Testy NIST

a) “Słaby” generator liczb losowych

Test name	Result value (P-value)	Status
1. Frequency (Monobit) Test	0.6921050282223633	Passed
2. Frequency Test within a Block	0.2814316573304481	Passed
3. Runs Test	0.9887048705999201	Passed
4. Test for the Longest Run of Ones in a Block	2.8603719771577458e-8	Failed
5. Binary Matrix Rank Test	7.302098054879341e-19	Failed
6. Non-overlapping Template Matching Test	0.8430128393186767	Passed
7. Overlapping Template Matching Test	1.3021860140165902e-9	Failed
8. Maurer's "Universal Statistical" Test	0.020739675974802863	Passed
9. Linear Complexity Test	0.6360858268287162	Passed
10. Serial Test	P-value 1: 0.9245205501339924 P-value 2: 0.9904256139112977	Passed
11. Approximate Entropy Test	0.995382559351598	Passed
12. Cumulative Sums (Cusum) Test	P-value Forward: 0.9814860915594663 P-value Reverse: 1	Passed
13. Random Excursions Test	0.00000249571223592112	Failed
14. Random Excursions Variant Test	0.039523370614497844	Passed

b) **“Dobry” generator liczb losowych**

1. Frequency (Monobit) Test	0.4237107971667935	Passed
2. Frequency Test within a Block	0.16037942404130748	Passed
3. Runs Test	0.6436222198366819	Passed
4. Test for the Longest Run of Ones in a Block	0.13751420178276527	Passed
5. Binary Matrix Rank Test	0.9097497764370372	Passed
6. Non-overlapping Template Matching Test	0.5425249647165342	Passed
7. Overlapping Template Matching Test	0.3300057234913645	Passed
8. Maurer's "Universal Statistical" Test	0.9766297754893561	Passed
9. Linear Complexity Test	0.6535102487452833	Passed
10. Serial Test	P-value 1: 0.6532469784731042 P-value 2: 0.6455162381736883	Passed
11. Approximate Entropy Test	0.6670695500299224	Passed
12. Cumulative Sums (Cusum) Test	P-value Forward: 0.7495846569565325 P-value Reverse: 0.5947211419319975	Passed
13. Random Excursions Test	0.30198519493328113	Passed
14. Random Excursions Variant Test	0.18210609497479158	Passed

c) **Output funkcji hashującej SHA-1 dla nazwiska**

1. Frequency (Monobit) Test	0.7503158528871919	Passed
2. Frequency Test within a Block	0.11161176275290287	Passed
3. Runs Test	1.2437176699105235	Failed
4. Test for the Longest Run of Ones in a Block	0.8061121721583822	Passed
5. Binary Matrix Rank Test		Error
6. Non-overlapping Template Matching Test		Error
7. Overlapping Template Matching Test		Error
8. Maurer's "Universal Statistical" Test		Error
9. Linear Complexity Test		Error
10. Serial Test		Error
11. Approximate Entropy Test	0.34855913109112635	Passed
12. Cumulative Sums (Cusum) Test	P-value Forward: 0.30425108162486036 P-value Reverse: 1	Passed
13. Random Excursions Test		Error
14. Random Excursions Variant Test		Error

OMÓWIENIE TESTÓW

'Dobry' generator w większości przypadków pozytywnie przechodził wszystkie testy. Dla ponownie generowanych miliona bitow wyniki testów bywały jednak różne. Czasami lepsze, a czasami nawet gorsze od 'słabego' generatora. 'Słaby' generator jednak zawsze osiągał te same wyniki negatywne ponieważ przez jego ograniczoną losowość output prawie wcale się nie zmieniał. 'Dobry' generator okazał się znacznie skuteczniejszy jednak jego skuteczność i tak nie jest perfekcyjna. Output funkcji hashującej dla wielu testów był za krótki.

