```
function numOfElements = valueCounter(array)

unique_arr = unique(array);%массив содержащий
уникальные элементы исходного
buff = zeros(size(unique_arr));%буфер для определения
кол-ва каждого из уник. эл-ов

    for i = 1:1:length(unique_arr)
        counter = 0;  %счётчик текущего i-го элемента
        for j = 1:1:length(array)
            if(array(j) == unique_arr(i)) %если есть
совпадение, счётчик увеличивается
                counter = counter + 1;
            end
        end
        buff(i) = counter; %записываем в буффер кол-во
i-го элемента
    end

    numOfElements = buff; % вывод массива с количеством
уникальных элементов (по возрастанию)
end

function [n] = countInRange(x1, x2, arr)
    n = 0;
    for i = 1:1:size(arr)
       if( x1 <= arr(i) && arr(i) <= x2 )
          n = n + 1;
       end
    end
end
////////////////////////////////////////////////////////////////////
```

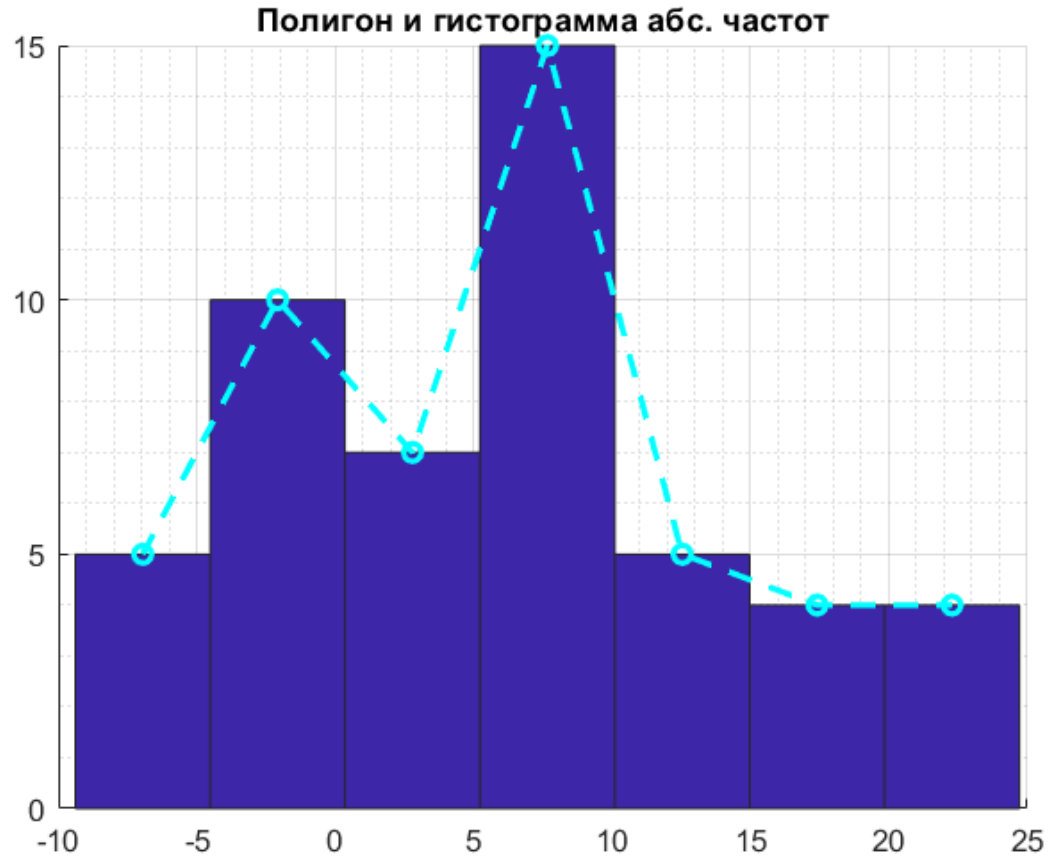
```
clc; clear all; format long
X = [
13.2412291533284
22.7459361740644
-0.205363009592022
1.05828227443028
-2.72985099725213
10.094231465859
3.47582202566441
4.18025412642885
13.7517283035358
-6.60887647365981
11.489240445001
21.3341359873376
-0.0382875129856135
-4.94976364017314
18.1234877946281
-1.98581279197455
```

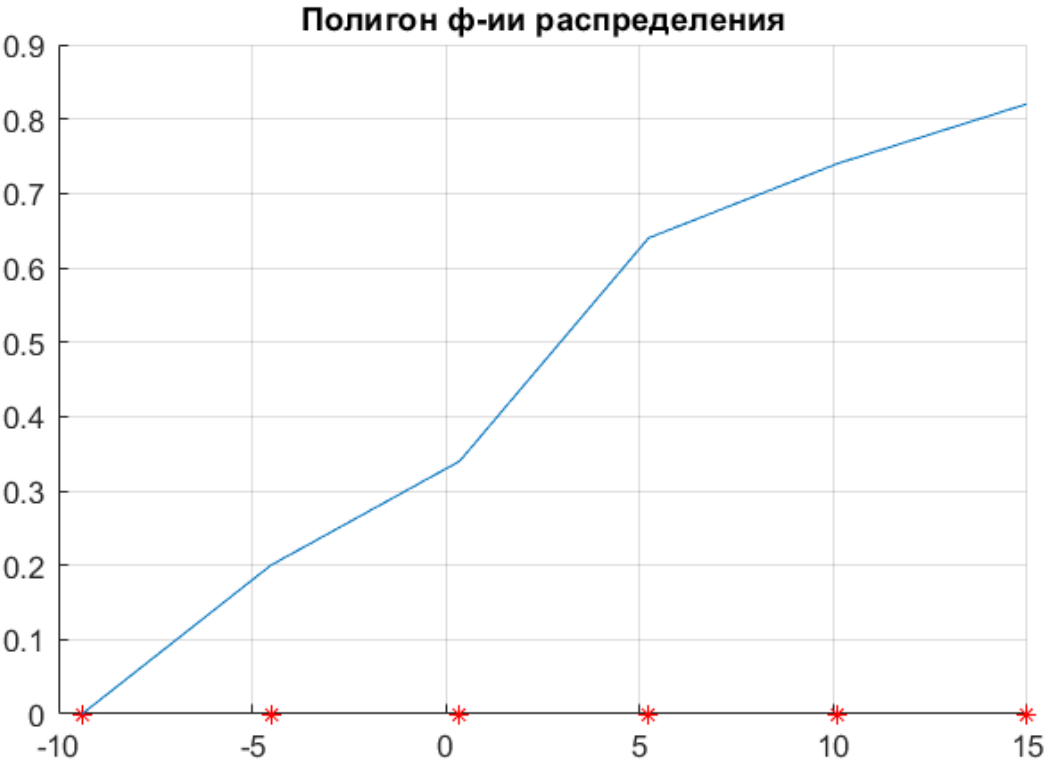
```
9.65941365588863
13.9442486241863
5.91862991795089
-6.26383153173474
-4.1523902031996
15.7951427334884
9.85479078979673
8.67518658005299
-9.38753013869102
23.6937791323226
-4.69729867111843
6.07388413875055
24.7182564746056
9.59763924989031
7.23660263618077
2.8674007012321
-3.61512853717142
18.6428576378922
-1.06868932457706
-0.70955887246984
4.49343431276244
1.92639851154489
2.10151312286222
-1.56049919172551
7.50830251663972
6.34241338328046
13.2629862553959
-2.71171564866184
8.8702810051634
9.61773883716106
5.66425091377702
8.48840434611646
7.74645207795784
17.7499427207833
];
X_sort = sort(X);

X_size = size(X);
n = max(X_size);
