**Functions *histfit* and *histogram (see HistEDFConstruct2025.mlx)***

Full sample is known

```
clear
load sample % sample.mat and your script (*.mlx) you need put in the same folder
figure
```

sample.mat contains X variable (sample)

```
subplot(2,3,1),  h=histfit(X,6), hold on
```

```
h =
  2×1 graphics array:

  Bar
  Line
```

Here h - descriptor of Graphical Object

h(1) - bar, h(2) - line of Normal probability density function on effective parameters.

Change the Bins color:

```
set(h(1),'FaceColor','y')
```

Here we determine Bins Width and the highest points in the middle of each bins and add this points with its coordinates on plot:

```
Xc=get(h(1),'XData'); Yc=get(h(1),'YData');
               BinWidthi=(Xc(2)-Xc(1));

r=plot(Xc',Yc','m*','MarkerSize',12, 'LineWidth',2); XY=[Xc',Yc'];

for i=1:length(Yc)
     text(Xc(i),Yc(i)+2,...
     ['(',num2str(Xc(i),2),',',num2str(Yc(i),2),')'],...
     'FontSize',12,'color',[.1,.1,.8])
end

     title(['histfit:  BinWidth(i)=',num2str(BinWidthi,3)] ,'fontsize',12);
     set(gca,'FontSize',12);
```

***Histogram*** function based on *full* sample and value of Nbins (WidthBins is the same for all bins):

```
    subplot(2,3,4), histogram(X,6),
    title('''count'' - default','fontsize',12)
                       set(gca,'FontSize',12);
```

Property of 'Normalization' has different otions: 'count' (default) | 'probability' | 'countdensity' | 'pdf' | 'cumcount' | 'cdf'

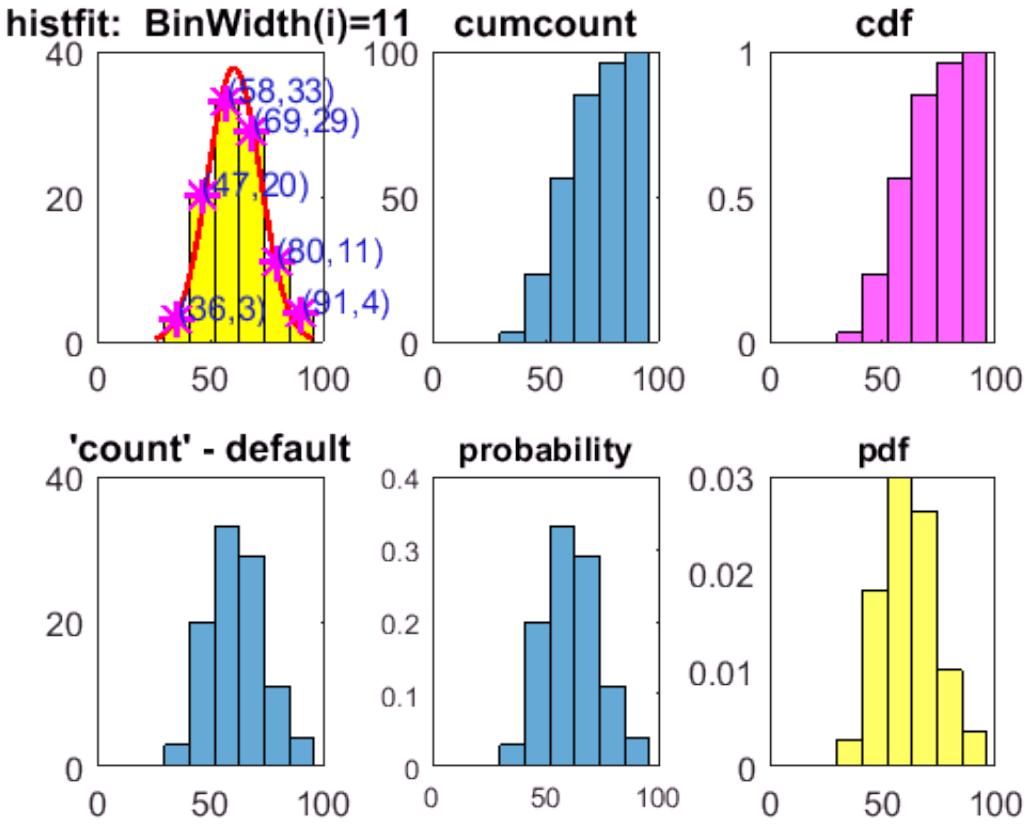
X - data for this example, for histogram we choose 6 bins of the same length

```
   subplot(2,3,2), histogram(X,6,...
              'Normalization', 'cumcount'),
```

```
    title('cumcount','FontSize',12)
            set(gca,'FontSize',12);
 subplot(2,3,5), histogram(X,6,'Normalization', 'probability' ),
    title('probability','FontSize',12) % Ni/N - relative frequency
 subplot(2,3,6), histogram(X,6,'Normalization',  'pdf', 'FaceColor','y'),
            set(gca,'FontSize',12);
    title('pdf','FontSize',12) % HbarI=Ni/(BinWidth*N) ~ pdf
```