1. **Monobit test**
   Wynik tego testu powinien być jak najbliższy wartości 0.5 dla dobrego generatora liczb losowych, ponieważ jest to stosunek wylosowanych zer do wszystkich wylosowanych bitów. Dla „słabego" generatora oraz outputu funkcji hashującej testy przeszły pozytywnie lecz wartości są podobnie dalekie od 0.5 w porównaniu do wyniku dla 'dobrego' generatora. W outpucie 'słabego' generatora oraz funkcji hashującej wystąpiło więcej zer niż jedynek.

2. **Frequency Test within a Block**
   Test ten polega na tym aby sprawdzić czy częstotliwość występowania jedynek ciągu zer i jedynek podzielonym na M-bitowe bloki. W każdym bloku częstotliwośc powinna wynosić M/2. Dla 'dobrego' generatora wynik testu powinien być liczbą bliską 0, najlepiej z przedziału (0.01 , 0.1). Dla używanego w testach 'dobrego' generatora widać że otrzymany wynik jest zdecydowanie lepszy od wyniku 'słabego' generatora. Najlepiej wypada output funkcji SHA-1 lecz dla innego tekstu, np. 'Hello World' wynikem jest ok. 0.85 więc output ten nie jest odpowiednio dobrze losowy.

3. **Runs Test**
   Test polega na obliczniu różnicy ilości ciagów jedynek i ciągów zer w outpucie z wartością oczekiwaną tej ilości oraz podzieleniu otrzymanego wyniku przez odchylenie standardowe tej liczby. Dla 'dobrego' generatora liczb losowych wartość bezwzględna otrzymanego wyniku powinna być jak najmniejsza w zależności jak duży poziom losowości badamy. Dla użytego 'dobrego' generatora liczb losowych wynik był wystarczająco mały aby przejść test natomiast dla funkcji sha-1 i 'słabego' generatora wynik był zbyt duzą liczbą co oznacza, że ciągów częstotliwośc występowania ciągów rosnących była znacznie większa niż malejących lub odwrotnie.

4. **Test of Longest Run of Ones**
   Output funkcji hashującej jest za krótki aby mógł zostać poddany temu testowi. Dla 'dobrego' generatora widać, że najdłuższy ciąg jedynek który wystąpił w outpucie był zdecydowanie krótszy od najdłuższego ciągu w outpucie 'słabego' generatora co świadczy o jego większej losowości.

5. **Binary Matrix Rank**
   Z otrzymanego ciągu zer i jedynek tworzy się macierze i liczy ich rząd, czyli maksymalny możliwy stopień niezerowego minora tej macierzy. Czestotliwość występowania. Odchylenie standardowe otrzymanych rzędów nie powinno być bliskie 0. Dla ‘ dobrego’ generatora odchylenie było odpowiednie aby przejść test pozytywnie. ‘Słaby’ generator wypadł bardzo negatywnie.

6. **Non-overlapping Template Matching**
   Celem testu jest wykrycie częstotliwości występowania ciągów różnych od reszty w outpucie generatorów. Output funkcji SHA-1 jest za krótki aby go przetestować. W ‘słabym’ oraz ‘dobrym’ generatorze występowanie takich samych ciagów było wystarczająco wysokie aby przejść test pozytywnie.

7. **Overlapping Template Matching**
   Celem testu jest wykrycie powtarzających się ciągów w outpucie generatorów. Dla ‘słabego’ generatora czestotliwośc występowania była zbyt wysoka aby przejść test pozytywnie. W ‘dobrym’ generatorze takie same ciągi powtarzały się znacznie rzadziej. Świadczy to o ‘lepszej’ losowości ‘dobrego’ generatora.

8. **Maurer’s Universal Statistical Test**
   Test pokazuje wariancje miedzy podanym outputem generatora, a outputem prawdziwego bardzo skutecznego generatora liczb losowych. Jak widąc na załączonych obrazkach zarówno ‘dobry’ jak i ‘ słaby’ generator losowy przeszły testy pozytywnie, jednak ten drugi wypadł w tym przypadku lepiej. Po wielu testach i generowaniu różnych ciagów ‘dobry’ generator potrafił osiągnąć w tym teście wynik np. 0,05.

