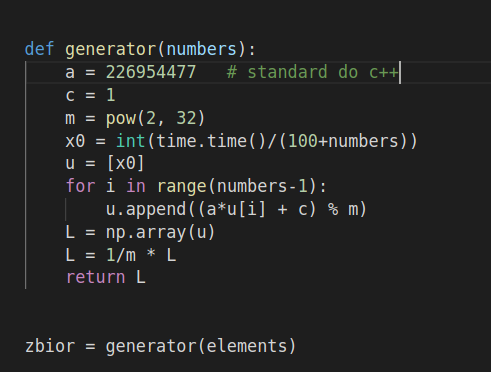
Maja Szymajda, 254313

Sprawozdanie lista 3

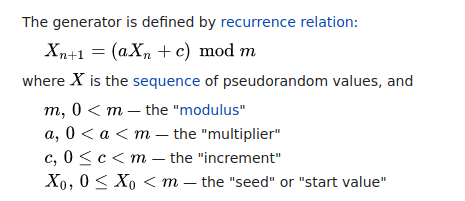
Do realizacji listy trzeciej użyłam środowiska Python3, napisałam w nim program wykorzystujący następujące biblioteki: matplotlib.pyplot, scipy, statistics, numpy, time oraz math.



Zadanie 1 i 2

Do wykonania dwóch pierwszych zadań stworzyłam funkcje o nazwie generator, która przyjmuje jako argument ilość elementów w tym przypadku 20 i 100. Jest to generator liniowy liczb pseudolosowych typu LCG (ang. pseudorandom number linear congruential generator). Funkcja ta wykorzystuje zmienne: a – mnożnik, c – przyrost, m – moduł, x0 – ziarno, u[x0] – tablica liczb pseudolosowych oraz L – zbiór naszych liczb pseudolosowych.

Aby wygenerować odpowiednia ilość elementów w tablicy funkcja dodaje do tablicy u odpowiedni element tablicy u [i] (gdzie i jest iteratorem od 0 do ilość elementów – 1) mnożona przez mnożnik a, dodaje do niej przyrost. Z otrzymanej liczby licznymy modulo m . Po przejściu przez cały for mamy w tablicy u mamy 20 lub 100 liczb pseudolosowych, nie są one jednak z zakresu [0,1). Aby otrzymać taki zakres mnożymy naszą tablice przez 1/m, co w efekcie da nam liczby pseudolosowe z rządanego zakresu. Aby wykonać to zadanie posiłkowałam się poniższym wzorem oraz tabelką.



Przykładowy zbiór 20 elementów:

[0.00311208 0.13226321 0.67886042 0.29317098 0.95341206 0.63367712

0.53089164 0.94798236 0.85321562 0.83640141 0.18401362 0.52638306

0.98922659 0.396518 0.50390778 0.64628325 0.54174544 0.1314209

0.38800524 0.23543422]

Przykładowy zbiór dla 100 elementów:

[0.00186725 0.0809568 0.39041862 0.93905605 0.50141336 0.9904841

0.96458503 0.79107867 0.96296681 0.09339545 0.38500452 0.31349546

0.04885892 0.87222882 0.63549396 0.90942812 0.57945251 0.98563532

0.70821295 0.83813469 0.58019996 0.98365019 0.43538991 0.39658911

0.12850599 0.57667473 0.55054034 0.49070927 0.5472702 0.06754417

0.40105228 0.58060845 0.24198417 0.30605647 0.48401391 0.73424525

0.22825835 0.27533708 0.72715841 0.2738469 0.14665362 0.16941461

0.18680098 0.24948107 0.31169248 0.57138247 0.584643 0.72415106

0.09192499 0.09152141 0.85531843 0.60478947 0.66792437 0.57475803

0.62655741 0.50352526 0.96508731 0.73683419 0.30384199 0.86359932

0.30811392 0.83856243 0.18142909 0.33430339 0.09404242 0.75754222

0.75161566 0.12931752 0.5307135 0.56671223 0.13412095 0.85282084

0.23753934 0.47911354 0.29746108 0.02775089 0.25545236 0.76823872

0.49419071 0.31058632 0.76290862 0.02334191 0.08188691 0.12702801

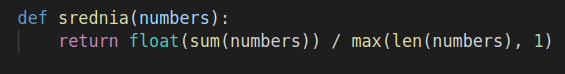
0.54482095 0.84908482 0.65175029 0.21295374 0.01129672 0.86889364

