

Projekt 1 - Monte Carlo

Marta Gacek

November 2024

1 Wprowadzenie

Celem niniejszego projektu jest wykorzystanie generatorów liczb pseudolosowych do przeprowadzenia testów statystycznych w symulacjach. Chcemy sprawdzić czy dany generator losuje liczby z rozkładu jednostajnego i jak duża jest losowość obserwacji.

1.1 Testy statystyczne

Zanim przejdziemy do części programistycznej, opiszemy stosowane przez nas testy od strony teoretycznej, aby wiadomo było jak funkcjonują.

1.1.1 Test χ^2

Test *chi kwadrat* jest dość popularnym testem statystycznym. Jego działanie opiera się na wygenerowaniu pewnej liczby obserwacji, powiedzmy n , z których każda dopasowana jest do jednej z k kategorii (obrazowo możemy opisać tę sytuację jako wrzucanie n kul do k kubeków).

Zakładając, że Y_i oznacza liczbę obserwacji z i -tej kategorii, natomiast p_i jest prawdopodobieństwem dopasowania obserwacji do i -tej kategorii, możemy się spodziewać, że dla dużych wartości n znajdzie zależność

$$Y_i \approx np_i.$$

Ze względu na założenie o niezależności obserwacji, statystyka

$$\hat{\chi}^2 = \sum_{j=1}^k \frac{(Y_j - np_j)^2}{np_j}$$

ma rozkład χ^2 z $k - 1$ stopniami swobody, co oznaczamy jako $\chi^2(k - 1)$.

1.1.2 Test Kołmogorowa-Smirnowa

Kolejnym testem, którego będziemy używać, jest test Kołmogorowa-Smirnowa. Zakładamy w nim, że mamy n obserwacji X_i , $i = 1, \dots, n$, gdzie zmienna

losowa X ma ciągły rozkład z dystrybuantą $F_X(x)$. Statystyka Kołmogorowa-Smirnowa ma postać

$$D_n = \sup_x |F_n(x) - F_X(x)|,$$

gdzie $F_n(x) = \frac{1}{n} \sum_{i=1}^n 1(X_i \leq x)$ oznacza dystrybuantę empiryczną. Sprawdzamy zatem czy rozkład naszej próby różni się w istotny sposób od rozkładu teoretycznego.

1.1.3 Test serii

Celem tego testu jest sprawdzenie czy dane są rozmieszczone w losowy sposób. Przyjmujemy, że seria to ciąg kolejnych wartości tego samego typu w danych. Dzielimy dane na dwie kategorie, na przykład wartości poniżej i powyżej mediany lub średniej. Następnie liczymy serie w ciągu i porównujemy ich liczbę z wartością oczekiwaną.

Dla dużych prób liczba serii (K) jest zbliżona do rozkładu normalnego ze średnią oraz wariancją odpowiednio

$$\mu = \frac{2xy}{x+y} + 1, \quad \sigma^2 = \frac{2xy(2xy - (x+y))}{(x+y)^2(x+y-1)}.$$

Oznaczenia x i y to liczby wartości w odpowiednio pierwszej i drugiej kategorii. Statystyka testowa ma zatem postać

$$Z = \frac{K - \mu}{\sigma}.$$

Pozostaje porównać ją z odczytaną z tablic wartością Z_α dla wybranego poziomu istotności. Jeśli $|Z| > Z_\alpha$, to odrzucamy hipotezę zerową, czyli ciąg nie jest losowy.

1.1.4 Frequency monobit test

Chcemy sprawdzić losowość ciągu bitów, który oznaczamy jako $b_1, \dots, b_i \in \{0, 1\}$. Przekształcamy bity na x_1, \dots, x_i , przyjmujące wartości $\{-1, 1\}$, według wzoru $x_k = 2b_k - 1$. Statystyka testowa ma postać

$$S_n = \frac{1}{\sqrt{n}} \sum_{i=1}^n x_i.$$

Jeśli b_k , $k = 1, 2, \dots$ są niezależne i rozłożone jednostajnie na $\{0, 1\}$, to x_k , $k = 1, 2, \dots$ są niezależne i rozłożone jednostajnie na $\{-1, 1\}$. Z tego względu statystyka S_n ma w przybliżeniu rozkład $N(0, 1)$. Wobec tego

$$p\text{-value} = P(|N| > |S_n|) = 2(1 - \phi(|S_n|)),$$

gdzie N jest zmienną losową standardowego rozkładu normalnego, a ϕ jego dystrybuantą. Możemy porównać wartość statystyki S_n z tablicami statystycznymi dla rozkładu normalnego, odczytując Z_α dla konkretnego poziomu istotności. Jeśli $|S_n| > Z_\alpha$, to ciąg nie jest losowy.

