Eksploracja danych

Sprawozdanie nr 1 z ćwiczeń laboratoryjnych

Michał Orlewski

Szymon Chadam

Wydział Fizyki i Informatyki Stosowanej

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

13 listopada 2023

Laboratorium 1

Celem pierwszych zajęć laboratoryjnych była analiza rozkładu normalnego.

Zadanie 1

Pierwszym zadaniem było wygenerowanie *m* punktów z rozkładu normalnego i przedstawienie ich na wykresie oraz wyliczenie dla każdego z punktów wartości funkcji gęstości prawdopodobieństwa. Wykresy wygenerowano za pomocą poniższego skryptu.

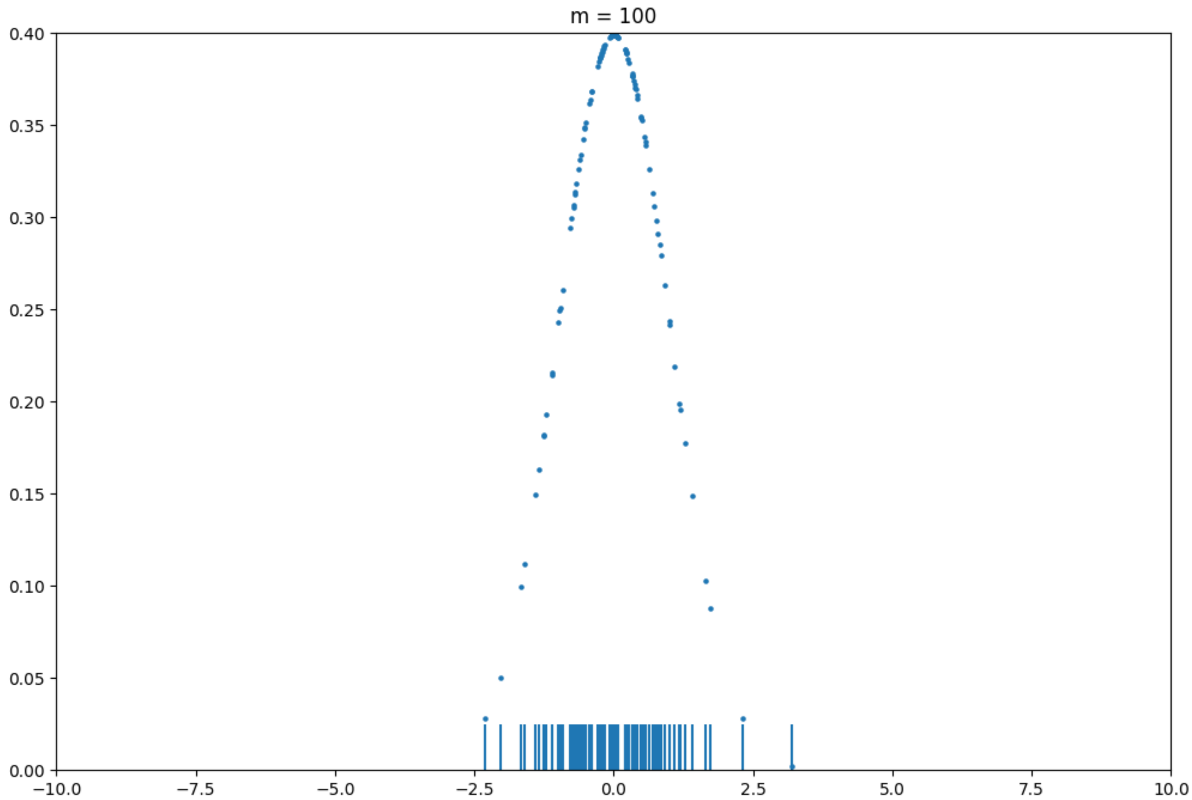
```
X = np.random.normal(mean, std, m)
Y = norm.pdf(X, mean, std)

ax = plt.subplot()
ax.set_title(f'm = {m}')
ax.set_xlim(left=-10, right=10)
ax.set_ylim(bottom=0, top=0.4)
ax.scatter(X, Y, s=5)
ax.vlines(X, 0, 0.025)
```

Przy pomocy funkcji *np.random.normal* wygenerowano *m* punktów z rozkładu normalnego o wartości średniej *mean* (=0) i odchyleniu standardowym *std* (=1).

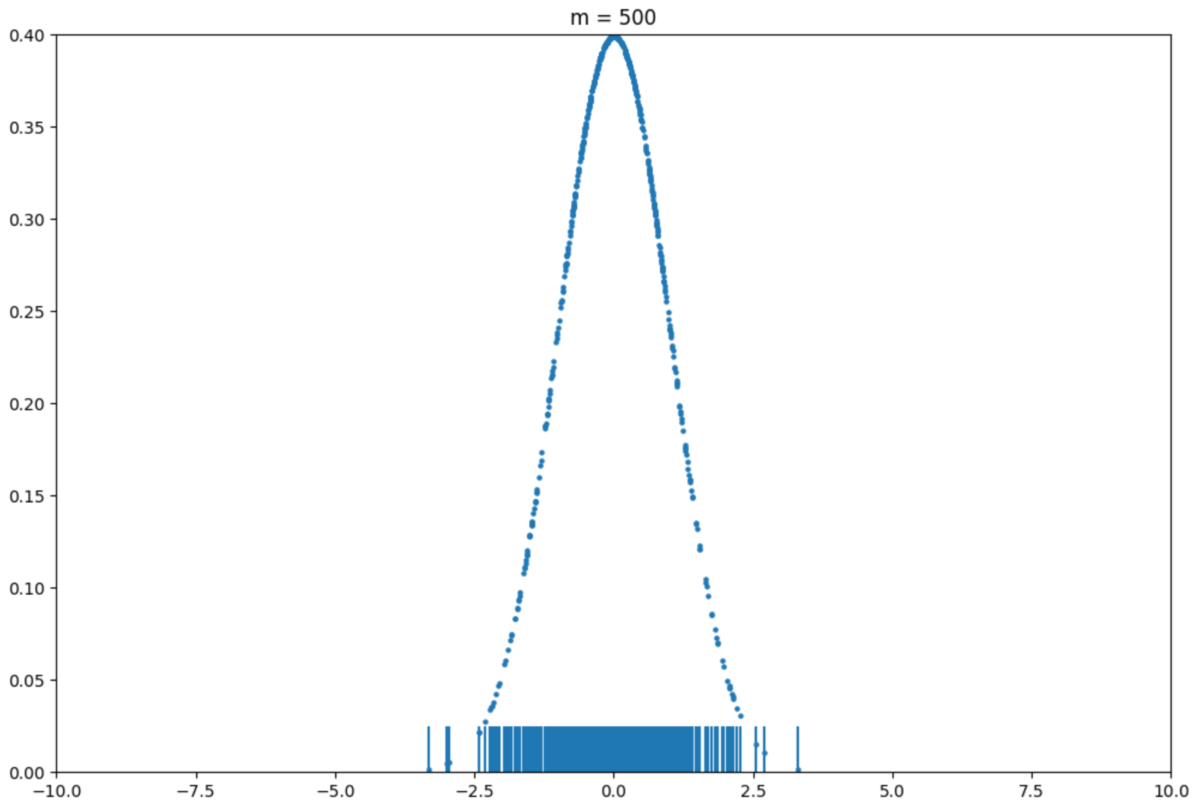
Gęstość prawdopodobieństwa dla każdego z punktów obliczono za pomocą funkcji *scipy.stats.norm.pdf.* Poniżej przedstawiono wzór funkcji gęstości prawdopodobieństwa

