Maja Szymajda, 254313

Sprawozdanie lista 4

Do realizacji listy czwartej 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 oraz rozkład naturalny. 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.

```
def generator(numbers):
    a = 226954477   # standard do c++
    c = 1
    m = pow(2, 32)
    x0 = int(time.time()/(100+numbers))
    u = [x0]
    for i in range(numbers-1):
        u.append((a*u[i] + c) % m)
    L = np.array(u)
    L = 1/m * L
    return L


zbior = generator(elements)
```

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. Które zostały przypisane do tabeli „zbiór1”

Source	modulus m	multiplier a	increment c
Numerical Recipes	2^{32}	1664525	1013904223
Borland C/C++	2^{32}	22695477	1
glibc (used by GCC)[15]	2^{31}	1103515245	12345
ANSI C: Watcom, Digital Mars, CodeWarrior, IBM VisualAge C/C++ [16] C90, C99, C11: Suggestion in the ISO/IEC 9899,[17] C18	2^{31}	1103515245	12345
Borland Delphi, Virtual Pascal	2^{32}	134775813	1
Turbo Pascal	2^{32}	134775813 (8088405_{16})	1
Microsoft Visual/Quick C/C++	2^{32}	214013 ($343FD_{16}$)	2531011 ($269EC3_{16}$)
Microsoft Visual Basic (6 and earlier)[18]	2^{24}	1140671485 ($43FD43FD_{16}$)	12820163 ($C39EC3_{16}$)
RtlUniform from Native API[19]	$2^{31} - 1$	2147483629 ($7FFFFFED_{16}$)	2147483587 ($7FFFFFC3_{16}$)
Apple CarbonLib, C++11's `minstd_rand0` [20]	$2^{31} - 1$	16807	0
C++11's `minstd_rand` [20]	$2^{31} - 1$	48271	0
MMIX by Donald Knuth	2^{64}	6364136223846793005	1442695040888963407
Newlib, Musl	2^{64}	6364136223846793005	1
VMS's **MTH$RANDOM**,[21] old versions of glibc	2^{32}	69069 ($10DCD_{16}$)	1
Java's java.util.Random, POSIX [ln]rand48, glibc [ln]rand48[_r]	2^{48}	25214903917 ($5DEECE66D_{16}$)	11
`random0` [22][23][24][25][26]	$134456 = 2^3 7^5$	8121	28411

The generator is defined by recurrence relation:

$$X_{n+1} = (aX_n + c) \bmod m$$

where X is the sequence of pseudorandom values, and

$m,\ 0 < m$ — the "modulus"

$a,\ 0 < a < m$ — the "multiplier"

$c,\ 0 \le c < m$ — the "increment"

$X_0,\ 0 \le X_0 < m$ — the "seed" or "start value"

Aby wykonać to zadanie posiłkowałam się poniższym wzorem oraz tabelką.

Do stworzenia zbioru o rozkładzie naturalnym skorzystałam z Transformacji Boxa-Mullera. Jest to metoda generowania liczb losowych o rozkładzie normalnym na podstawie dwóch wartości zmiennej o rozkładzie jednostajnym na przedziale (0,1]. W tej funkcji przyjmujemy x1 i x2 za niezależne zmienne o rozkładzie jednostajnym na przedziale (0, 1].
Funkcja jako argumenty przyjmuje ilość elementów oraz zbiór prób o rozkladzie normalnym. W pętli for przechodzi po każdych dwóch liczbach ze zbioru1 (zbioru powstałym w generatorze) i za pomocą poniższego wzoru dodaje liczby o rozkładzie naturalnym do zbioru prób o nazwie „zbiór”

```
def rozklad_normalny(numbers, zbior1):
    zbior = []
    for i in range(int(elements/2)):
        x1 = zbior1[i+i]
        x2 = zbior1[i+i+1]
        # print(x1, x2)
        y1 = np.sqrt(-2 * np.log(x1)) * np.cos(2 * np.pi * x2)
        y2 = np.sqrt(-2 * np.log(x1)) * np.sin(2 * np.pi * x2)
        zbior.append(y1)
        zbior.append(y2)
    # print(zbior)
    return zbior
```

$$Z_1 = R\cos\Theta = \sqrt{-2\ln U_1}\,\cos(2\pi U_2)$$

oraz

$$Z_2 = R\sin\Theta = \sqrt{-2\ln U_1}\,\sin(2\pi U_2).$$

Przykładowy zbiór 20 elementów:
[-2.5591607704089565, -2.234920092280576, -0.950194338431816, -0.19078207961084107, 0.048594342004369066, 0.06965340201025566, 0.6752498986008619, 1.8111683477132754, 0.1289148936026669, 0.37433585528824004, -0.06736619781096112, -0.784128960068903, 0.6330534807574332, -0.08795367536861498, -0.5841010241642031, -0.7539545878604862, -0.9701734393055241, -0.2895066137451158, -0.5288416684749007, -0.6582428146437724]