1.2 Generatory liczb pseudolosowych

Krótko opiszemy generatory, których będziemy używać, aby lepiej zrozumieć jak działają.

1.2.1 LCG (Linear Congruential Generator)

LCG jest popularnym, prostym w implementacji generatorem. W danych wejściowych wymagany jest początkowy warunek x_0 (pojedyncza wartość). Działanie algorytmu opiera się na wyliczaniu kolejnych liczb w sposób rekurencyjny, ze wzoru

$$x_i = (ax_{i-1} + c) \text{ modulo } M,$$

gdzie a oraz c są pewnymi parametrami, a M określoną wartością modulo.

1.2.2 GLCG (Generalized Linear Congruential Generator)

Ten generator jest podobny w budowie do poprzednika, ale uogólniony. Tym razem w danych wejściowych mamy k początkowych wartości, a kolejne liczby wyznaczane są rekurencyjnie przy użyciu k poprzedników:

$$x_i = (a_1x_{i-1} + \dots + a_kx_{i-k}) \text{ modulo } M.$$

Oczywiście a_1, \dots, a_k są pewnymi stałymi.

1.2.3 RC4

Generator RC4 ma bardziej złożoną budowę. Na wejściu bierze on "klucz", czyli liczby ze zbioru wybranej wielkości. Najpierw tworzona jest permutacja identycznościowa, która następnie zostaje przekształcona za pomocą klucza na inną permutację. Odbywa się to w obrębie funkcji *KSA*. Kolejnym etapem jest użycie funkcji *PRGA*, która bierze wspomnianą permutację i odpowiednio ją aktualizuje, zwracając komplet liczb pseudolosowych.

1.2.4 Xorshift

Ostatnim z generatorów jest Xorshift, który opiera się na przekształcaniu liczby za pomocą kilku czynności. Zaczynamy od pojedynczej niezerowej wartości początkowej. Następnie aktualizujemy ją za pomocą operacji XOR i kilku przesunięć bitowych w celu "wymieszania" bitów. Zaktualizowana wartość w każdej takiej iteracji jest liczbą pseudolosową.

2 Symulacje

Gdy już znamy teoretyczną stronę naszego projektu, możemy przejść do symulacji. Każdy z testów zostanie przeprowadzony na liczbach otrzymanych poprzez działanie różnych generatorów. W ten sposób będziemy mogli porównać rezultaty.

2.1 Wstępna analiza

Zacniemy od przeprowadzenia testów na liczbach uzyskanych za pomocą każdego z generatorów. Chcemy sprawdzić hipotezy za pomocą obliczonych p-wartości. Zgodnie z definicją możemy odrzucić hipotezę zerową, jeśli p-wartość jest mniejsza od ustalonego poziomu istotności. Założymy jego najczęściej używaną wersję, czyli $\alpha = 0.05$.

Będziemy sprawdzać trzy różne wielkości próby: $n = 2^{10}$, $n = 2^{15}$ oraz $n = 2^{20}$. Pierwszym z przeprowadzanych testów jest test χ^2 . Przyjmijmy hipotezę zerową, że obserwacje są niezależne. Poniższa tabelka przedstawia przybliżone p-wartości otrzymane po uruchomieniu kodu.

	LCG	GLCG	RC4	Xorshift
2^{10}	8.7e-105	1.6e-26	4.3e-17	0.9
2^{15}	0	0	0	0.55
2^{20}	0	0	0	-

Table 1: P-wartości dla testu χ^2

Jak widać, hipoteza zerowa jest odrzucana za każdym razem w przypadku generatorów LCG, GLCG oraz RC4, co wskazuje na zależność między obserwacjami. Dla coraz większych rozmiarów próby, p-wartość zbliża się coraz bardziej do 0.

Sytuacja wygląda odmiennie w przypadku generatora Xorshift. P-wartości są duże, co uniemożliwia odrzucenie hipotezy zerowej. Przy analizowaniu generatorów przyjmuje się zazwyczaj, że te z dużymi p-wartościami są lepsze. Możemy więc uznać, że Xorshift jest dobrym generatorem pod względem sprawdzania losowości próby. Ma on jednak wady. Jak można zauważyć, w tabeli brakuje p-wartości dla próby o rozmiarze 2^{20} . Jest to spowodowane bardzo długim czasem wykonywania algorytmu - znacznie dłuższym niż w przypadku pozostałych generatorów. Przy każdej kolejnej potędze liczby 2 czas działania Xorshift coraz gwałtowniej rośnie. Z tego względu mocno utrudnione jest korzystanie z generatora, gdy potrzebujemy większych prób.