$$\frac{1}{\sigma\sqrt{2\pi}}\ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

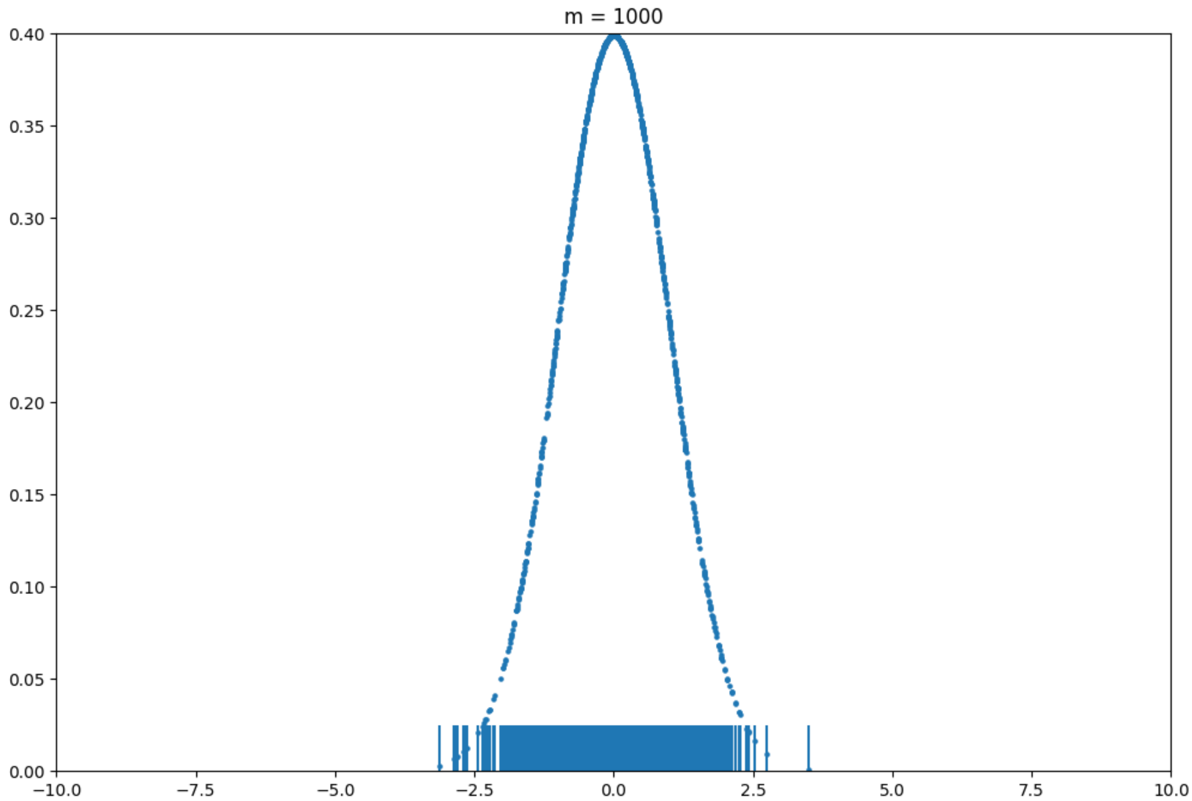
Poniżej przedstawiono otrzymane wyniki dla różnych wartości *m.*