0.13669819 0.29502113 0.31305534 0.64140194 0.32082518 0.89529414

0.85040041 0.84935446 0.9221057 0.99426799]

Zadanie 3

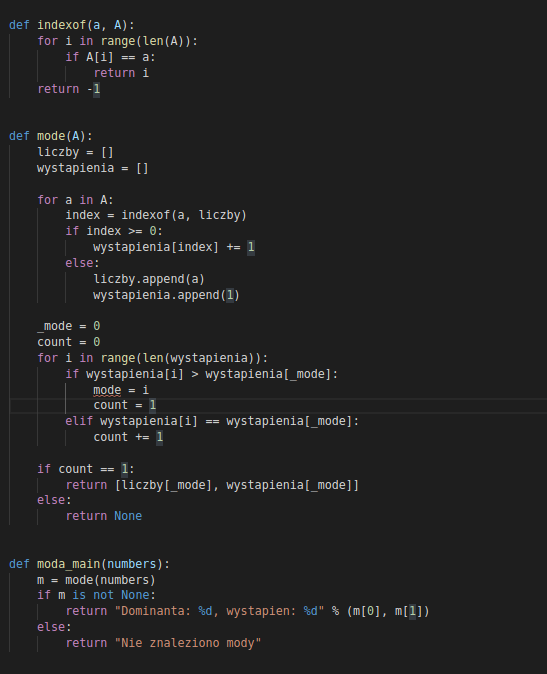
Aby wykonać drugie zadanie musiałam wyliczyć: średnią, medianę, modę, odchylenie standardowe, wariancję, skośność, kurtozę oraz co najmniej jeszcze jedną wybraną przez studenta – średnią harmoniczna.

 Średnia – pierwszą częścią zadania drugiego było wyliczenie średniej arytmetycznej ze zbioru 20 i 100 elementowego. Zrobiłam to za pomocą działania: float(sum(numbers)) / max(len(numbers), 1). Gdzie sum(numbers) to suma wszystkich wygenerowanych liczb a funkcja len(numbers) sprawdza długość tablicy, która później w funkcji max(len(numbers),1) ulega sprawdzeniu. Jeżeli jest mniejsza od 1 to algorytm wybierze większą z nich. Cała funkcja srednia wygląda następująco:

przyjmuje ona jako argument zbiór liczby wygenerowanych przez generator.

Aby obliczyć Medianę czyli wartość środkową można posortować wylosowane liczby i wziąć średnią z dwóch z nich które znajdują się w środku posortowanej listy lub wykorzystać funckje wybudowana w biblioteke numpy: np.median(zbior). Wyliczyła ona wartość środkowa dla zbioru 20 i 100 elementowego.

Do obliczenia mody wykorzystuje funkcje która sprawdza czy jakas liczba powtarza się więcej niż raz i ile razy. Wskazuje ona najczęściej powtarzającą się wartość w zbiorze. W z wziązku z tym ze

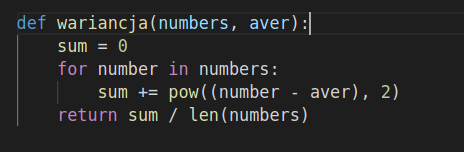
w losowanym przez nas przedziale jest olbrzymia ilość małych liczb to prawdopodobieństwo że jakaś się powtórzy jest bardzo małe. Kod do funkcji:

1. Funkcja indexofia(a,A) – sprawdza czy w zbiorze liczb powtarza się jedna liczba a. Argumentami: A - jest macierz z 20 i 100 elementami, natomiast a – jedna sprawdzana liczba.

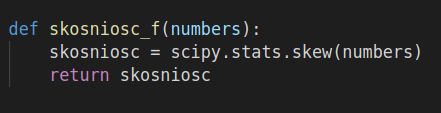
2. Funkcja mode(A) – jej zadaniem jest przejscie po tablicy i zliczenie wystąpień danych liczb oraz wybrania liczby z maksymalną ilością powtórzeń. Jej argumentem jest zbiór prób.

3. Funkcja moda\_main(numbers) – sprawdza czy funkcja mode(numbers) zwrociła liczbe inną niż 0, wtedy zwraca dominante, natomiast jeżeli zwrociła ona 0 to funkcja informuje o braku mody w danym zbiorze.