Kolejnym testem, jakim się zajmiemy, jest test Kołmogorowa-Smirnowa. Hipoteza zerowa zakłada, że próba pochodzi z rozkładu teoretycznego (o dystrybucji $F_X(x)$). P-wartości wyglądają w tej sytuacji następująco:

	LCG	GLCG	RC4	Xorshift
2^{10}	8.06e-06	5.7e-05	0.03	0.77
2^{15}	3.3e-169	1.8e-154	5.4e-32	0.27
2^{20}	0	0	0	-

Table 2: P-wartości dla testu Kołmogorowa-Smirnowa

Sytuacja wygląda analogicznie do poprzedniej - odrzucamy hipotezę zerową dla liczb otrzymanych z pierwszych trzech generatorów, a w przypadku Xorshift nie mamy podstaw do jej odrzucenia.

Jako ostatnim zajmiemy się testem serii. Przyjmujemy hipotezę zerową, że obserwacje są losowe. P-wartości przedstawiają się jak poniżej:

	LCG	GLCG	RC4	Xorshift
2^{10}	0	0.15	0.45	0.66
2^{15}	0	6.3e-14	0.23	0.43
2^{20}	-	-	-	-

Table 3: P-wartości dla testu serii

Tym razem nastąpiły już znaczące zmiany. Przede wszystkim uwagę zwraca fakt, że dla próby rozmiaru 2^{20} nie mamy żadnych rezultatów - otrzymaliśmy w wynikach "nan". Oznacza to, że dla większych rozmiarów próby kod natrafia prawdopodobnie na dzielenie przez liczby bliskie 0, co uniemożliwia zwrócenie wyniku. Test serii nie jest więc przydatny dla dużych wielkości n .

W przypadku mniejszych rozmiarów próby mamy z kolei większą różnorodność wyników w porównaniu z dwoma poprzednimi testami. Przy testach χ^2 i Kołmogorowa-Smirnowa p-wartości były tak małe, że hipoteza zerowa była odrzucana za każdym razem dla wszystkich generatorów poza Xorshift. Tutaj jest inaczej - wielkość próby gra istotną rolę w decyzji odrzucenia hipotezy. Jak widzimy, dla próby rozmiaru 2^{15} hipoteza zerowa pozostaje nieodrzucona już nie tylko w przypadku Xorshift, ale także RC4. Z kolei dla rozmiaru 2^{10} dołącza do nich również GLCG.

W każdym z powyższych testów generator LCG odrzuca hipotezę zerową niezależnie od rozmiaru próby.

2.2 Second-level testing

Zajmiemy się teraz tak zwanym "second-level testingiem". Jego celem jest głębsza analiza i większa wiarygodność wyników. Działanie takiej operacji polega na podzieleniu wygenerowanego ciągu liczb pseudolosowych na podciągi, a następnie przeprowadzeniu testów na każdym z nich. Jeśli prawdziwa jest hipoteza, że obserwacje pochodzą z rozkładu jednostajnego, to otrzymane p-wartości są niezależnymi zmiennymi losowymi o tymże rozkładzie.

Przeprowadzimy second-level testing dla prób rozmiaru $1000 * 2^{15}$ w przypadku generatorów LCG, GLCG i RC4 oraz 2^{18} w przypadku generatora Xorshift (ze względu na długość wykonywania kodu). Nie jest możliwe podwyższenie rozmiaru dla pierwszych trzech prób do $1000 * 2^{20}$, ponieważ po wywołaniu funkcji następuje awaria pamięci. Przy generowaniu liczb pseudolosowych widać bardzo dobrze, jak powolny jest Xorshift w porównaniu do pozostałych - mimo mniejszego rozmiaru wymaga do wygenerowania liczb około 380 sekund, podczas gdy przykładowo LCG potrzebuje jedynie niecałych 10. Z tego też względu długości podciągów będą mniejsze (2^{10}) niż dla pozostałych generatorów (2^{15}).

Po otrzymaniu zbioru p-wartości przedstawimy je na histogramach oraz sprawdzimy ile spośród nich odrzuca hipotezę zerową. Zaczniemy od testu χ^2 .