9. **Linear Complexity Test**
   Test ten jest przeprowadzany na rejestrze zbudowanym z przerzutników połącznoych ze sobą w taki sposób że w takt impulsów zegarowych przechowywana informacja bitowa przemieszcza się. Bitem wejściowym rejestru jest funkcja linowa z jego poprzedniego stanu. Za pomocą

algorytmu Berlekampa Masseya można zbadać długość tych rejestrów. ‘Dobry’ losowy generator charakteryzuje się długimi rejestrami.

10. Serial Test

Celem testu jest określenie czy liczba nakładających się wzorców jest w przybliżeniu taka sama, jak w oczekiwaniu od prawdziwie dobrego generatora losowego.
Zarówno ‘słaby’ jak i ‘dobry’ generator losowy przeszły test pozytywnie. Drugi wypadł w tym przypadku lepiej tzn. wartość była bliższa oczekiwanej.

11. Approximate Entropy Test

Celem jest zbadanie częstości nakładanie się sąsiadujących bloków z oczekiwanym wynikiem dla sekwencji ‘dobrego’ generatora. W tym przypadku użyty ‘dobry’ generator nakładał się częściej niż ‘słaby’. Funkcja SHA-1 dla mojego nazwiska również nakładała się częściej niż sekwencja ze ‘słabego’ generatora.

12. Cusum Test

Test polega na symulacji błądzenia losowego gdzie (bit 1)= +1 a (bit 0)= -1. Test będzie pozytywny dla sekwencji w których maksymalne oddalenie od 0 podczas przeprowadzania symulacji będzie jak najmniejsze. ‘Dobry’ generator wypadł tutaj najlepiej.

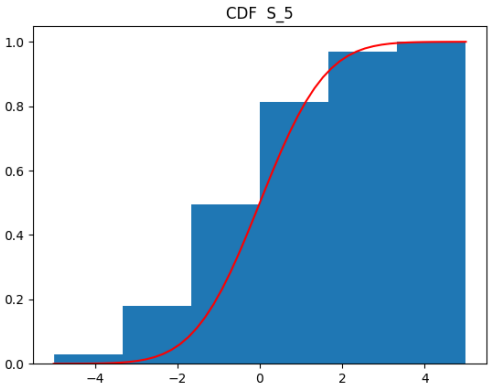
13. Random Excursion

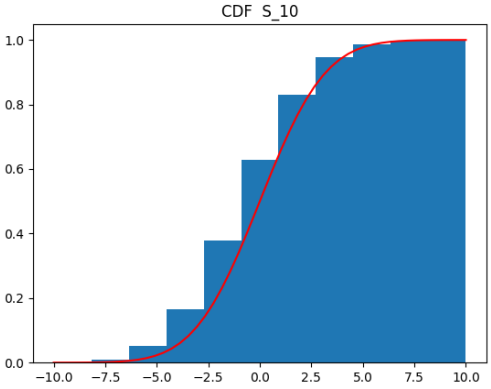
W tym teście bada się częstotliwość znajdowania się wykresu na danym poziomie podczas błądzenia losowego. ‘Dobry’ generator występował na tym samym poziomie z odpowiednia częstotliwością, która pozwoliła mu na pozytywne przejście testu.
Dla ‘słabego’ generatora ta częstotliwośc była za mała.