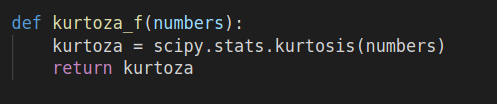
Odchylenie standardowe mówi nam o tym jak bardzo nasze losowe liczby są od siebie oddalone. Im mniejsze jest ono tym bliższe średniej są wartości. Do policzenia ochylenia standardowego użyłam funkcji: statistics.stdev(zbior), która policzyła ochylenie standardowe z podanego zbioru 20 lub 100 elementowego.

Wariancja to inaczej średnia arytmetyczna kwadratów odchyleń od ich średniej arytmetycznej. Aby obliczyć ją dla obu macierzy 20 i 100 elementowych wykorzystałam funkcje którą nazwałam wariancja, której argumentami jest zbiór wygenerowanych elementów oraz policzona wczesniej średnia arytmetyczna.

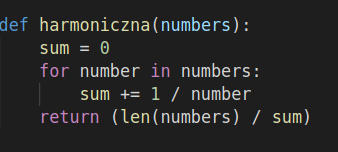
Funkcja w forze sumuje kwadrat różnicy między poszczególnym elementem a średnia arytmetyczna dla całego zbioru, a następnie całą sume dzieli przez ilość elementów.

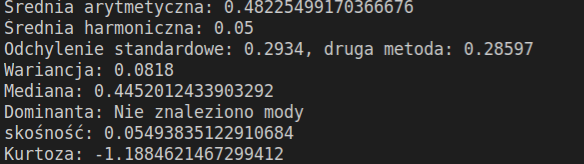
 Skośność jest miarą symetrii rozkładu, jeżeli rozkład jest idealnie symetryczny wynosi ona 0. Dla rozkładów o lewostronnej asymetrii przyjmuje wartości ujemne a dla prawostronnej dodatnie. Aby obliczyć ją w pythonie wykorzystałam funkcje: scipy.stats.skew(numbers), która wyliczyła mi skośność dla wylosowanego zbioru.

Kurtoza jest jedną z miar koncentracji wyników wokół średniej inaczej mówiąc spłaszczenia rozkładu wartości. Jej dodatnia wartość wskazuje na istnienie dużej ilości wartości bliskiej średniej, ujemna wartość kurtozy odpowiada rozproszonym wynikom wokół średniej. Aby ją obliczyc w Pythonie wykorzystałam funkcje scipy.stats.kurtosis(numbers), która znajduje się w bibliotece scipy.



Jako wybraną przeze mnie funkcje obliczyłam średnia harmoniczna stanowi ona odwrotność średniej arytmetycznej odwrotności danych statystycznych. Aby ją obliczyć stworzyłam funkcje harmoniczna która przyjmuje jako argument zbiór wygenerowany w pierwszym zadaniu a następnie sumuje odwrotności tych liczb. By następnie zwrócić ilość liczb podzieloną na wczeniej obliczoną sume odwrotności.

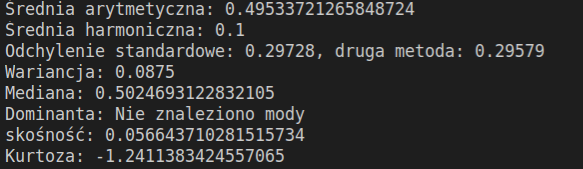


Otrzymane wyniki dla grupy 20 prób:

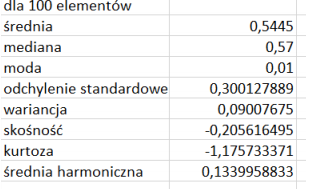
Dla porównania wyniki z zadania 2 listy 1

 W obu przypadkach średnia arytmetyczna jest bliska 0,5. Natomiast średnia harmoniczna w przypadku listy 3 wynosi 0,05 a listy 1 0,23. Jeżeli chodzi o modę to nie będzie ona występować w liscie 3 ponieważ jest zbyt mała próba by któraś z liczb mogłaby się powtórzyć w przeciwieństwie do listy 1 gdzie zakres był dużo mniejszy, ponieważ pracowaliśmy na liczbach tylko z dwoma miejscami po przecinku. Odchylenie standardowe jest bardzo podobne w obu przypadkach co mówi o oddaleniu liczb od średniej .