Przykładowy zbiór dla 100 elementów:
[0.4299343333761449, -3.5186614552832816, -0.7282821670703841, 0.9741661020159551, 0.25357153821873407, -1.9591718026409377, 0.7654316447189414, 0.28108240502610204, 0.9947760916907115, 0.23916211290442474, -0.48054792237206273, 2.1466648956607597, 3.5279116522455665, 1.2335497678443856, -0.8364423237336535, -0.24100129855388472, -1.4968925413017131, 1.3932935184566897, 1.7839392691282068, 0.41910805025925274, -0.4842277611638494, -0.11623720550877834, 1.0799337773701372, -0.4548298553573879, -1.2249476011347893, 1.0125102378083057, 1.3469503443284565, -0.36705454508783525, 0.9853526817566132, 0.27186419867449113, -1.5072107738202793, -0.6422299223698572, 0.12879969010869388, -0.03091748616338776, 1.0795420377487994, -0.4293691834057226, 0.45091381372336925, -0.9630871043708314, -0.7849238113709113, -0.39108292331828454, -0.286606722202527, 1.366256137748525, 1.056807096576235, 0.5026062358694073, -0.002157633170770868, -0.3223206747048368, -0.6456278002608681, -0.20153641479434717, -0.04739480881681405, -1.2809950530643253, -0.21278699936199175, -1.3738242060124277, -1.4654320417828202, 0.17625073352598547, -0.5787126561424549, -0.2660801806757908, 0.4555976789251188, -1.2825497541635722, 1.0230934275817694, 0.7088725057990707, -1.7073035410114015, -0.12629275397392775, -0.44509602228762496, -1.7860398200499692, 0.16179499450740856, -0.7930064197568445, 0.39157583898080806, 0.6083442325259412, 0.6216792203465822, 1.7853475514019785, 0.6492258956344688, -0.06907925203393114, -0.44969510004709623, -0.27927073732272306, 0.8712053732245754, 0.45595044142691815, -0.3371064360627359, -1.0722614304352542, -0.3928834318617775, -1.1149518945495893, -0.6880008009049131, 0.517164512232426, 1.362491817558544, 0.37748165562093255, 1.8055891514301095, -1.4694856909558895, -0.17289609541759332, 1.0424986473933269, -0.02009251275935779, 1.1335128712724463, -1.247521253407382, -0.49508124639574447, 1.297578812166941, 1.9416841240746212, -0.8933561348104088, -0.06900943811199138, -0.05654328457542896, -0.04894241883315923, 0.8920765212501516, 1.3343289662769318]

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:

```
def srednia(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
```

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.

```
def indexof(a, A):
    for i in range(len(A)):
        if A[i] == a:
            return i
    return -1


def mode(A):
    liczby = []
    wystapienia = []

    for a in A:
        index = indexof(a, liczby)
        if index >= 0:
            wystapienia[index] += 1
        else:
            liczby.append(a)
            wystapienia.append(1)

    _mode = 0
    count = 0
    for i in range(len(wystapienia)):
        if wystapienia[i] > wystapienia[_mode]:
            mode = i
            count = 1
        elif wystapienia[i] == wystapienia[_mode]:
            count += 1

    if count == 1:
        return [liczby[_mode], wystapienia[_mode]]
    else:
        return None


def moda_main(numbers):
    m = mode(numbers)
    if m is not None:
        return "Dominanta: %d, wystapien: %d" % (m[0], m[1])
    else:
        return "Nie znaleziono mody"
```

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.

```
def wariancja(numbers, aver):
    sum = 0
    for number in numbers:
        sum += pow((number - aver), 2)
    return sum / len(numbers)
```

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.

```
def skosniosc_f(numbers):
    skosniosc = scipy.stats.skew(numbers)
    return skosniosc
```

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.

```
def kurtoza_f(numbers):
    kurtoza = scipy.stats.kurtosis(numbers)
    return kurtoza
```

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

```
def harmoniczna(numbers):
    sum = 0
    for number in numbers:
        sum += 1 / number
    return (len(numbers) / sum)
```

Otrzymane wyniki dla grupy 20 prób:

```
Średnia arytmetyczna: -0.3459178021098784
Średnia harmoniczna: 5.84
Odchylenie standardowe: 0.96426, druga metoda: 0.93984
Wariancja: 0.8833
Dominanta: Nie znaleziono mody
skośność: -0.334363102928953
Kurtoza: 0.9176654831644386
```

Dla porównania wyniki z zadania 2 listy 2

```
Średnia arytmetyczna: 0.6162069230526729
Średnia harmoniczna: 0.54
Odchylenie standardowe: 0.19736, druga metoda: 0.1921
Wariancja: 0.0369
Dominanta: Nie znaleziono mody
skośność: -0.19838017849075873
Kurtoza: -0.6720437422466925
```

Ciężko porównać nam obie próby ponieważ wygenerowany rozkład odbiega zbiorem wartości od prób z listy 2. Jednak patrząc na umiejscowienie watrości średnich obie znajdują się w pobliżu środków przedziałów. Jeżeli chodzi o modę to nie występuje ona w liscie 4 i 2, ponieważ jest zbyt mała próba by któraś z liczb mogłaby się powtórzyć. Odchylenie standardowe jest bardzo podobne w obu przypadkach co mówi o oddaleniu liczb od średniej .