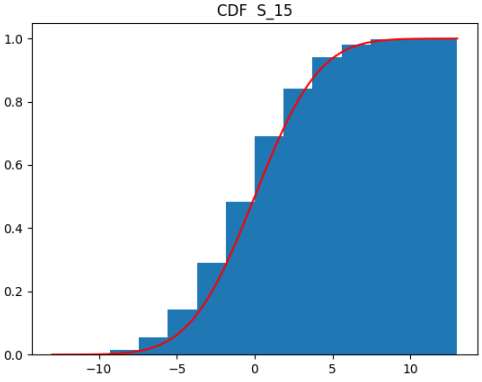
14. Random Excursions Deviation

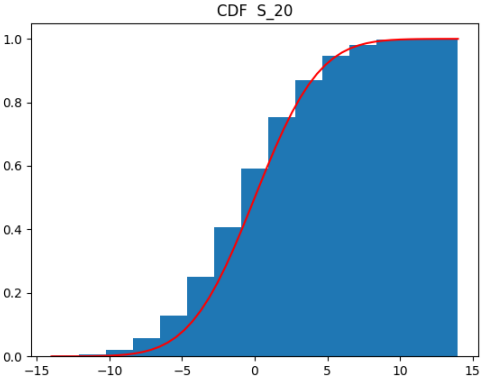
Test polega na policzeniu odchylenia między ilością razy kiedy wykres błądzenia losowego przebywał na danym poziomie. Dla ‘dobrego’ generatora odchylenie było mniejsze od ‘słabego’ generatora więc wypada on lepiej w tym teście.