Wyniki dla grupy 100 prób z listy 3:



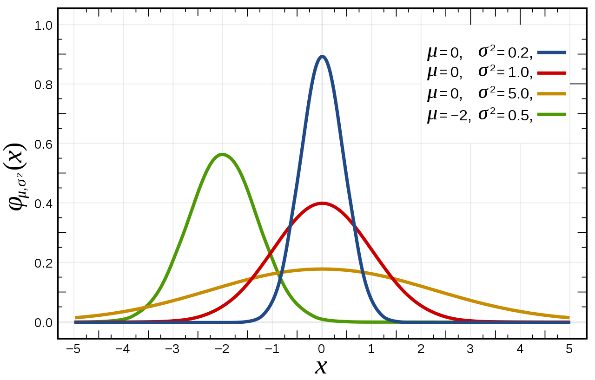
Wyniki dla 100 prób z listy 1 :

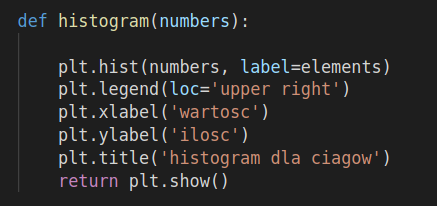
Wyniki w obu przypadkach są do siebie bardzo podobne.

Zadanie 4

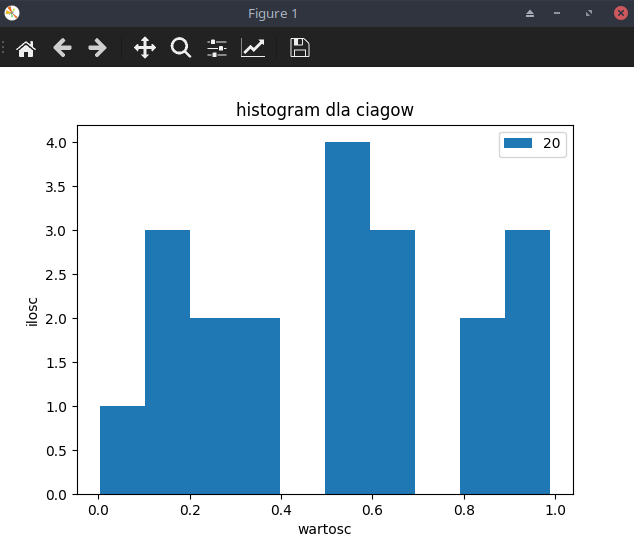
Aby wykonać zadanie 4 musiałam wygenerować histogramy dla obu zbiorów 20 i 100 elementowych a następnie porównać je do wykresu gęstości rozkładu normalnego.

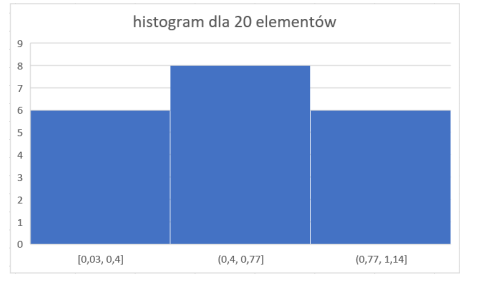
Wykonałam je za pomocą poniżej funkcji histogram:



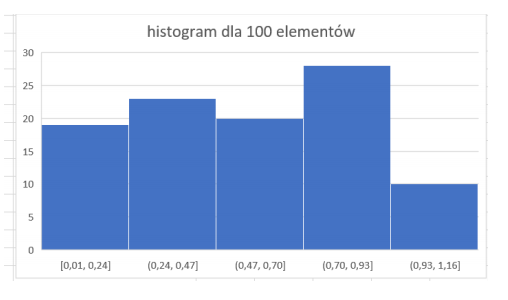
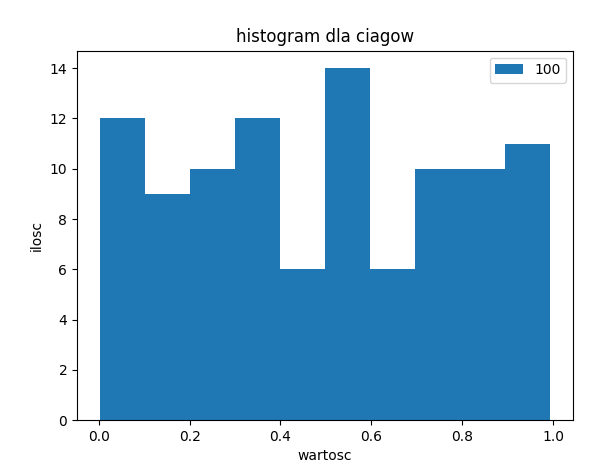


histogram dla 20 wartosci w liscie 3 : histogram dla 20 wartości z listy 1:





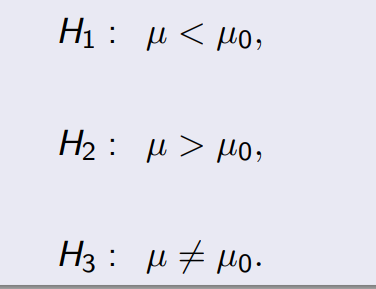
histogram dla 100 wartosci: histogram dla 100 wartości z listy 1:



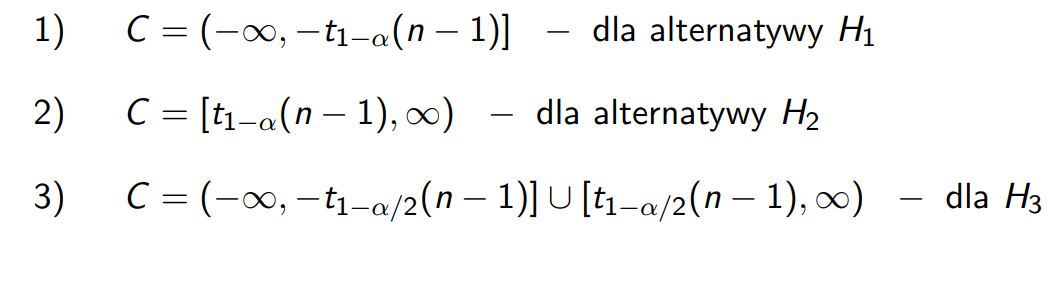
Jak widać wykres dla 20 wartości mocno odbiega od wykresu gęstości dla rozkładu normalnego. Spowodowane to jest małą ilością prób więc mogą wypaść dane ze skajnych obszarów.

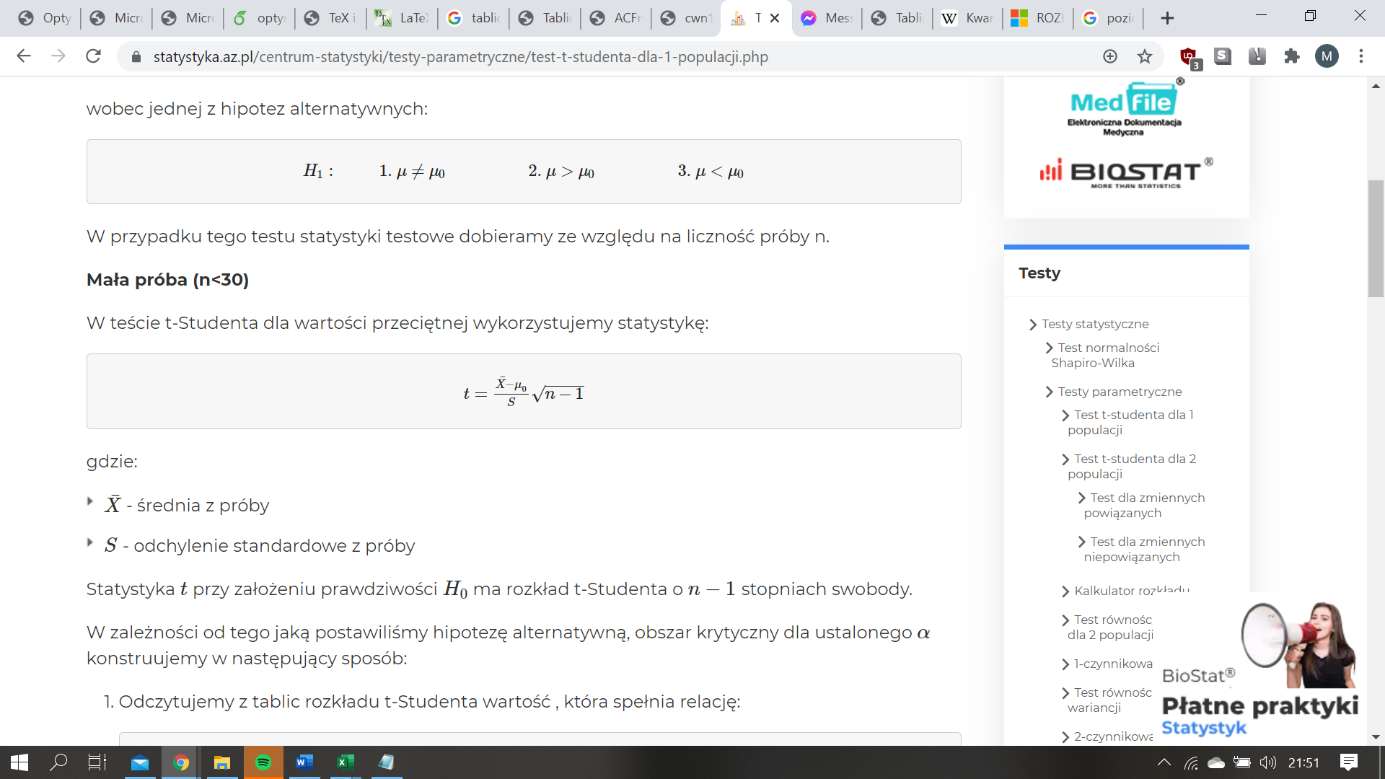
Jednak patrząc na wykres dla 100 wartości odmiennie niż, który w liscie 1 już kształtem przypominał wykres gęstości z powodu większej ilości prób. Wykres nie przypomina nam wykresu gęstości dla rozkładu normalnego, wręcz przeciwnie, ma on bardzo wiele wartość w skrajnym obszarze