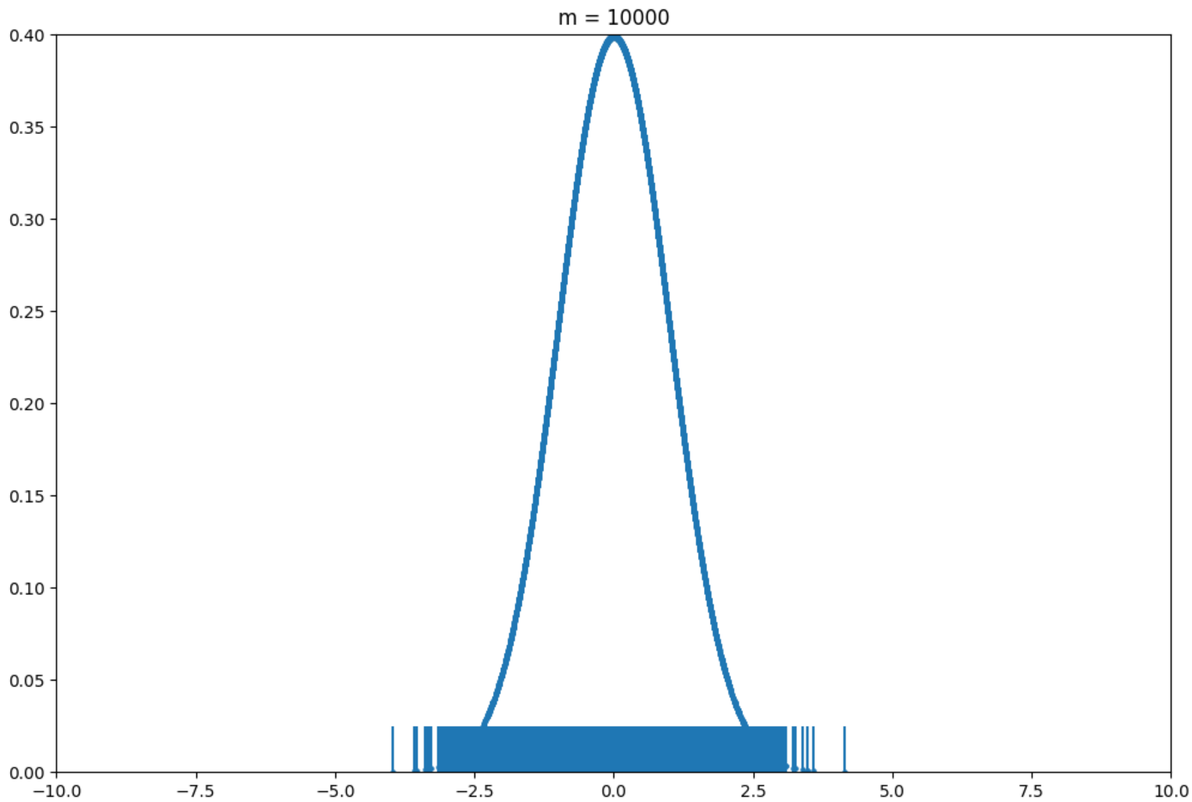
Rys. 1: Rozkład normalny dla m=100 próbek



Rys. 2 Rozkład normalny dla m=500 próbek



Rys. 3 Rozkład normalny dla m=1000 próbek



Rys. 4 Rozkład normalny dla m=10000 próbek

Zgodnie z oczekiwaniami możemy zauważyć że wraz ze zwiększeniem liczności próbek *m*, otrzymane wykresy coraz bardziej przypominają rozkład Gaussa.

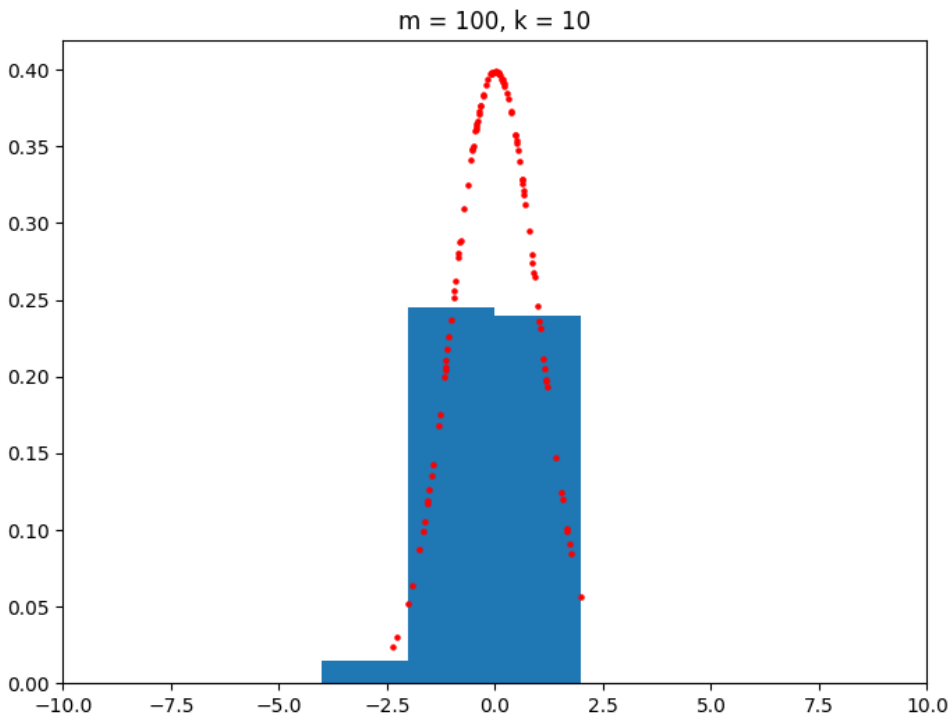
Zadanie 2

Kolejnym zadaniem było wyrysowanie histogramów wygenerowanych punktów z rozkładu normalnego, a następnie manipulowanie wybranymi parametrami tak aby jak najbardziej dopasować histogramy do funkcji gęstości prawdopodobieństwa.