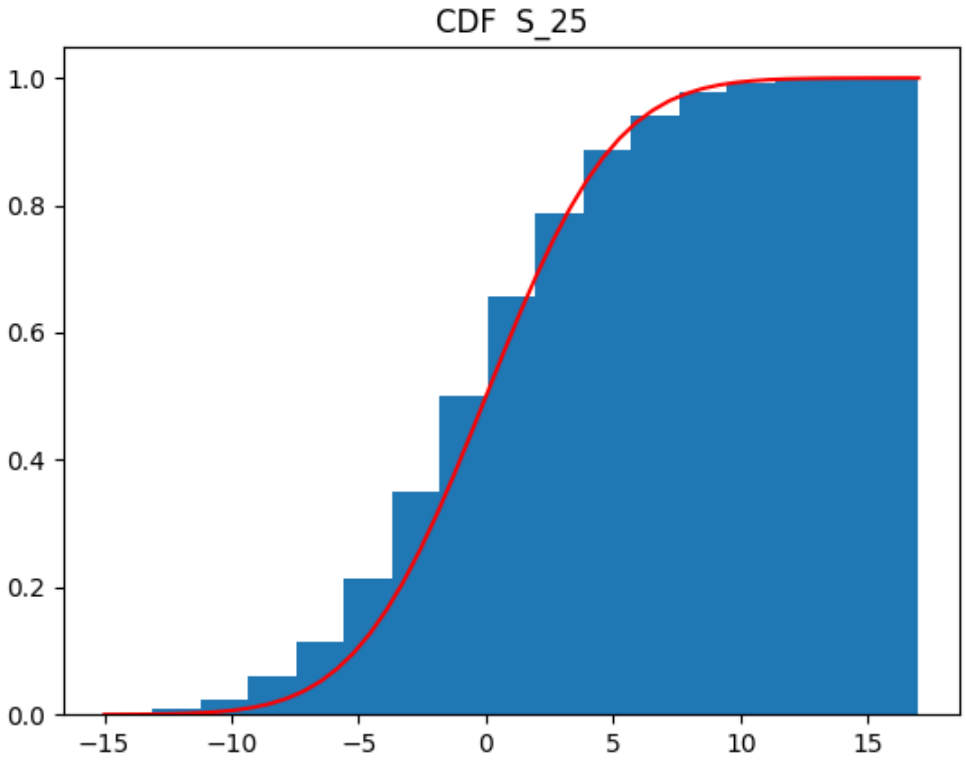
Zadanie 2

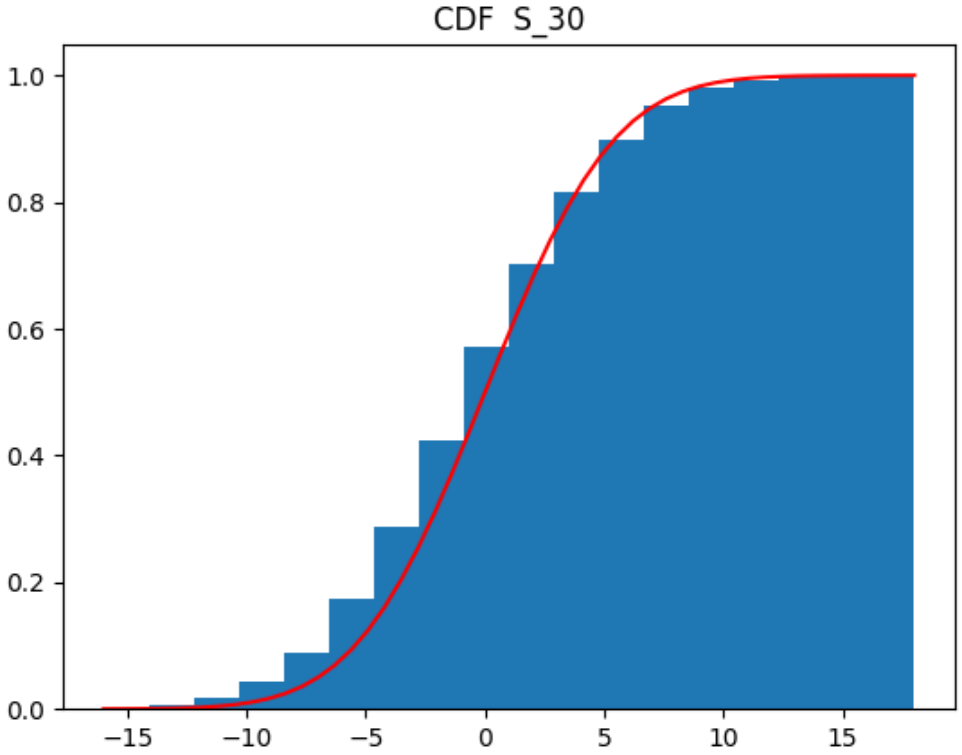

CDF S_5
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

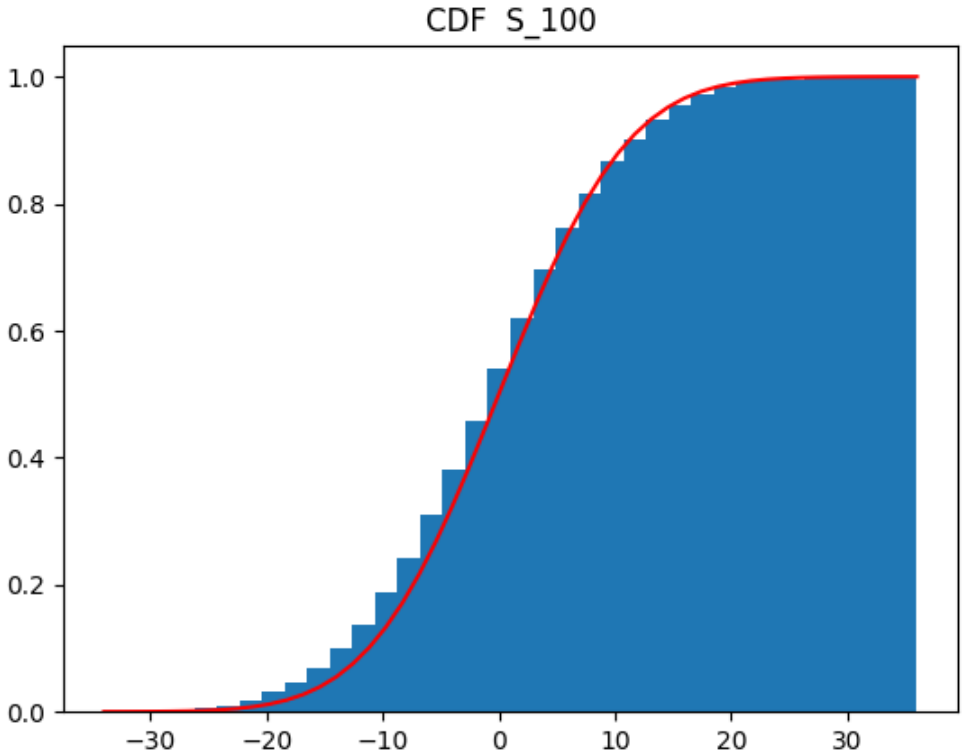

CDF S_10
1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


CDF S_15
1.0
0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10


CDF S_20
1.0
0.8
0.6
0.4
0.2
0.0
−15
−10
−5
0
5
10
15


CDF S_25
1.0
0.8
0.6
0.4
0.2
0.0
−15
−10
−5
0
5
10
15





Powyższe wykresy przedstawiają wyznaczone dystrybuanty zmiennych losowych S_N zdefiniowanych poleceniu zadania 2. Czerwonym kolorem