Zadanie 5

 W pierwszej części zadania należy porównać średnie są mniejsze od 0.5, dla 20 elementów wykorzystałam do tego test T-studenta. Aby przeprowadzić rozkład T studenta wyznaczam hipotezę: h0 – średnia z prób jest równa 0.5. Hipotezy alternatywne: h1 – średnia z prób nie jest mniejsza od 0.5, h2 – średnia z prób jest większa niż 0.5, h3 - średnia z prób nie jest równa 0.5. Jako poziom istotności wybrałam α1 = 0.05.

Na poziomie istotności zbiory krytyczne testów dla poszczególnych alternatyw mają odpowiednio postać:

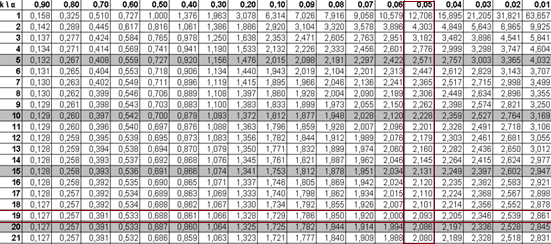




Rozkład t studenta wyliczyłam z następującego wzoru:

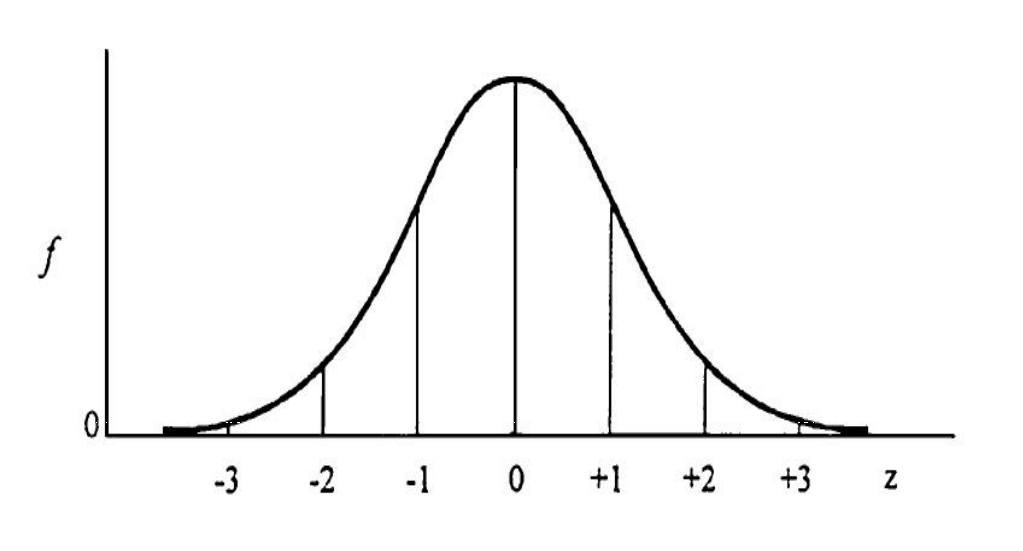
gdzie X – to średnia z próby, μ0 – wartość hipotezy h0,