Wyniki dla grupy 100 prób z listy 4:

```
Średnia arytmetyczna: 0.04980414261669843
Średnia harmoniczna: -0.15
Odchylenie standardowe: 1.05244, druga metoda: 1.04717
Wariancja: 1.0966
Dominanta: Nie znaleziono mody
skośność: 2.7488548825665453e-06
Kurtoza: 1.018423282352213
```

Wyniki dla 100 prób z listy 2 :

```
Srednia arytmetyczna: 0.5117078288143222
Srednia harmoniczna: 0.61
Odchylenie standardowe: 0.24233, druga metoda: 0.24111
Wariancja: 0.0581
Dominanta: Nie znaleziono dominanty
skośność: -0.12291261581538047
Kurtoza: -0.022874548569957742
```

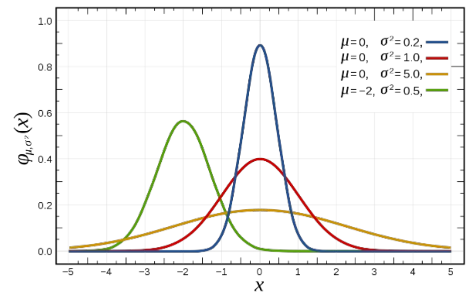
Jak w przykładzie z dwudziestoa próbami, ciężko nam porównać średnie z powodu różnych zbiorów wartości funkcji. Jednak średie arytmetyczne są bardzo bliskie środka zbioru. W obu przypadkach nie znaleziono dominanty ponieważ zbiór liczb tych przedziałów jest nieskończenie duża. Skośność w obu przypadkach jest bardzo mała. Podobnie odchylenie standardowe.

Zadanie 4

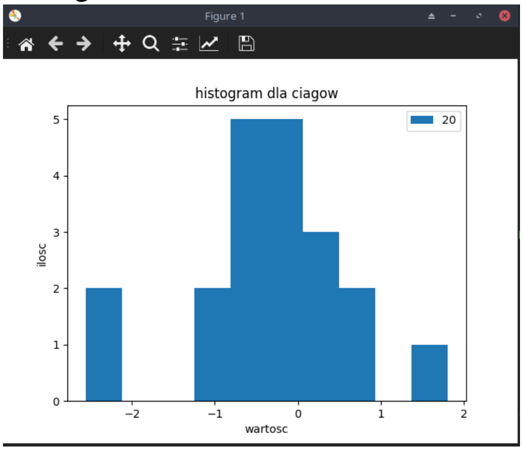
Aby wykonać zadanie 4 musiałam wygenerować histogramy dla obu zbiorów 20 i 100 elementowych a następnie porównać je do wykresu gęstości rozkładu normalnego.

Wykonałam je za pomocą poniżej funkcji histogram:

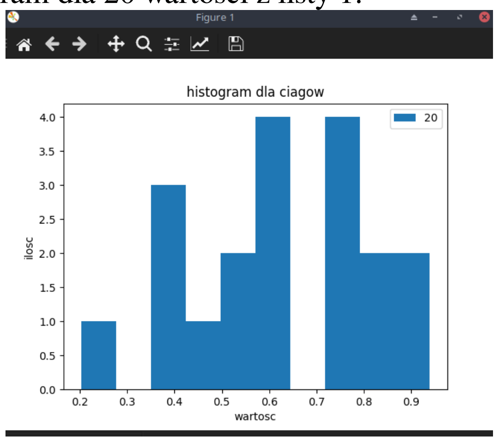
```python
def histogram(numbers):

    plt.hist(numbers, label=elements)
    plt.legend(loc='upper right')
    plt.xlabel('wartosc')
    plt.ylabel('ilosc')
    plt.title('histogram dla ciagow')
    return plt.show()
```



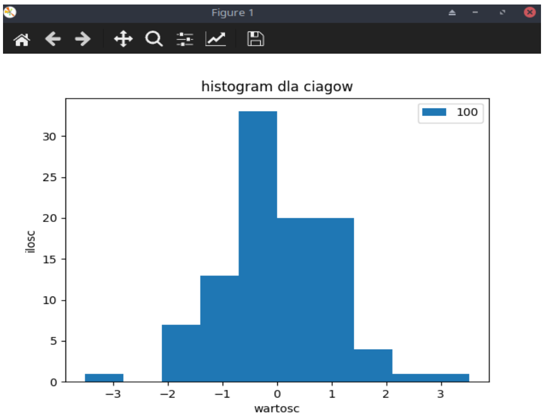
histogram dla 20 wartosci w liscie 3 :