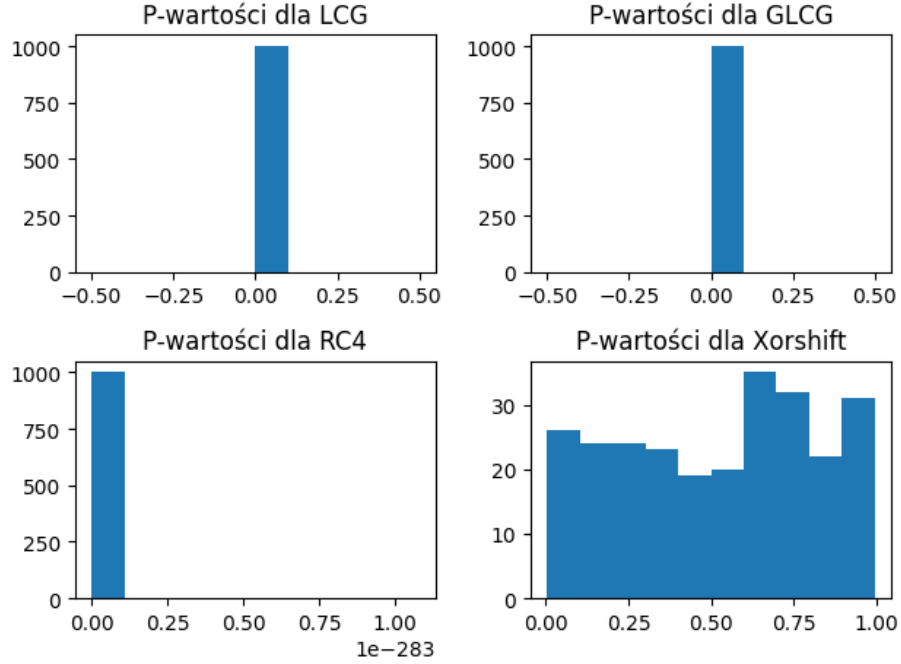


Figure 1: P-wartości dla testu χ^2

P-wartości dla pierwszych trzech generatorów prezentują się niemal identycznie. Wszystkie oscylują w okolicy 0. Widać tutaj związek z obserwacjami z poprzedniego podrozdziału - im większa jest próba, tym bardziej p-wartości zbiegają do zera. We wszystkich podciągach, których jest 1000, hipoteza zerowa jest odrzucana. Sytuacja wygląda odmiennie w przypadku ostatniego z generatorów, czyli Xorshift. W tym przypadku wielkości są zdecydowanie bardziej zróżnicowane, a hipoteza zerowa zostaje odrzucona jedynie 10 razy przy 256 podciągach. Należy pamiętać, że mamy do czynienia z mniejszym rozmiarem próby, co może mieć wpływ na wyniki, więc nie powinno się ich porównywać z pozostałymi. Warto mieć jednak na uwadze, że w pojedynczym testowaniu (w podrozdziale 2.1) sprawdzaliśmy również mniejsze rozmiary dla pozostałych generatorów, a i tak hipoteza zerowa była najczęściej odrzucana.

Gdy pierwszy etap second-level testingu mamy już za sobą, zgodnie z wytycznymi projektu przeprowadzamy test χ^2 już nie na wygenerowanych liczbach pseudolosowych, ale na otrzymanym zbiorze p-wartości. Dzięki temu otrzymujemy nową, pojedynczą p-wartość i sprawdzimy czy prowadzi ona do odrzucenia hipotezy zerowej. W poniższej tabeli przedstawione są właśnie te p-wartości, uzyskane dla wszystkich czterech generatorów.

	LCG	GLCG	RC4	Xorshift
$1000 * 2^{15}$	0	0	0	-
2^{18}	-	-	-	0.28

Table 4: P-wartości dla testu χ^2 - second-level testing

Rezultaty są porównywalne do tych z pojedynczych testów i pierwszej części second-level testingu. Odrzucamy hipotezę testową dla generatorów LCG, GLCG i RC4, dla których p-wartości wynoszą 0, nie możemy jej natomiast odrzucić dla Xorshift.

Kolejnym obiektem analizy jest test Kołmogorowa-Smirnowa, dla którego histogramy przedstawione są poniżej.