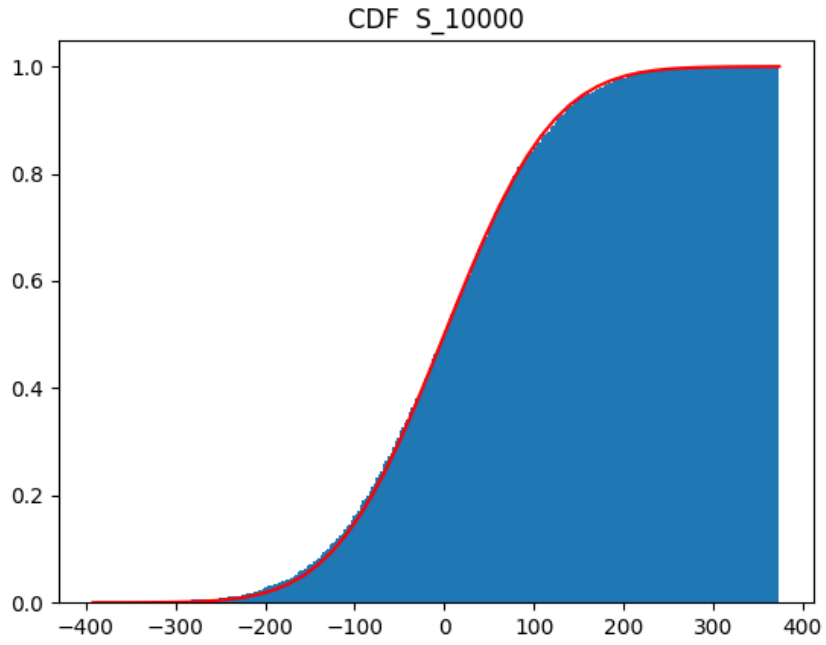
zaznaczony jest wykres funkcji dystrybuanty rozkładu normalnego o wartości oczekiwanej 0 oraz wariancji równej x. Można ją wyznaczyć ze wzoru:

$$0 \leq i \leq N$$

$$x = \frac{max\{S_i\} - min\{S_i\}}{8}$$

```
x2 = np.linspace(min(allsn), max(allsn))
y2 = ss.norm(loc=0, scale=(max(allsn)-min(allsn))/8).cdf(x2)
```

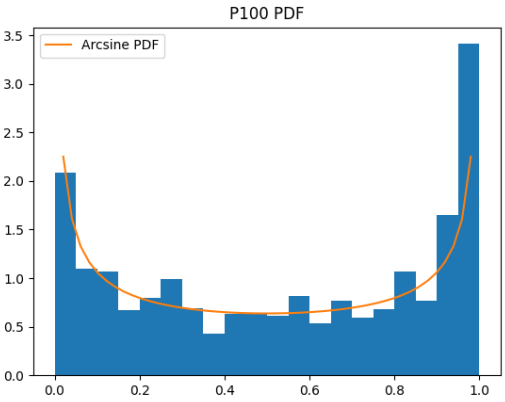
Na wykresach możemy zauważyć że dystrybuanta zmiennej losowej S_N aproksymuje dystrybuantę rozkładu normalnego o powyżej podanych parametrach z coraz większą dokładnością wraz z rosnącym N.