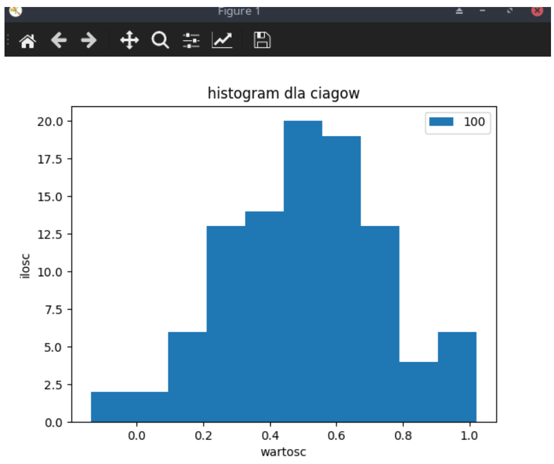
histogram dla 20 wartości z listy 1:



histogram dla 100 wartosci z listy 4:



histogram dla 100 wartości z listy 2:



Jak widać wykres dla 20 wartości nie odbiega mocno odbiega od wykresu gęstości dla rozkładu normalnego. Jednak porównując go z histogramem rozkładu nauralnego z zadania 3 listy 2 to jest on dużo bardziej podobny do wykresu gęstośći. Jednak może być to spowodowane to jest małą ilością prób, więc mogą wypaść nam dane ze skajnych obszarów.
Jednak patrząc na wykres dla 100 wartości podobniej jak w liscie 2 już kształtem przypominał wykres gęstości z powodu większej ilości prób.

Zadanie 5

W pierwszej części zadania należy porównać średnie są mniejsze od 0.6, dla 20 elementów wykorzystałam do tego test T-studenta. Aby przeprowadzić rozkład T studenta wyznaczam hipotezę: h_0 – średnia z prób jest równa 0.6. Hipotezy alternatywne: h_1 – średnia z prób nie jest mniejsza od 0.6, h_2 – średnia z prób jest większa niż 0.6, h_3 - średnia z prób nie jest równa 0.6. Jako poziom istotności wybrałam $\alpha_1 = 0.05$.

$$H_1: \ \mu < \mu_0,$$

$$H_2: \ \mu > \mu_0,$$

$$H_3: \ \mu \neq \mu_0.$$

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

1) $C = (-\infty, -t_{1-\alpha}(n-1)]$ – dla alternatywy H_1

2) $C = [t_{1-\alpha}(n-1), \infty)$ – dla alternatywy H_2

3) $C = (-\infty, -t_{1-\alpha/2}(n-1)] \cup [t_{1-\alpha/2}(n-1), \infty)$ – dla H_3

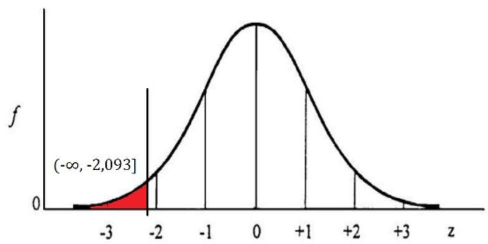
Rozkład t studenta wyliczyłam z następującego wzoru:
gdzie X – to średnia z próby, μ_0 – wartość hipotezy h_0,
S – odchylenie standardowe, n – liczność próby.