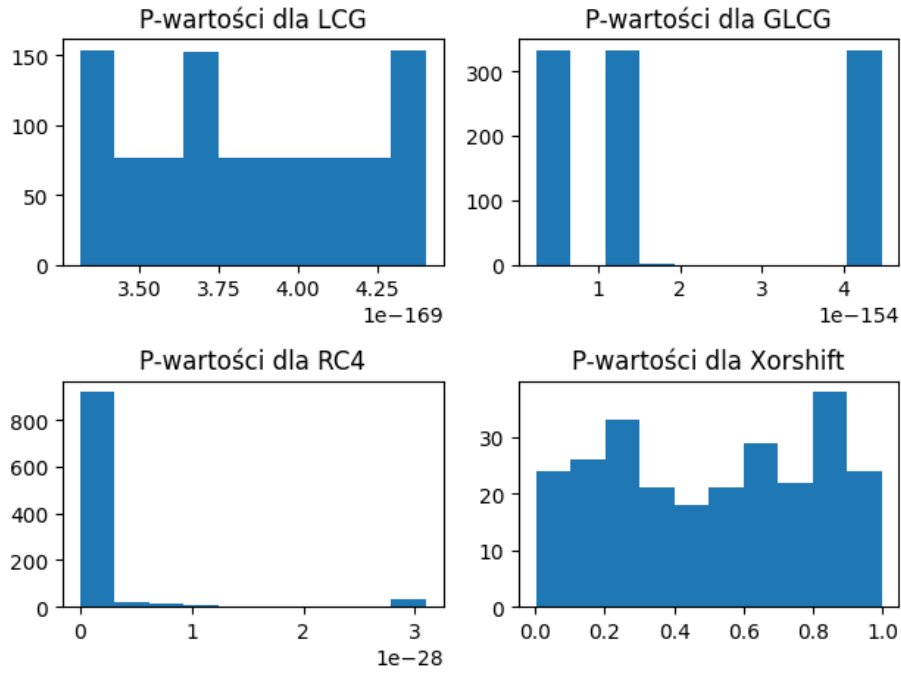


Figure 2: P-wartości dla testu Kołmogorowa-Smirnowa - second-level testing

P-wartości są tutaj bardziej zróżnicowane w stosunku do testu χ^2 . Różnice widać przede wszystkim w pierwszych trzech generatorach. P-wartości nie pozostają już w okolicach zera, ale przyjmują różne wielkości. Trzeba jednak zwrócić uwagę na skalę osi X na wykresach - mimo że wartości są większe niż poprzednio, to i tak pozostają stosunkowo małe, zatem nadal odrzucają hipotezę testową. Możemy się więc spodziewać, że "zbiórce" p-wartości, które za chwilę obliczymy, także będą prowadzić do jej odrzucenia. W przypadku generatora Xorshift sytuacja jest podobna do tej z poprzedniego

testu - odrzucamy hipotezę jedynie 8 razy. Poniżej znajduje się tabela obrazująca "zbiorcze", pojedyncze p-wartości.

	LCG	GLCG	RC4	Xorshift
$1000 * 2^{15}$	4.6e-102	0	0	-
2^{18}	-	-	-	0.11

Table 5: P-wartości dla testu Kołmogorowa-Smirnowa - second-level testing

Zgodnie z przypuszczeniami odrzucamy hipotezę zerową dla pierwszych trzech generatorów. Nie możemy jej odrzucić przy generatorze Xorshift.

Ostatnim z testów w tej części jest test serii, który miał najbardziej różnorodne rezultaty w pojedynczym testowaniu. Sprawdźmy czy w second-level testingu możemy uzyskać podobne.

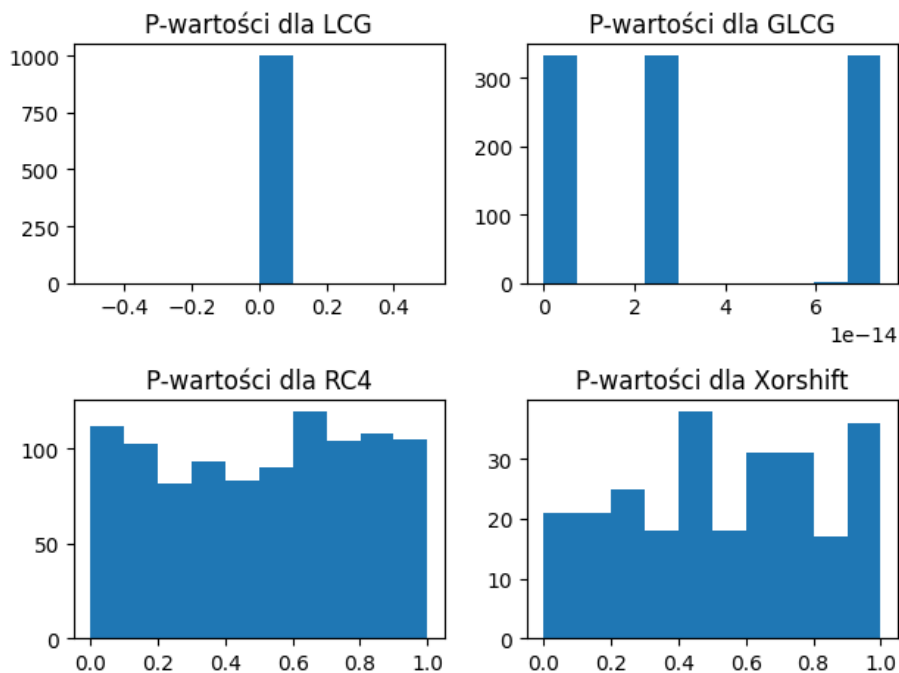


Figure 3: P-wartości dla testu serii - second-level testing

Łatwo zauważyć, że ze względu na bardzo małe p-wartości zarówno generator LCG jak i GLCG odrzucają hipotezę zerową. Xorshift zachowuje się podobnie do poprzednich testów (hipoteza odrzucona 13 razy). Jednakże w przeciwieństwie do second-level testingu dla testu χ^2 i Kołmogorowa-Smirnowa, tym razem generator RC4 ma dużo większe i bardziej różnorodne p-wartości, zbliżające się nawet do 1. Z tego względu hipoteza zostaje odrzucona tylko 63 razy przy 1000 podciągach, co jest istotną różnicą, zważywszy, że we wcześniejszych testach

100% p-wartości dla RC4 powodowało odrzucenie. Mamy tu jednak analogię do pojedynczego testowania, gdzie w teście serii p-wartość była wystarczająco duża, by uniemożliwić odrzucenie hipotezy zerowej. Sprawdźmy jeszcze p-wartości powstałe po przeprowadzeniu testu na powyższych zbiorach.