%1. Построить вариационный и статистические ряды, найти размах выборки

numOfElements = valueCounter(X); %Частоты значений Статистического ряда

Prob = zeros(size(numOfElements));
for i = 1:1:length(X_sort)
    Prob(i) = numOfElements(i) / length(X); % (Кол-во i-го уник. эл-та) / (кол-во
элементов) = вероятность
end
disp('Вариационный ряд: ')
Prob' %Вариационный ряд

disp('Статистический ряд: ')
statisticalSeries = [X_sort'; numOfElements'; Prob'] %Статистический ряд

disp('Размах выборки: ')
rangeOfSelection = abs(max(X_sort) - min(X_sort))
%figure(1)
%grid on, hold on, grid minor;
%stairs(X_sort)
%hist(X_sort)
%xlabel('n'); ylabel('X')
```

```
%2 Построить таблицу частот групированной выборки, сделав 7 равных
%интервалов
disp('Точки семи равных по длине интервалов: ')
X_linspaced = linspace(X_sort(1), X_sort(length(X_sort)), 8)

absFreq = [0 0 0 0 0 0 0];

for i = 1:1:7
    absFreq(i) = countInRange(X_linspaced(i), X_linspaced(i+1), X_sort);
end

disp('Абсолютная частота')
absFreq
%table(absFreq) %Абсолютная частота
format short

disp('Относительная частота')
relativeFreq = absFreq ./ n %Относительная частота
%table(relativeFreq)
format long

%3

figure(2)
hold on, grid on, grid minor
title('Полигон и гистограмма абс. частот')

midpoints = midPoints(X_linspaced);
hist(X, 7);
[h, x] = hist(X, 7);
%plot(midpoints, absFreq, 'c', 'LineWidth',2)
plot(x, h, '--co', 'LineWidth', 2)


%hist(absFreq, 7)
%[h, x] = hist(absFreq, 7);
%plot(x, h, '--ro')

figure(4)
title('Полигон ф-ии распределения')
hold on, grid on, %grid minor
Fx_i = [0 0 0 0 0 0];
for i = 1:1:6
    if(i ~= 1)
        Fx_i(i) = Fx_i(i-1) + (relativeFreq(i));
    end

    plot(X_linspaced(i), 0,'r*')
    plot(X_linspaced(i), Fx_i(i)) %%Построение



end

for i = 1:1:5
   line( [X_linspaced(i) X_linspaced(i+1)], [Fx_i(i) Fx_i(i+1) ])
end

%figure(5)
%title('Табличная функция')
%hold on, grid on, grid minor
%for i = 1:1:n
```

```
%    plot(i, X(i), 'm*', 'LineWidth', 2);
%end
disp('Мат. ожидание: ')