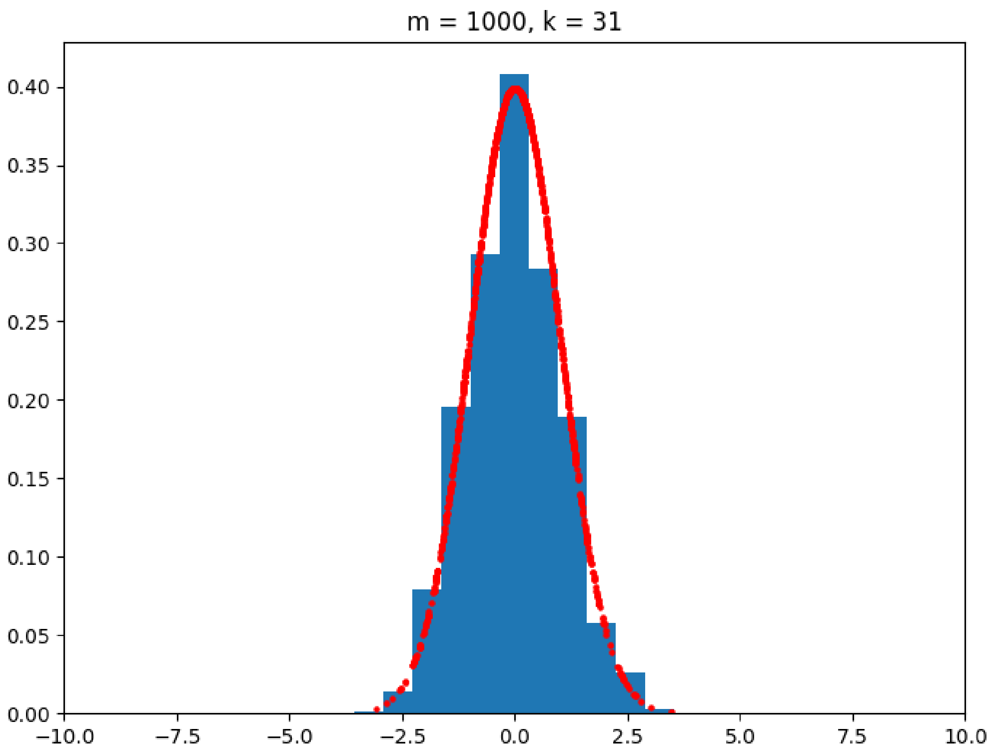
```
ax = plt.subplot()
ax.set_title(f'm = {m}, k = {k}')
ax.set_xlim(left=-10, right=10)
ax.hist(X, k, range=(-10, 10), weights=[(k/20.)/m]*m) # k/20. -> bins/range
ax.scatter(X, Y, s=5, color='red')
```

Histogram generowany jest przy pomocy funkcji *matplotlib.pyplot.hist*. Parametr *k* oznacza tutaj ilość słupków w histogramie, o równej szerokości w zakresie *range*. Parametr *weights* zdefiniowano w taki sposób aby wartości histogramu zostały przeskalowane do wartości funkcji gęstości prawdopodobieństwa. Następnie funkcją *matplotlib.pyplot.scatter* narysowano ‘oczekiwaną’ funkcję gęstości prawdopodobieństwa (dla porównania z histogramem).

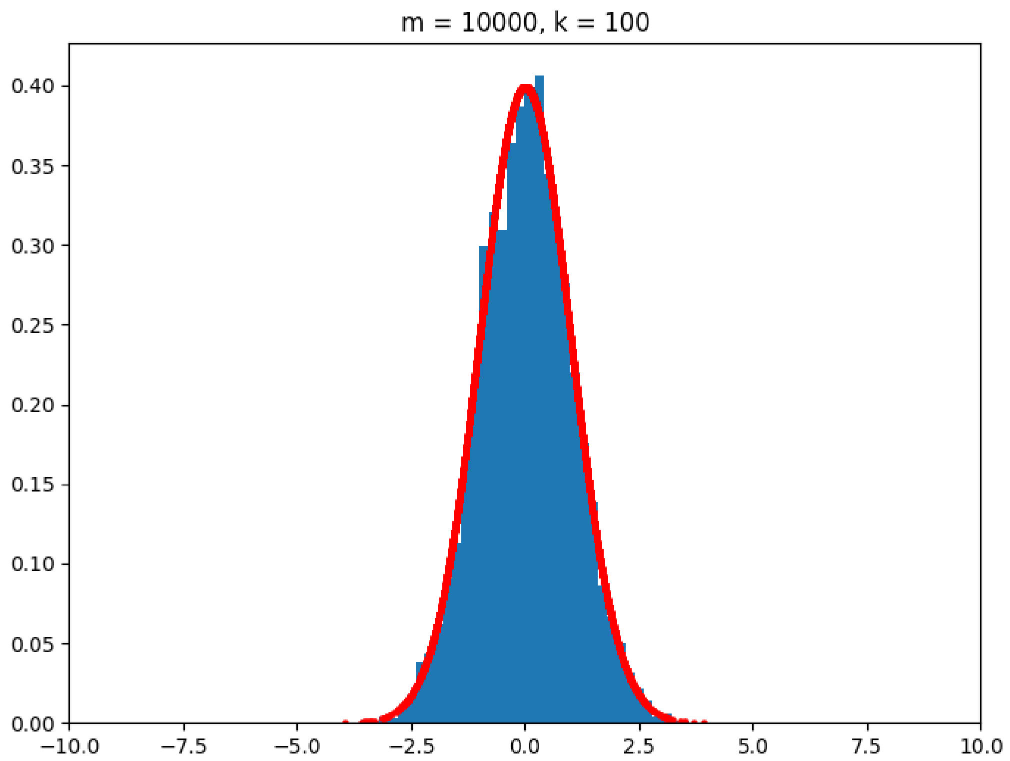
Poniżej przedstawiono wyniki dla różnych parametrów *m.* Jako regułę doboru parametru *k*, wybrano *k = sqrt(m).*