	LCG	GLCG	RC4	Xorshift
$1000 * 2^{15}$	0	0	0.06	-
2^{18}	-	-	-	8.7e-05

Table 6: P-wartości dla testu serii - second-level testing

Wyniki są dość zaskakujące. O ile zgodnie z przewidywaniami algorytmy LCG i GLCG odrzuciły hipotezę, o tyle dołączył do nich Xorshift, który dotąd jej nie wykluczał. Jak widać nie zawsze cechy zbioru odzwierciedlają cechy jednostki. Generator RC4 potwierdził natomiast poprzednie wyniki.

Podobne do siebie wyniki w różnych testach, zwłaszcza gdy p-wartości są małe, mogą świadczyć o zachodzeniu pewnych wzorców i sugerować, że nie mamy pełnej losowości. Praktyczne stosowanie generatorów może być więc utrudnione, gdy pracujemy w dziedzinie wymagającej, by losowość była bardzo duża i niezaburzona. Widać więc, że przykładowo algorytm LCG może nie dawać wystarczająco losowych wyników, by stosować go do bardziej zaawansowanych zadań, gdzie losowość ma kluczowe znaczenie. Podobnie jest z generatorem GLCG, który w większości analizowanych przez nas przypadków także odrzucał hipotezę o niezależności obserwacji.

Pod względem losowości najlepiej sprawdza się Xorshift, który najczęściej zwraca wysokie p-wartości, które potwierdzają losowość. Jest on niestety znacznie wolniejszy w wykonywaniu operacji, co jest minusem gdy potrzebujemy dużych prób. RC4 z kolei wydaje się być czymś "pomiędzy" - jego losowość jest lepsza niż algorytmów LCG i GLCG, jest w miarę szybki, ale wciąż odrzuca hipotezę zerową znacznie częściej niż Xorshift. Poza tym znaczący wpływ na p-wartości ma tutaj dobór testu statystycznego.

2.3 Frequency monobit test dla wybranych liczb niewymiernych

Kolejnym zadaniem, jakie przed nami stoi, jest zastosowanie testu frequency monobit dla kilku liczb - a dokładniej dla π , $\sqrt{2}$ oraz stałej Eulera. Test operuje na zapisie rozwinięcia liczby w systemie binarnym i opiera się na analizie częstości występowania 0 i 1. Jeśli liczba zer i jedynek jest mniej więcej podobna, to możemy wnioskować, że mamy do czynienia z losowością.

Zacniemy podobnie jak w przypadku testowania generatorów, czyli od pojedynczych prób, a następnie przeprowadzimy second-level testing. Poniżej przedstawione są przybliżone p-wartości wygenerowane w pojedynczym teście.

π	e	$\sqrt{2}$
0.61	0.93	0.82

Table 7: P-wartości dla testu frequency monobit

Widzimy, że przy żadnej z liczb zdecydowanie nie możemy odrzucić hipotezy zerowej, która zakłada losowość ich rozwinięć. Wysoka p-wartość dla stałej Eulera może sugerować, że lepiej spełnia ona warunek losowości niż pozostałe, a najwięcej potencjalnych odchyłeń można znaleźć w rozwinięciu liczby π .

Teraz przeprowadzimy second-level testing. Zrobimy to dla dwóch różnych długości podciągów. Poniżej zobrazowane są p-wartości dla liczby π .