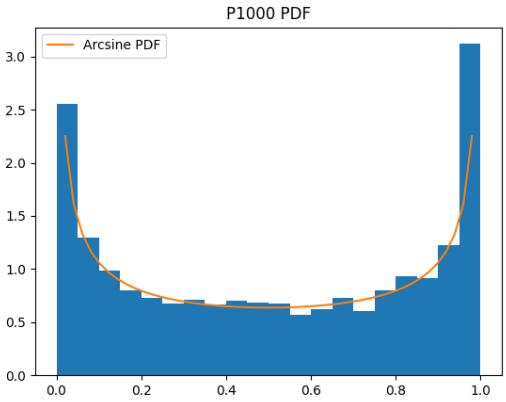


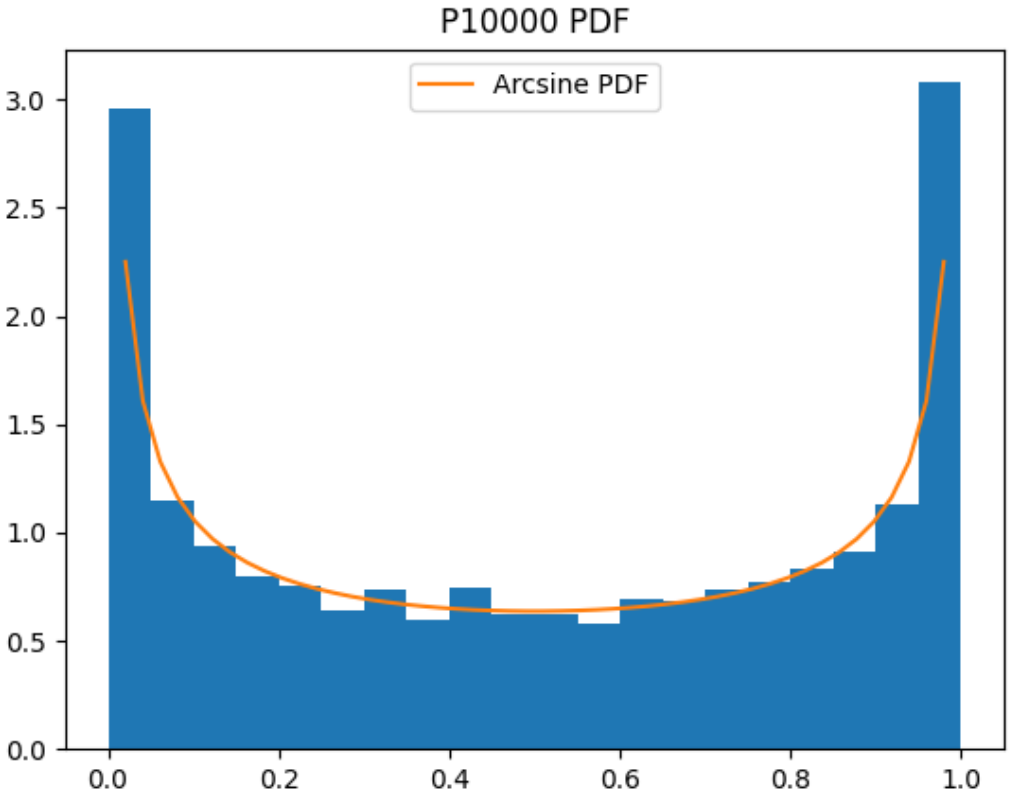
Jest to dobra metoda aproksymacji CDF rozkładu normalnego.

