6	7	7	8	8	9	9							**14**
15	2	3	3	4	5	6	6	7	7	8	8	9	9	10						**15**
16	2	3	4	4	5	6	6	7	8	8	9	9	10	10	11					**16**
17	2	3	4	4	5	6	7	7	8	9	9	10	10	11	11	11				**17**
18	2	3	4	5	5	6	7	8	8	9	9	10	10	11	11	12	12			**18**
19	2	3	4	5	6	6	7	8	8	9	10	10	11	11	12	12	13	13		**19**
20	2	3	4	5	6	6	7	8	9	9	10	10	11	12	12	13	13	13	14	**20**
n_2	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**n_2**

										$\alpha=0,975$										
n_2	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	n_2
2	4																			**2**
3	5	6																		**3**
4	5	7	8																	**4**
5	5	7	8	9																**5**
6	5	7	8	9	10															**6**
7	5	7	9	10	11	12														**7**
8	5	7	9	10	11	12	13													**8**
9	5	7	9	11	12	13	13	14												**9**
10	5	7	9	11	12	13	14	15	15											**10**
11	5	7	9	11	12	13	14	15	16	16										**11**
12	5	7	9	11	12	13	15	15	16	17	18									**12**
13	5	7	9	11	13	14	15	16	17	18	18	19								**13**
14	5	7	9	11	13	14	15	16	17	18	19	19	20							**14**
15	5	7	9	11	13	14	15	17	17	18	19	20	21	21						**15**
16	5	7	9	11	13	15	16	17	18	19	20	20	21	22	22					**16**
17	5	7	9	11	13	15	16	17	18	19	20	21	22	22	23	24				**17**
18	5	7	9	11	13	15	16	17	18	19	20	21	22	23	24	24	25			**18**
19	5	7	9	11	13	15	16	17	19	20	21	22	22	23	24	25	25	26		**19**
20	5	7	9	11	13	15	16	17	19	20	21	22	23	24	24	25	26	26	27	**20**
n_2	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	n_2

Testowanie generatora G:

```
Mediana : 1814098701
Serie :
[0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0]
Ilość serii (statystyka testu) : 19
Ilość ogólna symboli 0 : 18
Ilość ogólna symboli 1 : 16
```

$$K = (-\infty, 11] \cup [24, +\infty)$$
$$U = 19 \notin K$$

Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora J:

```
Mediana : 0.46330990025307983
Serie :
[0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1]
Ilość serii (statystyka testu) : 16
Ilość ogólna symboli 0 : 16
Ilość ogólna symboli 1 : 14
```

$$K = (-\infty, 10] \cup [21, +\infty)$$
$$U = 16 \notin K$$

Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora D:

```
Wpisz wartosc parametru p dla rozkładu dwumianowego (0<=p<=1):
0.7
Wpisz wartosc parametru n dla rozkładu dwumianowego (n>0):
20
Mediana : 14.0
Serie :
[1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0]
Ilość serii (statystyka testu) : 12
Ilość ogólna symboli 0 : 14
Ilość ogólna symboli 1 : 13
```

$$K = (-\infty, 9] \cup [19, +\infty)$$
$$U = 12 \notin K$$

Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora P:

```
Wpisz wartosc parametru lambda dla rozkładu Poissona (lambda>0):
4
Mediana : 3
Serie :
[0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1]
Ilość serii (statystyka testu) : 12
Ilość ogólna symboli 0 : 9
Ilość ogólna symboli 1 : 16
```

$$K = (-\infty, 7] \cup [17, +\infty)$$
$$U = 12 \notin K$$

Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora W:

```
Wpisz wartosc parametru lambda dla rozkładu wykładniczego (lambda>0):
0.5
Mediana : 1.0976704043905527
Serie :
[0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0]
Ilość serii (statystyka testu) : 19
Ilość ogólna symboli 0 : 18
Ilość ogólna symboli 1 : 16
```

$$K = (-\infty, 11] \cup [24, +\infty)$$
$$U = 19 \notin K$$

Nie mamy podstaw do odrzucenia hipotezy zerowej.