$$t = \frac{\bar{X} - \mu_0}{S}\sqrt{n-1}$$

k \ α	0,90	0,80	0,70	0,60	0,50	0,40	0,30	0,20	0,10	0,09	0,08	0,07	0,06	0,05	0,04	0,03	0,02	0,01
1	0,158	0,325	0,510	0,727	1,000	1,376	1,963	3,078	6,314	7,026	7,916	9,058	10,579	12,706	15,895	21,205	31,821	63,657
2	0,142	0,289	0,445	0,617	0,816	1,061	1,386	1,886	2,920	3,104	3,320	3,578	3,896	4,303	4,849	5,643	6,965	9,925
3	0,137	0,277	0,424	0,584	0,765	0,978	1,250	1,638	2,353	2,471	2,605	2,763	2,951	3,182	3,482	3,896	4,541	5,841
4	0,134	0,271	0,414	0,569	0,741	0,941	1,190	1,533	2,132	2,226	2,333	2,456	2,601	2,776	2,999	3,298	3,747	4,604
5	0,132	0,267	0,408	0,559	0,727	0,920	1,156	1,476	2,015	2,098	2,191	2,297	2,422	2,571	2,757	3,003	3,365	4,032
6	0,131	0,265	0,404	0,553	0,718	0,906	1,134	1,440	1,943	2,019	2,104	2,201	2,313	2,447	2,612	2,829	3,143	3,707
7	0,130	0,263	0,402	0,549	0,711	0,896	1,119	1,415	1,895	1,966	2,046	2,136	2,241	2,365	2,517	2,715	2,998	3,499
8	0,130	0,262	0,399	0,546	0,706	0,889	1,108	1,397	1,860	1,928	2,004	2,090	2,189	2,306	2,449	2,634	2,896	3,355
9	0,129	0,261	0,398	0,543	0,703	0,883	1,100	1,383	1,833	1,899	1,973	2,055	2,150	2,262	2,398	2,574	2,821	3,250
10	0,129	0,260	0,397	0,542	0,700	0,879	1,093	1,372	1,812	1,877	1,948	2,028	2,120	2,228	2,359	2,527	2,764	3,169
11	0,129	0,260	0,396	0,540	0,697	0,876	1,088	1,363	1,796	1,859	1,928	2,007	2,096	2,201	2,328	2,491	2,718	3,106
12	0,128	0,259	0,395	0,539	0,695	0,873	1,083	1,356	1,782	1,844	1,912	1,989	2,076	2,179	2,303	2,461	2,681	3,055
13	0,128	0,259	0,394	0,538	0,694	0,870	1,079	1,350	1,771	1,832	1,899	1,974	2,060	2,160	2,282	2,436	2,650	3,012
14	0,128	0,258	0,393	0,537	0,692	0,868	1,076	1,345	1,761	1,821	1,887	1,962	2,046	2,145	2,264	2,415	2,624	2,977
15	0,128	0,258	0,393	0,536	0,691	0,866	1,074	1,341	1,753	1,812	1,878	1,951	2,034	2,131	2,249	2,397	2,602	2,947
16	0,128	0,258	0,392	0,535	0,690	0,865	1,071	1,337	1,746	1,805	1,869	1,942	2,024	2,120	2,235	2,382	2,583	2,921
17	0,128	0,257	0,392	0,534	0,689	0,863	1,069	1,333	1,740	1,798	1,862	1,934	2,015	2,110	2,224	2,368	2,567	2,898
18	0,127	0,257	0,392	0,534	0,688	0,862	1,067	1,330	1,734	1,792	1,855	1,926	2,007	2,101	2,214	2,356	2,552	2,878
19	0,127	0,257	0,391	0,533	0,688	0,861	1,066	1,328	1,729	1,786	1,850	1,920	2,000	2,093	2,205	2,346	2,539	2,861
20	0,127	0,257	0,391	0,533	0,687	0,860	1,064	1,325	1,725	1,782	1,844	1,914	1,994	2,086	2,197	2,336	2,528	2,845
21	0,127	0,257	0,391	0,532	0,686	0,859	1,063	1,323	1,721	1,777	1,840	1,909	1,988	2,080	2,189	2,328	2,518	2,831

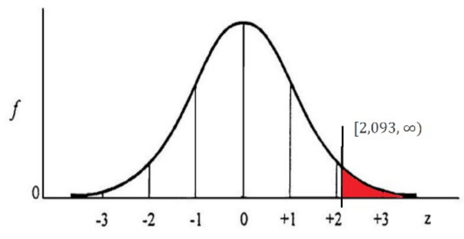
Dla hipotezy h_0 oraz hipotezy alternatywnej h_1:

Obszar krytyczny dla 20 stopni swobody i poziomie istotności α = 0,05 wynosi (-∞, -2,093] dla t = -4,73 jest to więc obszar krytyczny lewostronny, mamy zatem podstawy do odrzucenia hipotezy h_0, na korzyść hipotezy alternatywnej h_1 – średnia z prób nie jest mniejsza od 0.6 ponieważ znajdujemy się po lewej stronie obszaru krytycznego.

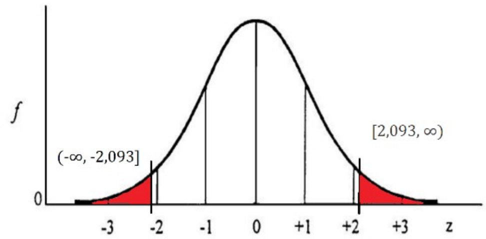


Dla hipotezy $h_0 = 0,6$ oraz hipotezy alternatywnej $h_2 > 0,6$:

Obszar krytyczny dla 20 stopni swobody i poziomie istotności α = 0,05 wynosi [2,093, ∞) dla t = -4,73 zatem t należy do obszaru 1- α zatem nie mamy podstawy do odrzucenia hipotezy h_0, więc nieprawdziwa jest hipoteza alternatywnej h_2 – średnia z prób nie jest większa od 0.6.



Dla hipotezy $h_0 = 0{,}6$ oraz hipotezy alternatywnej $h_3 \neq 0{,}6$:



Obszar krytyczny dla 20 stopni swobody i poziomie istotności $\alpha = 0{,}05$ wynosi $(-\infty, -2{,}093] \cup [2{,}093, \infty)$ dla $t = -4{,}73$ więc t należy do obszaru krytycznego, mamy zatem podstawy do odrzucenia hipotezy h_0, na korzyść hipotezy alternatywnej h_3 – średnia z prób nie równa się 0.6.