Rys. 5 Histogram dla m=100 i k=10

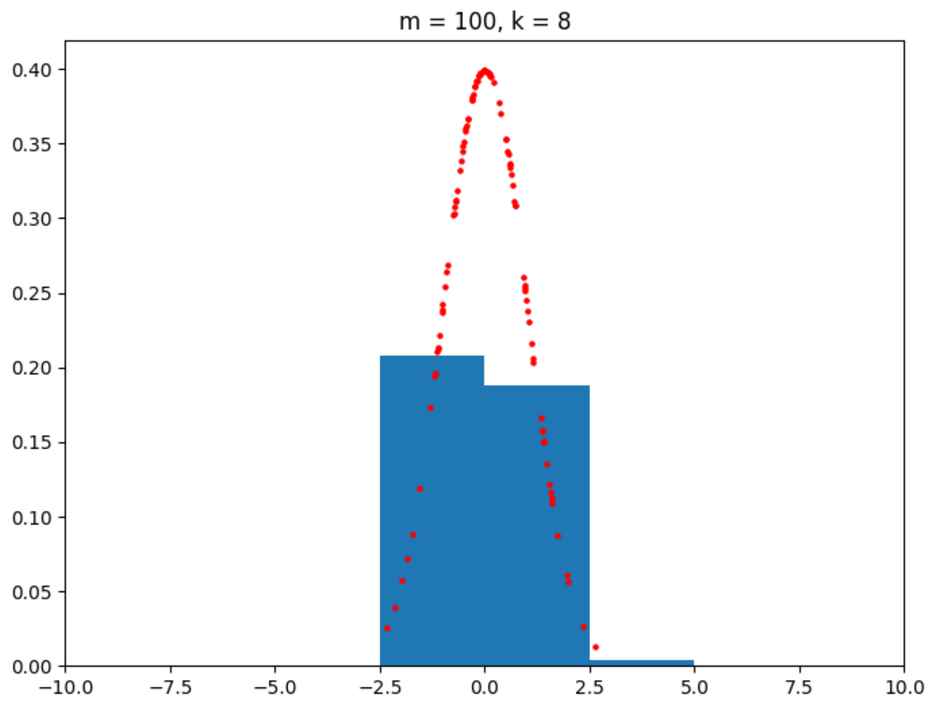


Rys. 6 Histogram dla m=1000 i k=31

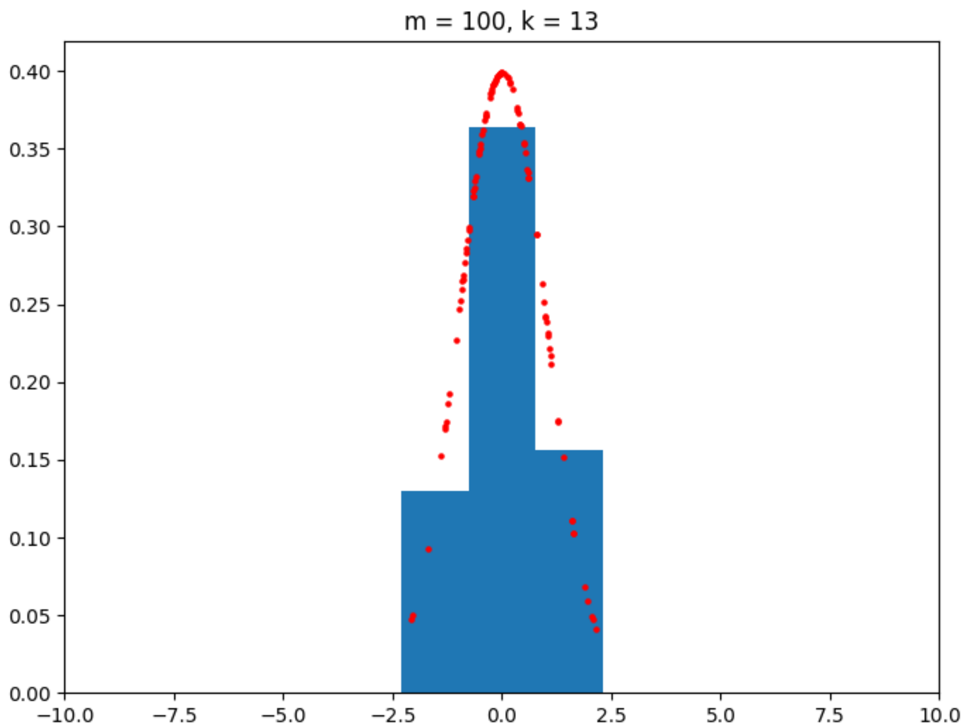


Rys. 7 Histogram dla m=10000 i k=100

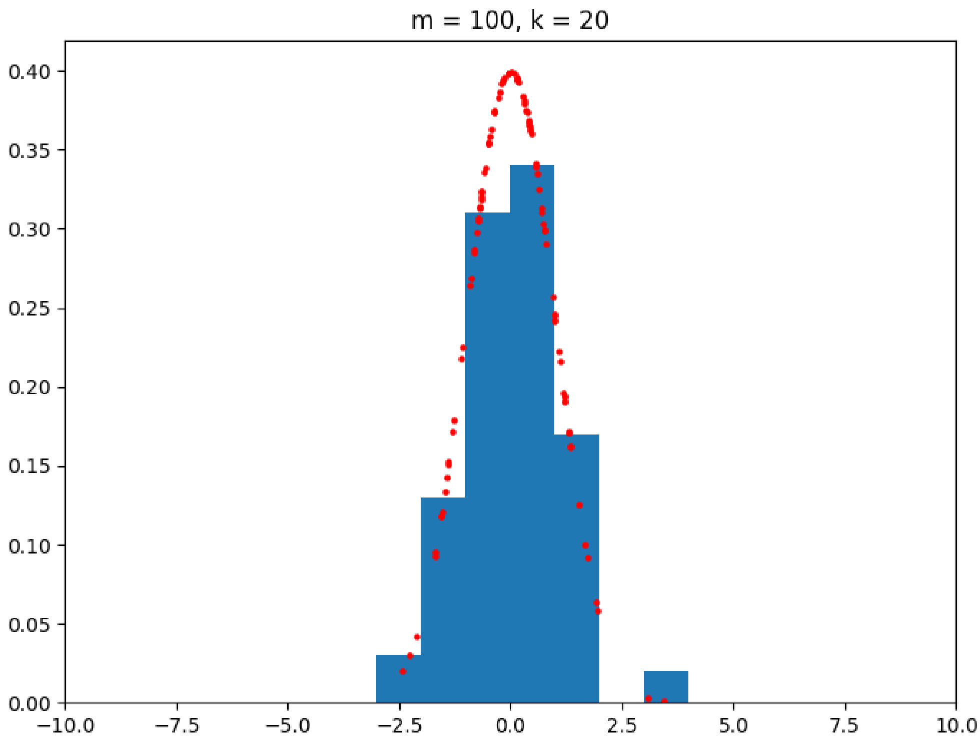
Na wykresie otrzymanym dla *m=100* widać, że taki dobór parametru *k* nie zawsze jest idealny. Na potrzeby analizy wykonano kilka prób z innymi wartościami parametru *k.*