Testowanie generatora N:

```
Wpisz wartosc parametru mi (wartosc oczekiwana) dla rozkładu normalnego :
0
Wpisz wartosc parametru sigma (odchylenie standardowe) dla rozkładu normalnego:
1
Mediana : 0.03803598298425944
Serie :
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0]
Ilość serii (statystyka testu) : 15
Ilość ogólna symboli 0 : 18
Ilość ogólna symboli 1 : 16
```

$$K = (-\infty, 11] \cup [24, +\infty)$$

$$U = 15 \notin K$$

Nie mamy podstaw do odrzucenia hipotezy zerowej.

Możemy twierdzić o losowości próby generowanej przez każdy zaimplementowany generator.

<u>Test pokerowy 4 bitowy dla rozkładu Bernoulliego</u>

Dla generatora B losowość ciągu sprawdzimy przy pomocy testu pokerowego 4 bitowego. Przeprowadzamy go w następujący sposób[5]:

Generujemy długi binarny ciąg losowy długości n przy pomocy generatora B z parametrem p=0.5. Dzielimy go na bloki 4-bitowe. Niech k – liczba możliwych wartości w bloku, dla bloków 4-bitowych k = 16 liczb czterobitowych. Wyznaczamy liczbę n_i – ilość wystąpień i-tej liczby we wszystkich blokach, i = 0,1,…, 15. Statystykę testu obliczamy według następującego wzoru:

$$U = \frac{16}{n} \sum_{i=0}^{15} (n_i^2) - n$$

Zbiór krytyczny definiujemy następująco:

$$K = (-\infty, k_1] \cup [k_2, +\infty)$$

Gdzie k_1 i k_2 wyznaczamy z tabeli rozkładu χ^2 dla odpowiednio α=0.95 oraz α=0.05 i k-1 stopni swobody. W naszym przypadku k_1 = 7.261 , a k_2 = 25.00.

Hipoteza zerowa – elementy próby mają charakter losowy.

Jeżeli U ϵ K – odrzucamy hipotezę zerową, w przeciwnym razie - nie mamy podstaw do odrzucenia hipotezy zerowej.

Wyniki testu:

```
Wartość statystyki testu =  16.220799999999144
```

Więc, U = 16.22 , co jest mniejsze od k_2 i większe od k_1. Nie mamy podstaw dla odrzucenia hipotezy zerowej.

Podsumowanie

W ramach projektu w języku Python zostały zaimplementowane w pełni działające generatory liczb pseudolosowych o rozkładach, wskazanych we wstępie. Ich jakość była sprawdzona przez przeprowadzenie testów zgodności rozkładu (testy zgodności wartości oczekiwanej i wariancji, testy χ^2) oraz testów losowości (testy serii, wyznaczanie współczynnika autokorelacji, test pokerowy 4-bitowy).

Na mocy przeprowadzonego testowania możemy twierdzić o tym, że nie mamy podstaw dla odrzucenia hipotezy o zgodności ciągów generowanych przez odpowiednie generatory ze spodziewanymi rozkładami. Analogicznie nie mamy takich podstaw i dla odrzucenia hipotezy o losowości próby generowanej przez zaimplementowane generatory.

Możemy podsumować, że cel projektu została osiągnięta.

Źródła:

[1] - THE ART OF SCIENTIFIC COMPUTING (źródło internetowe, URL: https://www.unf.edu/~cwinton/html/cop4300/s09/class.notes/LCGinfo.pdf) [str. 284]

[2] – Wykład „Generatory liczb pseudolosowych" (źródło internetowe, URL: http://home.agh.edu.pl/~chwiej/mn/generatory_1819.pdf) [str. 46]

[3] - Wykład „Generatory liczb pseudolosowych" (źródło internetowe, URL: http://home.agh.edu.pl/~chwiej/mn/generatory_1819.pdf) [str. 10]

[4] – Wykład „TESTY LOSOWOŚCI" (źródło internetowe, URL: http://statystyka.rezolwenta.eu.org/Materialy/Testy-los.pdf) [str. 1]

[5] – Wykład „TESTY LOSOWOŚCI" (źródło internetowe, URL: http://statystyka.rezolwenta.eu.org/Materialy/Testy-los.pdf) [str. 50]