Note: For categorical histograms, 'pdf ~'probability', because of BinWidth=1 in this case

```
 subplot(2,3,3), histogram(X,6,'Normalization', 'cdf','FaceColor','m'),
    title('cdf','FontSize',12)
            set(gca,'FontSize',12);
```




Empirical Distribution Function (EDF) has Matlab analogue ***ecdf*** - empirical cumulative distribution function.

Function *cdfplot* helps to represent result as graphics of empirical and theoretical distribution functions

theoretical distribution function we need construct

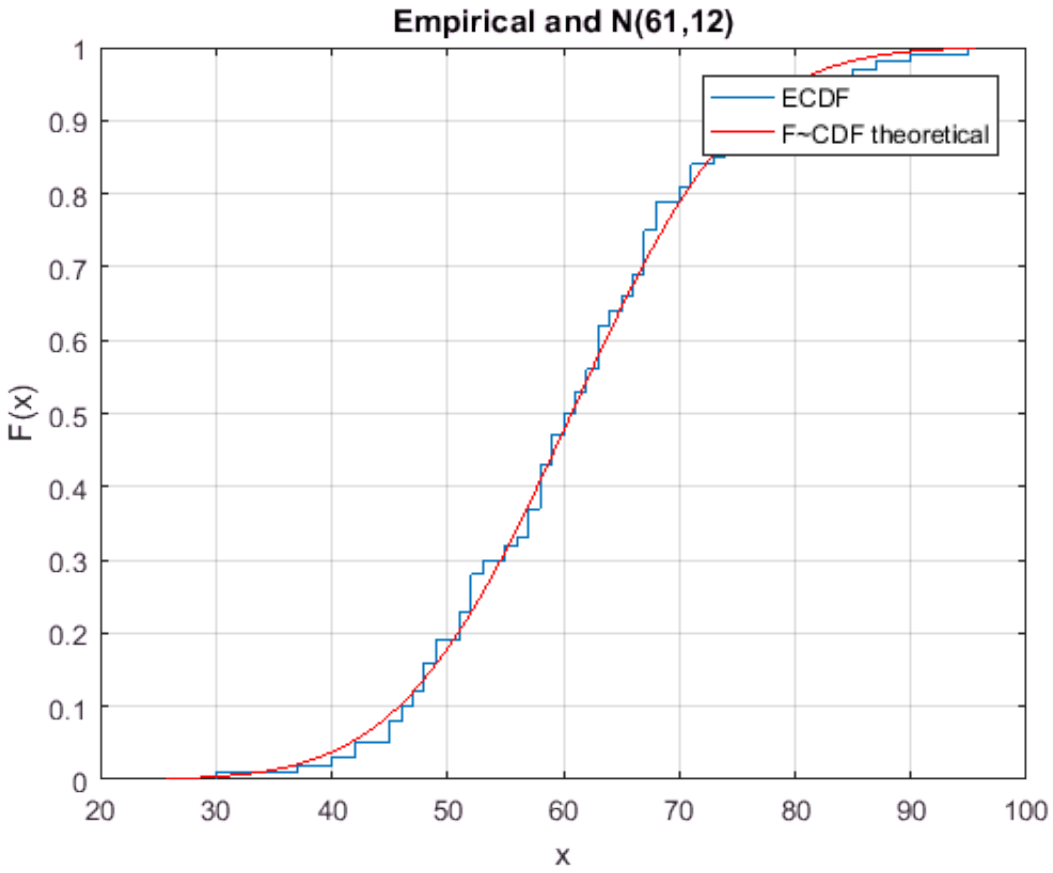
```
figure
cdfplot(X); hold on
```

theoretical distribution function:

$x_i \in [\mu - 3\sigma, \mu + 3\sigma]$, s=$\sigma$, $\mu$ = mx:

```
    mx=mean(X); s=std(X);

 XX=mx-3*s:0.05:mx+3*s;  % use rule x in
 YY=cdf('norm',XX,mx,s);
 plot(XX,YY,'r-');
 title(['Empirical and N(',num2str(mx,2),',',num2str(s,2),')'])
 legend('ECDF','F~CDF theoretical')
```




***Histogram*** function based on parts of the *sample along* with points of BinEdges and values of BinCounts. Let's look at the example of given BinEdges and BinCounts in histogram (different options)

```
 clear;  % HistEDFConstruct2025.mlx
         subplot(1,3,1)
histogram('BinEdges',[10000,14000,22000,30000,36000,70000],...
          'BinCounts',[18 22 30 22 8]);
            title('count');
         subplot(1,3,2)
histogram('BinEdges',[10000,14000,22000,30000,36000,70000],...
            'BinCounts',[18 22 30 22 8],'Normalization','pdf');
            title('pdf')
        subplot(1,3,3)
histogram('BinEdges',[10000,14000,22000,30000,36000,70000],...
            'BinCounts',[18 22 30 22 8],'Normalization', 'cdf');
            title('cdf')
```

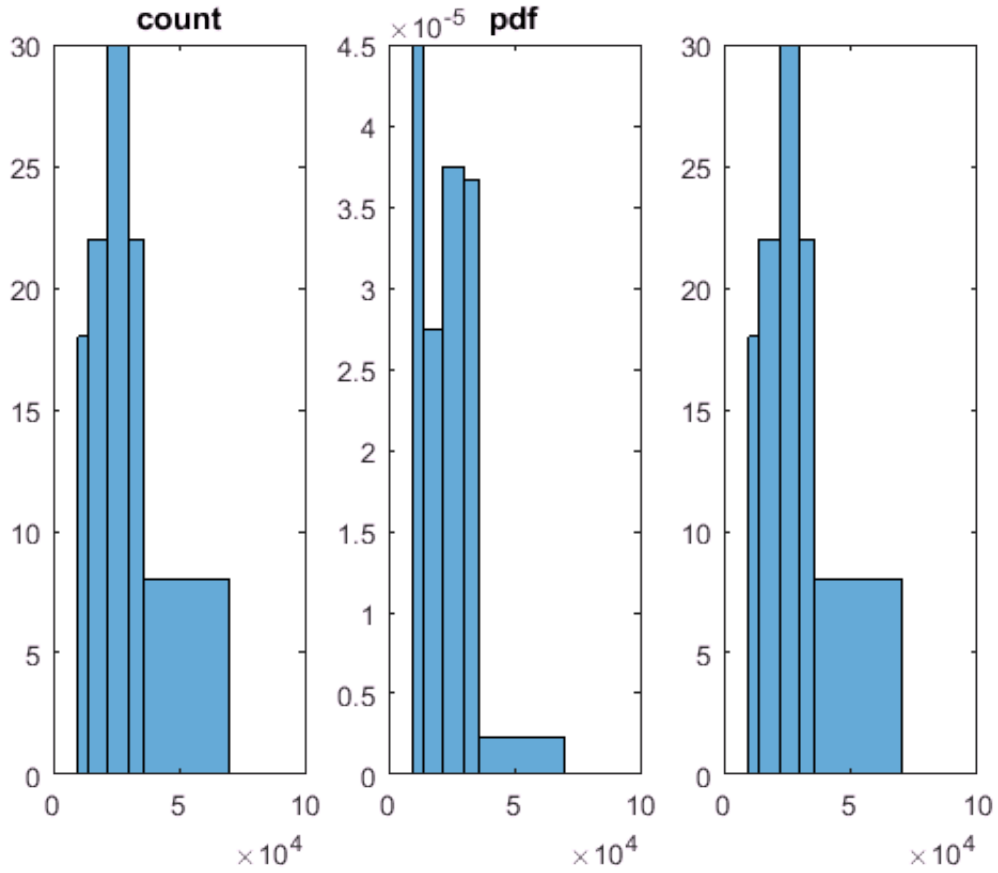
Here we generate subsamples at each given interval with given number of points and combine them into a general sample.