Rys. 8 Histogram dla m=100 i k=8



Rys. 9 Histogram dla m=100 i k=13



Rys. 10 Histogram dla m=100 i k=20

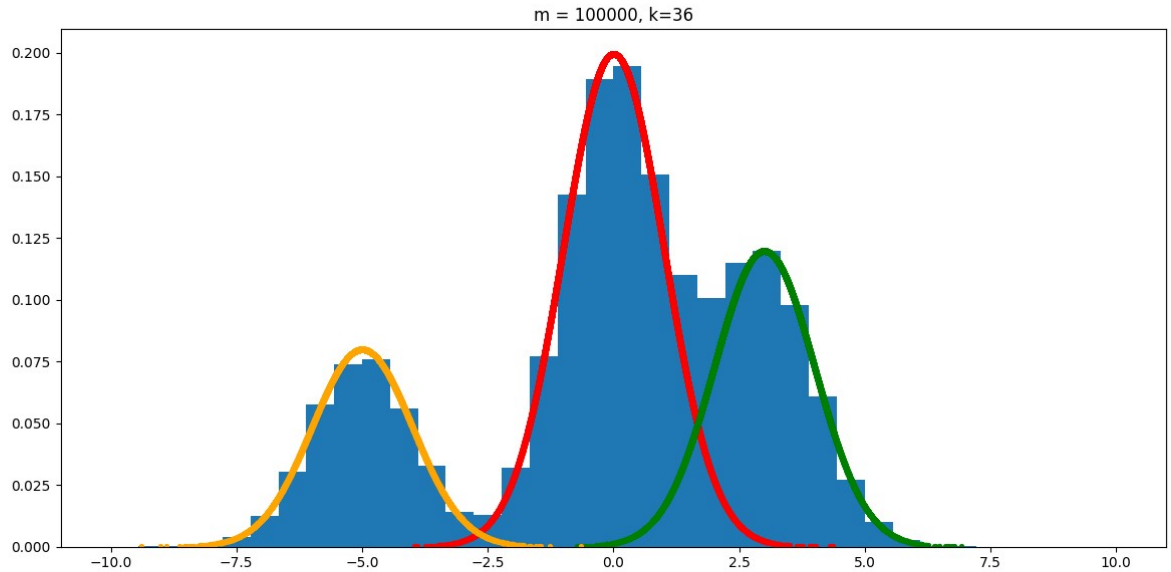
Porównując otrzymane wykresy można zauważyć, że im większy parametr *k,* tym ostrzejsze są zmiany w wysokościach poszczególnych słupków histogramu. Ostateczne określenie który histogram najlepiej oddaje funkcję gęstości jest trudnym zadaniem i może być inaczej interpretowane przez człowieka niż przez komputer.

Zadanie 3

Następnym zadaniem było utworzenie wykresu składającego się z trzech rozkładów normalnych w różnych przesunięciach i udziale procentowym:

- 50% zbioru z rozkładu (0, 1)
- 30% zbioru z rozkładu (3, 1)
- 20% zbioru z rozkładu (-5 1)

Wynikowy wykres został przedstawiony poniżej:



Rys. 11: Wykres przedstawiający histogram oraz wykres gęstości rozkładu normalnego dla parametrów m = 100000 oraz k=36

Kod generujący wykresy został przedstawiony poniżej:

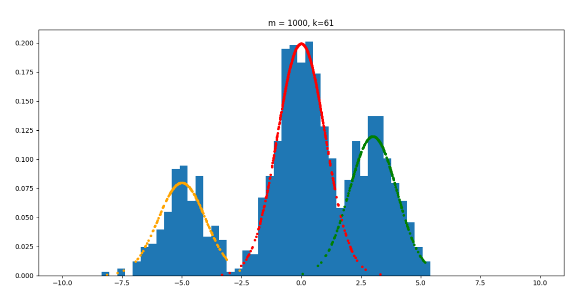
```
def generate_data(m):
    mean = 0
    std = 1
    m1 = int(0.5*m)
    m2 = int(0.3*m)
    m3 = m - m1 - m2
    X1 = np.random.normal(mean, std, m1)
    Y1 = norm.pdf(X1, mean, std)

    mean = 3
    std = 1
    X2 = np.random.normal(mean, std, m2)
    Y2 = norm.pdf(X2, mean, std)

    mean = -5
    std = 1
    X3 = np.random.normal(mean, std, m3)
```

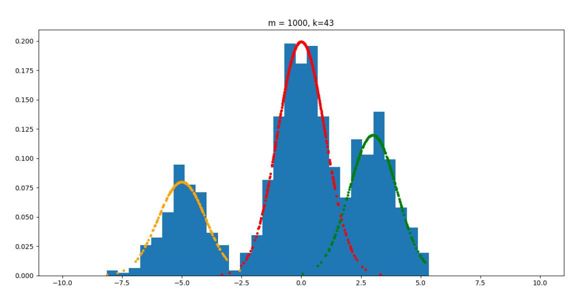
```
    Y3 = norm.pdf(X3, mean, std)
    return {
        "1": [X1, Y1],
        "2": [X2, Y2],
        "3": [X3, Y3]}


def display(m, k_min, k_max, directory, save):
    k_set  = [i for i in range(10, k_max)]
    plt.rcParams['figure.figsize'] = [12, 6]
    data = generate_data(m)
    X1, Y1 = data["1"]
    X2, Y2 = data["2"]
    X3, Y3 = data["3"]
    X4 = np.hstack([X1, X2, X3])
    for x, k in enumerate(k_set):
        fig, axs = plt.subplots(1, 1)
        fig.tight_layout(pad=2.0)
        axs.set_title(f"m = {m}, k={k}")
        axs.hist(X4, bins=k, range=[-10, 10], weights=[(k/20.)/m]*m)
        axs.scatter(X1, Y1*0.5, color="red", s=8)
        axs.scatter(X2, Y2*0.3, color="green", s=8)
        axs.scatter(X3, Y3*0.2, color="orange", s=8)
        if save == True:
            fig.savefig(f'{directory}{m}-{k}.png')
```