Test T studenta dla 20 prób z listy 4:

```
test t studenta: -4.728046187962077
test normalnosci rozkladu: 0.9436232632746296
test normalnosci rozkladu sprawdzenie: (0.9438977837562561, 0.2837821841239929)
```

Porównując średnie z tego zadania oraz z zadania 4 z listy 2, tam średnia również znajdowała się po lewej stronie obszaru krytycznego ale w dużo mniejszej odległości od środka. Jednak przy próbie tak małej jak 20 możemy się spodziewać że pojawia się wartości z przedziału krytycznego.

Test T studenta dla 20 prób z listy 2:

```
test t studenta: -1.8506640876672535
test normalnosci rozkladu: (0.9756880402565002, 0.8815097808837891)
```

Natomiast dla ciągu 100 wartości aby sprawdzić średnie z prób wykorzystałam Test Z. Aby przeprowadzić rozkład Z wyznaczam hipotezę: h_0 – średnia z prób jest równa 0,6. Hipoteza alternatywna: h_1 – średnia z prób nie jest mniejsza od 0,6, h_2 – średnia z prób jest większa niż 0,6, h_3 - średnia z prób nie jest równa 0.6. Jako poziom istotności wybrałam $\alpha_1 = 0{,}05$.

$$Z = \frac{\bar{x} - \mu_0}{s}\sqrt{n}$$

Rozkład Z wyliczyłam z następującego wzoru:
gdzie X – to średnia z próby, μ_0 – wartość hipotezy h_0,
S – odchylenie standardowe, n – liczność próby.
Otrzymaliśmy następujące wyniki:

```
test normalnosci rozkladu: 0.9954988884656983
test normalnosci rozkladu sprawdzenie: (0.9854516983032227, 0.34214693307876587)
test Z: -6.17796058462749
```

Porównując do listy 2:

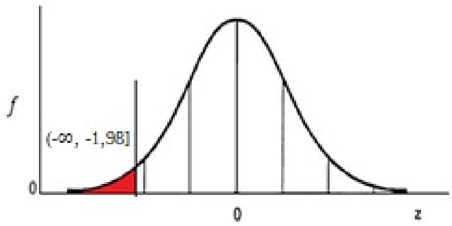
```
test normalnosci rozkladu: (0.9868707060813904, 0.4363394677639007б)
test Z: -7.76996978061411
```

Następnie skorzystałam z tabelki dla k = 100 i $\alpha = 0{,}05$ oraz $t_\alpha = -6{,}18$ wyliczonego z testu Z.