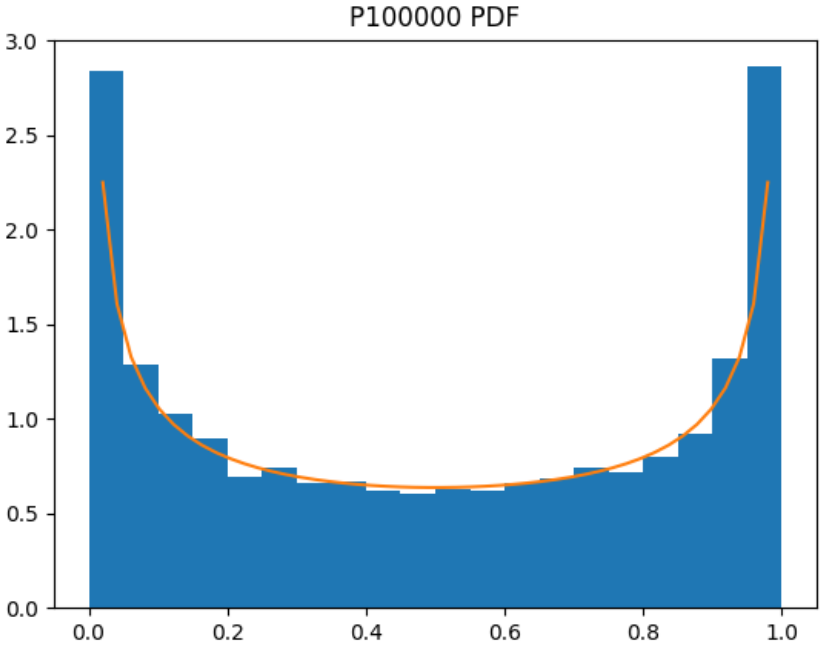
Poza tym możemy wyciągnąć wnioski dotyczące zmiennej losowej Sn. Zgodnie z rozkładem normalnym zmienna najczęściej przyjmuje wartości bliskie wartości oczekiwanej czyli bliskie 0.

Zadanie 3


P100 PDF
Arcsine PDF
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0

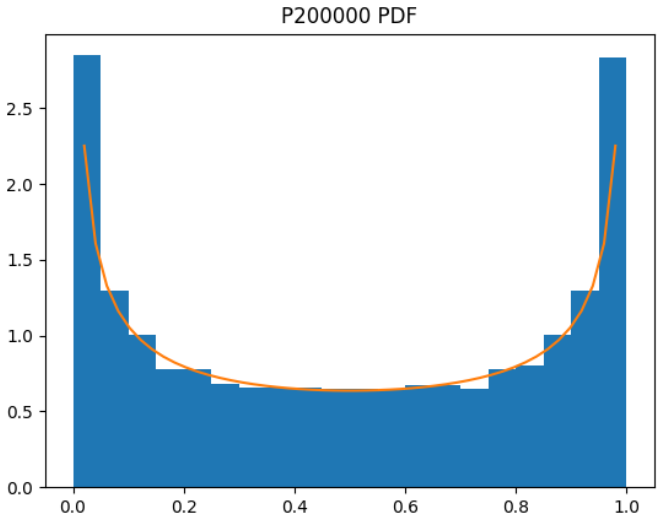

P1000 PDF
Arcsine PDF
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0


P10000 PDF
Arcsine PDF
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0



Na powyższych wykresach znaduje się wyznaczona estymacja funkcji gęstości prawdopodobieństwa zmiennej losowej będącą frakcją czasu, która błądzenie losowe ‘spędziło nad osią OX’. Pomarańczowym kolorem przedstawiony jest wykres PDF zmiennej losowej z rozkładem arcusa sinusa. Wraz z rosnącą liczbą N czyli dlugościa błądzenia losowego oraz zwiększoną liczbą powtorzen symulacji PDF wykresy obu zmiennych losowych wyglądają asymptotycznie coraz podobniej. Wyznaczanie PDF zmiennej losowej frakcji czasu, którą błądzenie losowe ‘spędziło nad osią OX’ jest dobrą metodą aproksymacji PDF zmiennej losowej o rozkładzie arcusa sinusa.

Wykres dla N = 200 000 oraz k = 100 000:



Gęstośc dla skrajnych wartości przyjmowanych przez Ln czyli 0 i 1 nadal jest większa niż przy rozkładzie arcsin, lecz widać że otrzymywany histogram jest coraz dokładniejszy.