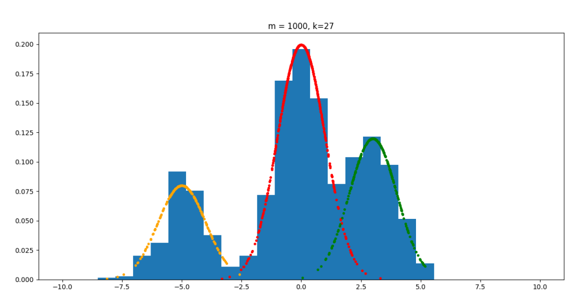
Zadanie 4

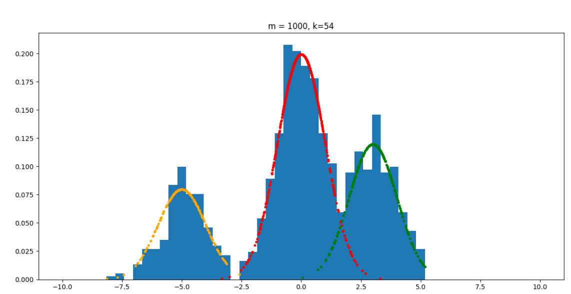
Ostatnim zadaniem było porównanie wpływu parametru *k* odpowiadającemu ilości binów na wykresie na czytelność wykresu dla różnych wartości parametru m. Przeprowadzono analizę dla m o wartościach 1000, 10000, 100000.

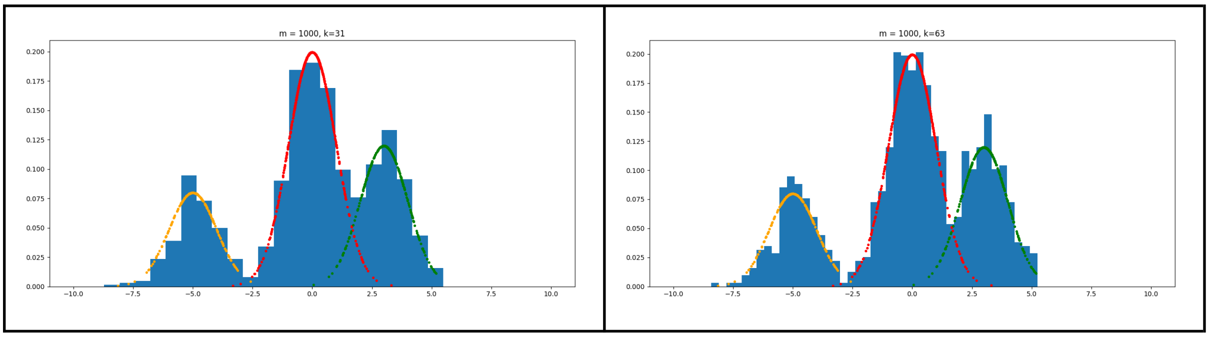
Wartości k dla wykresu o wartości m = 1000





