function [optN, C, N] = sshist(x,N)

% [optN, C, N] = sshist(x,N)

%

% Function `sshist' returns optimal number of bins in a histogram

% used for density estimation.

% Optimization principle is to minimize expected L2 loss function between

% the histogram and an unknown underlying density function.

% An assumption made is merely that samples are drawn from the density

% independently each other.

%

% The optimal binwidth D\* is obtained as a minimizer of the formula,

% (2K-V) / D^2,

% where K and V are mean and variance of sample counts across bins with width D.

% Optimal number of bins is given as (max(x) - min(x)) / D\*.

%

% Original paper:

% Shimazaki and Shinomoto, A method for selecting the bin size of a time histogram

% Neural Computation 19(6), 1503-1527, 2007

% http://dx.doi.org/10.1162/neco.2007.19.6.1503

%

% Example usage:

% optN = sshist(x); hist(x,optN);

%

% Input argument

% x: Sample data vector.

% N (optinal):

% A vector that specifies the number of bins to be examined.

% The optimal number of bins is selected from the elements of N.

% Default value is N = 2:50.

% \* Do not search binwidths smaller than a sampling resolution of data.

%

% Output argument

% optN: Optimal number of bins.

% N: Bin numbers examined.

% C: Cost function of N.

%

% See also SSKERNEL

%

% Copyright (c) 2009, Hideaki Shimazaki All rights reserved.

% http://2000.jukuin.keio.ac.jp/shimazaki

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Parameters Setting

x = reshape(x,1,numel(x));

x\_min = min(x);

x\_max = max(x);

if nargin < 2

buf = abs(diff(sort(x)));

dx = min(buf(logical(buf ~= 0)));

N\_MIN = 2; % Minimum number of bins (integer)

% N\_MIN must be more than 1 (N\_MIN > 1).

N\_MAX = min(floor((x\_max - x\_min)/(2\*dx)),50);

% Maximum number of bins (integer)

N = N\_MIN:N\_MAX; % # of Bins

end

SN = 30; % # of partitioning positions for shift average

D = (x\_max - x\_min) ./ N; % Bin Size Vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Computation of the Cost Function

Cs = zeros(length(N),SN);

for i = 1: length(N)

shift = linspace(0,D(i),SN);

for p = 1 : SN

edges = linspace(x\_min+shift(p)-D(i)/2,...

x\_max+shift(p)-D(i)/2,N(i)+1); % Bin edges

ki = histc(x,edges); % Count # of events in bins

ki = ki(1:end-1);

k = mean(ki); % Mean of event count

v = sum( (ki-k).^2 )/N(i); % Variance of event count

Cs(i,p) = ( 2\*k - v ) / D(i)^2; % The Cost Function

end

end

C = mean(Cs,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Optimal Bin Size Selection

[Cmin idx] = min(C);

optN = N(idx); % Optimal number of bins

%optD = D(idx); % \*Optimal binwidth

%edges = linspace(x\_min,x\_max,N(idx)); % Optimal segmentation