Tabela 1: Dystrybuanta odwrotna rozkładu t-studenta

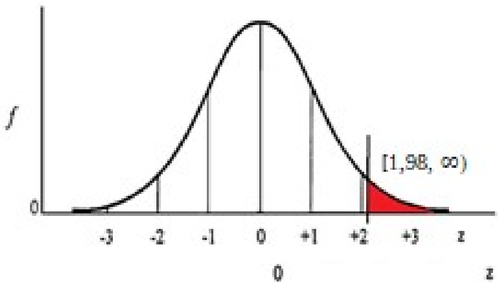
	0.4	0.3	0.2	0.1	0.05	0.04	0.02	0.01	0.002	0.001
1	1.3764	1.9626	3.0777	6.3138	12.7062	15.8945	31.8205	63.6567	318.3088	636.6192
2	1.0607	1.3862	1.8856	2.9200	4.3027	4.8487	6.9646	9.9248	22.3271	31.5991
3	0.9785	1.2498	1.6377	2.3534	3.1824	3.4819	4.5407	5.8409	10.2145	12.9240
4	0.9410	1.1896	1.5332	2.1318	2.7764	2.9985	3.7469	4.6041	7.1732	8.6103
5	0.9195	1.1558	1.4759	2.0150	2.5706	2.7565	3.3649	4.0321	5.8934	6.8688
6	0.9057	1.1342	1.4398	1.9432	2.4469	2.6122	3.1427	3.7074	5.2076	5.9588
7	0.8960	1.1192	1.4149	1.8946	2.3646	2.5168	2.9980	3.4995	4.7853	5.4079
8	0.8889	1.1081	1.3968	1.8595	2.3060	2.4490	2.8965	3.3554	4.5008	5.0413
9	0.8834	1.0997	1.3830	1.8331	2.2622	2.3984	2.8214	3.2498	4.2968	4.7809
10	0.8791	1.0931	1.3722	1.8125	2.2281	2.3593	2.7638	3.1693	4.1437	4.5869
11	0.8755	1.0877	1.3634	1.7959	2.2010	2.3281	2.7181	3.1058	4.0247	4.4370
12	0.8726	1.0832	1.3562	1.7823	2.1788	2.3027	2.6810	3.0545	3.9296	4.3178
13	0.8702	1.0795	1.3502	1.7709	2.1604	2.2816	2.6503	3.0123	3.8520	4.2208
14	0.8681	1.0763	1.3450	1.7613	2.1448	2.2638	2.6245	2.9768	3.7874	4.1405
15	0.8662	1.0735	1.3406	1.7531	2.1314	2.2485	2.6025	2.9467	3.7328	4.0728
16	0.8647	1.0711	1.3368	1.7459	2.1199	2.2354	2.5835	2.9208	3.6862	4.0150
17	0.8633	1.0690	1.3334	1.7396	2.1098	2.2238	2.5669	2.8982	3.6458	3.9651
18	0.8620	1.0672	1.3304	1.7341	2.1009	2.2137	2.5524	2.8784	3.6105	3.9216
19	0.8610	1.0655	1.3277	1.7291	2.0930	2.2047	2.5395	2.8609	3.5794	3.8834
20	0.8600	1.0640	1.3253	1.7247	2.0860	2.1967	2.5280	2.8453	3.5518	3.8495
21	0.8591	1.0627	1.3232	1.7207	2.0796	2.1894	2.5176	2.8314	3.5272	3.8193
22	0.8583	1.0614	1.3212	1.7171	2.0739	2.1829	2.5083	2.8188	3.5050	3.7921
23	0.8575	1.0603	1.3195	1.7139	2.0687	2.1770	2.4999	2.8073	3.4850	3.7676
24	0.8569	1.0593	1.3178	1.7109	2.0639	2.1715	2.4922	2.7969	3.4668	3.7454
25	0.8562	1.0584	1.3163	1.7081	2.0595	2.1666	2.4851	2.7874	3.4502	3.7251
26	0.8557	1.0575	1.3150	1.7056	2.0555	2.1620	2.4786	2.7787	3.4350	3.7066
27	0.8551	1.0567	1.3137	1.7033	2.0518	2.1578	2.4727	2.7707	3.4210	3.6896
28	0.8546	1.0560	1.3125	1.7011	2.0484	2.1539	2.4671	2.7633	3.4082	3.6739
29	0.8542	1.0553	1.3114	1.6991	2.0452	2.1503	2.4620	2.7564	3.3962	3.6594
30	0.8538	1.0547	1.3104	1.6973	2.0423	2.1470	2.4573	2.7500	3.3852	3.6460
35	0.8520	1.0520	1.3062	1.6896	2.0301	2.1332	2.4377	2.7238	3.3400	3.5911
40	0.8507	1.0500	1.3031	1.6839	2.0211	2.1229	2.4233	2.7045	3.3069	3.5510
45	0.8497	1.0485	1.3006	1.6794	2.0141	2.1150	2.4121	2.6896	3.2815	3.5203
50	0.8489	1.0473	1.2987	1.6759	2.0086	2.1087	2.4033	2.6778	3.2614	3.4960
60	0.8477	1.0455	1.2958	1.6706	2.0003	2.0994	2.3901	2.6603	3.2317	3.4602
70	0.8468	1.0442	1.2938	1.6669	1.9944	2.0927	2.3808	2.6479	3.2108	3.4350
80	0.8461	1.0432	1.2922	1.6641	1.9901	2.0878	2.3739	2.6387	3.1953	3.4163
90	0.8456	1.0424	1.2910	1.6620	1.9867	2.0839	2.3685	2.6316	3.1833	3.4019
100	0.8452	1.0418	1.2901	1.6602	1.9840	2.0809	2.3642	2.6259	3.1737	3.3905

Dla hipotezy $h_0 = 0,6$ oraz hipotezy alternatywnej $h_1<0,6$:



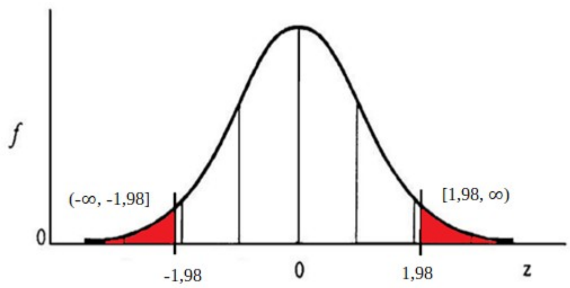
Obszar krytyczny dla poziomu istotności $\alpha = 0,05$ wynosi: (-∞, -1,98] co oznacza dla t = -6,18 że znajduje się on w obszarze krytycznym lewostronnym, czyli mamy podstawy do odrzucenia hipotezy h_0 na korzyść hipotezy alternatywnej: h_1 – średnia z prób nie jest mniejsza od 0.5 ponieważ znajdujemy się po lewej stronie obszaru krytycznego.

Dla hipotezy $h_0 = 0,6$ oraz hipotezy alternatywnej $h_2>0,6$:



W tym przypadku obszar krytyczny dla 100 stopni swobody i poziomie istotności $\alpha = 0,05$ wynosi [1,98, ∞) dla t = -6,18 zatem t należy do obszaru 1- α zatem nie mamy podstawy do odrzucenia hipotezy h_0, więc nieprawdziwa jest hipoteza alternatywnej h_2 – średnia z prób nie jest większa od 0.5.