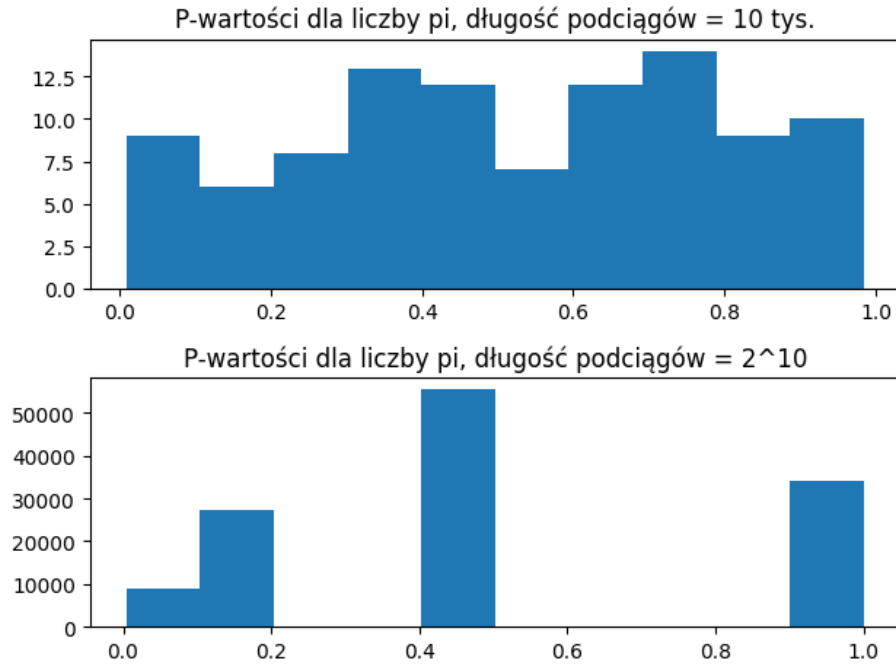


Figure 4: P-wartości dla liczby π - second-level testing

Jak można zauważyć, długość podciągów ma w tej kwestii znaczenie. P-wartości w pierwszym przypadku (długość 10 tys.) są znacznie bardziej zróżnicowane, co kontrastuje z drugim histogramem. W sytuacji pierwszej hipotezę zerową odrzucono 7 razy przy 100 podciągach, a w drugiej - 8768 razy przy 125610 podciągach. Interesujące są też wyniki drugiej części second-level testingu, gdy otrzymamy ostateczne pojedyncze p-wartości. Dla próby o mniejszej długości podciągów p-wartość wynosi 0.3, co nie pozwala odrzucić hipotezy, jednak w drugim przypadku mamy 0, co ją wyklucza. Jest to ciekawy rezultat, bo sugeruje, że losowość rozwinięcia liczby π zależy od tego jak długie podciągi

wyberzemy.

Zajmiemy się teraz liczbą e w analogiczny sposób.

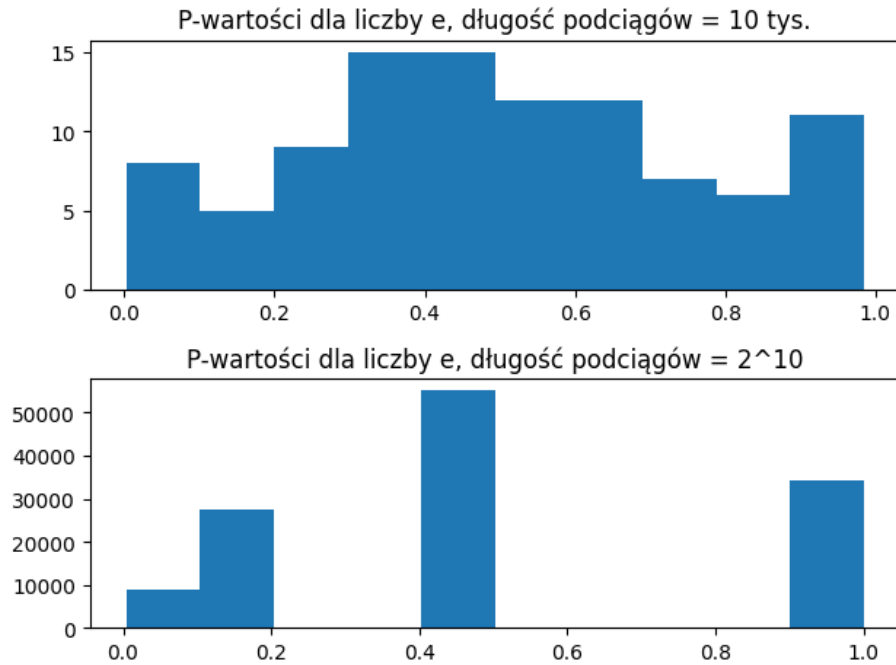


Figure 5: P-wartości dla liczby e - second-level testing

Jak widać sytuacja jest bardzo podobna do przypadku liczby π . Mniejsza długość podciągów powoduje większe zróżnicowanie p-wartości. Przy długości równej 10 tys. hipotezę testową wykluczono 5 razy, a przy długości 2^{10} - 8806 razy. Finalne p-wartości również zachowują się analogicznie: w przypadku krótszych podciągów nie odrzucamy hipotezy (p-wartość równa 0.23), ale dla tych dłuższych - już tak (p-wartość równa 0). Możliwe, że przy dłuższych podciągach test staje się bardziej czuły na odchylenia.

Ostatnią liczbą jaką się zajmiemy jest $\sqrt{2}$.