Mx = mathExpect(X_sort, Prob)

[S2_n, S2_0] = samplingVariances(X_sort, Prob);
disp('Оценка дисперсии(смещённая): ')
S2_n
disp('Оценка дисперсии(несмещённая): ')
S2_0

disp('Медиана: ')
median_X = median(X')

disp('Мода (исх.): ')
mode_X = mode(X')

 ////////////////////////////////////////////////////////////////////////

 function [Mx] = mathExpect(X, P)

    X_size = size(X);
    P_size = size(P);

    if(X_size(1) == P_size(1) && X_size(2) == P_size(2))
        Mx = 0;
       for i = 1:1:max(X_size)
           Mx = Mx + X(i) * P(i);
       end

    else
        disp('Wrong size of matrixes!')
        size(X)
        size(P)
        Mx = 'err';
    end
end
////////////////////////////////////////////////////////////////////////
function [Dx] = variance(X, P)
    X_size = size(X);
    P_size = size(P);

    if(X_size(1) == P_size(1) && X_size(2) == P_size(2))
       Mx = mathExpect(X, P);
       Dx = 0;
       for i = 1:1:max(X_size)
            Dx = Dx + (X(i)^2) * (P(i));
       end

       Dx = Dx - Mx;
    else
        disp('Wrong size of matrixes!')
        size(X)
        size(P)
        Dx = 'err';
    end
end
////////////////////////////////////////////////////////////////////////
function [S2_n, S2_0] = samplingVariances(X, P)
    X_size = size(X);
    P_size = size(P);
```

```
    if(X_size(1) == P_size(1) && X_size(2) == P_size(2))

       n = max(X_size);

       x_average = 0;
       for i = 1:1:n
           x_average = x_average + X(i);
       end
       x_average = x_average / n;

       S2_n = 0;
       for i = 1:1:n
          S2_n = S2_n + (X(i) - x_average)^2;
       end

       S2_0 = S2_n / (n-1);
       S2_n = S2_n / n;


    else
        disp('Wrong size of matrix!')
        size(X)
        size(P)
        S2_n = 'err';
        S2_0 = 'err';
    end
end
```


Полигон и гистограмма абс. частот
15
10
5
0
-10
-5
0
5
10
15
20
25


Полигон ф-ии распределения
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-5
0
5
10
15