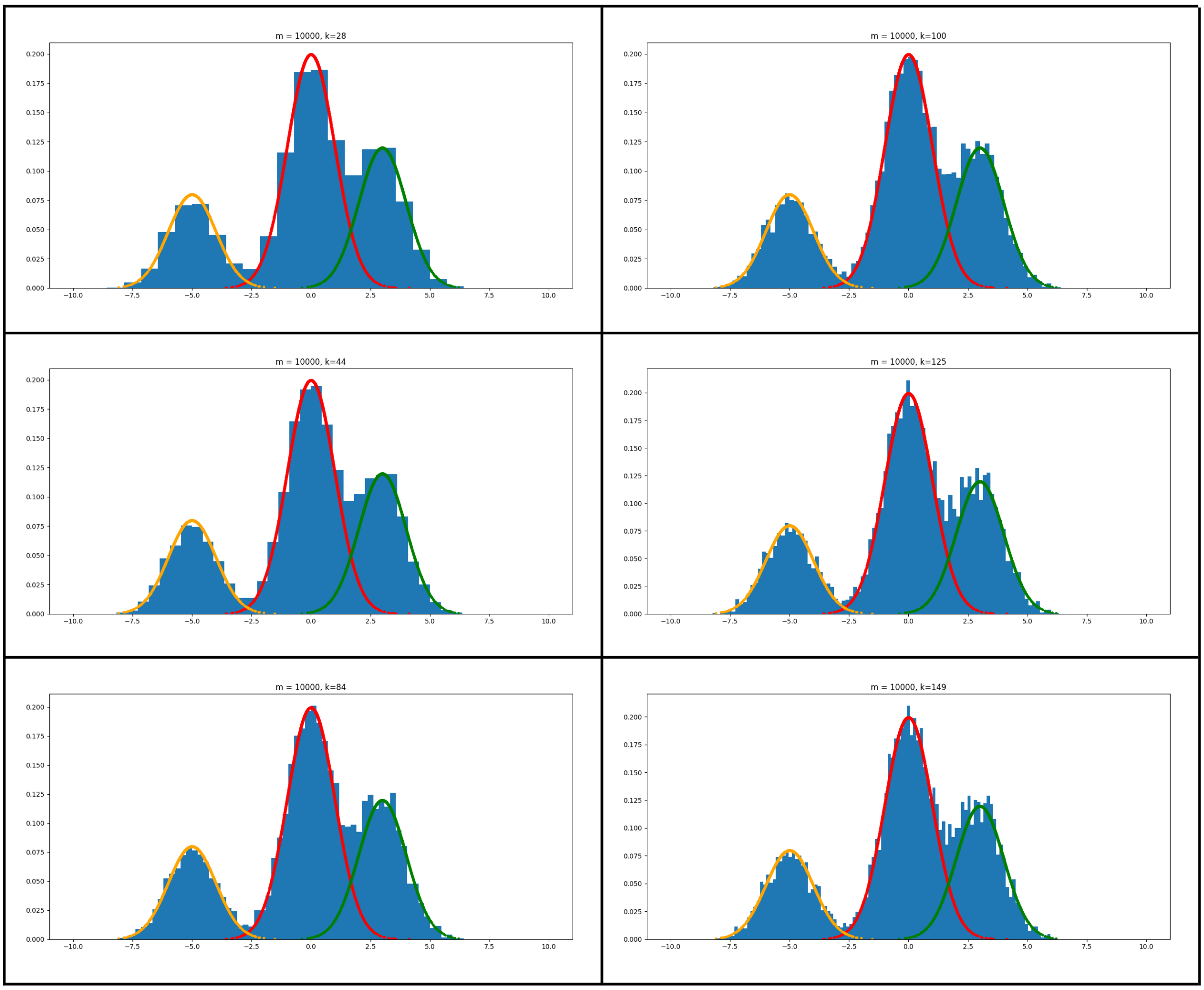






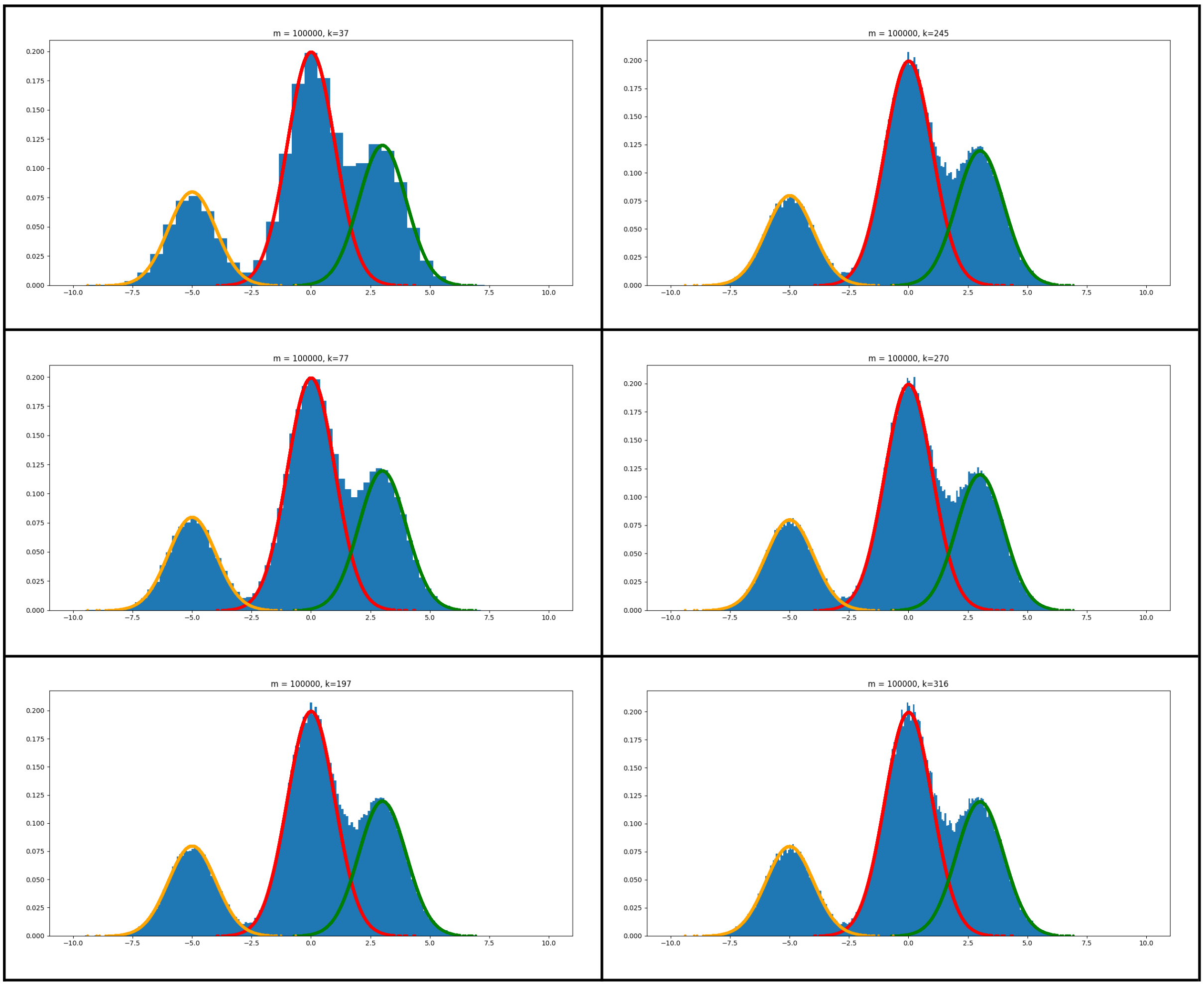
Rys. 12: Wykresy przedstawiające histogram oraz wykres gęstości rozkładu normalnego dla parametrów m = 100000 oraz różnych k

Wartości k dla wykresu o wartości m = 10000



Rys. 13: Wykresy przedstawiające histogram oraz wykres gęstości rozkładu normalnego dla parametrów m = 10000 oraz różnych k

Wartości k dla wykresu o wartości m = 100000



Rys. 14: Wykresy przedstawiające histogram oraz wykres gęstości rozkładu normalnego dla parametrów m = 100000 oraz różnych k

Wnioski:

Dla mniejszego m (poniżej 100000) dobrą regułą dla wartości k jest ustawienie go jako pierwiastek z liczby m. Dla bardzo dużej wartości m (> 100000) wartość k jako pierwiastek z liczby m zaczyna produkować nieznacznie gorsze wyniki - warto w tym przypadku zastosować regułę k = 0.7 * sqrt(m)

Wnioskując po wykresach, wraz ze wzrostem liczby k, histogram zaczyna coraz bardziej dopasowywać się do wykresu gęstości rozkładu normalnego. Po osiągnięciu liczby k przewyższającej pierwiastek z liczby m, histogram zaczyna być zbyt dokładny co utrudnia analizę i powoduje generowanie mniej przejrzystego wykresu.