We implement this in a loop.

```
SampleGenerated =[],  % empty before
```

```
SampleGenerated =
     []
```

```
 binedge=[10000,14000,22000,30000,36000,70000]; % you must know, where to find this points
 Ni=[18 22 30 22 8];
 for  i=1: length(Ni)
     SampleGenerated =[SampleGenerated(:);randi([binedge(i),binedge(i+1)],Ni(i),1)];
 end
histogram(SampleGenerated,'BinEdges',binedge);
```



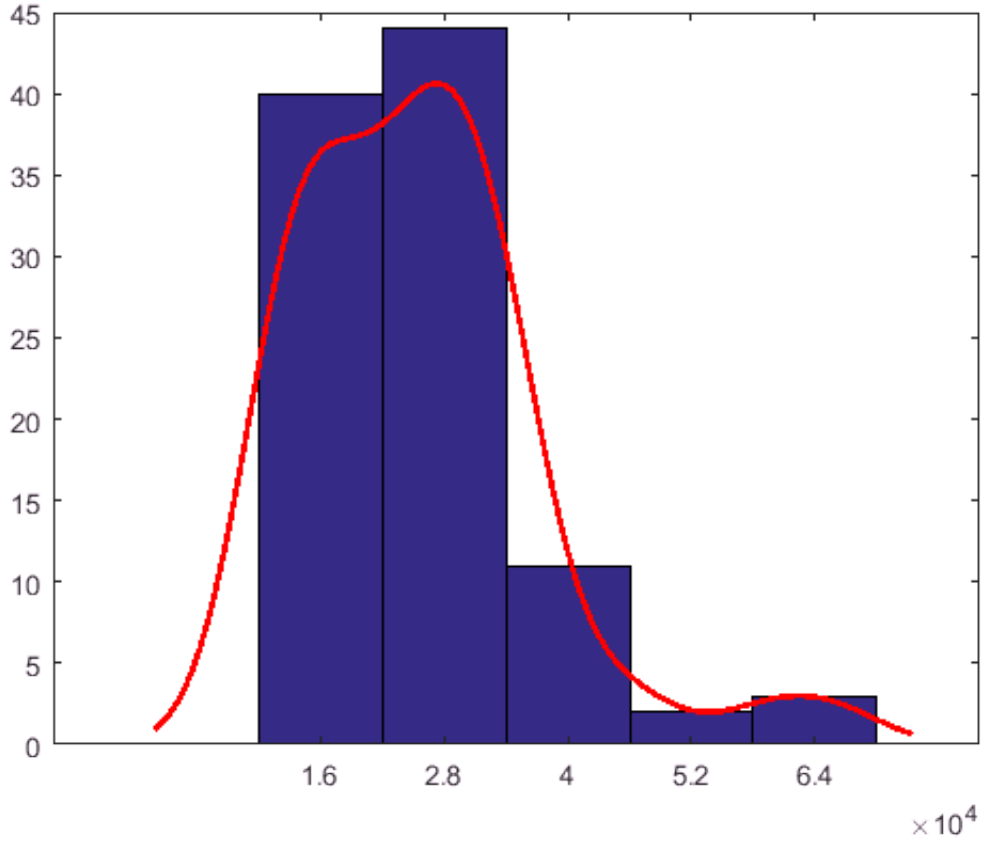

```

```

*histfit*  has bins of the same length, this function allow to choose any theoretical distribution: *beta, normal, poisson, gamma, exponentional*

*'kernel'* - Nonparametric kernel-smoothing distribution (for mixed, polymodal distribution)

```
figure
histfit(SampleGenerated,5,'kernel')
```


45
40
35
30
25
20
15
10
5
0
1.6
2.8
4
5.2
6.4
×10^4