Dla hipotezy $h_0 = 0,6$ oraz hipotezy alternatywnej $h_3 \neq 0,6$:



Obszar krytyczny dla 100 stopni swobody i poziomie istotności $\alpha = 0,05$ wynosi: (-∞, -1,98] ∪ [1,98, ∞) dla t = -6,18, t należy do obszaru krytycznego, mamy zatem podstawy do odrzucenia hipotezy h_0, na korzyść hipotezy alternatywnej h_3 – średnia z prób nie równa się 0.5.

Test Shapiro - Wilka

Aby wykonać porównanie zbioru elementów do rozkładu normalnego wykorzystałam test Shapiro-Wilka. Test ten służący do oceny, czy zebrane przez nas wyniki od badanych prób posiadają rozkład normalny. Jako hipoteze przyjełam h_0 : p>0,05 co oznacza że badana próba należy do rozkładu normalnego. Hipoteza alternetywna h_1 ; p <0,05 – badana próba nie jest w postaci rozkładu normalnego.

```python
def test_shapiro_wilka(numbers, a):
    numbers = sorted(numbers)
    n = len(numbers)

    sum1 = 0
    sum2 = 0
    for i in range(len(a)):
        sum1 += a[i] * (numbers[n-i-1] - numbers[i])
    for j in range(n):
        sum2 += (numbers[j] - np.mean(numbers))**2
    return sum1**2/sum2
```

Dla dwudziestu watrości wykonałam go za pomocą funkcji, która korzysta ze wrozu:

Wzór na **test normalności rozkładu Shapiro-Wilka** ma postać:

$$W = \frac{[\sum_i a_i(n)(X_{n-i+1} - X_i)]^2}{\sum_{j=1}^{n}(X_j - \bar{X})^2}$$

gdzie:

W - wynik testu Shapiro-Wilka

$a_i(n)$ - stała, wartości w tablicy

$X_{n-i+1} - X_i$ - różnica pomiędzy skrajnymi obserwacjami, przy czym i = 1 różnica dla min i max; dla i = 2 różnica dla min+1 i max - 1 itd..

j - kolejne obserwacje w próbie

i - kolejne różnice między skrajnymi obserwacjami

$\bar{X}$ - średnia

i tablicy:

i / n	12	13	14	15	16	17	18	19	20	21
1	0,5475	0,5359	0,5251	0,5150	0,5056	0,4968	0,4886	0,4808	0,4734	0,4643
2	0,3325	0,3325	0,3318	0,3306	0,3290	0,3273	0,3253	0,3232	0,3211	0,3185
3	0,2347	0,2412	0,2460	0,2495	0,2521	0,2540	0,2553	0,2561	0,2565	0,2578
4	0,1586	0,1707	0,1802	0,1878	0,1939	0,1988	0,2027	0,2059	0,2085	0,2199
5	0,0922	0,1099	0,1240	0,1353	0,1447	0,1524	0,1587	0,1641	0,1686	0,1736
6	0,0303	0,0539	0,0727	0,0880	0,1005	0,1109	0,1197	0,1271	0,1334	0,1399
7		0	0,0240	0,0433	0,0593	0,725	0,0837	0,0932	0,1013	0,1092
8				0	0,0196	0,0359	0,0496	0,0612	0,0711	0,0804
9						0	0,0130	0,0303	0,0422	0,0530
10								0	0,0140	0,0263
11										0

Otrzymałam w ten sposób p = 0.88, do sprawdzenia tych obliczeń wykorzystałam funkcje wbudowana w biblioteke scipy: scipy.stats.shapiro(numbers), który zwrocił mi test statystyczny oraz watość p – istotność statystyczna.
Ponieważ p = 0.88 > 0,05 to możemy przyjąć h_0 jako prawdziwą.
Tak więc dla 20 elementów

```
test t studenta: -4.728046187962077
test normalnosci rozkladu: 0.9436232632746296
test normalnosci rozkladu sprawdzenie: (0.9438977837562561, 0.2837821841239929)
```

p = 0.94, więc jest to rozkład naturalny .

Jeśli test Shapiro-Wilka osiąga istotność statystyczną ($p < 0{,}05$), świadczy to o rozkładzie oddalonym od krzywej Gaussa. W przypadku tego testu najczęściej chcemy otrzymać wartości nieistotne statystyczne ($p > 0{,}05$), ponieważ świadczą one o zgodności rozkładu zmiennej z rozkładem normalnym.

Dla 100 elementów policzyłam za pomocą powyższej funkcji:
Jako hipoteze przyjełam h_0 : p>0,05 co oznacza że badana próba należy do rozkładu normalnego.
Hipoteza alternetywna h_1 ; p <0,05 – badana próba nie jest w postaci rozkładu normalnego.

```
test normalnosci rozkladu: 0.99549888884656983
test normalnosci rozkladu sprawdzenie: (0.9854516983032227, 0.34214693307876587)
test Z: -6.17796058462749
```

p = 0,99, więc również jest to rozkład naturalny i nie mamy podstawy do odrzucenia hipotezy h_0.