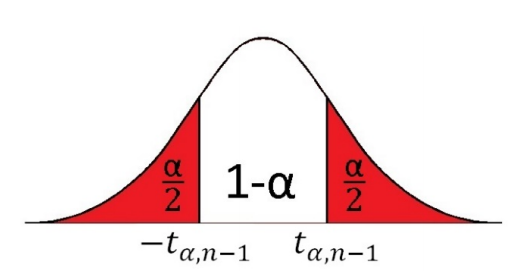
S – odchylenie standardowe, n – liczność próby.



Dla hipotezy h0 oraz hipotezy alternatywnej h1:

Więc obszar krytyczny dla 20 stopni swobody i poziomie istotności α = 0,05 wynosi

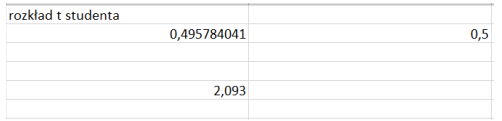
 (-∞, -2,093] ∪ [2,093, ∞) dla t = -2,62 jest to więc obszar krytyczny lewostronny, mamy zatem podstawy do odrzucenia hipotezy h0, na korzyść hipotezy alternatywnej. Dokładnie na korzyść hipotezy h1 – średnia z prób nie jest mniejsza od 0.5 ponieważ znajdujemy się po lewej stronie obszaru krytycznego.