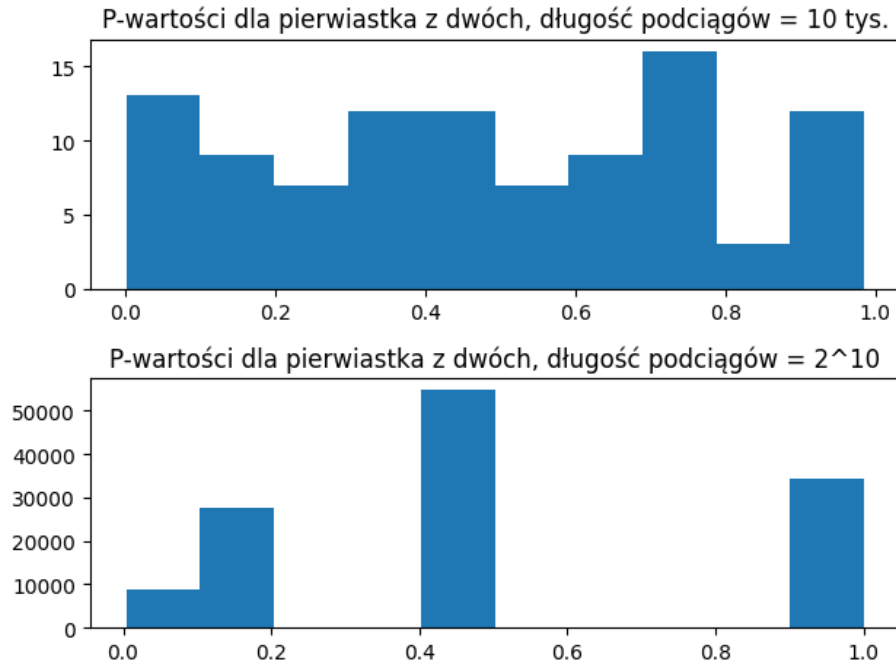


Figure 6: P-wartości dla liczby $\sqrt{2}$ - second-level testing

Ponownie napotykamy analogie do dwóch poprzednich liczb. Dla mniejszej długości podciągów hipotezę zerową odrzucono 6 razy, a dla większej - 8960 razy. Poza tym, tak jak zdarzyło się wcześniej, przy zbadaniu ostatecznych pojedynczych p-wartości okazuje się, że nie wykluczamy hipotezy przy pierwszym przypadku (p-wartość równa 0.1), ale odrzucamy ją w drugim (p-wartość równa 0).

Podsumowując, wszystkie trzy analizowane liczby spełniają warunki losowości, jednak przy dużych rozmiarach podciągów w second-level testingu możemy napotkać na pewne zaburzenia, co prowadzi do odrzucenia hipotezy o losowości.

2.4 Dodatkowe dane

Tabele poniżej przedstawiają czas potrzebny do wykonania poszczególnych algorytmów. Może to być przydatne do porównywania ich działania. Pierwsze dwie tabele dotyczą użytych wyżej generatorów.

Funkcja	LCG	GLCG	RC4	Xorshift
Czas				
wykonania	0.01	0.08	0.02	6.44

Table 8: Czas działania generatorów w sekundach (rozmiar próby: 2^{15})

Funkcja	LCG	GLCG	RC4	Xorshift
Czas				
wykonania	9.7	46.57	17.46	380.34

Table 9: Czas działania generatorów w sekundach (rozmiar próby: $1000 * 2^{15}$ dla LCG, GLCG i RC4 oraz 2^{18} dla Xorshift)

Ostatnia tabelka odnosi się do działania testu frequency monobit.

Liczba	π	e	$\sqrt{2}$
Czas			
wykonania	3.14	2.72	2.36

Table 10: Czas działania testu frequency monobit w sekundach dla różnych liczb

Poniżej zamieszczone są jeszcze dwie tabele, opisujące które funkcje i algorytmy zostały użyte do poszczególnych celów w tym projekcie. Pierwsza dotyczy tabel, a druga histogramów (czy też zestawów histogramów). Nie uwzględniamy tu generatorów, ponieważ są one używane przez cały czas. Dla ułatwienia odczytu poniżej napisany jest klucz z oznaczeniami poszczególnych funkcji.

Numer tabeli	1	2	3	4	5	6	7
Użyte funkcje	A,B	A,C	A,D	A,B	A,B	A,B	A,I

Table 11: Wykaz funkcji użytych do wygenerowania danych w tabelach

Numer wykresu	1	2	3	4	5	6
Użyte funkcje	E,H	F,H	G,H	H,J	H,J	H,J

Table 12: Wykaz funkcji użytych do wygenerowania danych na histogramach

Klucz do obu tabel:

- A - p_val_test
- B - chisquare (wbudowana)
- C - ks_1samp (wbudowana)
- D - runs_test
- E - second_level_testing_chi_square
- F - second_level_testing_ks
- G - second_level_testing_runs_test
- H - sec_level_test_p_val
- I - freq_monobit
- J - second_level_testing_freq_monobit