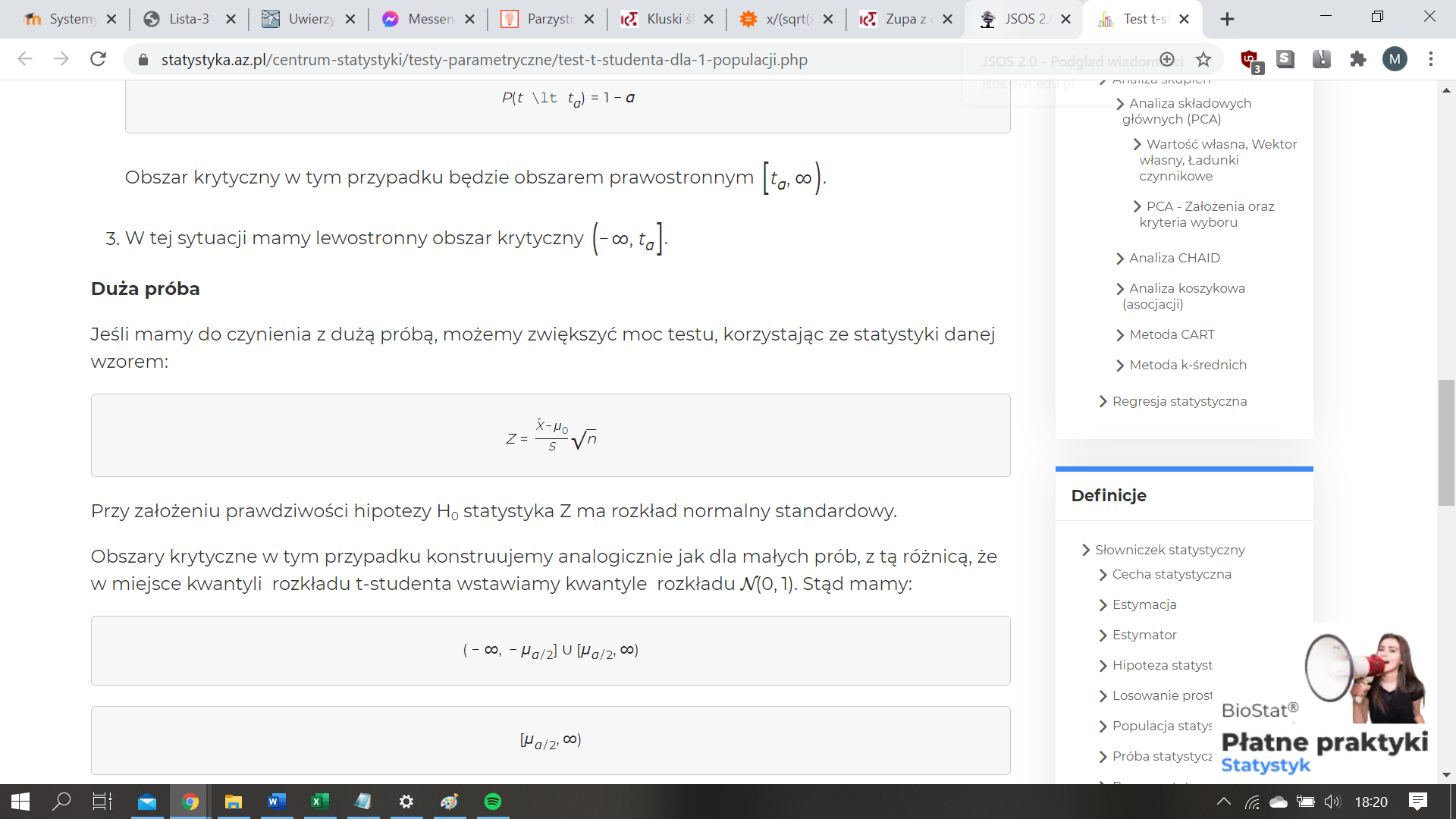


Oraz h3 ponieważ średnia nie jest równa 0,5 dla hipotezy dwustronnej alternatywnej.



Porównując średnie z tego zadania oraz z zadania 1 z listy 1, tam średnia znajdowała się po prawej stronie obszaru krytycznego ale w podobnej odległości.

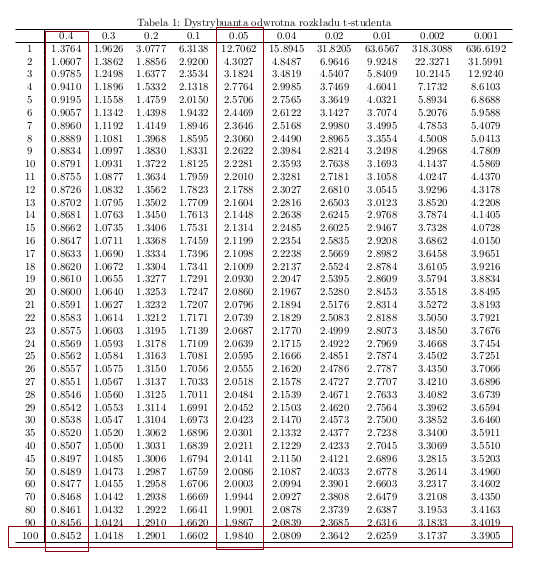


Natomiast dla ciągu 100 wartości aby sprawdzić średnie z prób wykorzystałam Test Z. Aby przeprowadzić rozkład Z wyznaczam hipotezę: h0 – średnia z prób jest równa 0,5. Hipoteza alternatywna: h1 – średnia z prób nie jest mniejsza od 0,5, h2 – średnia z prób jest większa niż 0,5, h3 - średnia z prób nie jest równa 0.5. Jako poziom istotności wybrałam α1 =0,05.

Rozkład Z wyliczyłam z następującego wzoru:

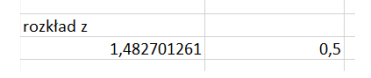
gdzie X – to średnia z próby, μ0 – wartość hipotezy h0,

S – odchylenie standardowe, n – liczność próby.





Porównując do listy 1:



Następnie skorzystałam z tabelki dla k = 100 i α = 0,05 co daje tα = -6,88. Więc obszar krytyczny dla poziomu istotności α = 0,05 wynosi: (-∞, -6,88] ∪ [6,88, ∞) co oznacza dla t = -6,88 że znajduje się on w obszarze krytycznym lewostronnym, czyli mamy podstawy do odrzucenia hipotezy h0 na korzyść hipotezy alternatywnej: h1 – średnia z prób nie jest mniejsza od 0.5 ponieważ znajdujemy się po lewej stronie obszaru krytycznego. Oraz h3 ponieważ średnia nie jest równa 0,5.